Welcome to the DB2, XML and Web Services e-learning workshop

Part 1: Intro to XML and DB2 XML Extender

IBM Software Group

Overview of education series

- Workshop 1 : DB2, XML and Web Services
  - Part 1 : Intro to XML and DB2 XML Extender
  - Part 2 : DB2 and Web Services

- Workshop 2 : Using the DB2 XML Extender
  - Part 1 : Mapping XML to DB2 databases, Building DADs
  - Part 2: Administration, Storing and Retrieving data

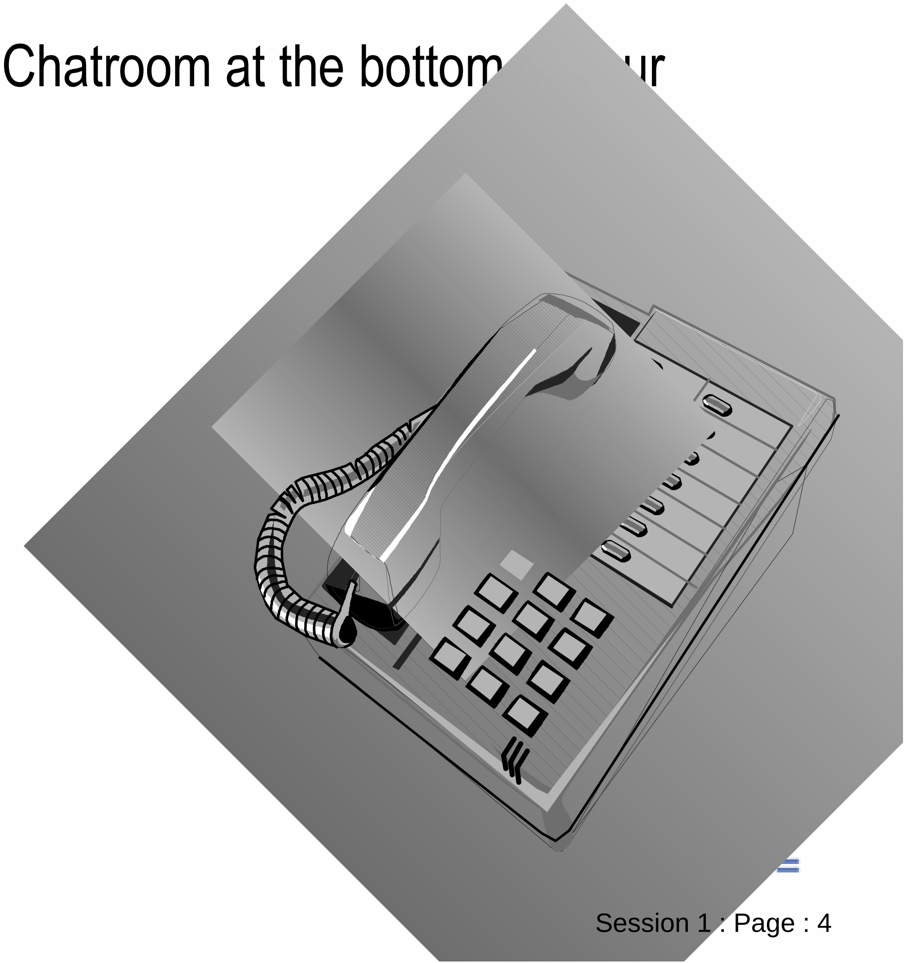
Product Introduction Center - Who are we?

- Run Introduction Programs to accelerate Customer Acceptance and increase Marketplace penetration of selected new Products, Systems and Solutions by:-
  - Enhancing their Quality and Marketability
  - Managing Customer Early Support Programs (ESP)
  - **Transferring early product skills and customer experience to the field**
  - Establishing references for our Products, Systems and Solutions

- Product Introduction Center Objectives include:
  - Improve customer enablement
  - Reduce risk and improve customer satisfaction
  - **Enhance field readiness**
  - Increase sales effectiveness

Classroom Etiquette - Telephone Conferencing

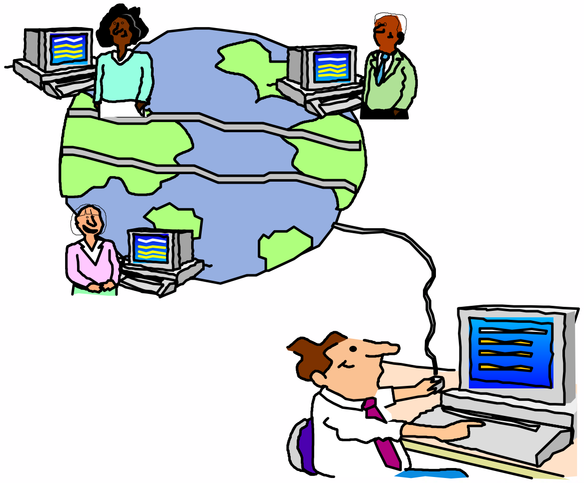
- To improve audio quality by reducing background noise - you will be placed on MUTE
- We will take questions in the Sametime Chatroom at the bottom [illegible] ur screen



Web Conferencing - Sametime Instant Messaging

- Using Sametime Instant Messaging - you can send a text message to a colleague, to the moderator or to the entire group
  - Type your message in the window at the bottom of the screen

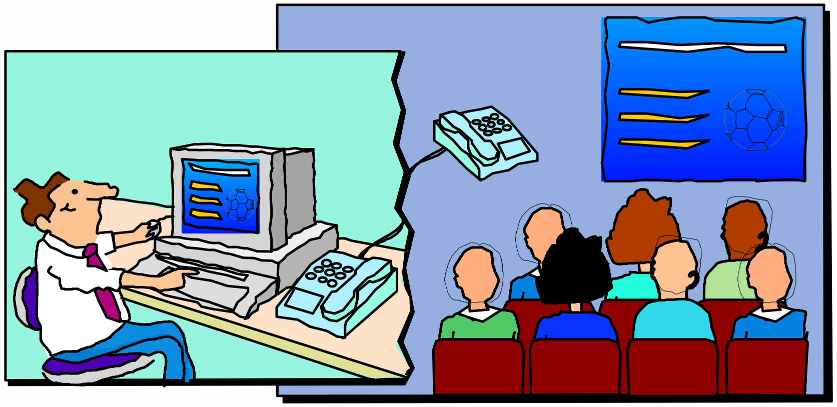
- We can't see your faces for visual clues - provide us with feedback to assist you.
  - You can ask us to speak louder / slower ....
  - You can ask us to give more details for a specific section
  - Ask a question



What happens if the Web Conferencing Technology fails during a presentation?

- □ Remain on the phone

- □ The foils have been made available for download
  - ▷ We have noticed that the Sametime Servers are not 100% reliable.
  - ▷ We will continue the lecture using the foils, returning to Web Conferencing when it becomes available.



Product Introduction Center .... providing timely, FREE education

- This eLearning education is FREE

- The Product Introduction Center's mission is to "accelerate the acceptance of new products", one of the ways this is achieved is by providing education to the field.

- Delivering education remotely (eLearning), enables us to reach a wider audience (both numerically and geographically).

- In the future, we would like to be able to deliver more eLearning education so your feedback is essential.

eLearning Workshop Evaluation (help us to help you!)

By the time you complete this eLearning workshop, we would like you to be "delighted".

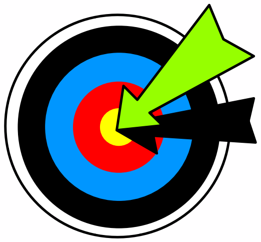
We hope you will get value for this education, increasing your skill levels in this area.
Please ask questions / ask for clarification as required.

We will ask for your evaluation - your evaluation is very important to us:

- ▷ Review Quality / Value of PIC Workshops
- ▷ Added to the PIC **N**et **S**atisfaction **I**ndex (NSI)
- ▷ Our Target is 85 (or higher!)

Your input to the PIC NSI Calculation

1(100)	**= VERY SATISFIED**
2 (75)	**= SATISFIED**
3 (50)	**= NEITHER SATISFIED NOR DISSATISFIED**
4 (25)	**= DISSATISFIED**
5 (0)	**= VERY DISSATISFIED**



PLEASE complete your Evaluation form which has been sent to your Notes Id

The Team

Susan Malaika
SVL

Mark Cathcart
IBM UK

Mark Colan
IBM US

Joan Haggarty
Toronto

Ian Cook
PIC

Sarah Ellis
PIC

Thanks also to:
Bruce Benfield
Alan Carpenter
Tina Gleisner
Grant Hutchison
Dirk Wollschied
XML Development team
And everyone else who helped!

Agenda

- Opening session : XML and the IBM Software Strategy for e-Business
  - By Mark Cathcart, IBM Distinguished Engineer

- Introduction to XML
  - Why XML? XML vs HTML, What is XML?
  - XML Terminology and concepts
  - XML Standards

- Overview of the DB2 XML Extender
  - Including XML Futures and Strategy
  - By Susan Malaika, Silicon Valley Lab

