

XML-based Web content management

Hans C. Arents
senior IT market analyst

I.T. Works
"Guiding the IT Professional"

Innovation Center, Technologiepark 3, B-9052 Gent (Belgium), Tel: +32 (0)9 241 56 21 - Fax: +32 (0)9 241 56 56

E-mail: hca@itworks.be - Site: <http://www.itworks.be/> - Home: <http://www.arents.be/>

XML-based Web content management

n What are the challenges?

- separating content / presentation / navigation
- targeting look and feel of different access devices
- managing site change and content production process

n How can XML address these challenges?

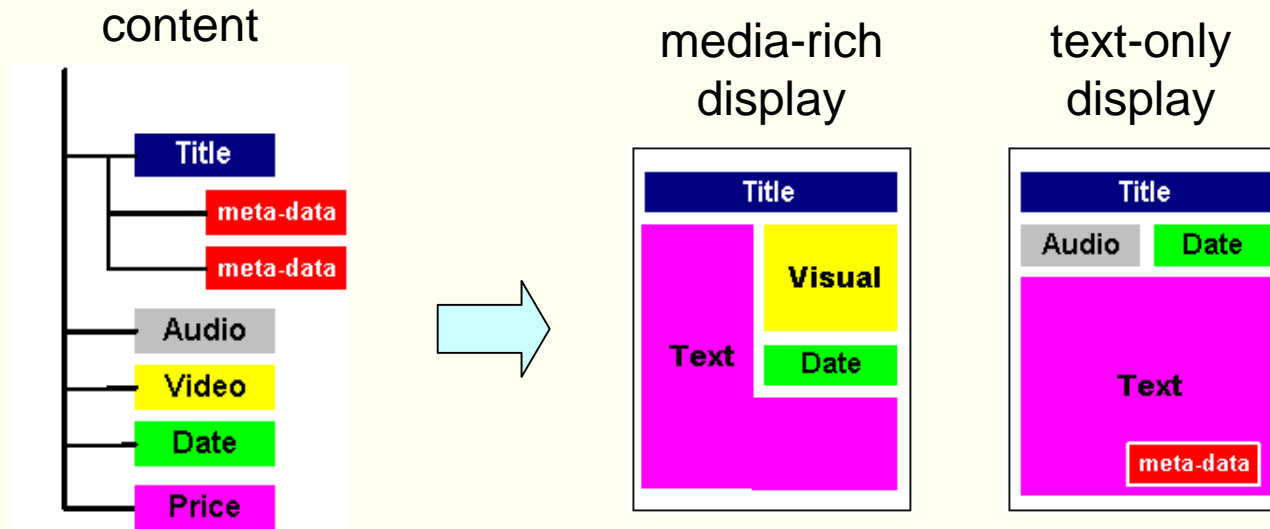
- XML for rich modeling of content
- XSL for flexible rendering of content
- XML for defining navigation of content

n Build or buy?

- do-it-yourself solutions
 - XML generating or storing databases / XML search engines
- Web publishing frameworks
- Web site management tools

n Conclusions

Separating content and presentation



n Semantically rich content

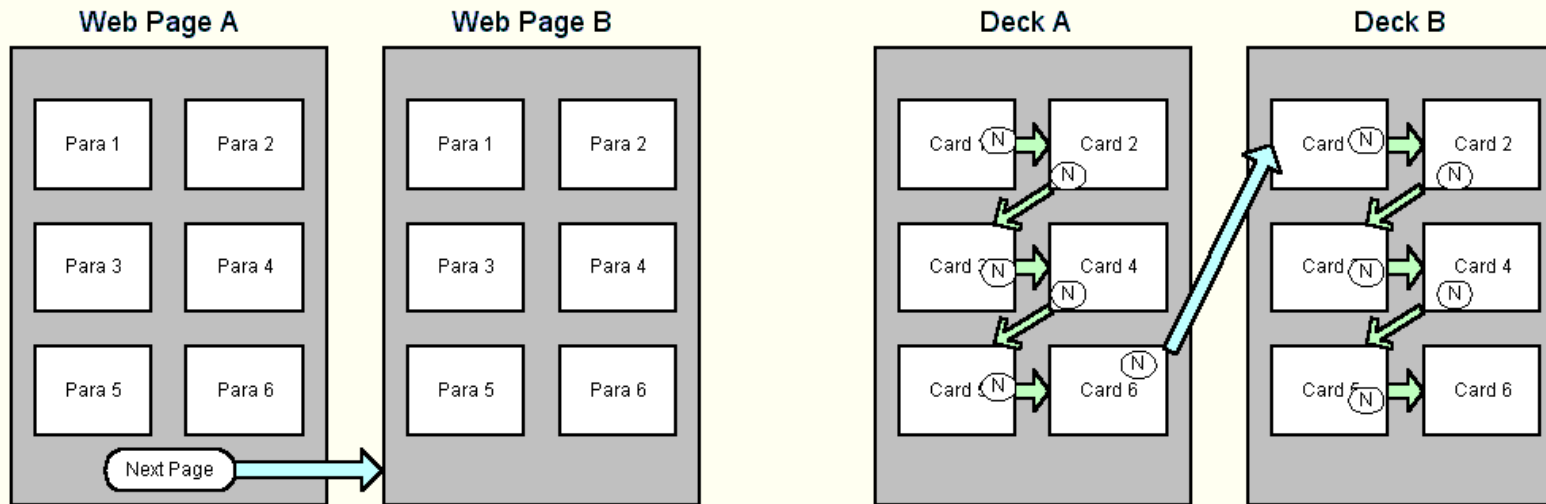
- knowledge about the meaning of the data, datatypes, metadata

presented in different ways ("look")

- restyling the presentation
- re-using the content

n Goal: *write once and display anywhere*

Separating content and navigation



- n Same rich document content adapted to different form factors ("feel")
 - different content chunking
 - e.g. page vs. paragraph granularity
 - different navigation behaviour
 - e.g. random vs. conversational access
- n Goal: *write once and navigate anywhere*

