

# Web services, DB2, WORF



Kris Van Thillo,  
kvanthillo@abis.be

ABIS Training & Consulting  
Diestsevest 32  
B - 3000 Leuven  
WWW.ABIS.BE

# Agenda



- ⌘ Web services defined ...
- ⌘ DB2 XML Extender
- ⌘ DB2 as a Web service consumer
- ⌘ DB2 as a Web service provider - WORF

# Web services defined ...



Web services are

“Software applications identified by a URI, whose interfaces and bindings are capable of being defined, described and discovered by XML artifacts, supporting direct interactions with other software applications using XML messages via internet-based protocols.”

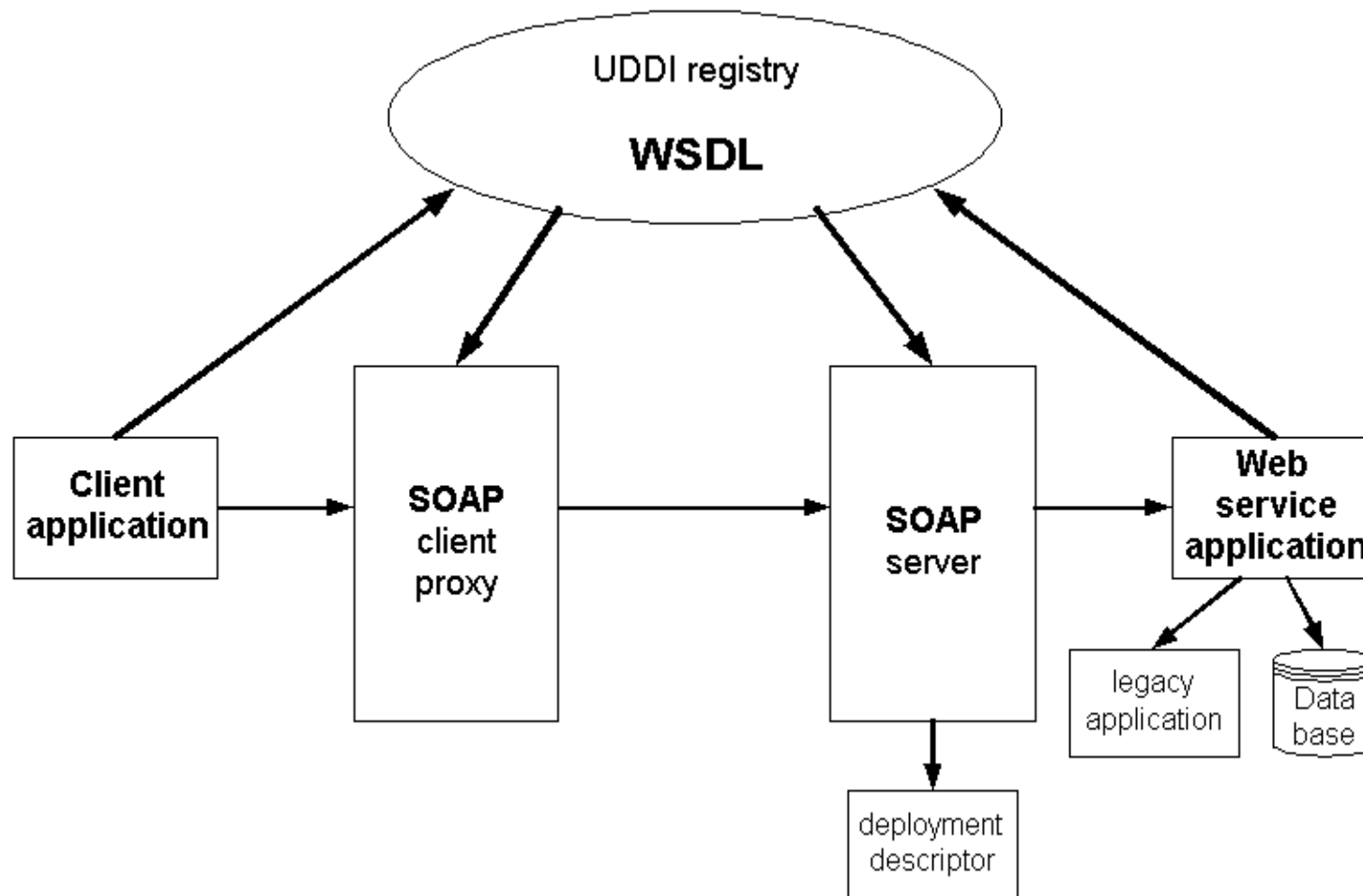
and based on open standards

XML

SOAP

WSDL

# Web services - execution



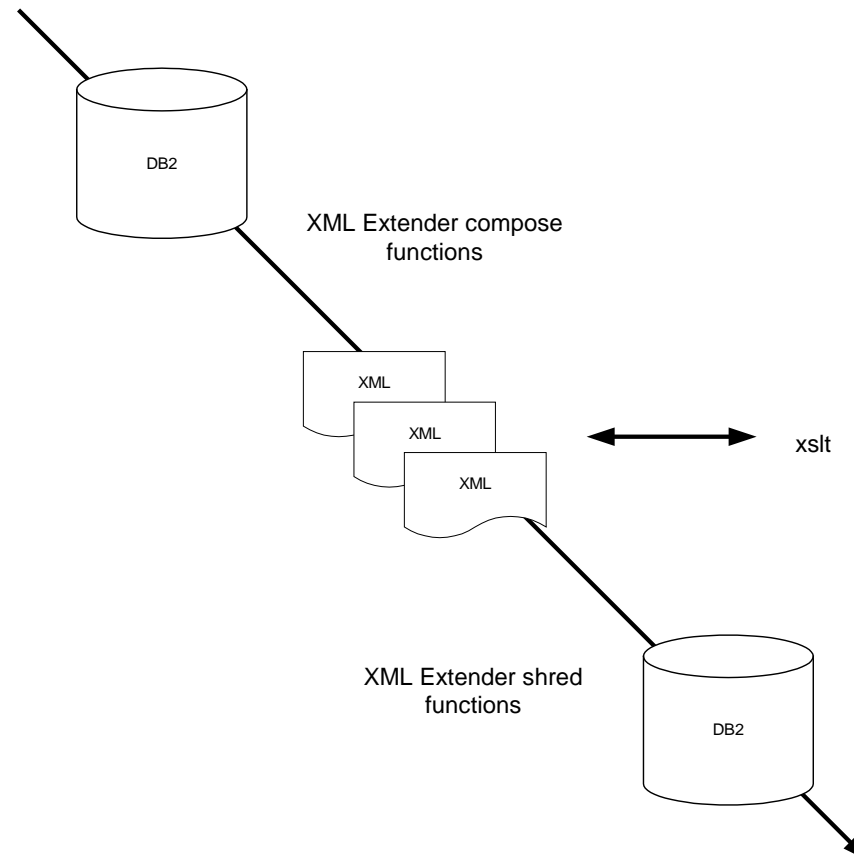
# Web services - why?



## Advantages:

- ⌘ loosely coupled and coarse-grained service granularity
- ⌘ programming language independent, interoperable
- ⌘ transport independent
- ⌘ multiple invocation styles: static or dynamic
- ⌘ multiple communication styles: synchronous or asynchronous
- ⌘ open, extensible, standards based: based on XML
- ⌘ composable