DB2 Data Management Software

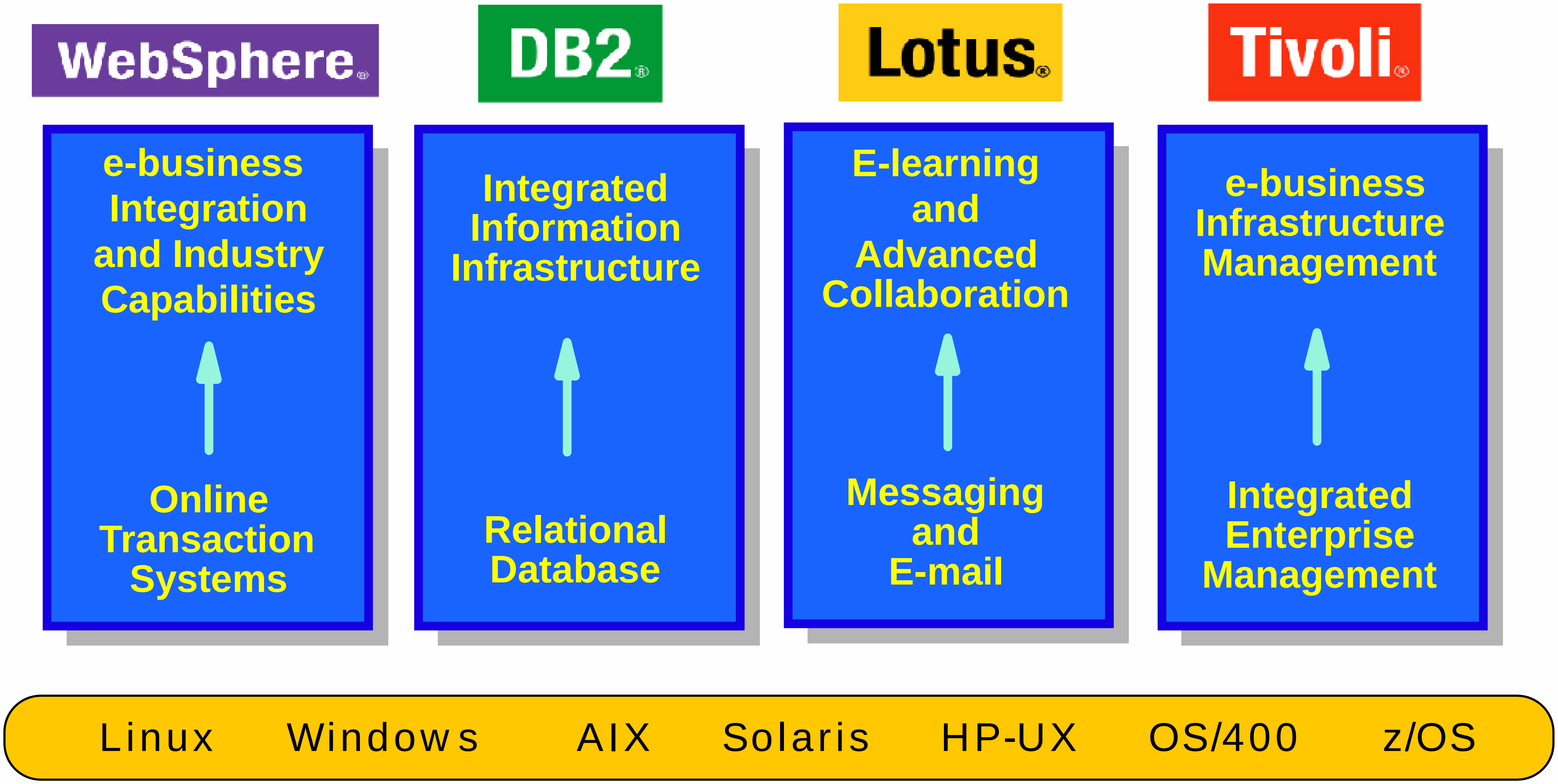
e business software

XML and the IBM Software Strategy for e-business

by Mark Cathcart, IBM Distinguished Engineer

IBM Software Group

Foundation Technologies to Meet Key IT Challenges



WebSphere

e-business Integration and Industry Capabilities

↑

Online Transaction Systems

DB2

Integrated Information Infrastructure

↑

Relational Database

Lotus

E-learning and Advanced Collaboration

↑

Messaging and E-mail

Tivoli

e-business Infrastructure Management

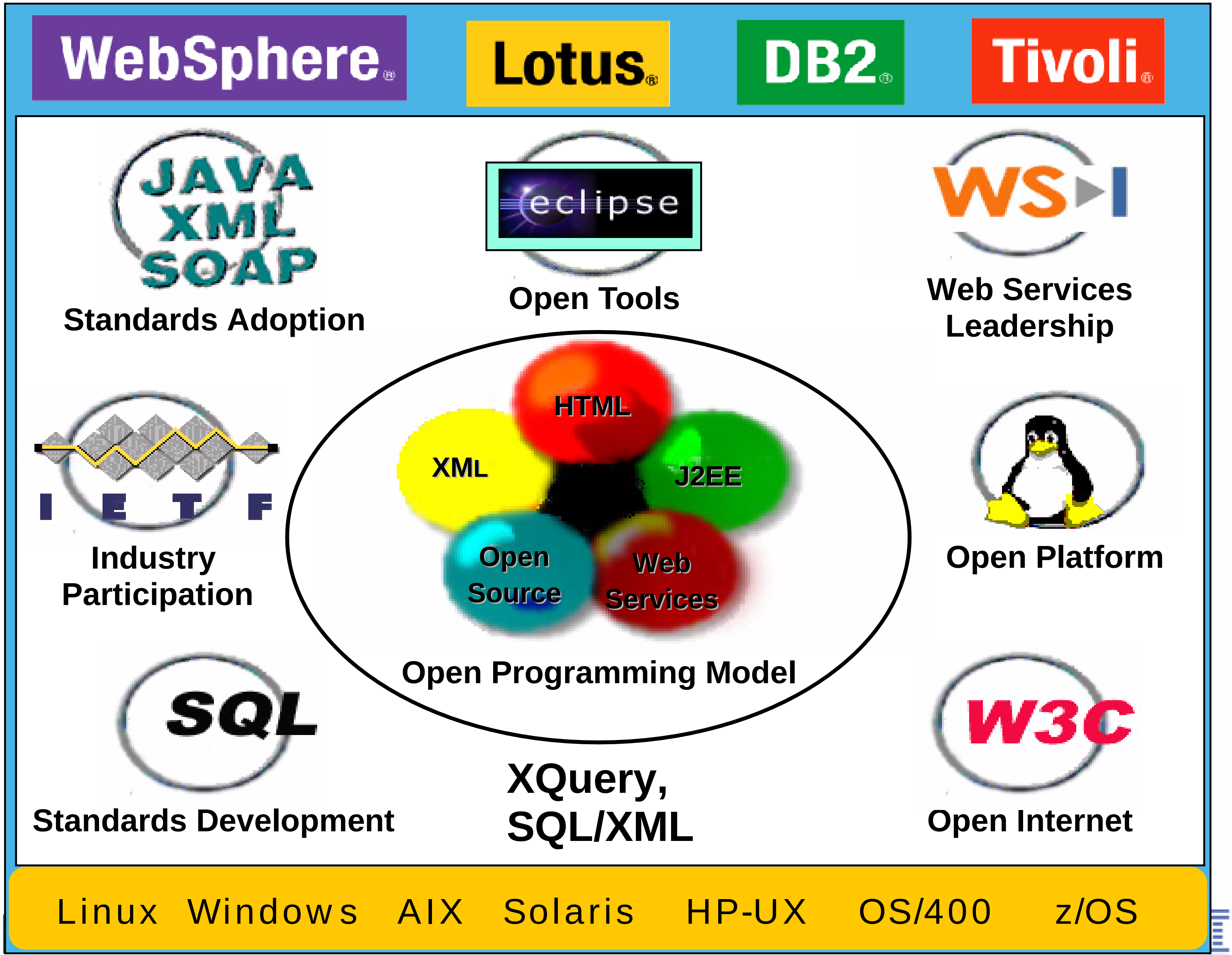
↑

Integrated Enterprise Management

Linux Windows AIX Solaris HP-UX OS/400 z/OS

- □ **Standards Based**
- □ **Scalable**
- □ **Reliable**
- □ **Cross-platform**

Open Standards Enable Integration



Standards Adoption

Open Tools

Web Services Leadership

Industry Participation

Open Programming Model

Open Platform

Standards Development

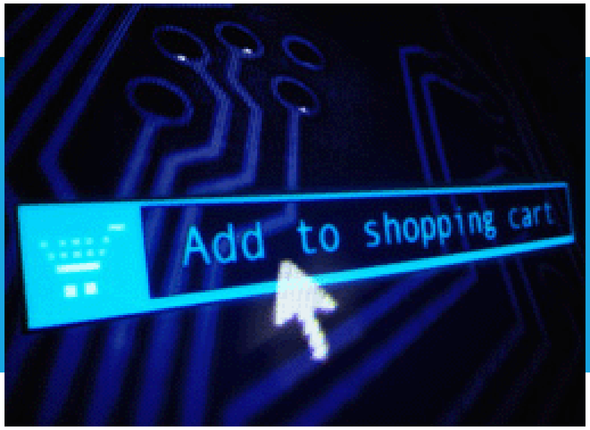
XQuery, SQL/XML

Open Internet

Linux Windows AIX Solaris HP-UX OS/400 z/OS

The Rules Continue to Change

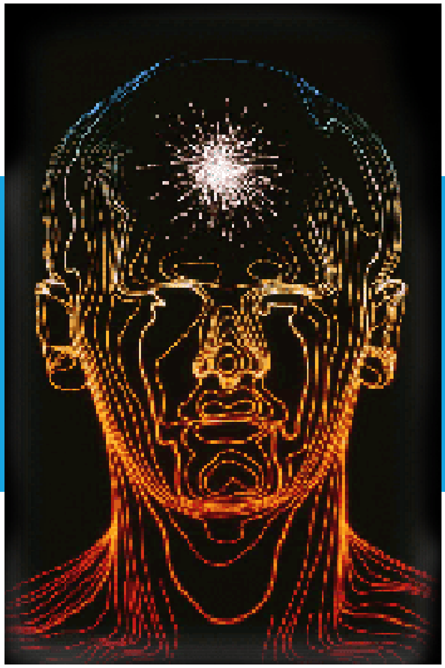
e-business Innovations



Business Process Integration

Web Services

Technology Innovations



Linux

Life Sciences

Infrastructure Innovations

Autonomic Computing

Pervasive Computing

Introduction to XML

(eXtensible Markup Language)

XML completes the puzzle!

XML brings data to the web, completing the necessary components for 90%+ of e-business applications

Universal Communication	Internet
Universal User Interface	Browsers
Universal Programming	Java
and now...	
Universal Information	XML

What is XML?

- □ Designed to easily represent data in a very portable manner
  - ▷ **Key message of Java = Code portability, XML = Data Portability**
  - ▷ Standard way of sharing structured and semi-structured data
- □ A text based tag language
  - ▷ Similar in style to HTML
  - ▷ Syntax standard with flexibility to extend definitions (user-defined tags)
- □ A language for defining tag languages
  - ▷ external or internal
  - ▷ private or public
  - ▷ industry specific or custom-built
- □ A W3C standard :
  - ▷ Worldwide Web Consortium http://www.w3.org/XML/

W3C WORLD WIDE WEB consortium

Leading the Web to its Full Potential...

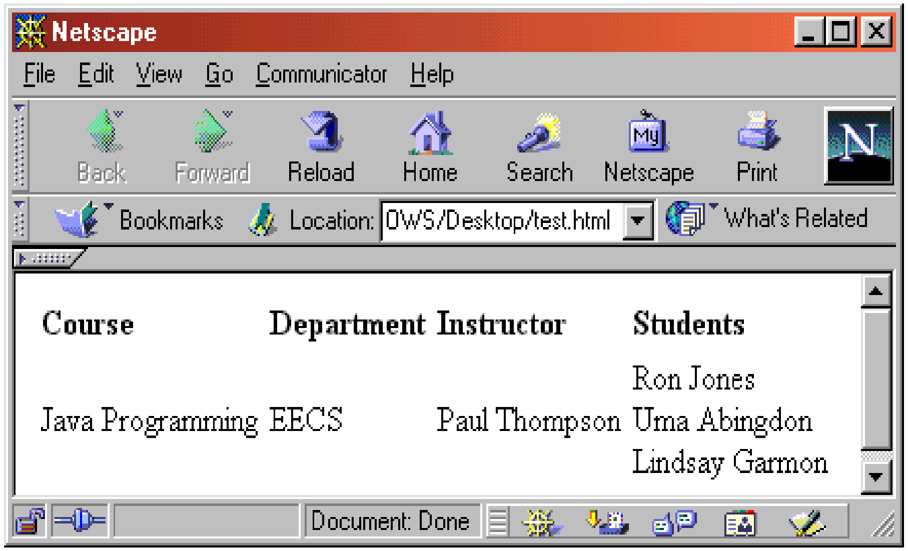
Characteristics of mark-up languages

- Applies some sort of form to particular data
- Stores information about the data within the data
- Data encapsulated within opening and closing tags
- HTML example
  - <title> Harry Potter and the Philosopher's Stone </title>
- SGML is orginal Markup language and much more complex than XML

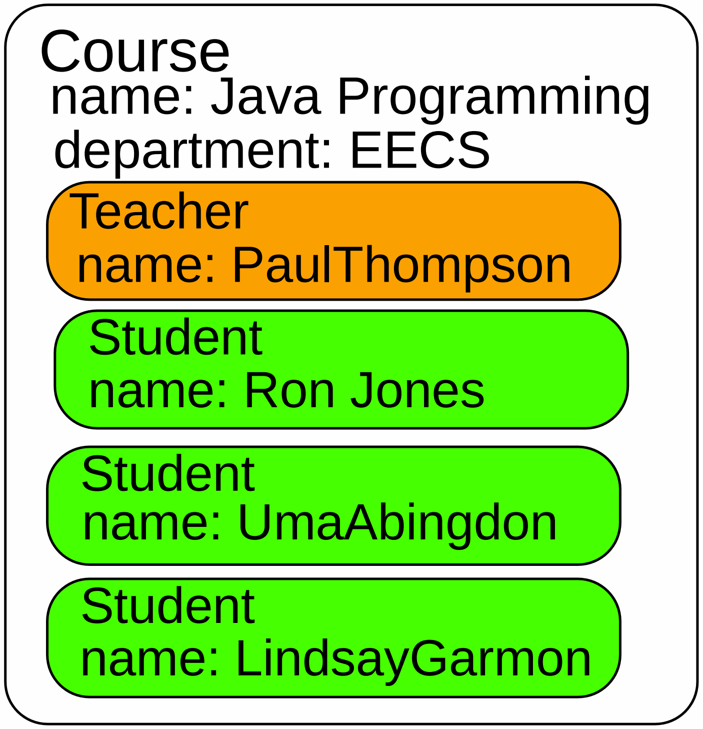
HTML vs. XML

- □ HTML is about presentation and browsing

- □ XML is about structured information interchange


Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print
Bookmarks
Location: OWS/Desktop/test.html
What's Related
Document: Done

Course	Department	Instructor	Students
Java Programming	EECS	Paul Thompson	Ron Jones Uma Abingdon Lindsay Garmon



Course
name: Java Programming
department: EECS

Teacher
name: PaulThompson

Student
name: Ron Jones

Student
name: UmaAbingdon

Student
name: LindsayGarmon

HTML vs. XML

```
<HTML>
<BODY>
<TABLE>
   <TR>
      <TD><b>Course</b></TD>
      <TD><b>Department</b></TD>
      <TD><b>Instructor</b></TD>
      <TD><b>Students</b></TD>
   </TR>
   <TR>
      <TD>Java Programming</TD>
      <TD>EECS</TD>
      <TD>Paul Thompson</TD>
      <TD>Ron Jones<BR>
             Uma Abingdon<BR>
             Lindsay Garmon</TD>
   </TR>
</TABLE>
</BODY>
</HTML>
```

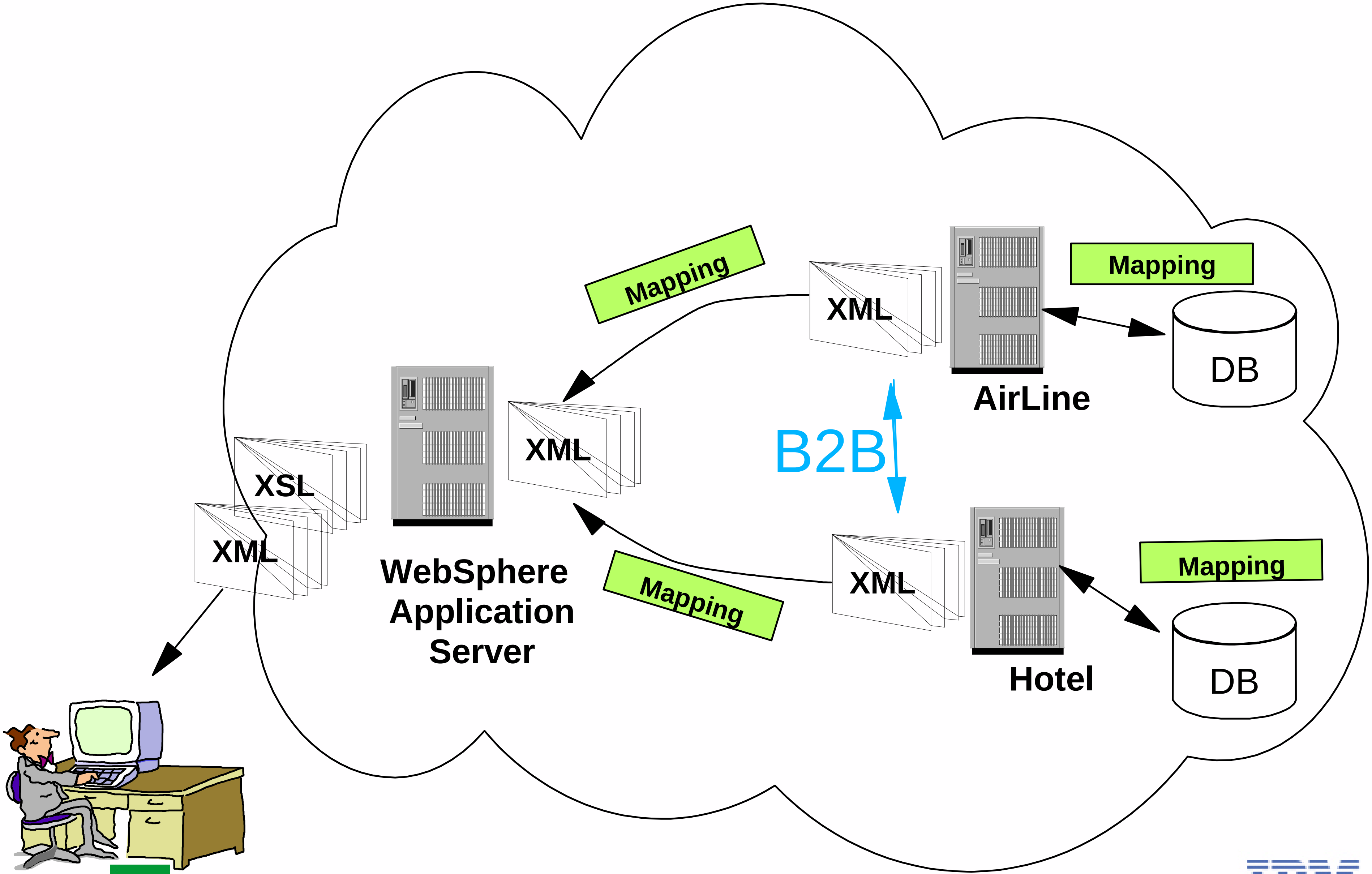
```
<?xml version="1.0"?>
<Course>
      <Name>Java Programming</Name>
      <Department>EECS</Department>
      <Teacher>
         <Name>Paul Thompson</Name>
      </Teacher>
      <Student>
         <Name>Ron Jones</Name>
      </Student>
      <Student>
         <Name>Uma Abingdon</Name>
      </Student>
      <Student>
         <Name>Lindsay Garmon</Name>
      </Student>
</Course>
```

HTML vs XML

- HTML provides a presentation solution:
  - For display to humans
  - It is a fixed tagset corresponding to an SGML DTD (Standard Generalized Markup Language)

- XML is a notation which allows users to define their own tags which:
  - Describe data
  - Describe the logical structure of data
  - Defines the vocabulary
  - Does not describe how data is presented
  - XML is a restricted form of SGML with some modifications

Where is XML?


Mapping
XML
Mapping
DB
AirLine
B2B
XSL
XML
WebSphere Application Server
XML
Mapping
XML
Mapping
Hotel
DB

How can XML be used?

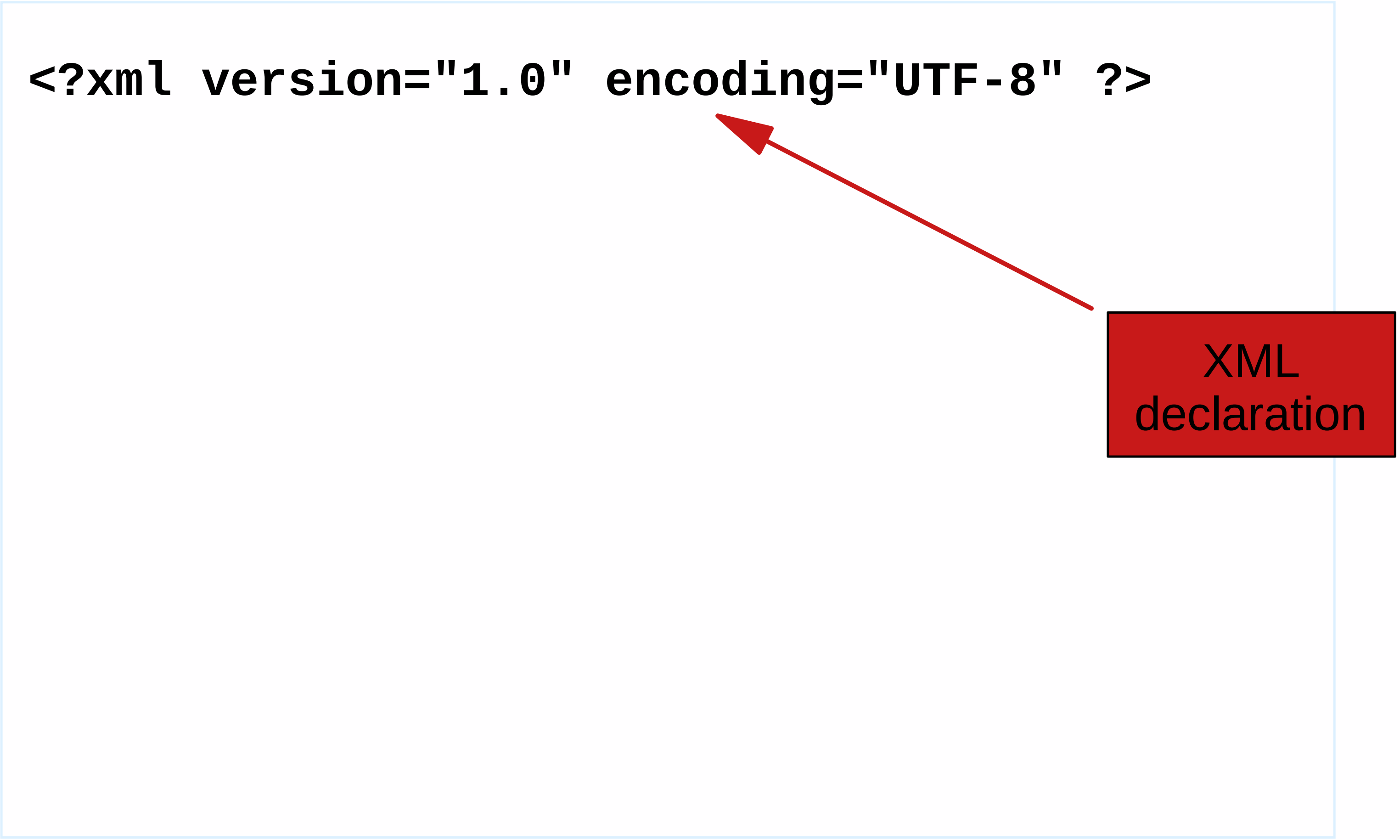
- Information interchange
  - B2B, B2C
- Integrate data from multiple sources, from different operating systems etc
- Deliver customized data to different types of client devices
  - separation of data and presentation increases flexibility
  - can use different XML style sheets (XSL)
- Enable automatic application to application interchange

Worldwide web ====> Worldwide database!