Targeting different access devices



fixed
PC



mobile
PC



handheld
PC



mobile
phone



voice



WebTV



WebPad

n by 2002: 75% of all Internet access using non-PC devices

n never-ending need to adapt and optimize presentation

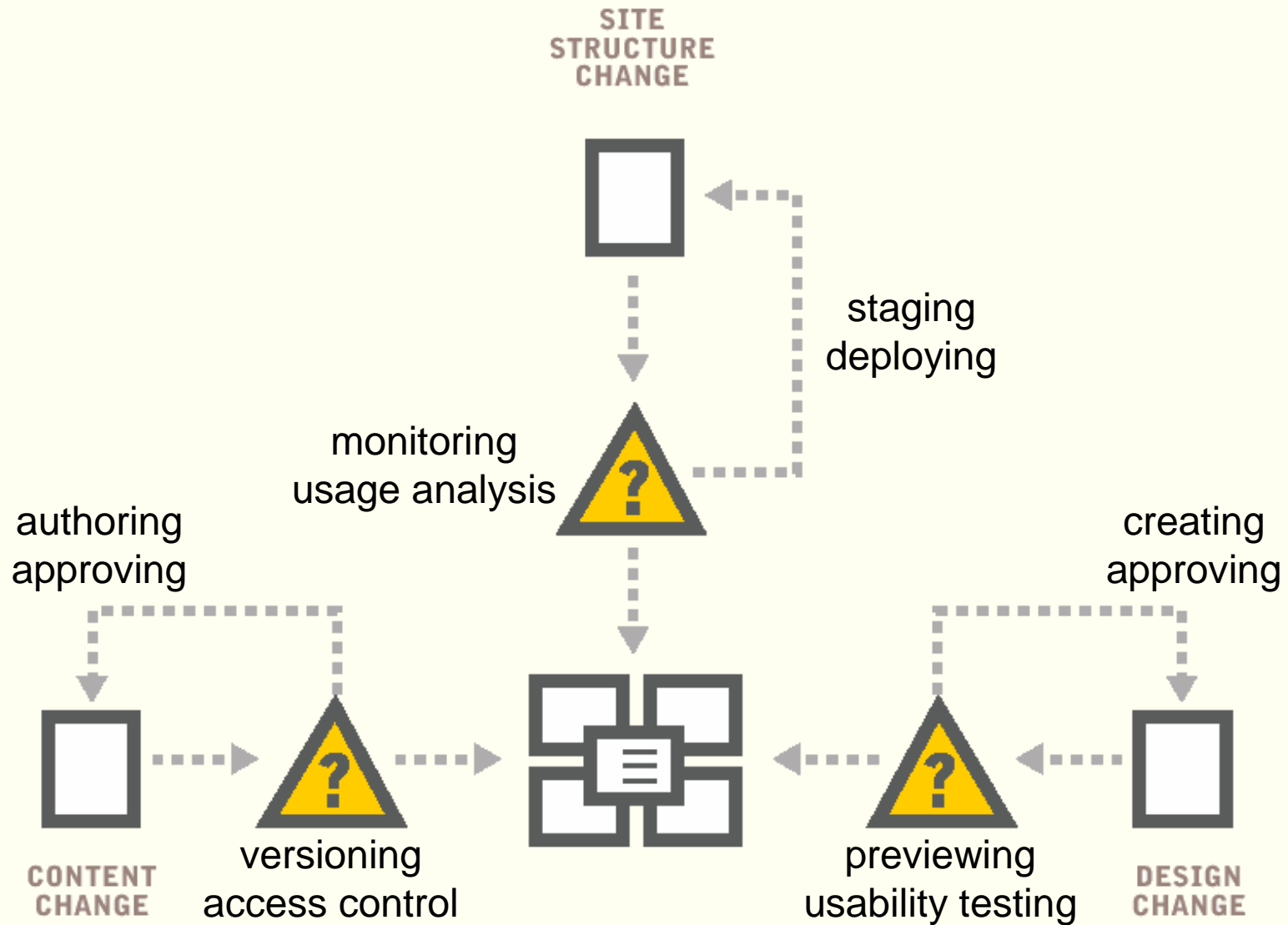
– different display markup languages:

- fixed and mobile PC: HTML, XHTML / handheld PC: HDML
- mobile phone: WML (WAP), cHTML (i-mode) / voice: VoxML

– different device characteristics:

- different sizes of screens / different screen resolutions
- new ways of interacting (e.g. "callto" hyperlink, talkback button)

Managing change and content production



XML for rich modeling of content



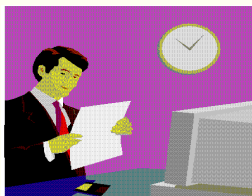
Design

```
<chip_design>
<description>
This part does many interesting and amazing things none of which any of us understand.
</description>
<pin_count>40
</pin_count>
</chip_design>
```



Testing

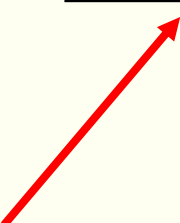
```
<test_report>
<failure_rate>
<overvoltage>10
</overvoltage>
</failure_rate>
</test_report>
```



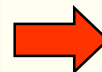
Marketing

```
<specsheat>
<new_partno>31231
</new_partno>
<pricing>
<individual>32.95
</individual>
<volume>23.95
</volume>
</pricing>
</specsheat>
```

semantically rich content

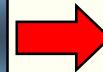


Document assembly process

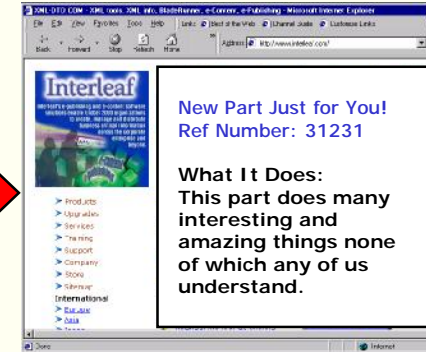
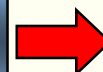


Document styling Process

Web style sheet



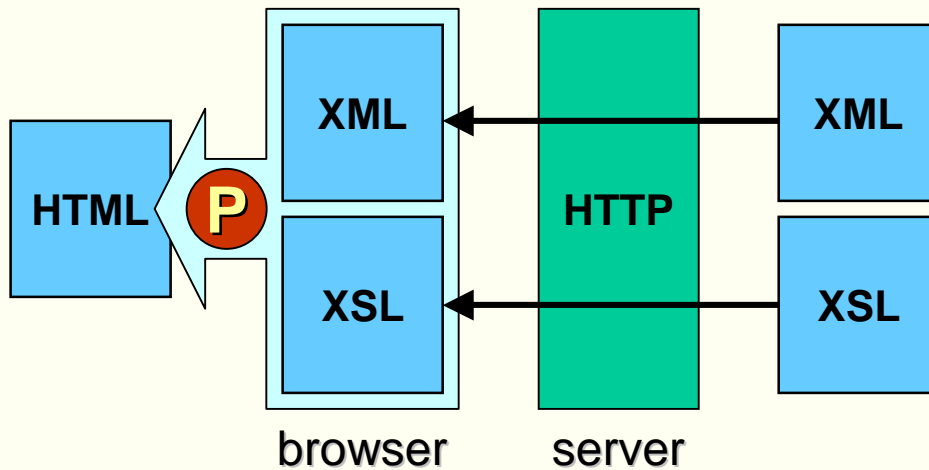
Device style sheet



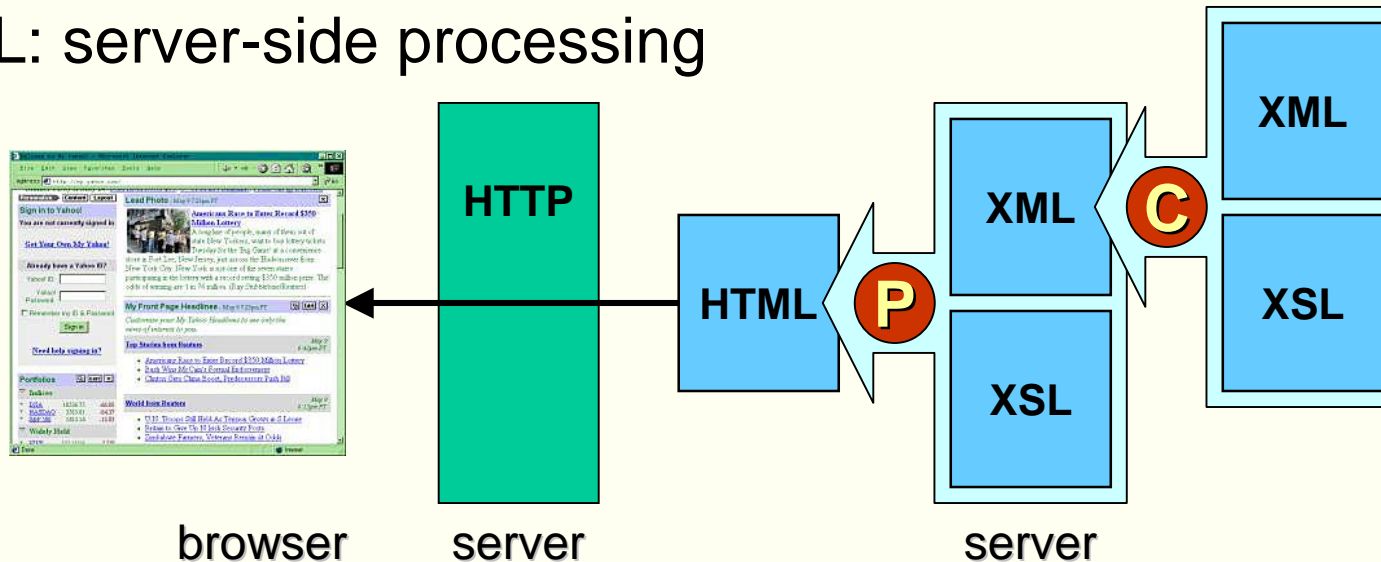
well-structured content (DTD, XML Schema)