# DB2 XML Extender



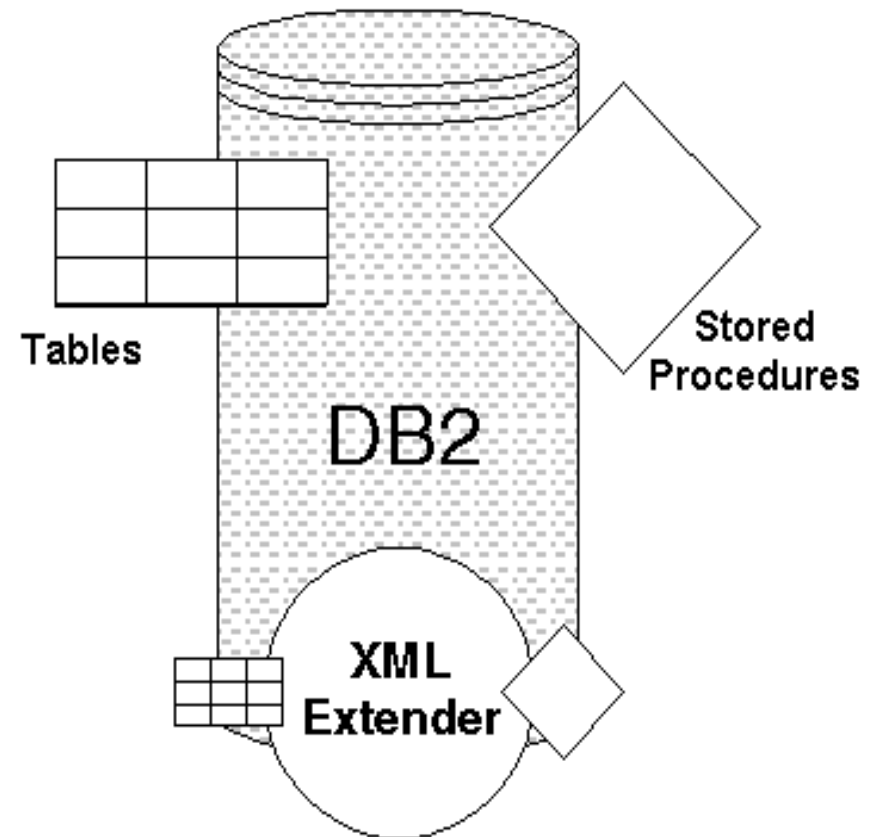
# DB2 XML Extender

It's all about XML!

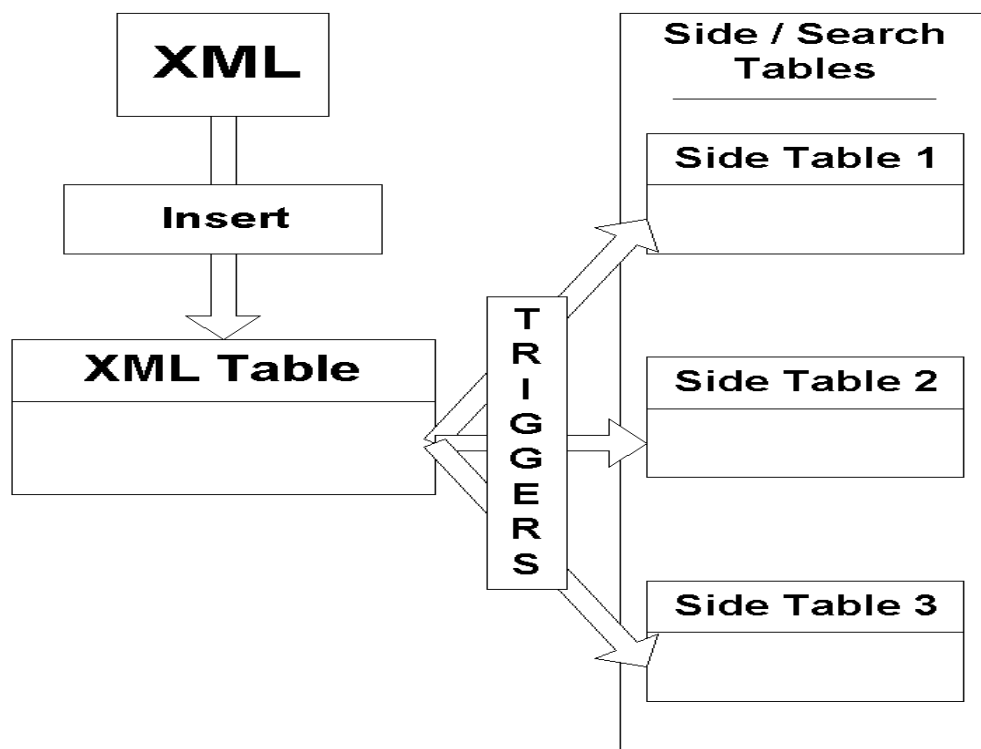
Offers ...

- ☒ Stored procedures
- ☒ Triggers
- ☒ User defined functions (UDF)
- ☒ User defined datatypes (UDT)
- ☒ Supporting tables

... to extend DB2 functionality!