Key Benefits of XML

- Vendor and Platform Independence
- Easily model data at any level of complexity
- Extremely Extensible
  - Define new tags as needed
- Easily validate data to check for structural correctness
- Easily transformed for publishing in different formats
  - e.g. HTML, WML, CompactHTML, etc.

Example XML File - Personnel data

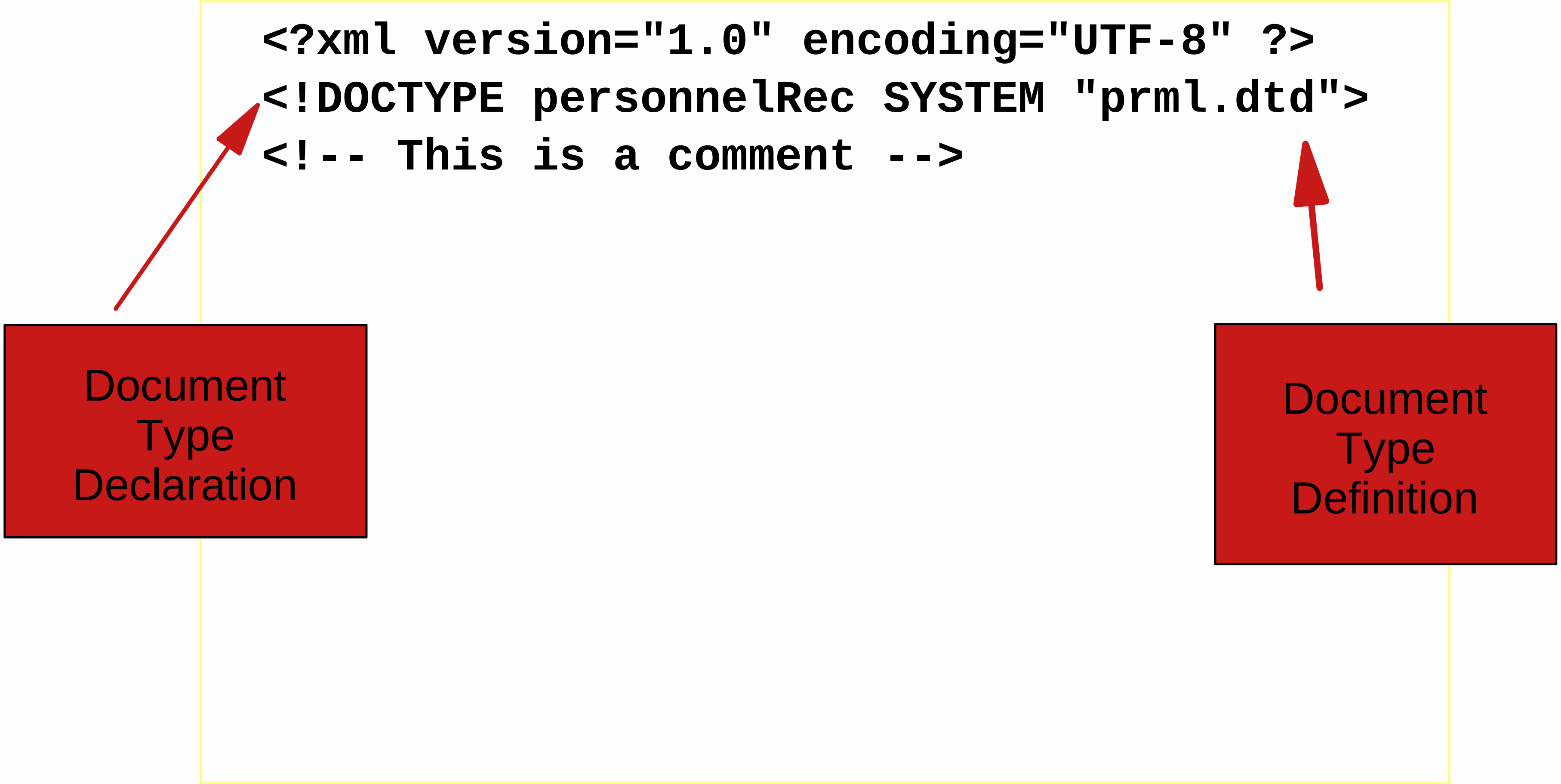


```
<?xml version="1.0" encoding="UTF-8" ?>
```

XML declaration

Encoding declaration default is UTF-8

Example XML File - Personnel data



```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE personnelRec SYSTEM "prml.dtd">
<!-- This is a comment -->
```

Document Type Declaration

Document Type Definition

Example XML File - Personnel data

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE personnelRec SYSTEM "prml.dtd">
<!-- This is a comment -->
<personnelRec>
  <person                  >
    <name>
      <family>Wallace</family>
      <given>Bob</given>
    </name>
    <email>bwallace@megacorp.com</email>
    <dept>&d1</dept>
  </person>
</personnelRec>
```

Elements

Example XML File - Personnel data

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE personnelRec SYSTEM "prml.dtd">
<!-- This is a comment -->
<personnelRec>
  <person salary="26350.00" band="D">
    <name>
      <family>Wallace</family>
      <given>Bob</given>
    </name>
    <email>bwallace@megacorp.com</email>
    <dept>&d1</dept>
  </person>
</personnelRec>
```

Attribute

Example XML File - Personnel data

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE personnelRec SYSTEM "prml.dtd">
<!-- This is a comment -->
<personnelRec>
  <person salary="26350.00" band="D">
    <name>
      <family>Wallace</family>
      <given>Bob</given>
    </name>
    <email>bwallace@megacorp.com</email>
    <dept>&d1</dept>
  </person>
</personnelRec>
```

Entity reference

XML Components

- Element (eg person, name)
  - Basic building blocks, delimited by angle brackets
  - Empty elements have a modified syntax
    - eg <optionaldata/> or <optionaldata></optionaldata>
- Attribute (eg salary, band)
  - Name-value pairs that occur inside start-tags after the element name
  - All attribute values must be in quotes, eg salary="26350.00"
- Entities
  - A shortcut used to represent complex strings or symbols that would otherwise by impossible, difficult or repetitive to include by hand
  - eg file content, special characters or some text referred to regularly eg copyright statement

Some XML rules

A small set of rules define the basic syntax that any XML document must follow (to be "well-formed")

Required first line   <?xml version="1.0"?>

Syntax of tags   <tag>data</tag>   (closing tag is mandatory)

Legal nesting of tags

```
<employee>
   <name>Fred Bloggs</name>
   <id>X04913</id>
</employee>
```

(XML document must be enclosed in one set of tags)

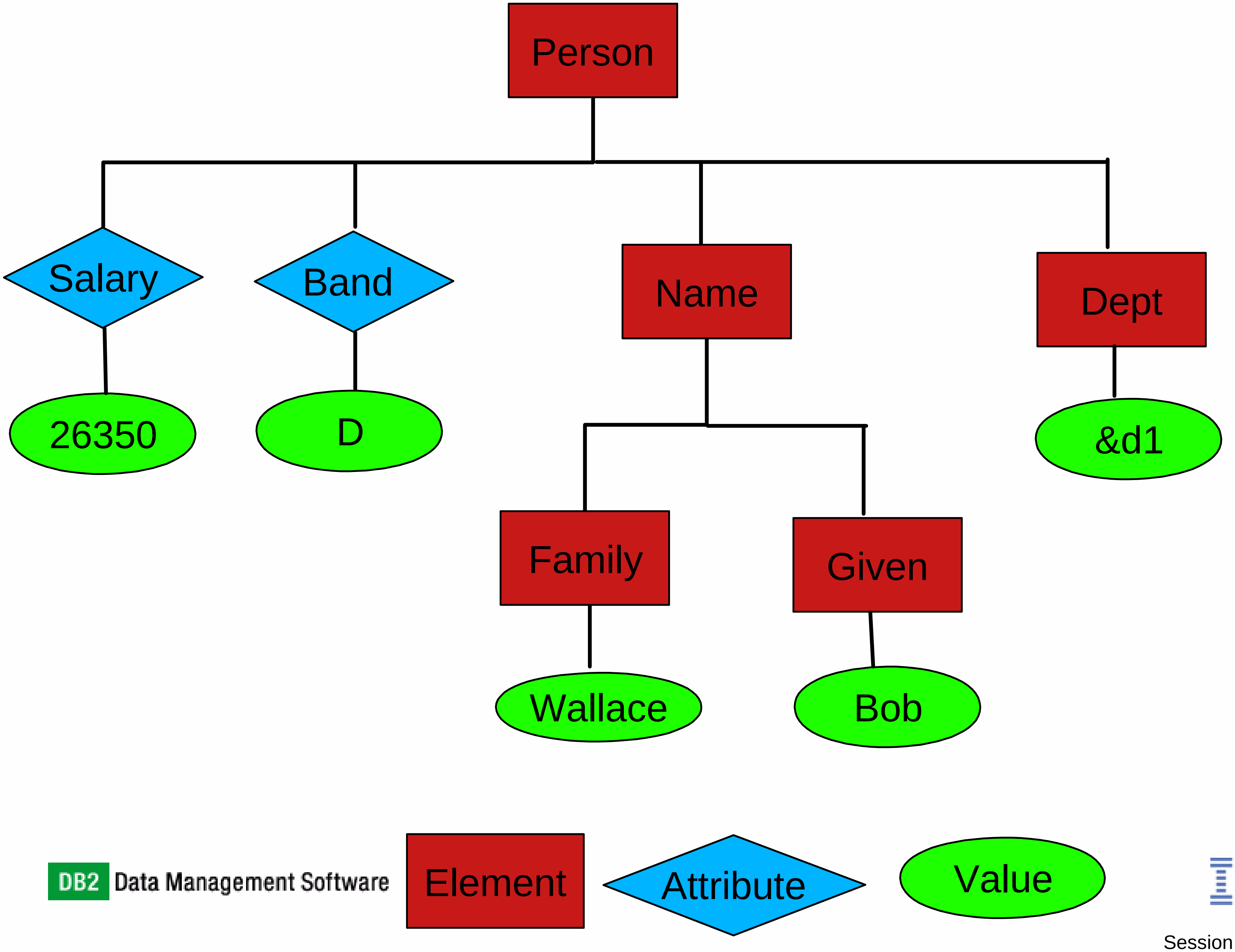
A "valid" XML document is well-formed AND complies to specific DTD or XML Schema (Checked by XML parsers)

Document Type Definition (DTD)

- Defines the structure of an XML document
  - Defines how elements relate to one another within the documents tree structure
  - Defines the tags that can or must appear
  - How often the tags can appear
  - How often the tags can be nested in the XML hierarchy
  - Allowable, required or default attributes
- XML references DTD in <!DOCTYPE> XML prolog
  - Allows a validating parser to detect deviations from a vocabulary
  - Use is optional - can parse a well-formed XML document without a DTD
- DTDs can be shared across organisations

Defines the rules of the language we create

Example : XML Hierarchy



Example DTD

optional first line regarding encoding

```
<?xml encoding="UTF-8"?>
```

Example DTD

```
<?xml encoding="UTF-8"?>
<!ENTITY d1 "salaries department">
<!ENTITY d2 "policies department">
```

Text substitution for &d1 in XML

(Supported by DB2 XML extender when validation is on)

Example DTD

```
<?xml encoding="UTF-8"?>
<!ENTITY d1 "salaries department">
<!ENTITY d2 "policies department">
<!ELEMENT personnelRec (person)+>
<!ELEMENT person (name,email*)>
```

Indicates one or more elements present (? denotes optional)

Complex element

Simple element

```
<!ELEMENT name (family, given)>
<!ELEMENT family (#PCDATA)>
<!ELEMENT given (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

Parsable character data
- any text: alpha, numeric, punctuation, whitespace

Example DTD

```
<?xml encoding="UTF-8"?>
<!ENTITY d1 "salaries department">
<!ENTITY d2 "policies department">
<!ELEMENT personnelRec (person)+>
<!ELEMENT person (name,email*)>

<!ATTLIST person salary CDATA #REQUIRED >
<!ATTLIST person band (A|B|C|D|E|F) #REQUIRED>
<!ATTLIST person active (true|false) "true"
#IMPLIED >

<!ELEMENT name (family, given)>
<!ELEMENT family (#PCDATA)>
<!ELEMENT given (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

Attribute definitions

valid values

DTD Considerations

- No type support - **#PCDATA** can be any string of parsable charcater data

- DTD syntax is different from XML syntax