XSL for flexible rendering of content

n XSL: client-side processing

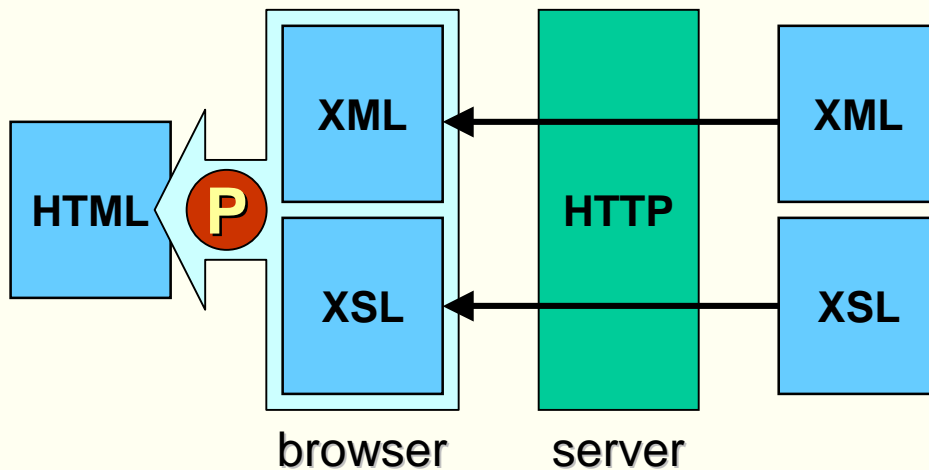


n XSL: server-side processing

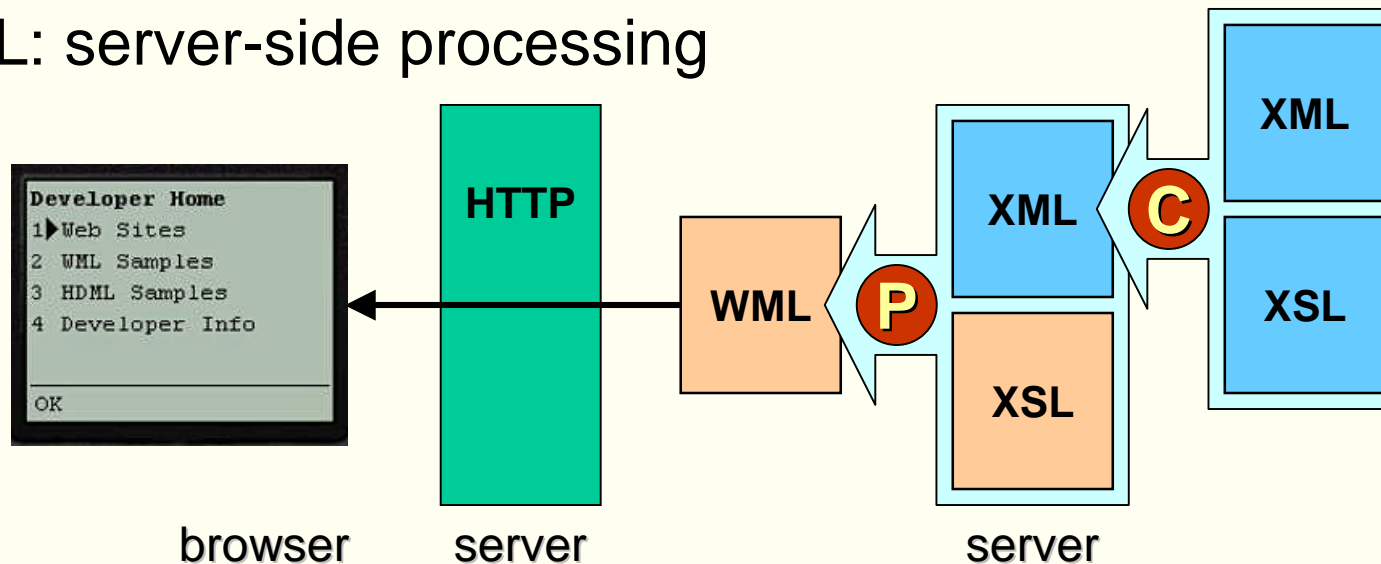


XSL for flexible rendering of content

n XSL: client-side processing



n XSL: server-side processing



XML for defining navigation of content

n Adapting content to different form factors

- in which order portions of information are shown (XDNL)
- in what style each portion of information is displayed (XSL)

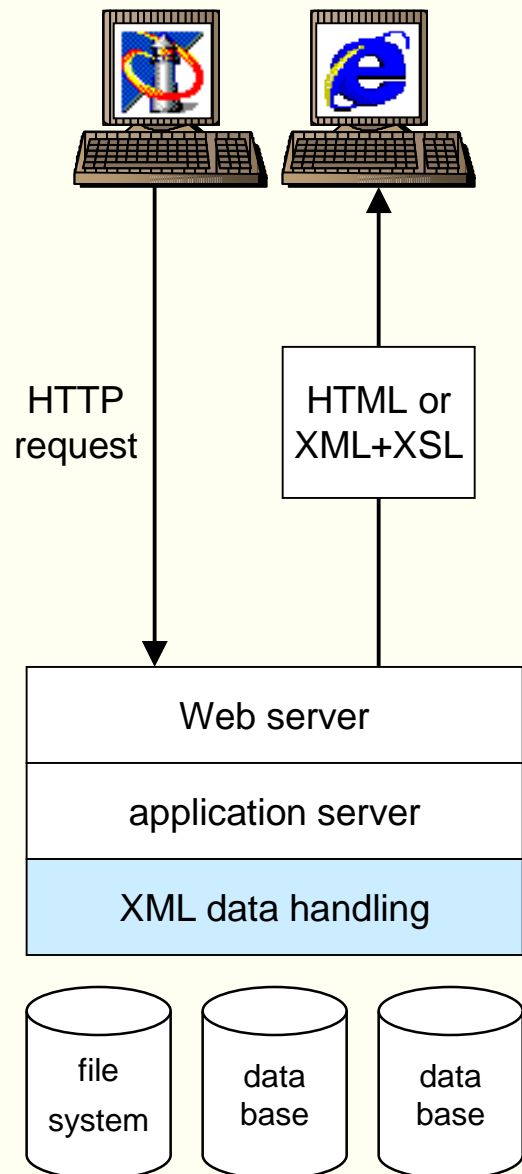
n XML Document Navigation Language (XDNL)

<http://www.w3.org/TR/xdnl/>

- navigation = set of instructions to create an appropriate flow through a document for a particular type of device
 - which portion of the document is shown at each time, as a *leaf-document*
 - from / to which leaf-document a *flow-link* should traverse
- flow-links:
 - links from outline leaf-document to detailed leaf-documents
 - links between sequentially paged leaf-documents
 - links between leaf-documents at the same level

n Note: XDNL is not a standard and no tool is using it (yet)

Do-it-yourself solutions



n Web browser

- sends HTTP requests
- receives HTTP responses
 - HTML
 - result of server-side XSL
 - XML + XSL
 - for client-side XSL