# Xcolumn

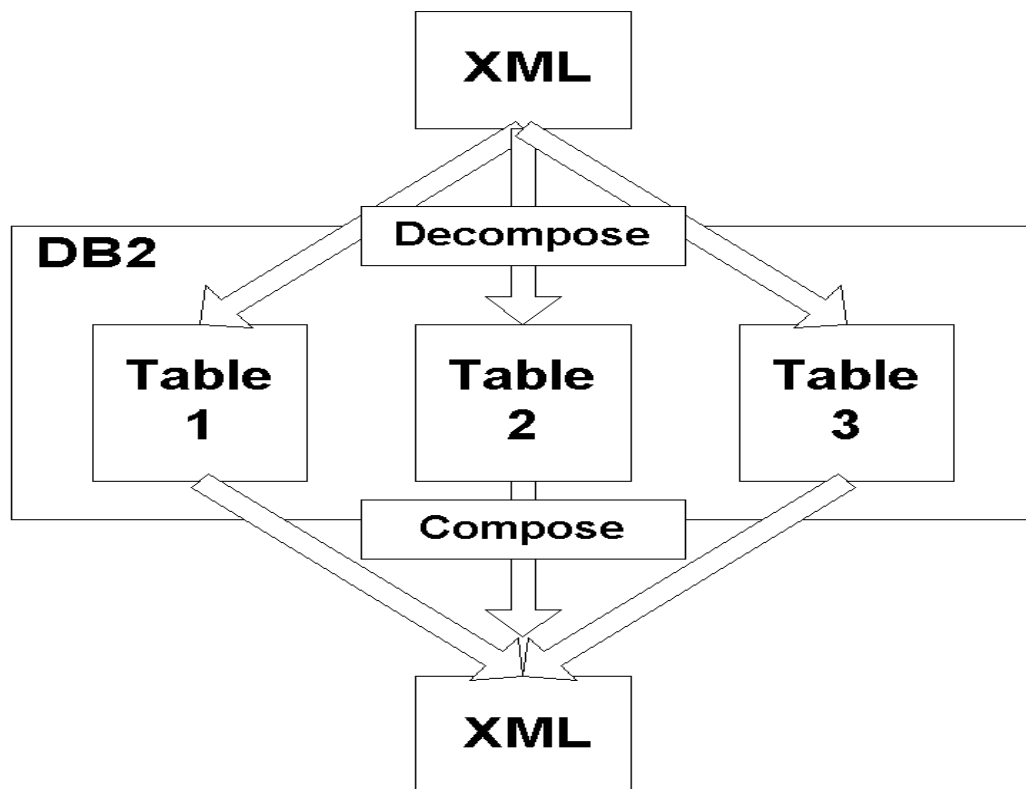


Column data = XML documents  
DB2 used as XML document repository

Validation of XML docs  
Storing XML docs  
Search through XML docs



# Xcollection



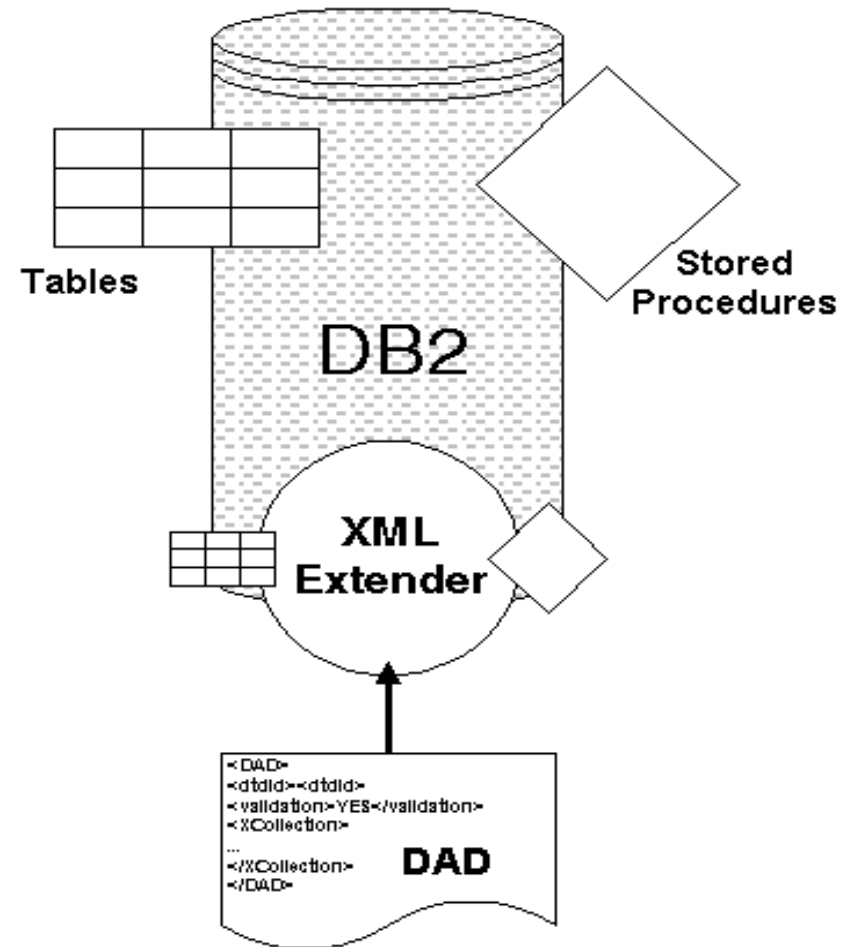
DB2 used as *database*  
(no XML in DB2)  
XML is transport language