  ```
  <!ELEMENT zip (#PCDATA)>
  ```

- DTDs cannot express specific constraints:
  - Eg element x can occur from 4 to 17 times
  - or if the type of element y is "decimal", the y element must contain an x element

A word on schemas.....

- XML schemas are designed to overcome weaknesses in DTD definitions
  - Explicit data type support
  - Created using XML syntax
  - Superset of DTD functionality

- Biggest improvement is specification of types
  - Enforcement of data type restrictions
  - Built-in simple types, derived types, creation of complex data types
  - Specification of allowable data ranges, masks etc

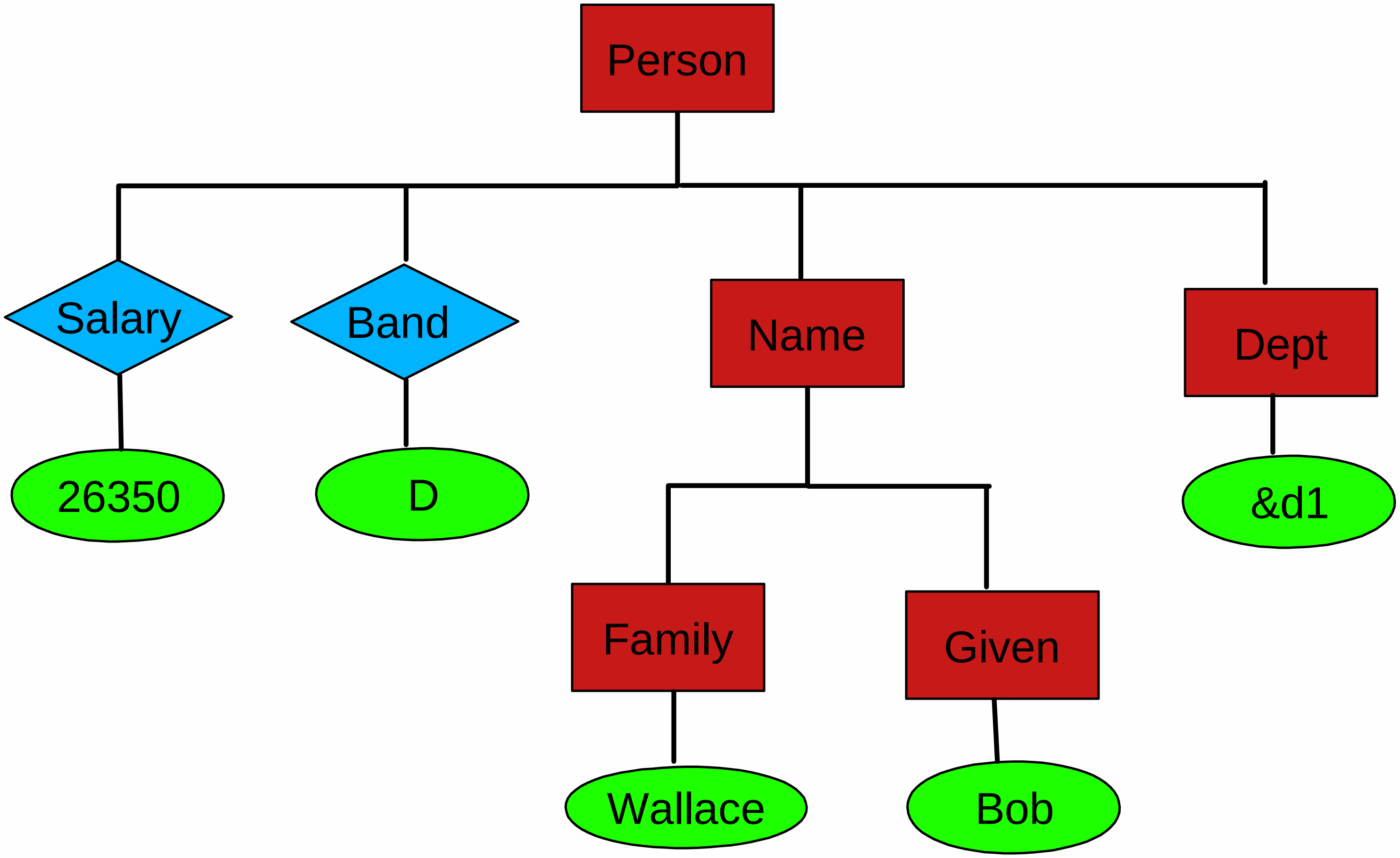
- Validation very useful for B2B as avoids programming to check data validity

- Schema support within the DB2 XML Extender is planned

XPATH

- Used to locate XML elements and attributes in a document
- Can be used as a simple query langage (prior to XQuery)
- Examples
  - /PersonnelRec/Person/Name/Family
  - /PersonnelRec/Person/@Salary
- Basic facilities for manipulation of strings, numbers and booleans
- W3C Recommendation

XML Path Examples

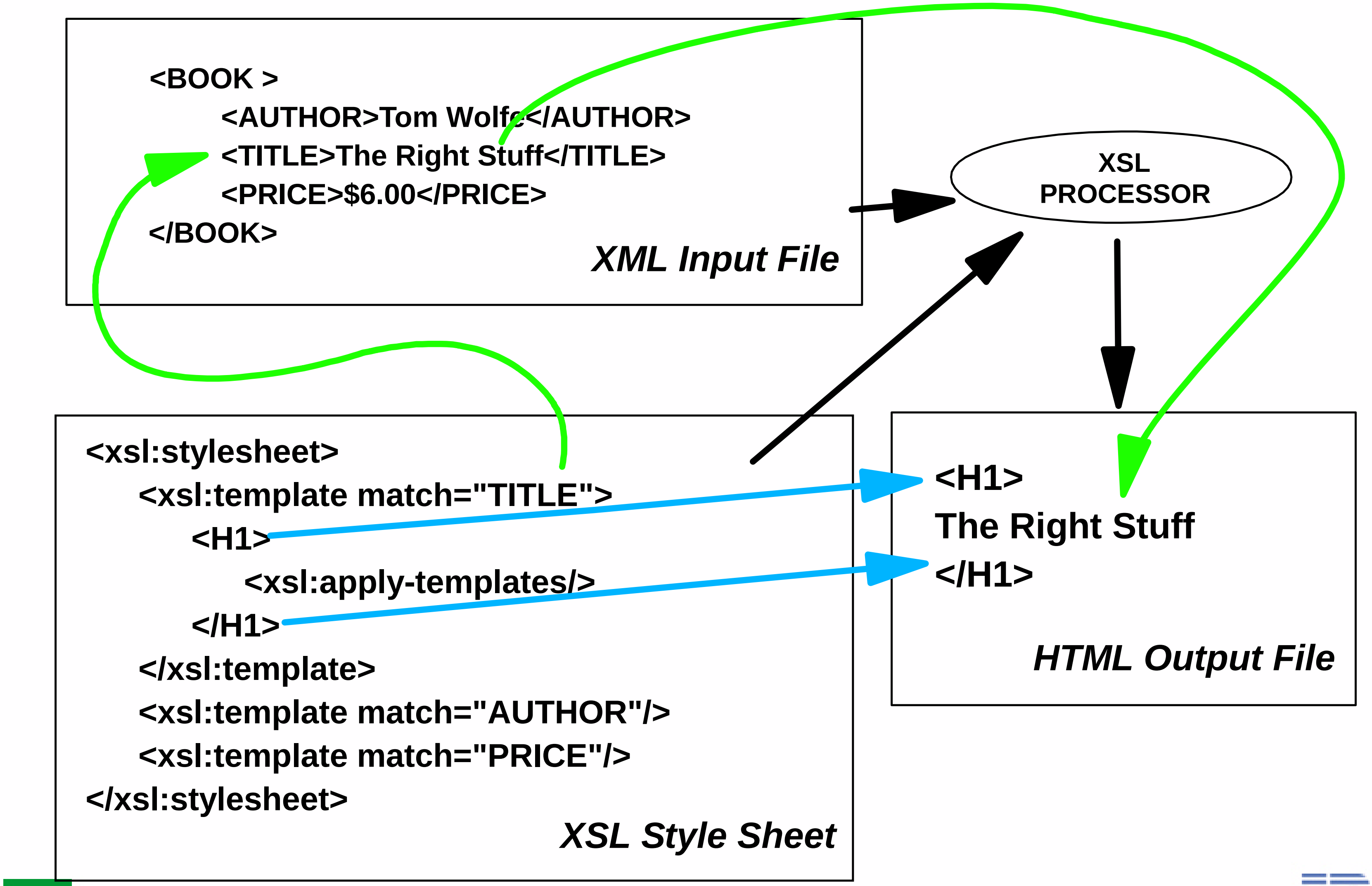


Examples

1) /Person/Name/Family
2) /Person/@Salary
3) //Family - finds family anywhere in the structure
4) /Person//Family - finds family anywhere under Person
5) /Person[@Salary='50000']

XSL: Simple display example

A W3C-defined language for expressing stylesheets and transformations



```
<BOOK >
     <AUTHOR>Tom Wolfe</AUTHOR>
     <TITLE>The Right Stuff</TITLE>
     <PRICE>$6.00</PRICE>
</BOOK>
```

XML Input File

XSL PROCESSOR

```
<xsl:stylesheet>
    <xsl:template match="TITLE">
        <H1>
            <xsl:apply-templates/>
        </H1>
    </xsl:template>
    <xsl:template match="AUTHOR"/>
    <xsl:template match="PRICE"/>
</xsl:stylesheet>
```

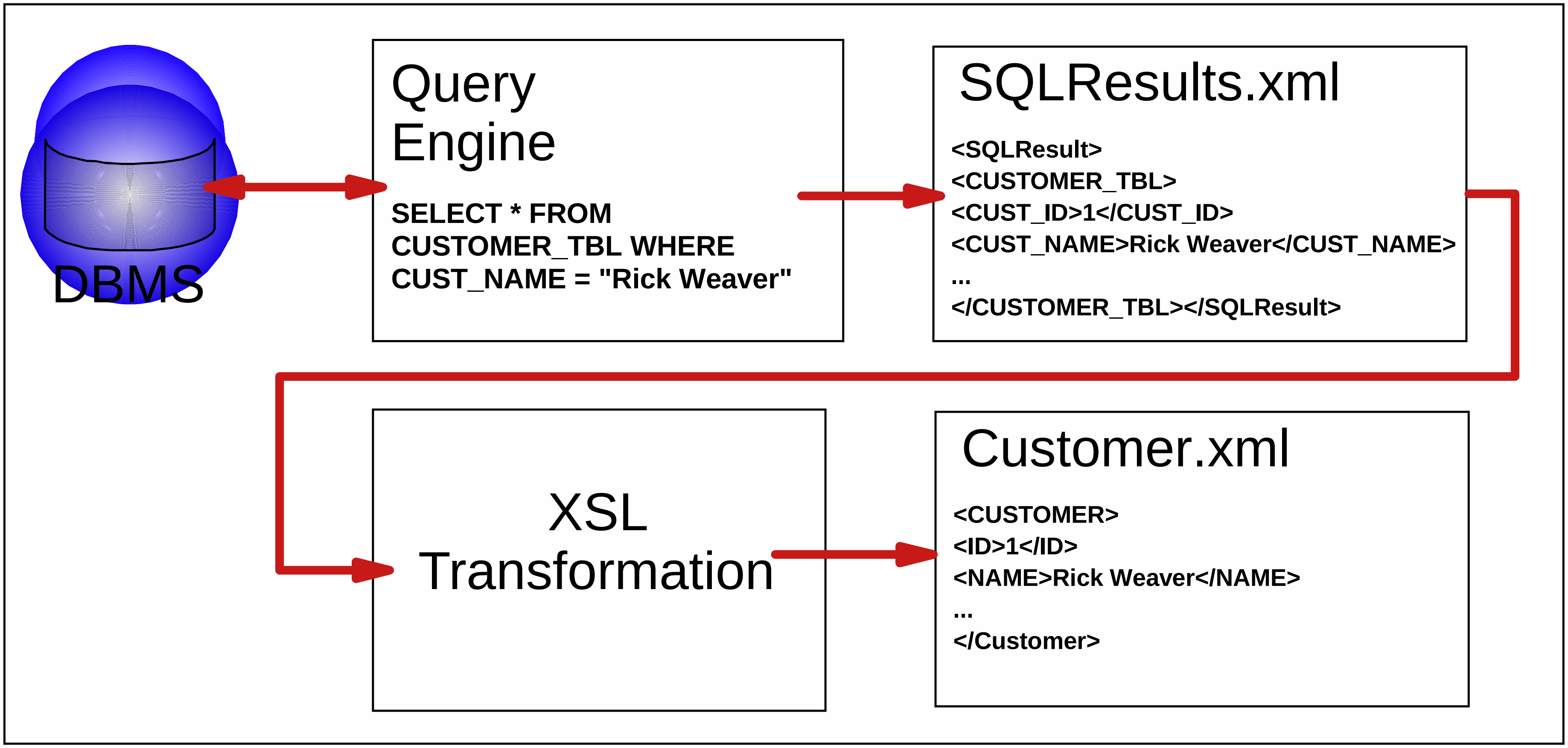
XSL Style Sheet

```
<H1>
The Right Stuff
</H1>
```

HTML Output File

XSL: Simple transform example

Also for transforming one XML document into another XML document



Query Engine

SELECT * FROM CUSTOMER_TBL WHERE CUST_NAME = "Rick Weaver"

SQLResults.xml

```
<SQLResult>
<CUSTOMER_TBL>
<CUST_ID>1</CUST_ID>
<CUST_NAME>Rick Weaver</CUST_NAME>
...
</CUSTOMER_TBL></SQLResult>
```

XSL Transformation

Customer.xml