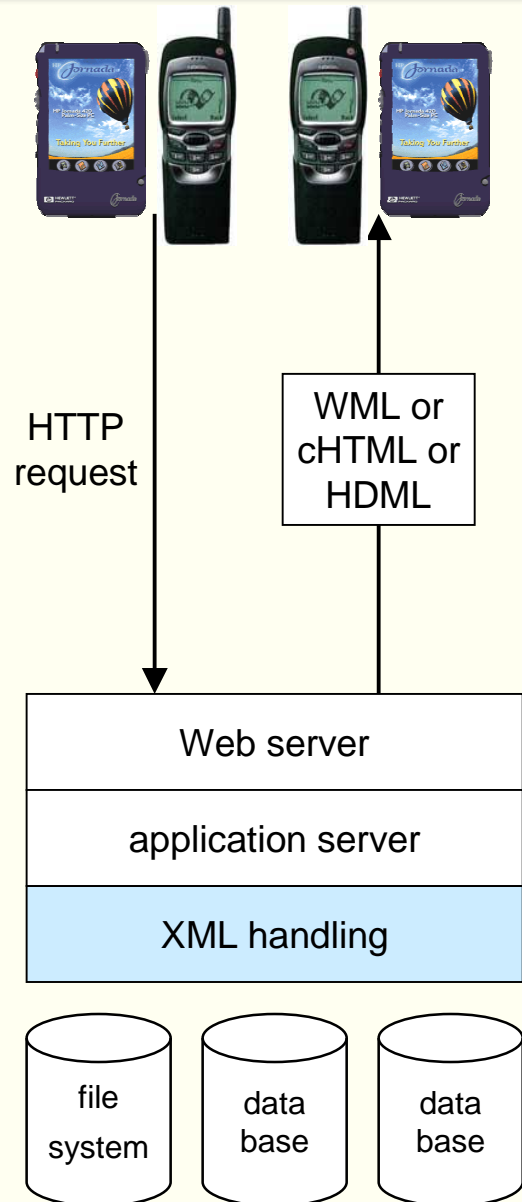
n application server

- intercepting HTTP requests
- generating HTTP responses
 - ASP, JSP pages

n XML data handling code

- retrieving XML files
- interpreting XML queries
 - generated / stored XML
- transforming results using XSL

Do-it-yourself solutions



n device browser

- sends HTTP requests
- receives HTTP responses
 - WML, cHTML, HDML, ...
 - result of server-side XSL (byte-code compiled)

n application server

- intercepting HTTP requests
- generating HTTP responses
 - ASP, JSP pages

n XML data handling code

- retrieving XML files
- interpreting XML queries
 - generated / stored XML
- transforming results using XSL

Transforming XML to HTML: server-side

ASP page

```
<%@ Language=VBScript %>
<%
  Dim oDoc
  Set oDoc = Server.CreateObject("Msxml2.DOMDocument")
  oDoc.load(Server.MapPath("books.xml"))

  Dim oXsl
  Set oXsl = Server.CreateObject("Msxml2.DOMDocument")
  oXsl.load(Server.MapPath("books2html.xsl"))

  Dim oHtml
  oDoc.transformNodeToObject oXsl, Response
%>
```

- Ⓔ Create a DOM tree object and load the XML document into it
 - Create a DOM tree object and load the XSL stylesheet into it
- Ⓕ Perform the XSL transformation on the XML source DOM tree
 - Send the result back as an HTTP response object

Transforming XML to HTML: client/server-side

ASP page with browser sniffing

```
<%@ Language=VBScript %>
<%
  Dim IsIE
  IsIE = InStr(1,Request.ServerVariables("HTTP_USER_AGENT"),"MSIE") > 0
  If IsIE Then
    Dim fso, file
    Set fso = Server.CreateObject("Scripting.FileSystemObject")
    Set file = fso.OpenTextFile(Server.MapPath("books.xml"))
    Response.ContentType = "text/xml"
    Response.Write file.ReadAll
    file.Close
  Else
    Dim oDoc
    Set oDoc = Server.CreateObject("Msxml2.DOMDocument")
    oDoc.load(Server.MapPath("Books.Xml"))

    Dim oXsl
    Set oXsl = Server.CreateObject("Msxml2.DOMDocument")
    oXsl.load(Server.MapPath("QueryResult.xsl"))

    Dim oHtml
    oDoc.transformNodeToObject oXsl, Response

  End If
%>
```

Transforming XML to HTML: client/server-side

- Ⓔ Detect whether the browser requesting the ASP page is a Microsoft Internet Explorer or not
 - If it is IE, then the stylesheet can be executed client-side, so ...
- Ž create a file system object and read the XML document into it
 - set the content type of the HTTP response object to XML
 - write the XML document into the HTTP response object
 - note: this XML document has to contain an `xml-stylesheet` directive!

```
<?xml-stylesheet href="file-name" type="text/xsl"?>
```
 - send the HTTP response object back to the browser
 - If it's not IE, then the stylesheet has to be executed server-side

Do-it-yourself solutions: content management

n Why not use the file system?

- ü similar to managing HTML files
- ü XML files can be indexed and searched
- ü XML files can be processed with XSL or DOM
- û size of XML data files à query and update problems
- û XML data is managed as an ordinary string of characters

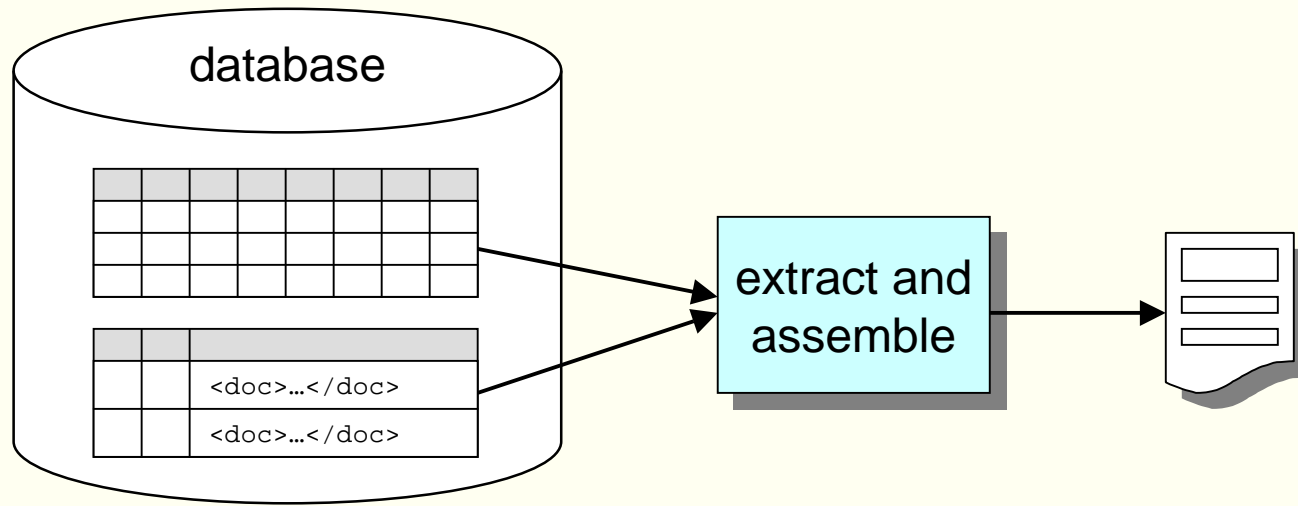
n Advantages of using databases:

- ü inherent reliability of database
- ü query and update handled by database
- ü XML data can be managed as a parsed tree of nodes

n Types of databases suited for XML:

- XML generating databases
- XML storing databases: native / fragments

XML generating database



n Information structured in the database, not as documents

- contains data defined by a database schema
- may contain XML documents stored as BLOB or CLOB

n XML generated on output from the database

- by special XML formatting logic in program code (e.g. ASP, JSP)
- using the built-in support for retrieving query results in XML format

Microsoft SQL Server 2000

n Support for the HTTP retrieval of query results in XML format:

- sending an URL-compliant SQL query
- , calling an SQL query template file with XML syntax
- f* sending an URL-compliant XPath expression using an XML View

n the **SELECT** statement has a new **FOR XML** clause

- FOR XML RAW
- FOR XML AUTO
 - use the table name as the element name instead of the default **row** name
- FOR XML EXPLICIT
 - specify exactly how the XML is structured as a tree when the results from the query are returned
- FOR XML ..., DTD or FOR XML ..., XMLData
 - include an XML Schema in DTD or XDR (XML-Data Reduced) format

Sending an SQL query

Query URL

<http://www.mysite.com/mydatabase?>

`sql=SELECT+*+FROM+Customers+WHERE+CustomerID='ALFKI'+FOR+XML+AUTO&root=result
&xsl=stylesheet.xsl`