Validation of XML docs  
Decompose XML docs into data  
Compose XML docs from data

# DAD

## XML document

- ☒ Validate
- ☒ DTD-id
- ☒ Method
- ☒ Detailed description of data mapping:  
DB2 vs. XML



# DAD - example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE DAD SYSTEM "E:\dxx\dtd\dad.dtd">
<DAD>
  <validation>NO</validation>
  <Xcolumn>
    <table name="visa_uittreksel">
      <column name="kaartnr" type="character(10)" path="/visakaart/@nr"/>
      <column name="startdate" type="character(10)" path="/visakaart/valid/startdate" />
      <column name="enddate" type="character(10)" path="/visakaart/valid/enddate"/>
    </table>
    <table name="visa_lijn_product">
      <column name="product" type="character(30)" path="/visakaart/entry/product"
        multi_occurrence="YES"/>
    </table>
  </Xcolumn>
</DAD>
```

# DB2 XML Extender - example



```
db2 create table xmlVisa (id smallint, XMLvisakaart db2xml.xmlvarchar)
```


```
db2 insert into db2xml.dtd_ref values('f:\library\udb\XmlExtender\xmlColumn\visa.dtd',  
db2xml.XMLClobFromFile('f:\library\udb\XmlExtender\xmlColumn\visa.dtd'), 0, 'db2', 'db2', 'db2')
```

```
dxxadm enable_column dbeb46 xmlVisa XMLvisakaart  
f:\library\udb\XmlExtender\xmlColumn\visa.dad
```

```
db2 insert into xmlvisa (id, XMLvisakaart) values(1 ,  
db2xml.XMLVarcharFromFile('f:\library\udb\XmlExtender\xmlColumn\visa1.xml') )
```

```
db2 insert into xmlvisa (id, XMLvisakaart) values(2 ,  
db2xml.XMLVarcharFromFile('f:\library\udb\XmlExtender\xmlColumn\visa2.xml') )
```

# DB2 as a Web service consumer (1)



- ⌘ Integrate SQL statements and Web service invocations
- ⌘ UDFs embed calls to Web services
- ⌘ Procedure:
  - ☒ create UDF
  - ☒ invoke UDF through SQL statement
- ⌘ What do we need?
  - ☒ The URI of the target object/service
  - ☒ The name of the of an operation to execute, including nput and output format
  - ☒ Binding info: protocol to be used, encoding style, etc

# DB2 as a Web service consumer (2)

```
db2xml.soaphttpv returns VARCHAR():  
    db2xml.soaphttpv (endpoint_url VARCHAR(256),  
                      soap_action VARCHAR(256),  
                      soap_body VARCHAR(3072))  
    RETURNS VARCHAR(3072)
```

```
db2xml.soaphttpc returns CLOB():  
    db2xml.soaphttpc (endpoint_url VARCHAR(256),  
                      soapaction VARCHAR(256),  
                      soap_body CLOB(1M))  
    RETURNS CLOB(1M)
```


# UDF - pseudo example

```
Create function get_courses(ccode varchar(20))
returns table (cno varchar(20),
              cdate varchar(20))

language
return
....
....
Soap(out) as
(values soaphttp('http://www.abis.be/soap/servlet/rpcrouter', ccode));
....
select ...
from TABLE (tableEXTRACT(.... soap) ....
;