```
<CUSTOMER>
<ID>1</ID>
<NAME>Rick Weaver</NAME>
...
</Customer>
```

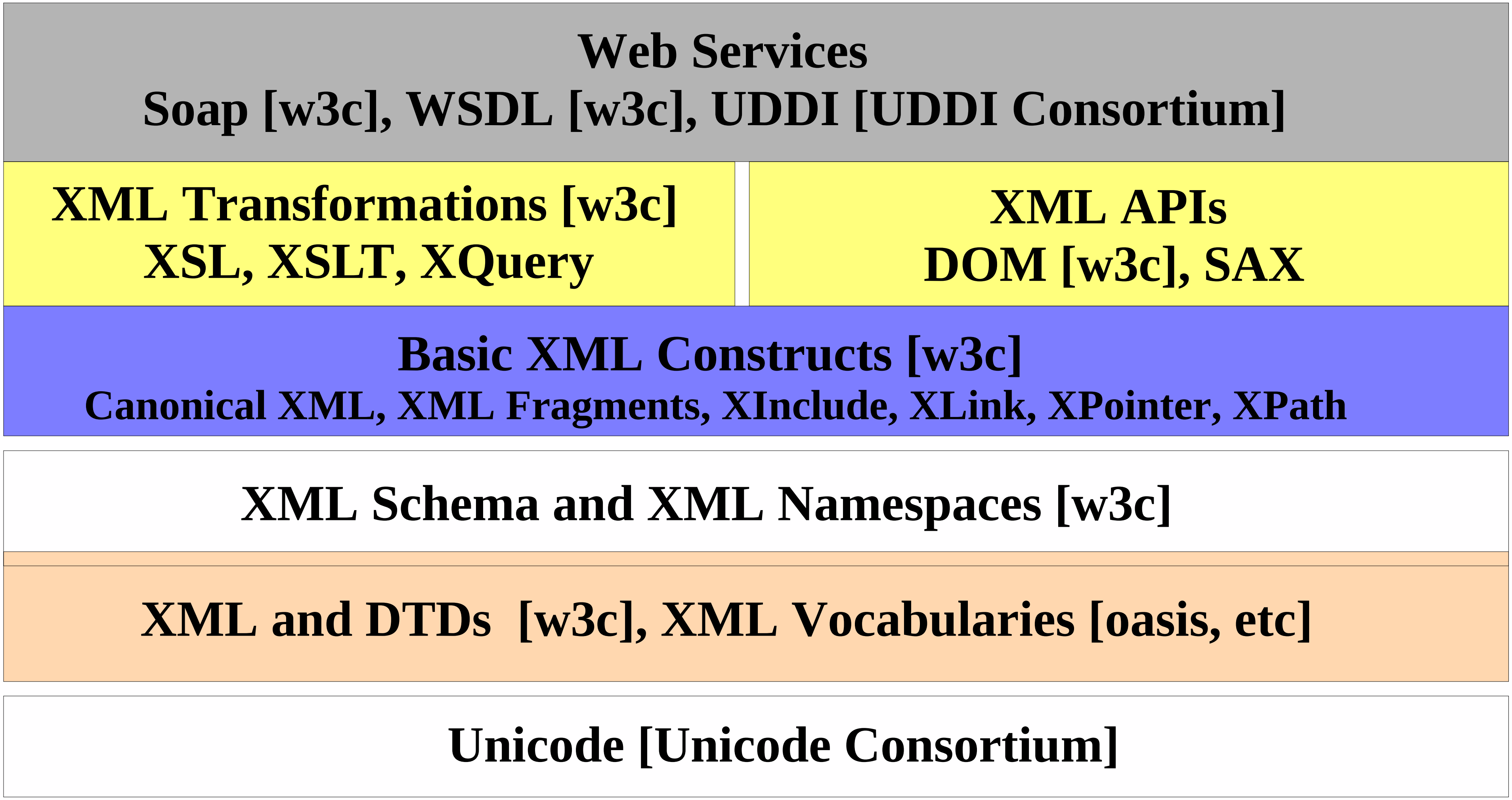
XQuery

- W3C's query language for XML
  - Way of intelligently querying the XML data
  - Better than XSL for data manipulation
  - Builds on XPATH
- Key feature is FLWR expressions (For-Let-Where-Return)
- Tracking towards Summer 2002 release

Some other XML concepts

- XLINK
  - XML elements for links between documents (similar to HTML hypertext links)
- XHTML
  - Reformulation of HTML 4.01 as XML
- VoiceXML
  - To make internet content and information accessible via voice and phone
- Namespace support
  - Simple method for qualifying names used in XML documents to avoid naming collisions

Some XML Standards



Web Services
Soap [w3c], WSDL [w3c], UDDI [UDDI Consortium]

XML Transformations [w3c]
XSL, XSLT, XQuery

XML APIs
DOM [w3c], SAX

Basic XML Constructs [w3c]
Canonical XML, XML Fragments, XInclude, XLink, XPointer, XPath

XML Schema and XML Namespaces [w3c]

XML and DTDs [w3c], XML Vocabularies [oasis, etc]

Unicode [Unicode Consortium]

XML Summary

- XML is data annotated by meaningful tags
- Presentation is separate; specified by stylesheet
- Data is easily accessed by a program
- Is perfect for structured data interchange
- Great for application integration
- Independent of programming language, platform, delivery device, operating system, ...
- International

⇨ **XML is a key component for e-business**

XML and DB2

At the heart of XML is DATA............

Utilizing DB2's Power to Support XML

- DB2 provides stability, scalability and security
- Your mission-critical business data is currently stored in DB2
- DB2s XML support enables your business data for XML
- With DB2s XML support, you can create applications to:
  - Store XML documents for fast search
  - Compose or decompose XML documents from or into relational tables
  - Query and access XML data based on XPath expressions
- You can build B2B and B2C applications with DB2 using XML Extender as the SDK

XML and DB2 - What can you do?

□ Build an XML document from DB2 data

□ Store an XML document in DB2

□ Store an XML document in an external file

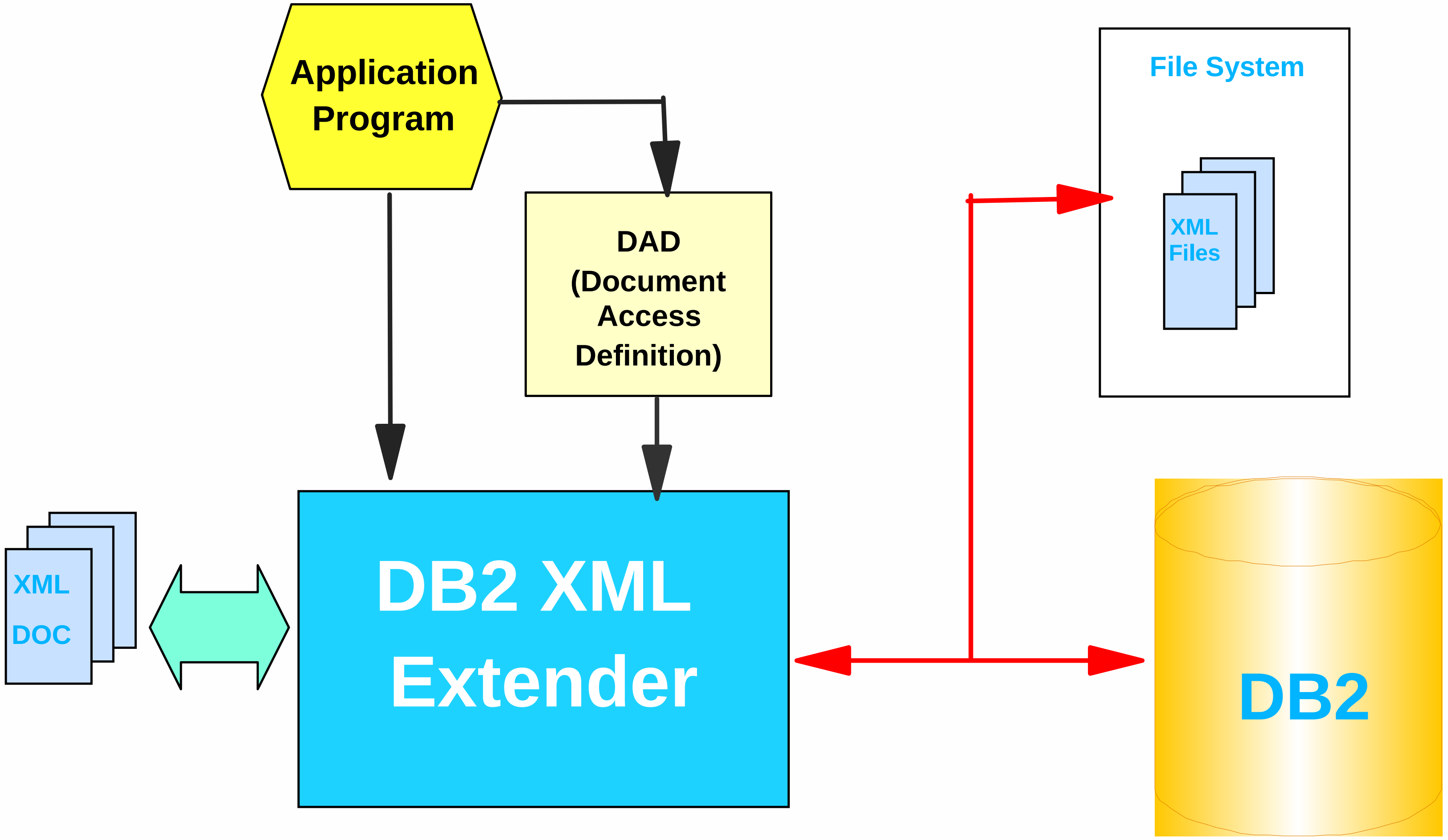
□ Retrieve an entire XML document from DB2

□ Reconstruct a decomposed document

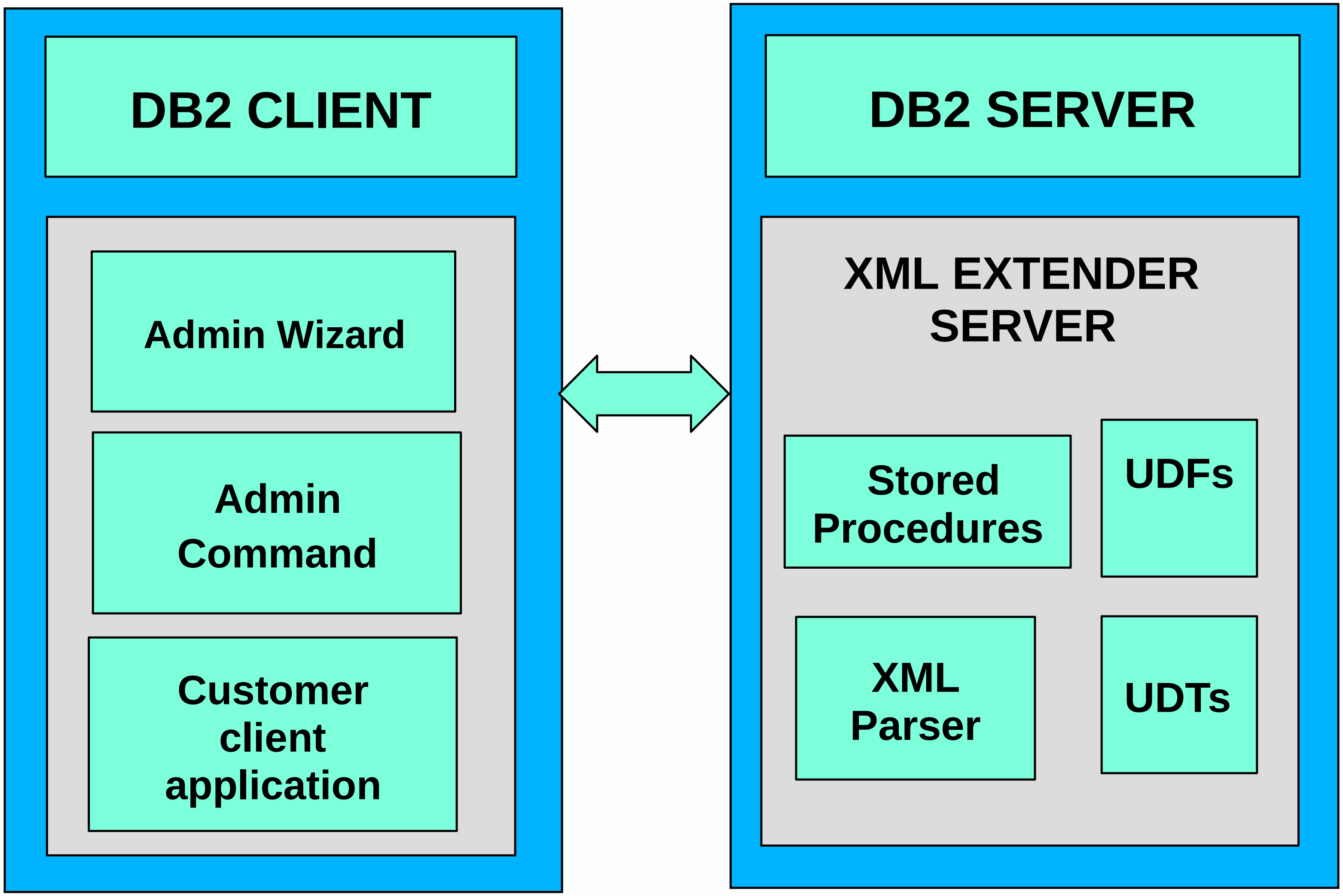
□ Extract XML elements or attribute values

□ Provide fast, powerful search and indexing on XML

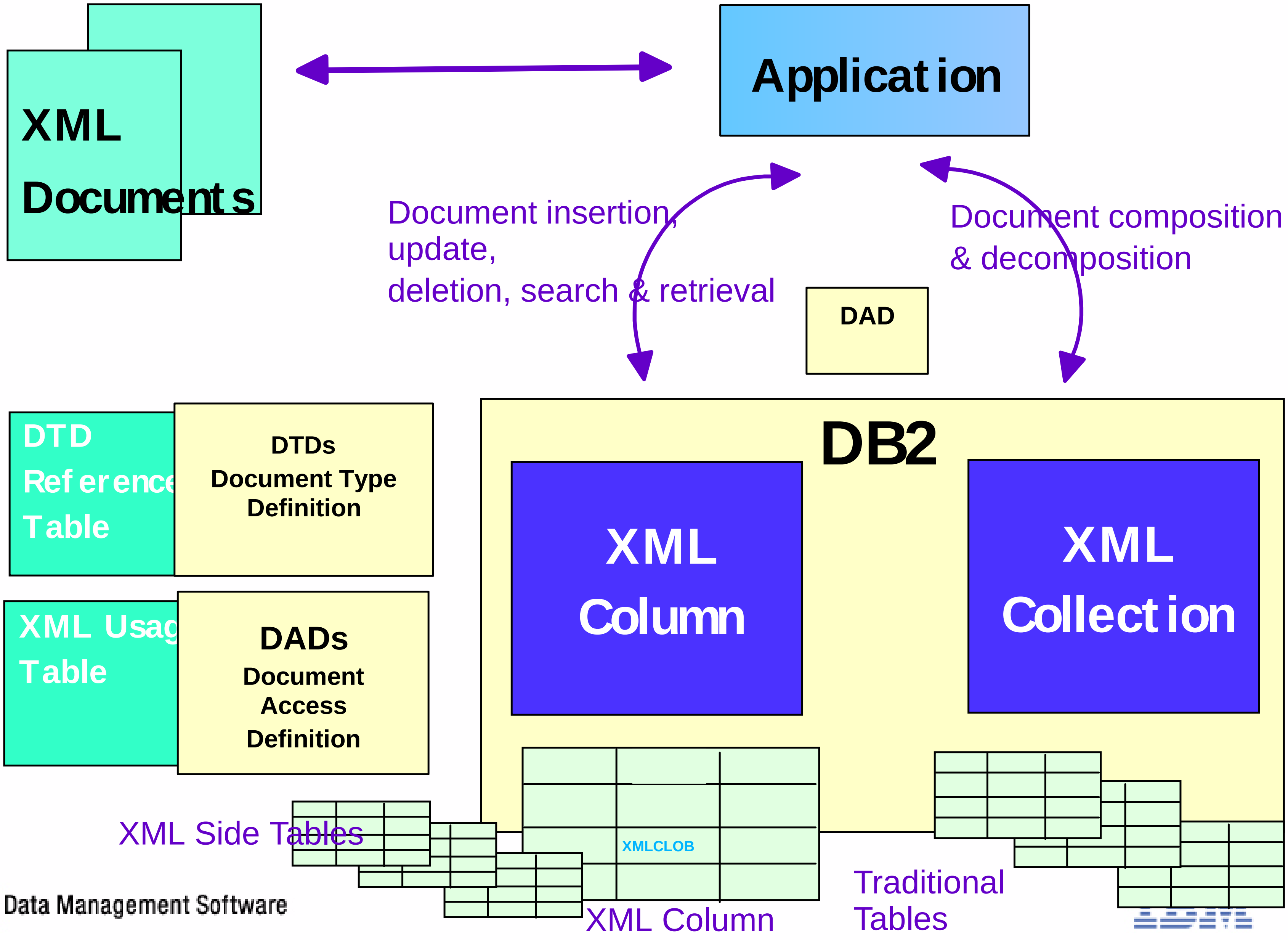
XML Extender Overview



XML Extender Architecture



DB2 Access and Storage Methods



DAD: Document Access Definition

- The DAD is an XML document
- Maps XML document structures to DB2 data structures
- For XML columns, it defines how documents are indexed by identifying the elements to be extracted into side tables
- For XML collections, it maps the structure of an XML document to DB2 tables and columns.
- WebSphere Studio assists in DAD creation

XML Collection - Two Mapping schemes

- **SQL mapping** (composition only)
  - SQL syntax
  - Scoping
    - specify the SQL select statement
  - Shaping
    - specify the mapping between column and XML data in attribute_node and text_node

```
<DAD>
<dtdid>c:\dxx\samples\dtd\getstart.dtd</dtdid>
<validation>YES</validation>
<Xcollection>
<SQL_stmt>
        SELECT  ........
</SQL_stmt>
<root_node>
        <element_node name="Order">
        <attribute_node name="key">
        <column name="order_key"/>
        </attribute_node>

        ........
        </element_node>
</root_node>
</Xcollection>
</DAD>
```

XML Collection - Two Mapping Schemes

- **RDB_node mapping** (composition and decomposition)
  - XML syntax
  - Scoping
    - specify tables and relationship among tables in the RDB_node of the root element_node
  - Shaping
    - specify RDB_node with table name, column name and optional condition for attribute_node and text_node