Query results

```
<?xml version="1.0"?>  
<result>  
  <Customers  
    CustomerID="ALFKI "  
    CompanyName="NewCom"  
    ContactName="Maria Anders"  
    ContactTitle="Sales Representative"  
    Address="Obere Str. 57 "  
    City="Berlin"  
    PostalCode="12209 "  
    Country="Germany"  
    Phone="030-0074321 "  
    Fax="030-0076545 "  
  />  
</result>
```

Calling an SQL query template

Query URL

<http://www.mysite.com/mydatabase/myquery.xml>

Query template

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="stylesheet.xsl"?>
<root xmlns:sql="urn:schemas-microsoft-com:xml-sql">
  <sql:query>
    SELECT Customers.CustomerID, OrderID
    FROM Customers, Orders
    WHERE Customers.CustomerID = Orders.CustomerID
    FOR XML AUTO
  </sql:query>
</root>
```

Query results

```
<?xml version="1.0"?>
<root>
<Customers CustomerID="ALFKI">
  <Orders OrderID="10643">
  <Orders OrderID="10643">
</Customers>
...
</root>
```

Sending an XPath expression

Query URL

[http://www.mysite.com/mydatabase/Employees\[@EmployeeID=1\]/@Photo](http://www.mysite.com/mydatabase/Employees[@EmployeeID=1]/@Photo)

Query results



n Query using XPath syntax on a table with an XML View

- tables as elements
- columns as attributes

n Purpose:

- to select a single value in a single column
 - value returned in native format, not XML
- to allow direct retrieval of objects from the database
 - e.g. images, OLE objects, etc.

Oracle 8i/9i

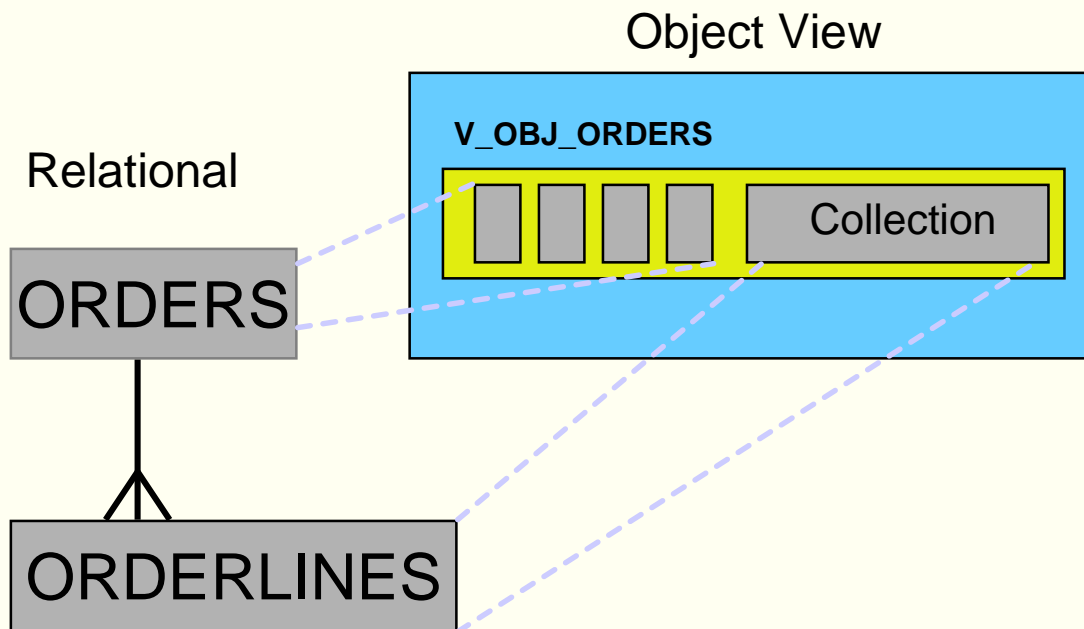
n XDK (XML Developer's kit)

- set of components, tools and utilities in Java, C, C++ and PL/SQL
 - XML parser, XSL processor, XML Schema processor
 - XML Class Generator, XML Transviewer Java Beans, ...
- JVM in the database kernel, Java code compiled to native code

n Useful for generating XML:

- XML SQL Utility: a set of Java classes that
 - pass an SQL query to the database and generate XML data (string representation or in-memory DOM tree for XSL processing)
 - write XML data to a database table (or tables by using object views)
 - inserting, updating or deleting values of the appropriate tables/columns
- XSQL Servlet: a Java servlet that
 - takes as its input an SQL query template file with XML syntax
 - transforms the SQL query results and any other XML data embedded in the query template file using a specified XSL stylesheet

XML SQL Utility



```
SELECT * FROM v_obj_orders
```

```
<?xml version="1.0" ?>
- <data>
- <table num="1">
  <id>1</id>
  <clt_id>2</clt_id>
  <order_dt>1999-06-25 23:28:43.0</order_dt>
  <total_amount>501000</total_amount>
  <status>X</status>
- <orderline>
  - <orderline_item itemno="1">
    <id>1</id>
    <ord_id>1</ord_id>
    <prd_id>1</prd_id>
    <qty>10</qty>
    <price>10000</price>
    <amount>100000</amount>
    <prd_name>Oracle 8i</prd_name>
    </orderline_item>
  + <orderline_item itemno="2">
  + <orderline_item itemno="3">
  + <orderline_item itemno="4">
  + <orderline_item itemno="5">
  + <orderline_item itemno="6">
  + <orderline_item itemno="7">
  + <orderline_item itemno="8">
    </orderline>
  </table>
- <table num="2">
  <id>2</id>
  <clt_id>3</clt_id>
  <order_dt>1999-06-28 23:28:44.0</order_dt>
  <total_amount>50500</total_amount>
  <status>X</status>
- <orderline>
  + <orderline_item itemno="1">
  + <orderline_item itemno="2">
```

XSQL Servlet

SQL query (using a template file)

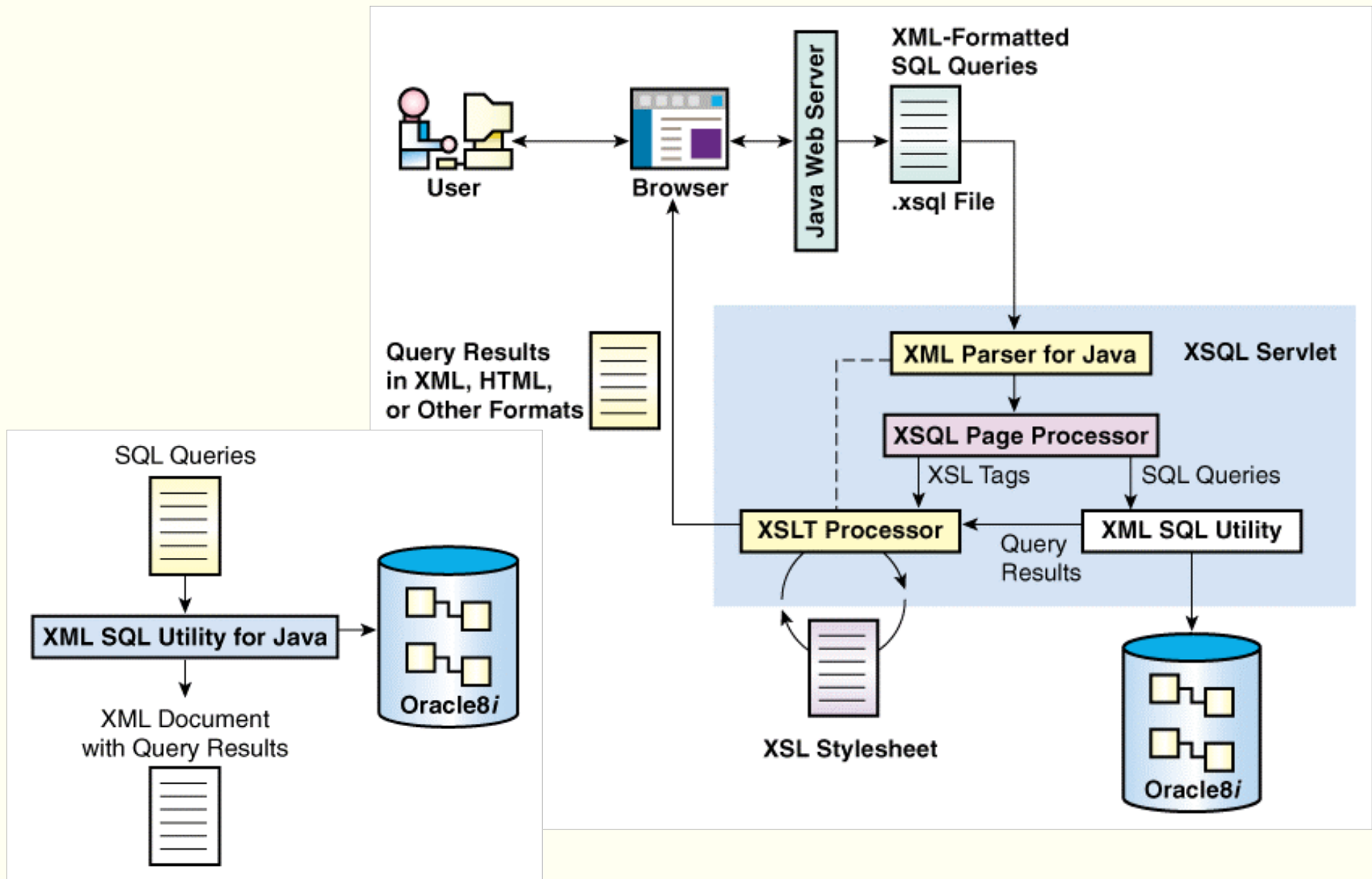
<http://www.mysite.com/mydatabase/employee.xsql?find=M&sort=EMPNO>

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="look-ie.xsl" media="msie 5.0" client="yes"?>
<?xml-stylesheet type="text/xsl" href="look-ns.xsl" media="ns 4.0"?>
<?xml-stylesheet type="text/xsl" href="look-any.xsl"?>
<xsql:query xmlns:xsql="urn:oracle-xsql"
  connection="demo" max-rows="10" skip-rows="5">
  SELECT * FROM EMP
  WHERE ENAME LIKE '%{@find}%' ORDER BY {@sort}
</xsql:query >
```