Select x.cno, x.cdate
from TABLE (get_courses("DB2")) as x;
```

# DB2 as a Web service provider - WORF



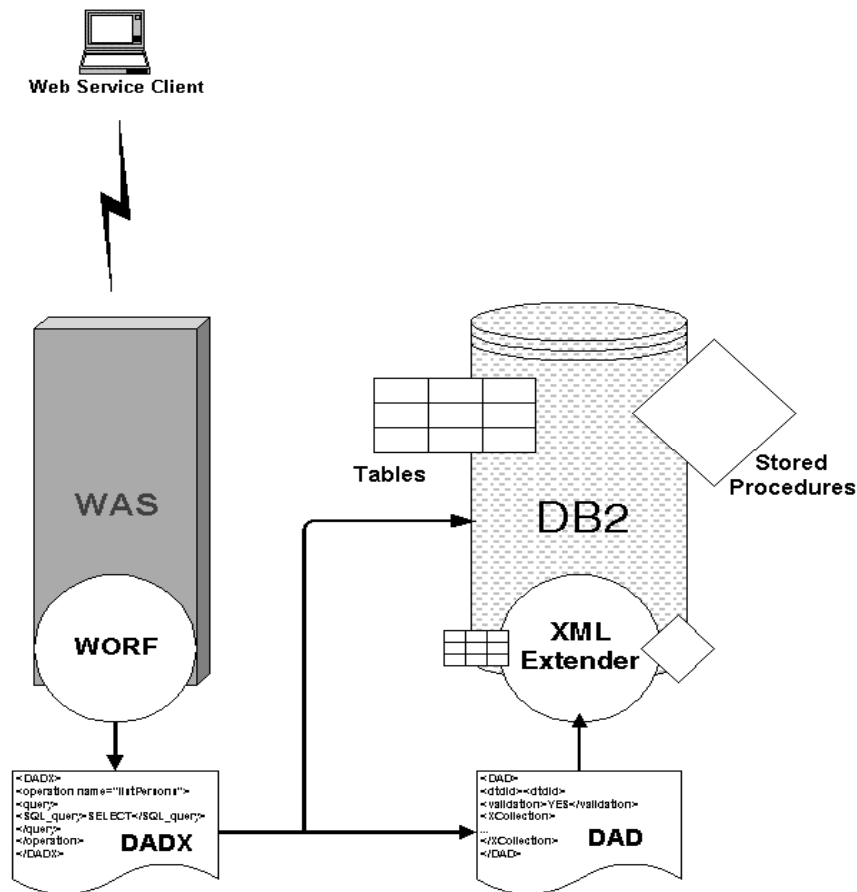
## Web Object Runtime Framework

Provides for:

- ☒ Resource-based deployment and invocation, i.e.
  - ☒ DADX based
  - ☒ optionally other resources that help define the web service
- ☒ Automatic service redeployment
- ☒ Automatic WSDL and XSD generation
- ☒ Automatic documentation