```
<DAD>
<Xcollection>
     ....
<element_node name="Order">
 <RDB_node>
  <table name="order_tab"/>
 <table name="part_tab"/>
   <condition>
      order_tab.order_key =
           part_tab.part_key
   </condition>
 </RDB_node>
......
<attribute_node name="key">
<RDB_node>
<table name="order_tab"/>
<column name="order_key"/>
</RDB_node>
</attribute_node>
....</Xcollection>
</DAD>
```

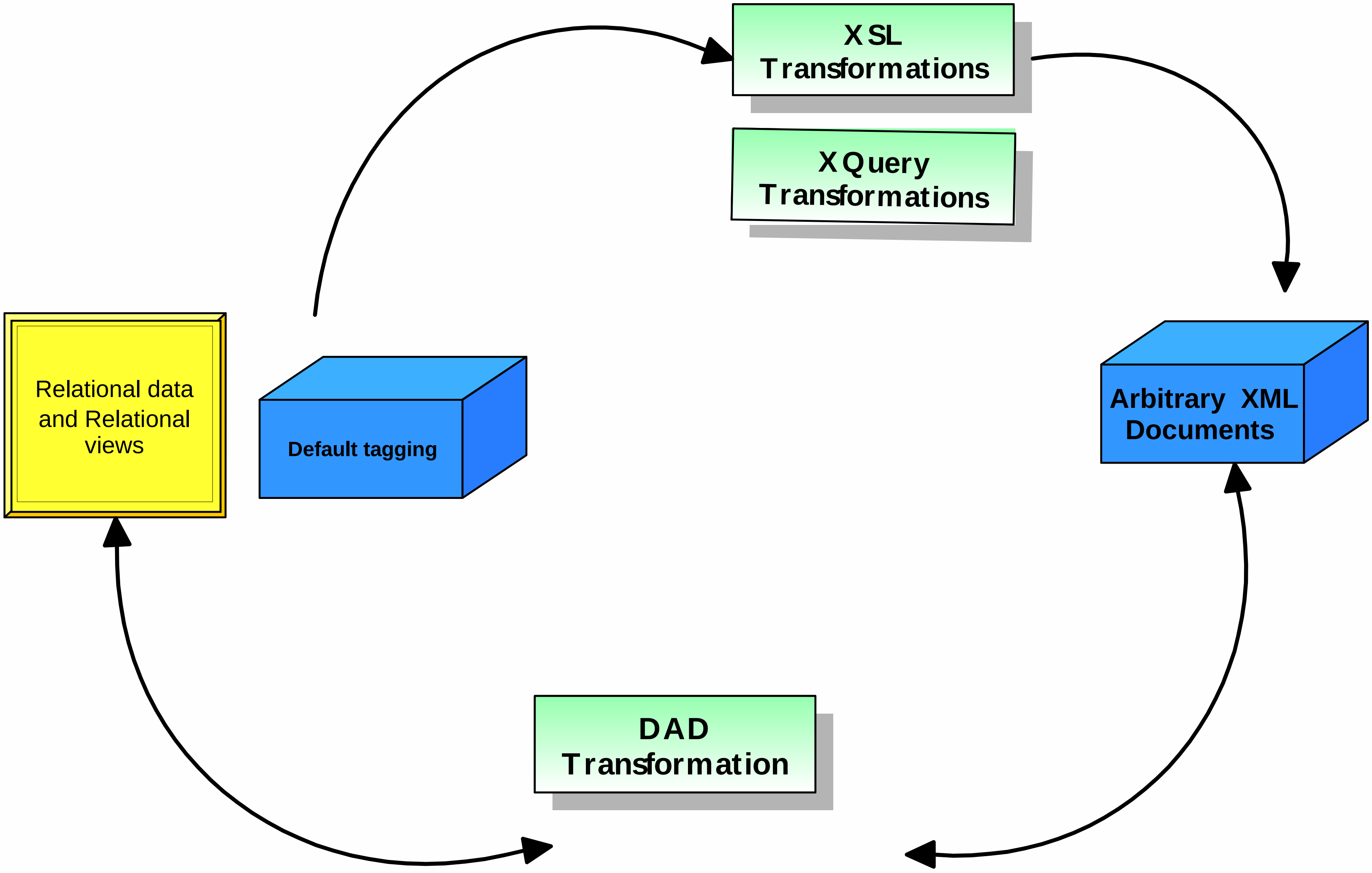
Dynamic Mapping

- Ability to take DAD to generate or shred XML documents, where DAD can be created by applications on the fly.

- Change query criteria via overriding query condition for document generation
  - SQL_OVERRIDE for SQL composition:
    - replace the SQL query
    - change conditions in the WHERE clause
    - designed for DBMS experts
  - XML_OVERRIDE for RDB node composition:
    - Use XPath syntax to define element or attribute,
    - Specify the constraints on XML elements or attributes on the documents to be generated
    - An example:

      /department/@id='E01'

The DB2 Document Access Definition



DAD: XML Columns

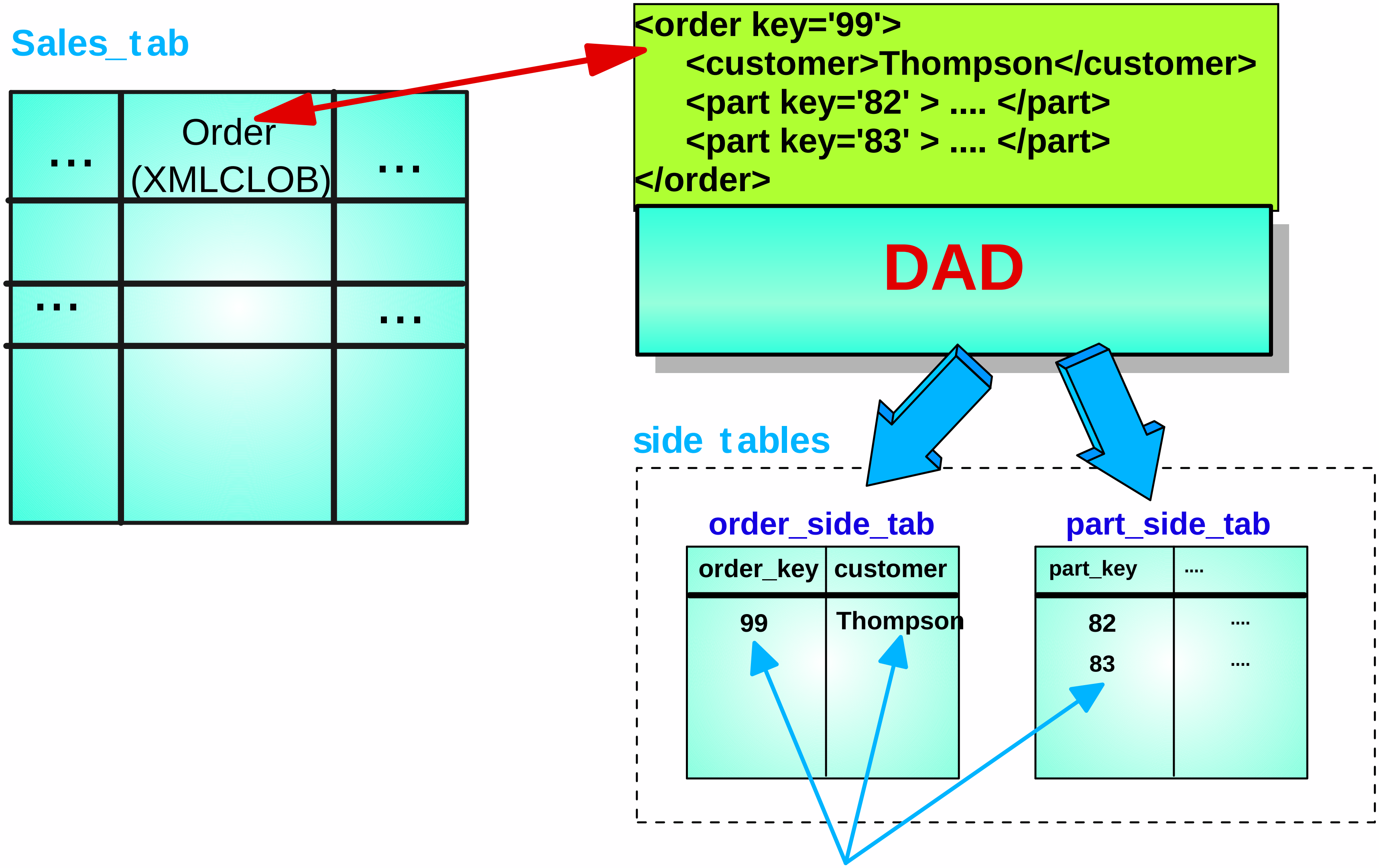
- For XML columns, the DAD defines how documents are indexed by identifying the elements to be extracted into side tables
- Uses XPath-like notation

```
<DAD>
<dtdid>c:\samples\dtd\getstart.dtd</dtdid>
<validation>YES</validation>
<Xcolumn>
  <table name="order_side_tab">
    <column name="order_key"
      type="integer"
      path="/Order/@key"
      multi_occurrence="NO"/>
    <column name="customer"
      type="varchar(50)"
      path="/Order/Customer/Name"
      multi_occurrence="NO"/>
  </table>
 </Xcolumn>
</DAD>
```

XML Columns - Side Tables

- Increases performance of searches
- Creates DB2 tables to store indexes and frequently searched data
- The tables are synchronized with the XML document with DB2 triggers
- Cannot enable a column without at least one side table being defined in the DAD

Using Side Tables for Fast Searches in XML Documents