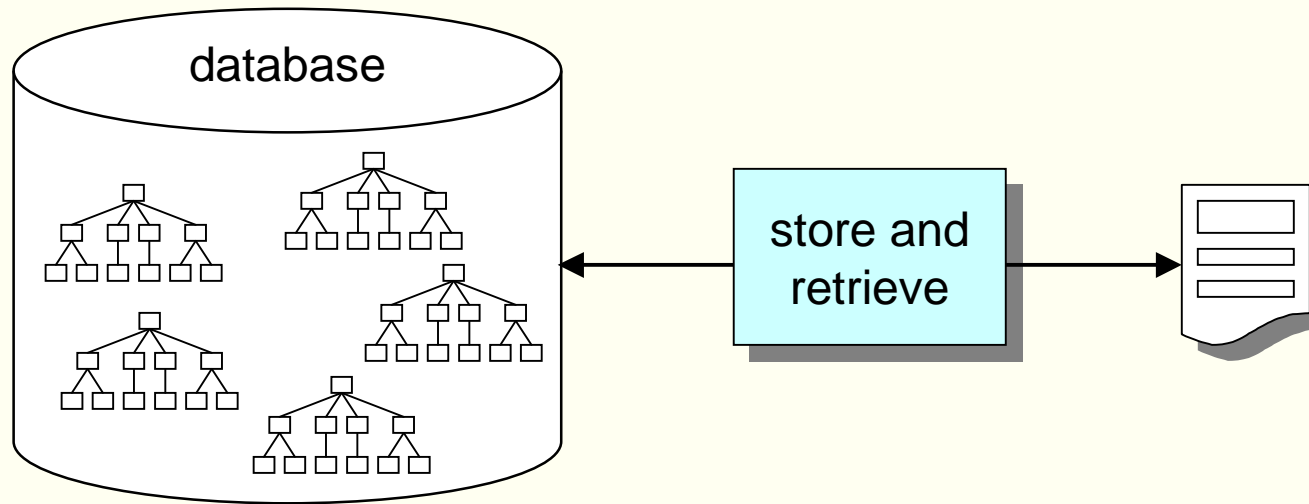
Query results

```
<?xml version="1.0"?>
<ROWSET>
  <ROW id="1">
    <EMPNO>7654</EMPNO>
    <ENAME>MARTIN</ENAME>
    <JOB>SALESMAN</JOB>
    <SALARY>1250</SALARY>
    <HIREDATE>1981-09-28 00:00:00.0</HIREDATE>
    <MGR>7698</MGR>
  </ROW>
  ...
</ROWSET>
```


Oracle 8i/9i XML features



XML storing database: native



- n** XML documents stored as parsed trees of nodes
 - contains DOM trees of nodes (or other optimized tree data model)
 - parts of XML documents can be retrieved using XPath query language
- n** XML is stored and retrieved natively
 - by sending XPath queries to the database using HTTP requests
 - by accessing the parsed trees using DOM program code (ASP, JSP)

XML storing databases: native

n Designed for native XML storage

- ü database kernel optimized for XML tree structures
- ü storage of XML and XSL, support for XPath as a query language
- ü act as data servers, accessible from application server or Web server
- û first generation XML database products
- û proprietary extensions for insert/update/delete

n Examples:

- Software AG *Tamino*
<http://www.softwareag.com/tamino/>
- eXcelon Corp. *eXcelon*
<http://www.exceloncorp.com/>
- X-Hive Corporation *X-Hive/DB*
<http://www.xhive.com/>