- ☒ Automatic test page generation



# WORF



## WORF:

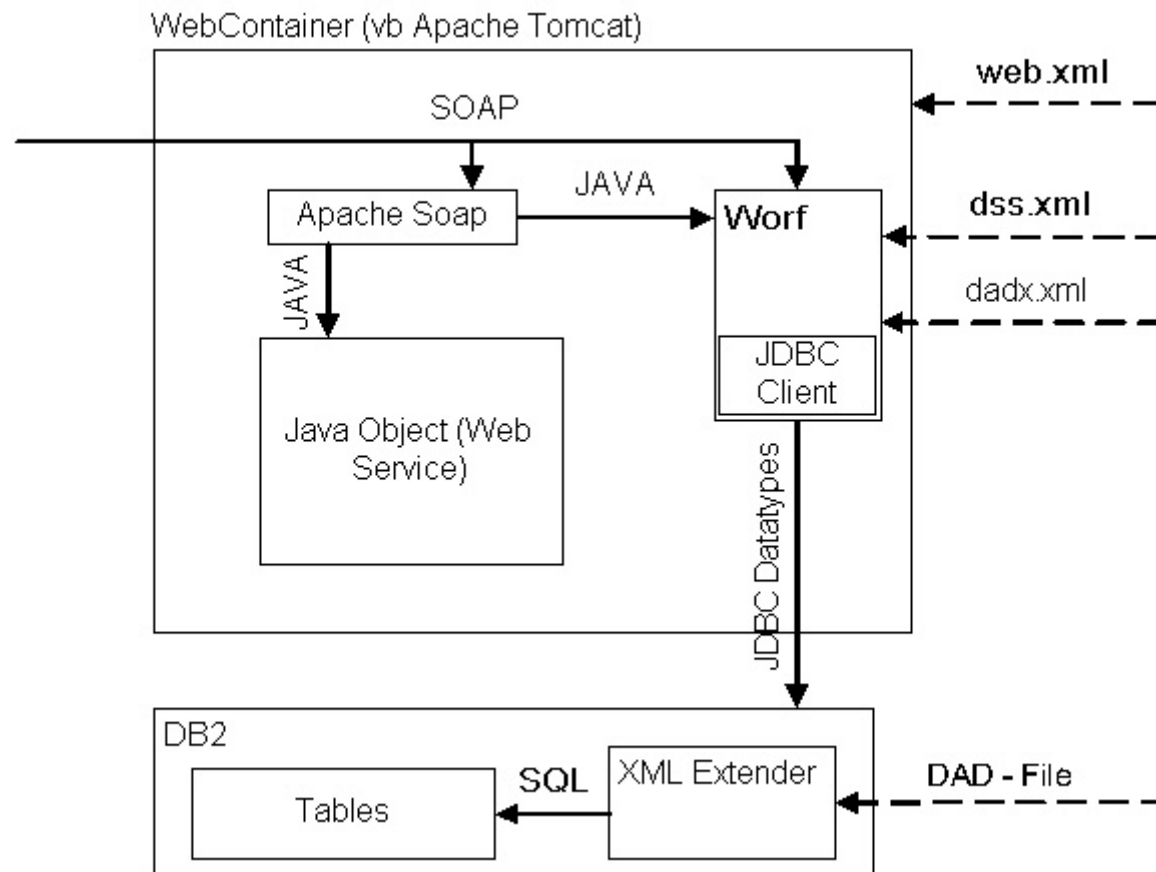
SQL based

select  
update/insert/delete  
call

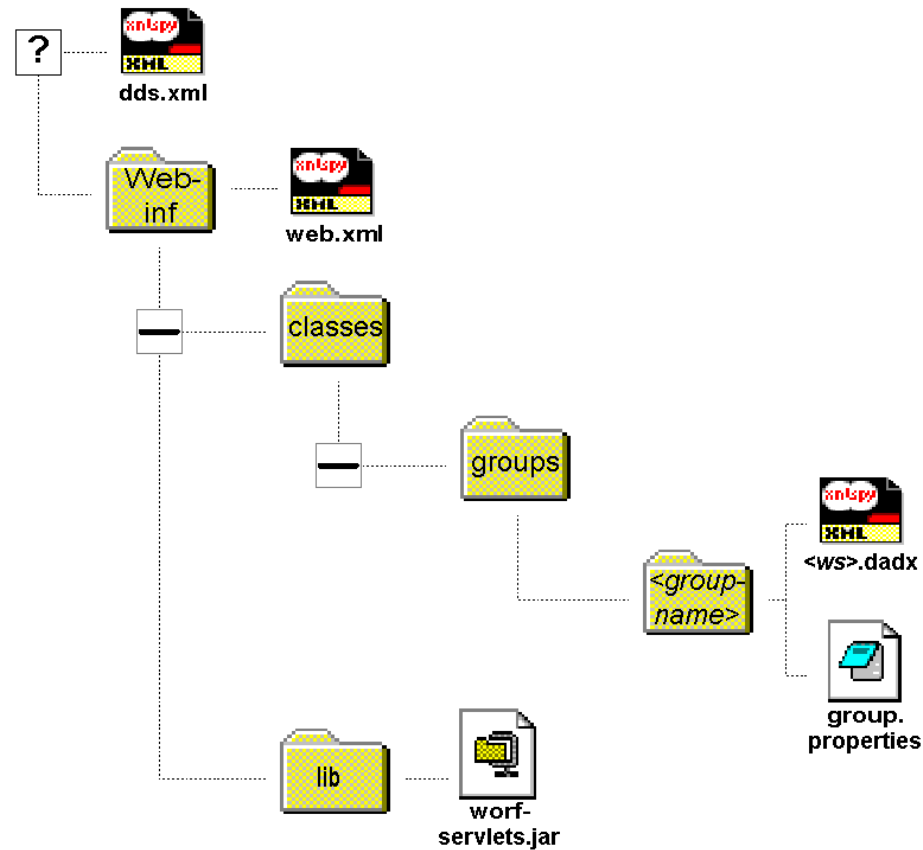
XML based

retrieveXML  
storeXML

# WORF architecture



# WORF configuration files



# DADX



- ☒ XML document
- ☒ DADX for each Web Service
- ☒ Web Service methods: DADX operations
- ☒ Web Service documentation
- ☒ Query description

# DADX - example 1