```
<order key='99'>
    <customer>Thompson</customer>
    <part key='82' > .... </part>
    <part key='83' > .... </part>
</order>
```

order_side_tab

order_key	customer
99	Thompson

part_side_tab

part_key	....
82	....
83	....

Indexes can be created for frequently accessed tables

DB2 XML Column

- XML documents stored intact in DB2
  - With optional validation against DTDs
  - Can also use text extender text search facilities

- Supplied User Defined Types
  - XMLVarchar, XMLCLOB, XMLFile

- Supplied User Defined Functions:
  - To store or retrieve entire XML documents into/from DB2
  - To extract or update XML elements or attribute values using XPath notation
  - To extract XML fragments

- Document Access Definition: To define special indexes {side tables}
  - For speedy structured search of XML document content
  - For use in conjunction with regular SQL data

DB2 XML Column

- XML Extract Example:
  - ○ EXEC SQL SELECT extractVarchar(pers_rec, '/personnelRec/person/email') from personnel_tab WHERE salary > 100000.00

- XML Update Example:
  - ○ EXEC SQL UPDATE personnel_tab set pers_rec = Update(pers_rec, '/personnelRec/person/email', 'bwallace@us.ibm.com') WHERE family = 'Wallace'

DB2 XML Collection

- XML documents decomposed
  - XML elements and atributes stored as SQL data types
  - with optional validation against DTDs
  - many rows in many tables can be created

- XML documents composed or reconstructed
  - from regular SQL content held in traditional tables
  - with optional validation against DTDs
  - multiple documents can be generated from one requests

- Document Access Definition: To define mappings between relational and XML data

DB2 XML Collection

- Example:
  - Document decomposition into traditional tables
    - EXEC SQL CALL db2xml.dxxShredXML
    - Input: dad, xml document
    - Output: Messages & return codes
    - Outcome: Elements & attributes from input document stored in DB2 tables
- Example
  - Document composition from traditional tables
    - EXEC SQL CALL db2xml.dxxGenXML
    - Input: dad, overrides
    - Output: Composed documents, messages & return codes

XML Columns vs XML Collections

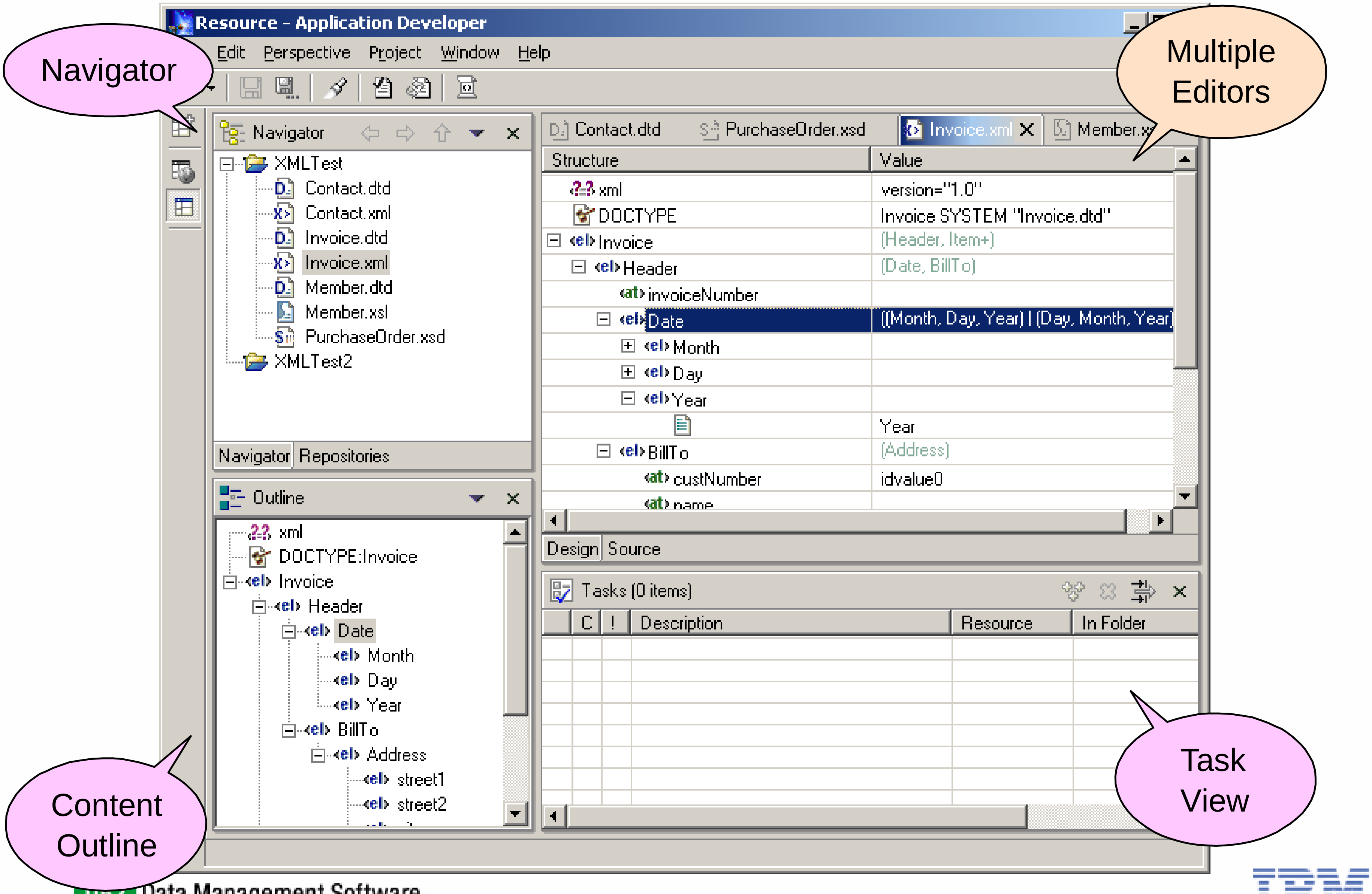
- **Reasons for choosing XML Column:**
  - XML documents already exist
  - Wish to store intact XML documents for audit purposes
  - Know which elements or attributes will be frequently searched
  - Know that XML documents are frequently read but rarely updated
  - Wish to keep XML documents external to DB2 on a file system
  - Tolerance to some DTD changes

- **Reasons for Choosing XML Collection:**
  - Require XML documents to be generated from your DB2 tables
  - Wish to store intact XML documents, but only un-tagged portions
  - Wish to incorporate document content in existing or new relational systems
  - Need to update parts of XML documents through SQL often
  - Performance of updates through SQL is critical
  - Need to process document content with analytical software
  - Some flexibility with respect to DTD changes

Tooling

- **DB2 XML Administration Wizard**
  - GUI based front-end for enabling columns and collections
  - DAD Editor component will be withdrawn
- **Websphere Studio**
  - An advanced development environment for J2EE application development
  - Includes comprehensive XML tooling including
    - XML and DTD editors
    - RDB to Database Mapping tool (aka DAD Builder)
  - Future plans to package a copy with DB2 XML Extender
- **Both will be discussed further and demonstrated in future sessions**

XML Tooling in Websphere Studio



Structure	Value
xml	version="1.0"
DOCTYPE	Invoice SYSTEM "Invoice.dtd"
Invoice	(Header, Item+)
Header	(Date, BillTo)
invoiceNumber	
Date	((Month, Day, Year) | (Day, Month, Year)
Month	
Day	
Year	
	Year
BillTo	(Address)
custNumber	idvalue0
name	

C	!	Description	Resource	In Folder

WSAD: XML Tooling Features

- **DTD/Schema Editor**
  - Visual tooling for working with DTDs/Schemas
  - Create DTDs/Schemas from existing XML documents
  - Convert to/from DTD or Schema
  - Generate JavaBean(s) for creating/manipulating XML documents from DTD/Schema
  - Generate an HTML form from a DTD
- **XML Source Editor**
  - Design/Source mode
  - DTD/Schema validation
  - Code Assist for building XML documents

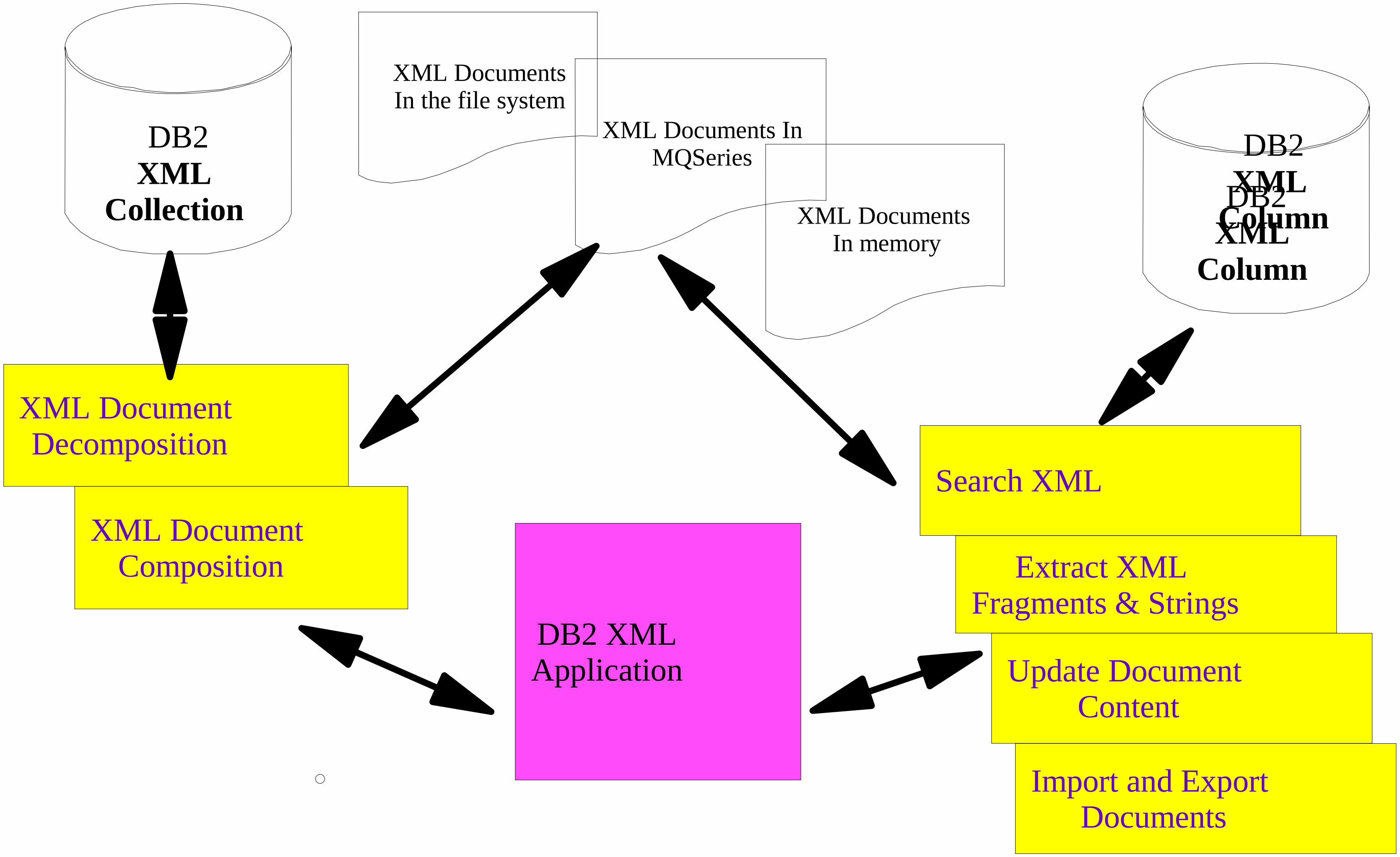
WSAD: XML Tooling Features

- **XML Mapping Editor**
  - Generate XSL to map XML between DTDs/Schemas
- **XML to/from Relational Databases**
  - Generate XML, XSL, XSD from an SQL Query
- **RDB/XML Mapping Editor**
  - Map columns in a table to elements/attributes to XML
  - Generate a Database Access Definition (DAD) script to compose/decompose XML documents to/from a database
    - DAD is used with DB2 XML Extender

DB2 XML Extender - futures and strategy

IBM Software Group

XML Data Source Integration with DB2



DB2 **XML Collection**

XML Documents In the file system

XML Documents In MQSeries

XML Documents In memory

DB2 **XML Column**

DB2 **XML Column**

XML Document Decomposition

XML Document Composition

DB2 XML Application

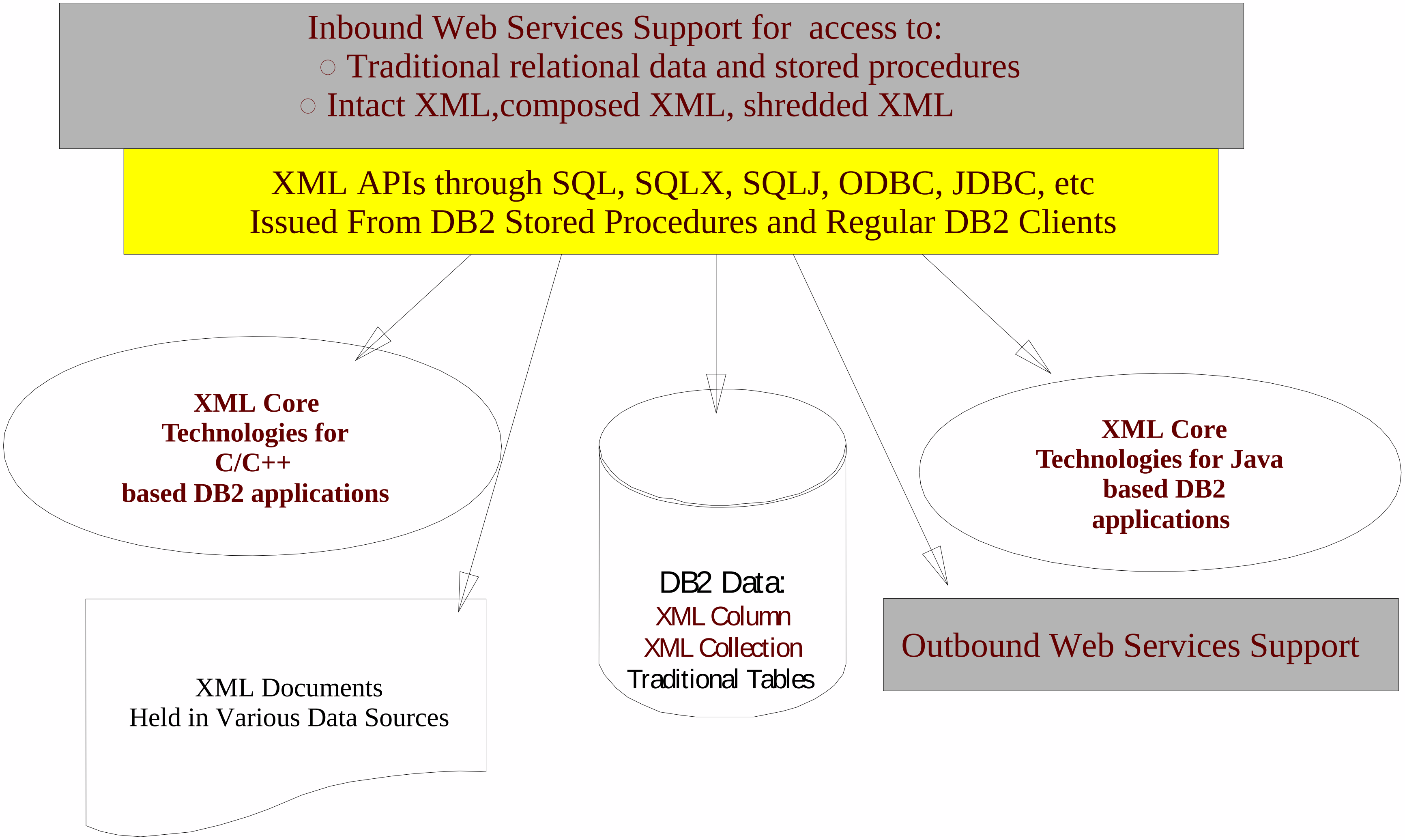
Search XML

Extract XML Fragments & Strings

Update Document Content

Import and Export Documents

XML Directions for DB2



Inbound Web Services Support for access to:

- ○ Traditional relational data and stored procedures
- ○ Intact XML,composed XML, shredded XML

XML APIs through SQL, SQLX, SQLJ, ODBC, JDBC, etc