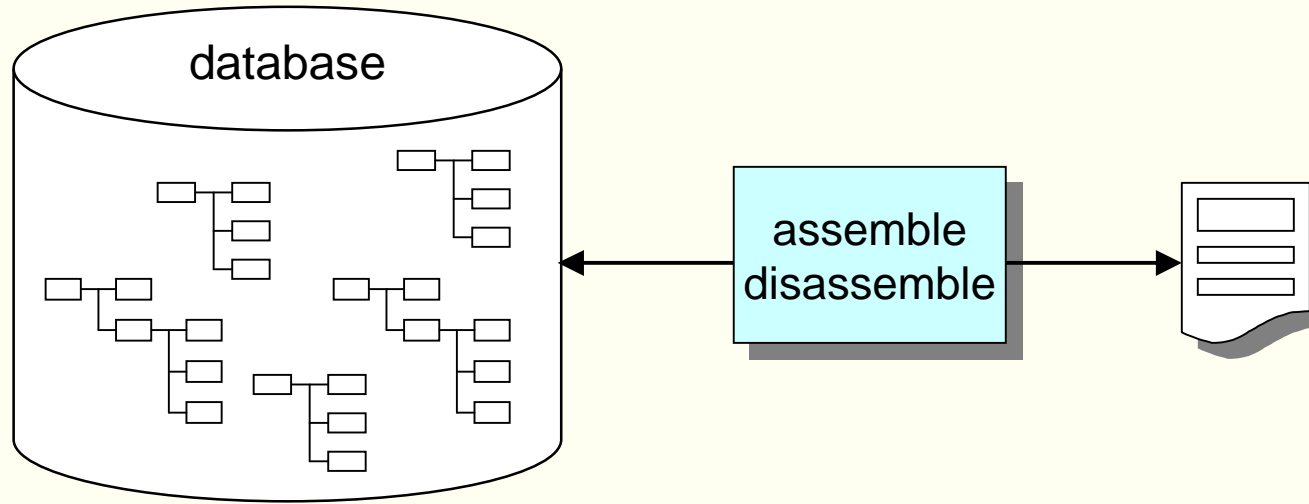
XML storing databases: native

The screenshot displays two windows. The top window, Microsoft Internet Explorer, shows the raw XML content of a file named 'memos.xml'. The XML contains three memo entries with varying priorities and subjects. The bottom window, eXcel Explorer, shows the same XML data in a tree view, with a corresponding table of tag values.

```
<?xml version="1.0" ?>
<!DOCTYPE memos (View Source for full doctype...)>
- <memos>
- <memo priority="Medium">
- <head>
  <from>hca@offis.be</from>
  <to>participant1@xmlseminar.itworks.be</to>
  <to>participant2@xmlseminar.itworks.be</to>
  <subj>What is XML?</subj>
</head>
- <body>
+ <para>
  <sign>Hans C. Arents - senior consultant</sign>
</body>
</memo>
- <memo priority="Low">
- <head>
  <from>paul@protext.be</from>
  <to>participant2@xmlseminar.itworks.be</to>
  <to>participant3@xmlseminar.itworks.be</to>
  <subj>XML seminar price</subj>
</head>
- <body>
+ <para>
  <sign>Paul Hermans - senior consultant</sign>
</body>
</memo>
- <memo priority="High">
- <head>
  <from>bmarchal@pineapplesoft.com</from>
  <to>participant1@xmlseminar.itworks.be</to>
  <to>participant3@xmlseminar.itworks.be</to>
  <subj>Why use XML for databases?</subj>
</head>
- <body>
+ <para>
  <sign>Benoit Marchal - consultant</sign>
</body>
</memo>
</memos>
```

Tag	Value
DOCTYPE	memos SYSTEM "memosxml.dtd"
memos	
memo	
priority	Medium
head	
from	hca@offis.be
to	participant1@xmlseminar.itworks.be
to	participant2@xmlseminar.itworks.be
subj	What is XML?
body	
para	
sign	Hans C. Arents - senior consultant
memo	
priority	High
head	
from	paul@protext.be
to	participant2@xmlseminar.itworks.be
to	participant3@xmlseminar.itworks.be
subj	XML seminar price
body	
para	
sign	Paul Hermans - senior consultant
memo	
priority	High
head	
from	bmarchal@pineapplesoft.com
to	participant1@xmlseminar.itworks.be
to	participant3@xmlseminar.itworks.be
subj	Why use XML for databases?
body	
para	
TEXT	Database developers are likely to be involved with XML for several reasons. You might develop applications that query databases and format...
bold	EDI
TEXT). You might also have to store XML documents in a database. Today's object-relational DBMS products from Oracle, Informix, and IBM supp...
bold	CODASYL
TEXT) model databases have no problem with hierarchies or trees. Those are legacy technologies, but object and object-relational databases also ...
sign	Benoit Marchal - consultant

XML storing database: fragments



n XML documents stored as fragments

- contains pieces of XML documents as distinct objects
- XML fragments can be retrieved using proprietary query language

n XML is stored and retrieved as objects

- by assembling / disassembling the XML document
- fragments have metadata: access rights, version, language, ...

XML storing databases: fragments

- n Layer on top of (object-oriented) database
 - ü decomposition granularity to be defined by the user
 - ü reuse, versioning and variants of XML document fragments
 - ü sophisticated management of document composition / linking
 - û designed for "classic" XML document management
 - û Web publishing support is an add-on afterthought

n Examples:

- Chrystal Software *Eclipse*
<http://www.chrystal.com/>
- Poet *Content Management Suite*
<http://www.poet.com/>
- Documentum *Documentum 4i WCM Edition*
<http://www.documentum.com/>

XML storing databases: fragments

The screenshot shows the POET Content Client interface. The main window displays a hierarchical tree structure of XML objects. The selected object is <SECT3> Change a User's Password or Real Name. The right pane shows the selected object's content, which is an XML fragment. The bottom pane shows the XML content of the selected object.

POET Content Client - [techdocs:1]

File Query View Window Help

New Db Open Db Checkout Checkin Undo Change Folder Edition Delete Up

<BOOK> POET Content Management Suite - Administration Utilities

- <CHAPTER> Administration Utilities
 - <SECT1> POET ADMINISTRATOR
 - <SECT2> User Authorization
 - <SECT2> Database Reorganization
 - <CHAPTER> User Authorization
 - <SECT1> Enabling User Authorization
 - <SECT2> Creating and Administering Users
 - <SECT3> Add a New Database User
 - <SECT3> Change a User's Password or Real Name
 - <SECT3> Delete a User
 - <SECT2> Creating and Administering Groups
 - <SECT1> Setting Rights for Groups and Users
 - <SECT1> An Example
 - <CHAPTER> Verity Full-text Indexing

Object

- <SECT3> Add a New Database User
- <SECT3> Change a User's Password or Real Name
- <SECT3> Delete a User

Log Query View Browse

```
<SECT3>
<TITLE>Change a User's Password or Real Name</TITLE>
<PARA>To change an existing user's password or real name, select <GUIMENU>
User</GUIMENU> | <GUIMENUITEM>User Administration</GUIMENUITEM> from the menu
of the POET Administrator. The <GUIMENUITEM>User Administration</GUIMENUITEM> dialog
```

Ready

XML-aware indexing and search

n XML generating databases:

- Microsoft *SQL Server 2000*: no XML-aware indexing or search
 - but upcoming *SharePoint Portal Server* will index XML files
- Oracle *8i*: tag-aware searching using *interMedia Text Search*
 - e.g. find “word WITHIN title” where “title” is an XML element
- Oracle *9i*: search using XPath queries for XMLType table columns
 - e.g.

```
SELECT extract(m.msgVal, "/po/cust/custname") FROM
msgs m WHERE existsNode(m.msgVal, "/po[@id='a123']") > 0;
```

n XML storing databases: native

- eXcelon Corp. *eXcelon*
 - indexing hints, search limited to XPath queries
- Software AG *Tamino*
 - indexing hints, search limited to XPath queries (with proprietary extensions)
- X-Hive Corporation *X-Hive/DB*
 - context conditioned indexing, search limited to XPath queries

XML search engines

n What's needed for searching XML content?

- XML is *text*
 - à full-text indexing of element/attribute names/values
- XML is *extensible*
 - à dealing with a large number of different well-formed/valid documents
- XML has a *hierarchical structure*
 - à keeping track of where content is found in the document tree
 - à controlling the structure and content of the query result tree

n Problems:

- an XML query algebra has only just been defined
 - what's the optimal way of querying an XML document?
 - so what's the optimal way of indexing an XML document?
- XPath: good for specifying structure searches, bad for content searches
 - wildcard, proximity, ... full-text search, data-type aware search?

XML search engines

n XYZFind Corporation *XYZFind* <http://www.xyzfind.com/>

- indexing of natively stored well-formed XML documents, no binary
 - common set of b-tree indices for all stored XML documents
 - element/attribute names/values (strings, numbers) and document structure
- search: structure-aware
 - full-text, boolean/numeric range search on element/attribute names/values
 - uses *XYZQL*, a proprietary template-based XML syntax for queries/results

n IXIA *TEXTML Server* <http://www.textmlserver.com/>

- indexing of natively stored parsed well-formed XML documents, binary
 - index definition document defines custom indices: full-text, date, list
 - dynamic background re-indexing as documents are added or deleted
- search: not structure-aware
 - full-text, boolean/proximity search on element/attribute names/values
 - uses a proprietary XML syntax for queries/results

Publishing frameworks

n What is it?

- a complete approach to dynamic content generation from XML

n What does it offer?