Issued From DB2 Stored Procedures and Regular DB2 Clients

XML Core Technologies for C/C++ based DB2 applications

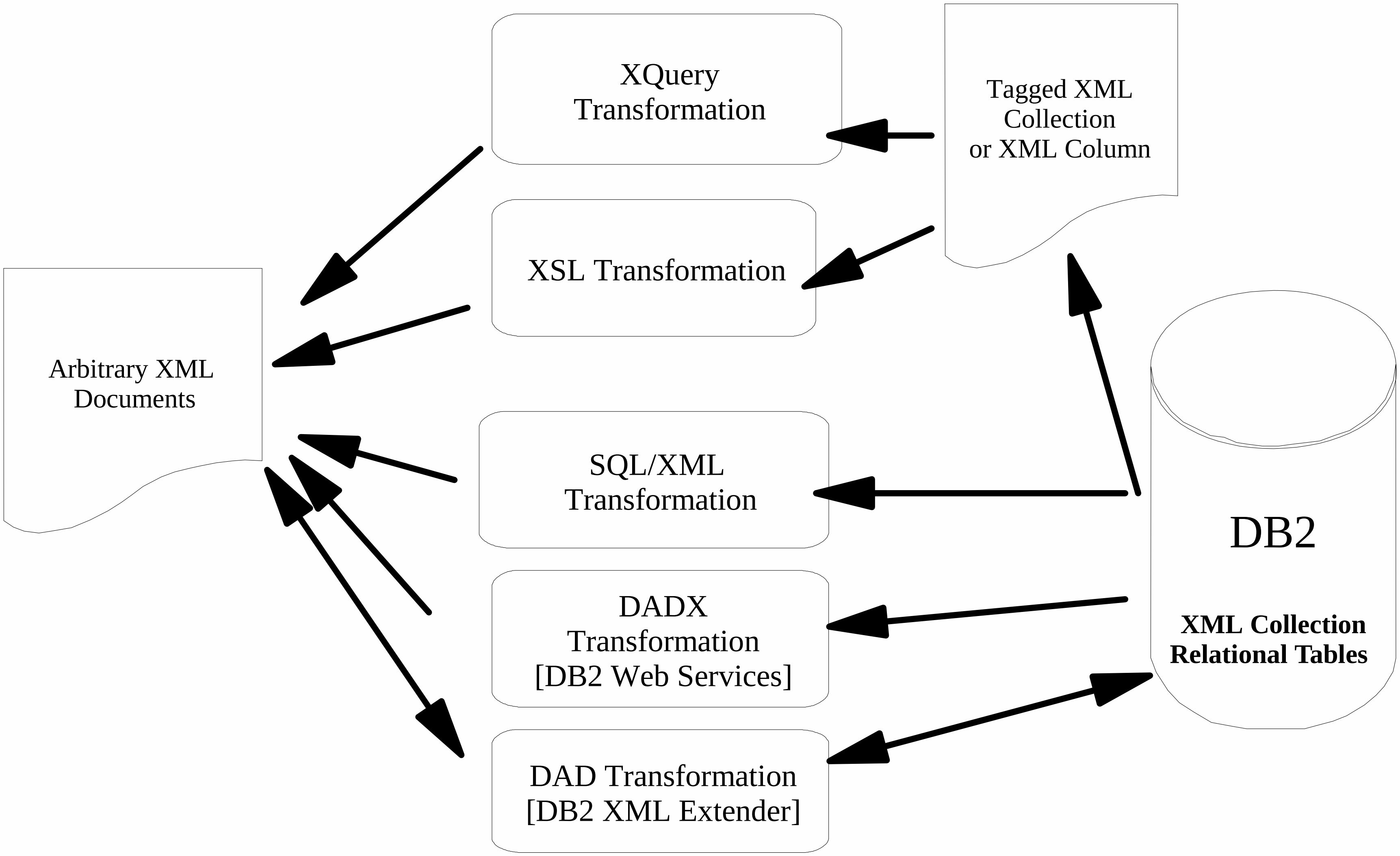
XML Documents
Held in Various Data Sources

DB2 Data:
XML Column
XML Collection
Traditional Tables

XML Core Technologies for Java based DB2 applications

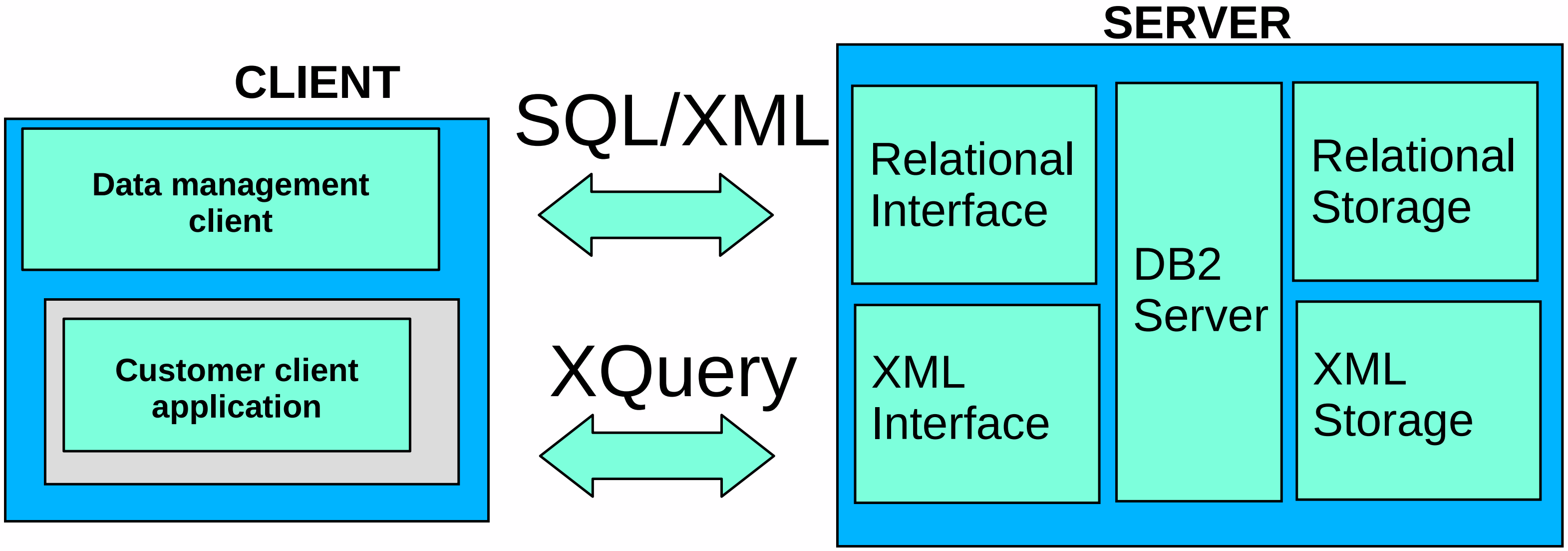
Outbound Web Services Support

XML Directions in DB2


XQuery Transformation
Tagged XML Collection or XML Column
XSL Transformation
Arbitrary XML Documents
SQL/XML Transformation
DB2
XML Collection Relational Tables
DADX Transformation [DB2 Web Services]
DAD Transformation [DB2 XML Extender]

XML in DB2 - The Future

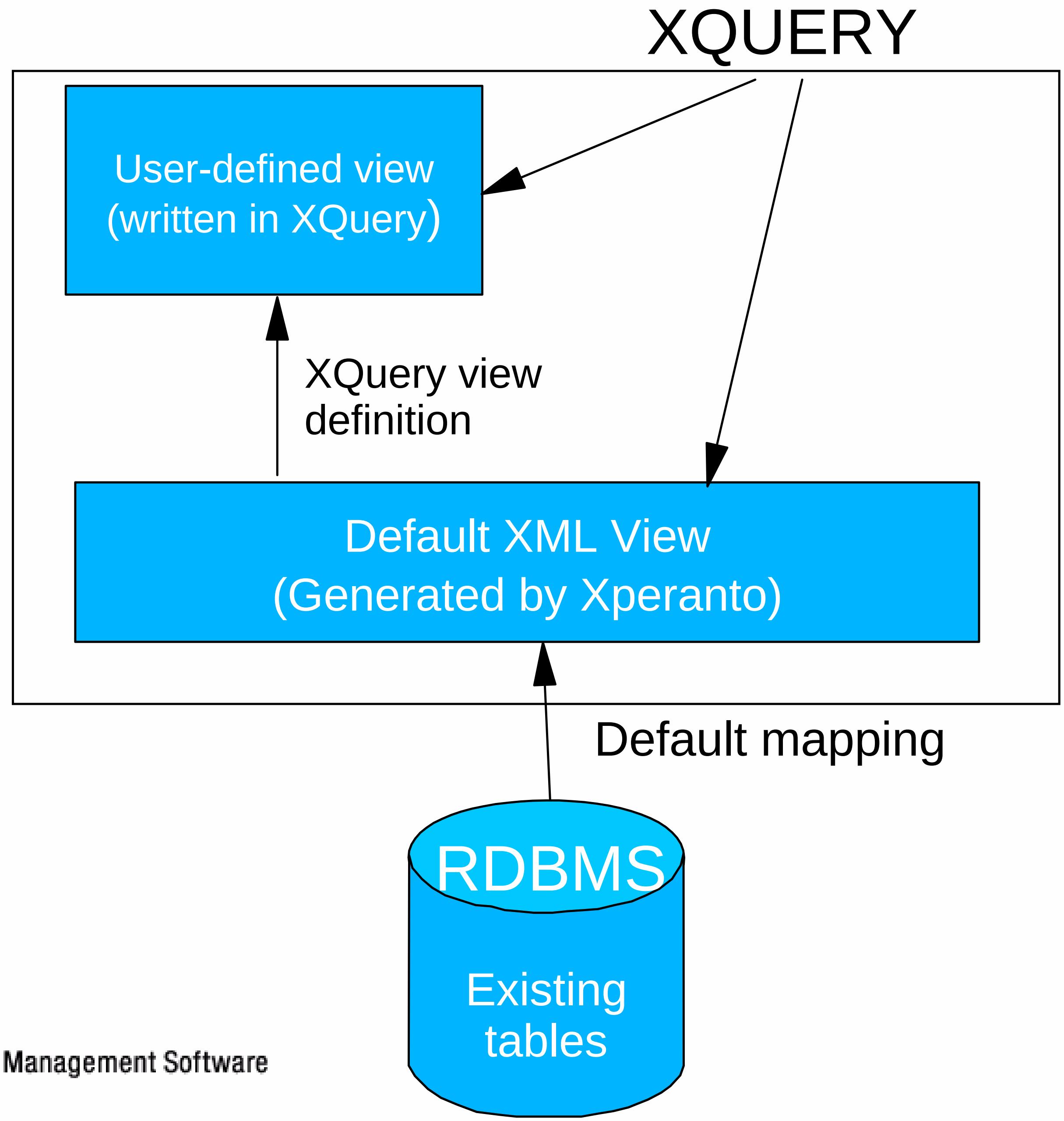
- ✓ XML capabilities "inside" the engine
  - ✓ Performance, performance, performance
- ✓ "Feels" relational and/or XML
- ✓ Both SQL flavor and fully XML flavor
- ✓ XML *is* DB2 internals
- ✓ XML Extender becomes one with the data engine



Xperanto

- A bridge between internet applications and relational databases
- Ability to create and query XML views of existing relational data
- Automatically creates default XML view
- Users can create views on top using XQuery
- Only desired relational data is materialised
  - Query rewrite and view composition
- Uses power of relational database engine
  - pushes computations down into engine

Xperanto Architecture

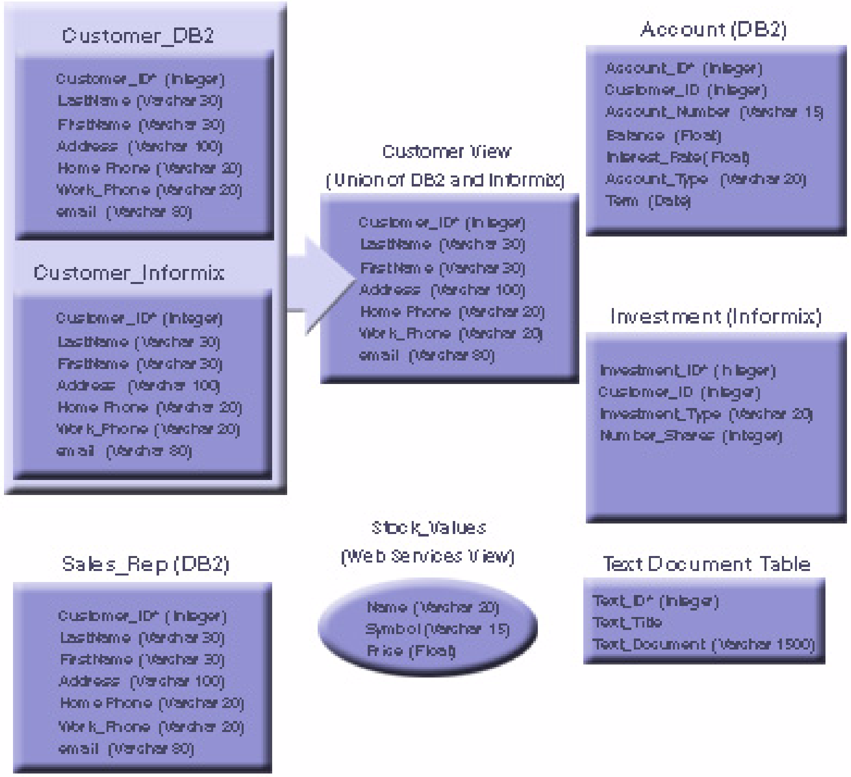


Xperanto Demo

- Available online at http://xperanto.dfw.ibm.com/demo/mainPage.jsp

- Financial scenario involving Bank acquisition of Financial Services
  - Objective is to give a unified view of the customer portfolio
  - Leverage new sales opportunities across divisions
  - Integrating database infrastructures with real-time data

Technology Underpinnings



Customer_DB2

Customer_ID* (Integer)
LastName (Varchar 30)
FirstName (Varchar 30)
Address (Varchar 100)
Home Phone (Varchar 20)
Work_Phone (Varchar 20)
email (Varchar 80)

Customer_Informix

Customer_ID* (Integer)
LastName (Varchar 30)
FirstName (Varchar 30)
Address (Varchar 100)
Home Phone (Varchar 20)
Work_Phone (Varchar 20)
email (Varchar 80)

Customer View
(Union of DB2 and Informix)

Customer_ID* (Integer)
LastName (Varchar 30)
FirstName (Varchar 30)
Address (Varchar 100)
Home Phone (Varchar 20)
Work_Phone (Varchar 20)
email (Varchar 80)

Account (DB2)

Account_ID* (Integer)
Customer_ID (Integer)
Account_Number (Varchar 15)
Balance (Float)
Interest_Rate (Float)
Account_Type (Varchar 20)
Term (Date)

Investment (Informix)

Investment_ID* (Integer)
Customer_ID (Integer)
Investment_Type (Varchar 20)
Number_Shares (Integer)

Sales_Rep (DB2)

Customer_ID* (Integer)
LastName (Varchar 30)
FirstName (Varchar 30)
Address (Varchar 100)
Home Phone (Varchar 20)
Work_Phone (Varchar 20)
email (Varchar 80)

Stock_Values
(Web Services View)

Name (Varchar 20)
Symbol (Varchar 15)
Price (Float)

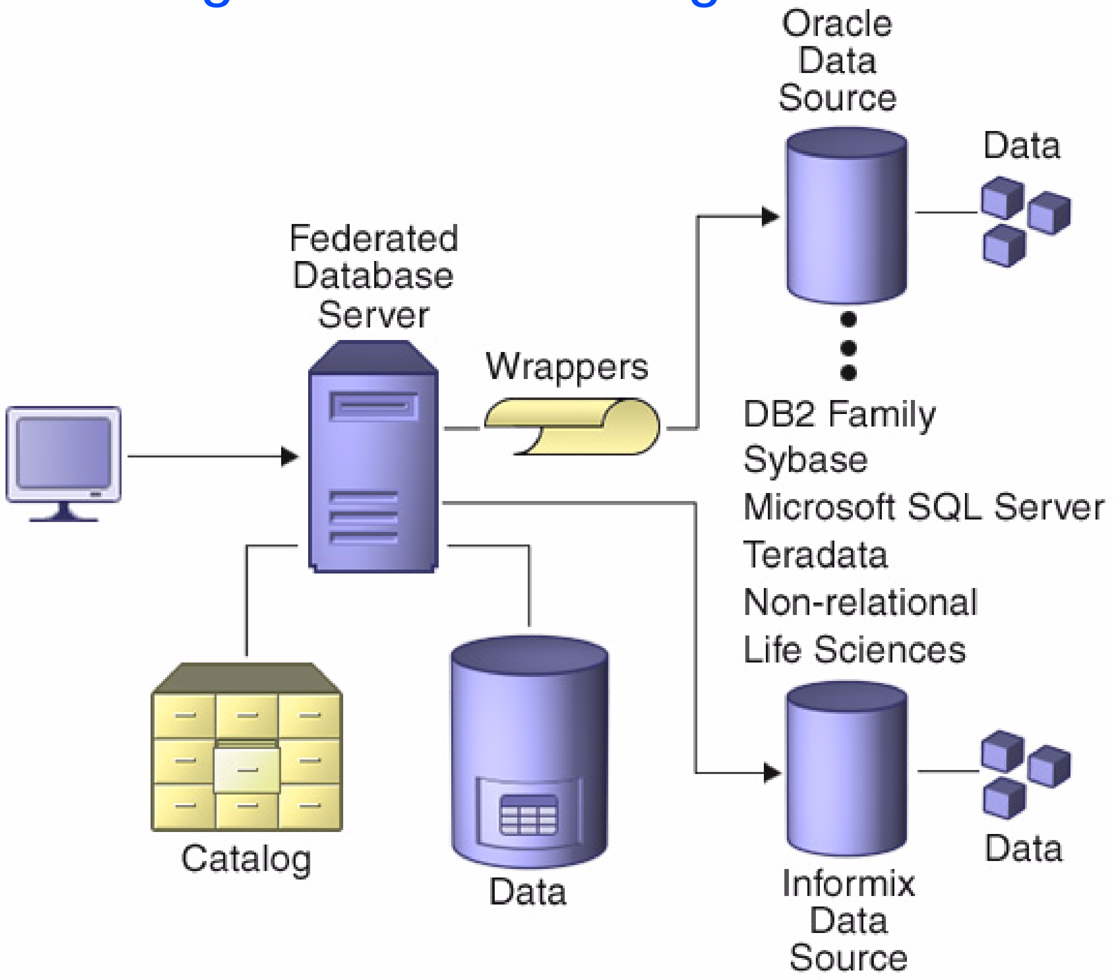
Text Document Table

Text_ID* (Integer)
Text_Title
Text_Document (Varchar 1500)

- □ XQuery provides an XML view of all data
- □ Federation enables a single virtual view of distributed, diverse data sources
- □ Web services consumer support enables applications to view web service as a scalar or table function
- □ Text search function is applied consistently across all data

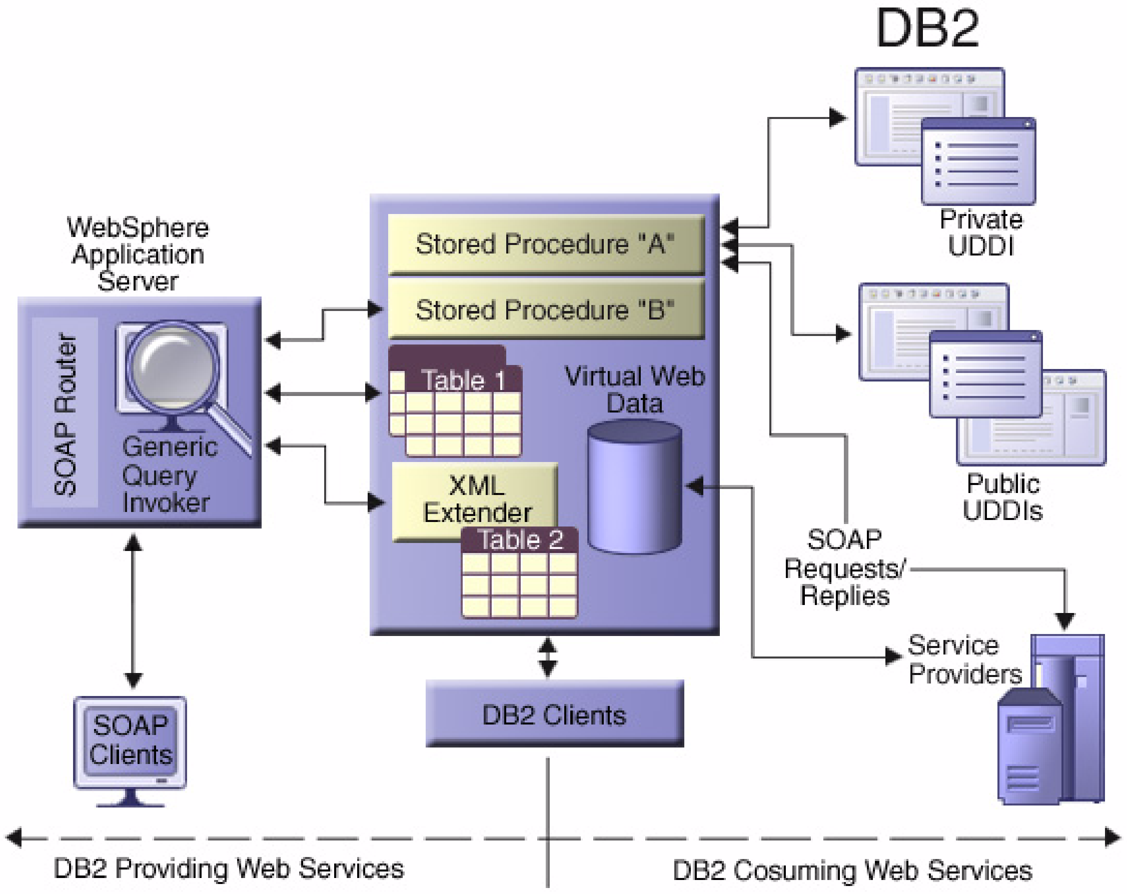
Federation technology

Single Database Image



- Resources are defined through nicknames
- Nicknames have associated properties used for optimization
- Push-down functions (release level)
- Join algorithms
- Table properties (e.g. carinality indices,...)
- Network speed
- Processor speed
- Garlic technology enables specification of source properties
- Single sign-on security
- Federation engine compensates for missing back-end function

Web Services Technology



- Tooling tranforms a WDSL into a scalar or table function
- C-based UDFs provide a SOAP client
- Application benefits from core database technologies applied to web services
- Query
- Federation
- Optimization/Parallelism
- Caching
- Search

Summary

- XML is key element of e-business
- XML data can be stored in DB2 tables
  - Either intact in columns or decomposed into relational tables
- XML documents can be retrieved from DB2
  - Via stored procedures or UDFs
- Available on many DB2 platforms NOW!
  - Introduced with Version 7

And finally don't forget ...

- Tomorrows session will cover **"DB2 and Web Services"**
- Next week's sessions will cover **"Using the DB2 XML Extender"**
  - End-to end process using available tooling
  - Hands-on labs

Information sources

- XML
  - General info : xml.coverpages.org, www.w3.org
  - Annotated XML spec : http://www.xml.com/pub/a/axml/axmlintro.html
  - IBM developerworks for XML http://www.ibm.com/developerworks/xml

- DB2 XML Extender
  - Home page : www.ibm.com/software/data/db2/extenders/xmlext
  - Hints and tips : http://www.ibm.com/software/data/db2/extenders/xmlext/support.htm
  - DB2 Developers Domian http://www.ibm.com/software/data/developer/
  - Redbooks
    - "Integrating XML with DB2 XML Extender and DB2 Text Extender" - SG24-6130-00
    - DB2 for OS/390 and z/OS Powering the World's e-business Solutions SG24-6257
  - DB2 XML Extender Hints and Tips for the IBM ~ iSeries Server
    - http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp0135.html?Open

- XML database marketing community (IBMers only)
  - email Tina Gleisner