- clean separation of content storage and page publishing logic
- automates the whole process of
 - generating XML content from different data sources
 - styled at predetermined times (static content generation)
 - or styled dynamically at runtime (dynamic content generation)

n Benefits:

- doing away with hand-made ASP or JSP pages
- handling the ugly details of stylesheet caching, load balancing, ...

n Drawbacks:

- does not include static content management (still do-it-yourself)

Publishing frameworks

n Apache Cocoon

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



- part of the open source Apache XML projects
- design goal: pipelining of XML content
 - generator (1): generate XML content
 - from SQL, LDAP, ... queries
 - using XSP (**X**ML **S**erver **P**age) pages
 - transformer (0...n): process XML content
 - modifying content / structure
 - serializer (1): format XML into a stream representation
 - to HTML, WML, ...
- for Web sites of medium to high complexity
 - precompiled dynamic page generation
 - caching of dynamically generated pages
- can be used together with Apache *FOP* (XSL FO to PDF formatter)

Web site management tools

n What is it?

- a complete solution for content management and site publishing

n What does it offer?

- content management
 - XML content storage, media asset management, content syndication, ...
- site publishing
 - static/dynamic page generation, site previewing/staging, site editions, ...
- collaboration
 - joint authoring, role-based managing, task planning, project workflow, ...

n Benefits:

- turning Web site management from an ad-hoc into a controlled process

n Drawbacks:

- the process model of the tool has to fit your own way of working

Web site management tools

n Interwoven *TeamSite*

<http://www.interwoven.com/products/teamsite/>

- XML-based page templating, content replication, workflow, ...
- uses XML throughout (configuration files, deployment messaging, ...)

n Mediasurface *Mediasurface*

<http://www.mediasurface.com/>

- non-XML based, but now integrating XML content
- uses Kinecta technology for ICE-based content syndication

n Tridion *DialogServer*

<http://www.tridion.nl/>

- uses XML for "transactive content management"
- Unicode support for automated content localisation
- multimedia content management using XML metadata

Web site management tools

The screenshot shows a Microsoft Internet Explorer browser window displaying the Tridion Dialog Server 4.0 interface. The browser title is "http://tdserver/ - Microsoft Internet Explorer". The address bar shows "http://tdserver/". The main content area is divided into two panes. The left pane shows a tree view of the site structure, with "Contents of Folder 28 March 2000" selected. The right pane shows a form for editing a flight entry. The form fields are as follows:

Title	A Moscow Experience
DTD	Discount Flight
Place of Departure	Amsterdam
Place of Arrival	Moscow
Discount price of the flight	f1 599,-
Normal Price	f1 1000,-
Conditions	<input type="checkbox"/> Children younger than 12: 33% discount <input type="checkbox"/> Children younger than 2: 90% discount <input checked="" type="checkbox"/> No changes allowed <input type="checkbox"/> No cancellations allowed <input type="checkbox"/> Return trip within a week <input checked="" type="checkbox"/> Return Trip within a month
Date of the flight	28 02 2000

The taskbar at the bottom shows the Start button and several open applications, including Test..., Visu..., FW..., Inbo..., Cont..., Micr..., UN..., W Micr..., Expl..., and Inte... The system clock shows 5:05 PM.

Web site management tools

n Allaire *Spectra*

<http://www.allaire.com/products/spectra/>

- XML stored transparently in ContentObject database
- uses WDDX for content syndication and system integration

n eBusiness Technologies *engenda*

<http://www.ebt.com/products/engenda/>

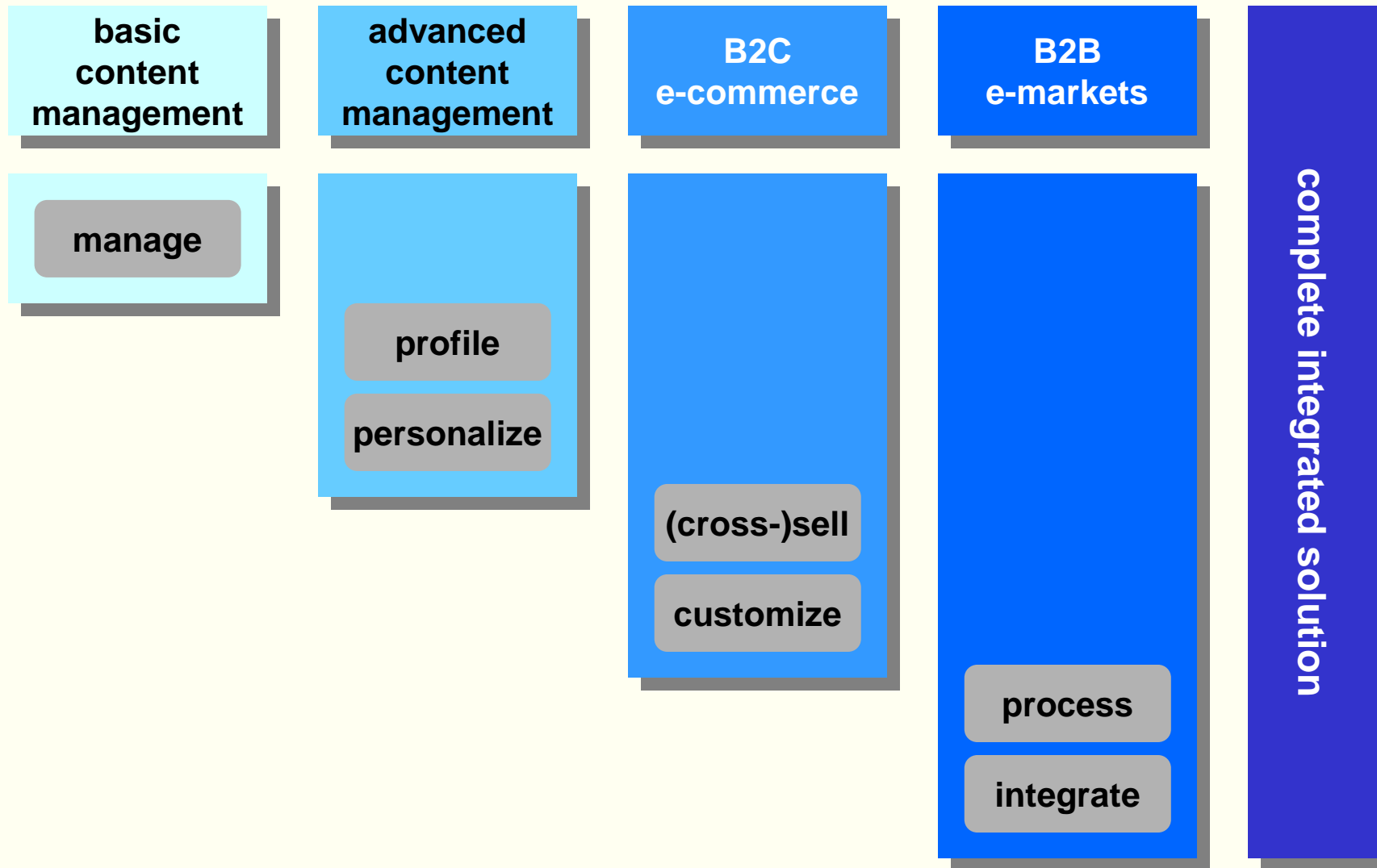
- forms-based authoring of XML documents
- automated Word-to-XML conversion, XML-based workflow

n Similar products:

- Eprise *Participant Server* <http://www.eprise.com/>
- NetObjects *Fusion* <http://www.netobjects.com/>
- Relevant Tools *Relevant* <http://www.relevanttools.com/>
- Rhythmyx *Content Manager* <http://www.percussion.com/>

n The big (and expensive) boys: Vignette, BroadVision, ATG

Web site management tools: the future



Conclusions

- n The challenges of Web site content management can be addressed effectively using the XML family of standards
 - XML for rich modeling of content, XSL for flexible rendering of content
- n You have a choice of:
 - do-it-yourself solutions
 - content management and page publishing as complete *as you build it*
 - Web publishing frameworks
 - rich support for page publishing, poor support for content management
 - Web site content management tools
 - complete content management and site publishing solution (and more)
- n Caveats:
 - XML standards (and the tools using them) never stop changing
 - XML support is broad, but first generation and sometimes rudimentary