```
<DADX xmlns="http://schemas...">
  <operation name="listPersons">
    <query>
      <SQL_query>SELECT pfname, pname FROM
                          db2.persons</SQL_query>
    </query>
  </operation>
  <operation name="InsertPerson">
    <call>
      <SQL_call>call db2.InPers(:lname, :com) </SQL_call>
      <parameter name="lname" type="xsd:string"/>
      <parameter name="com" type="xsd:string" kind="out"/>
    </call>
  </operation>
</DADX>
```

# DADX - example 2

```
<DADX xmlns="http://schemas...">
<operation name="InsertPerson">
  <retrieveXML>
    <DAD_ref>get_sessions.dad</DAD_ref>
    <no_override/>
  </retrieveXML>
</operation>
</DADX>
```

# SOAP request processing



Processing sequence:

- ☒ loads the DADX
- ☒ replaces query parameters
- ☒ connects to DB2
- ☒ runs the SQL statement
- ☒ commits the database transaction
- ☒ formats the result into XML
- ☒ returns the response in a SOAP envelope

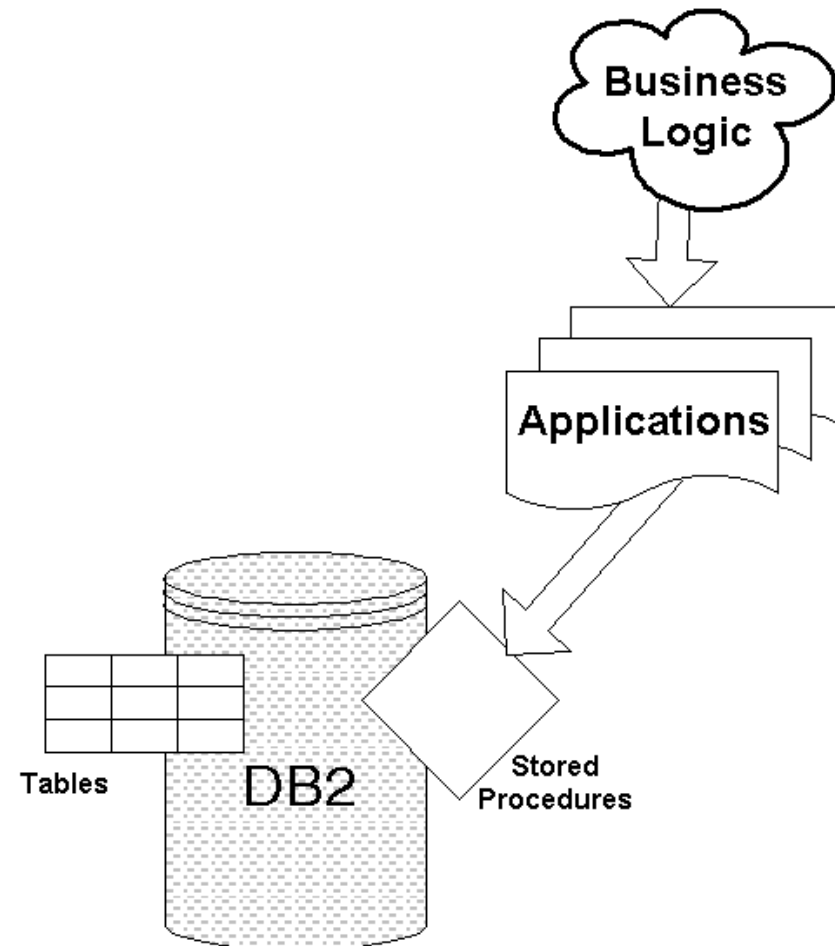
# Stored Procedures

## Generic

- ⌘ Faster execution
- ⌘ Reduced network traffic
- ⌘ Modular programming
- ⌘ Increased security

## Specific

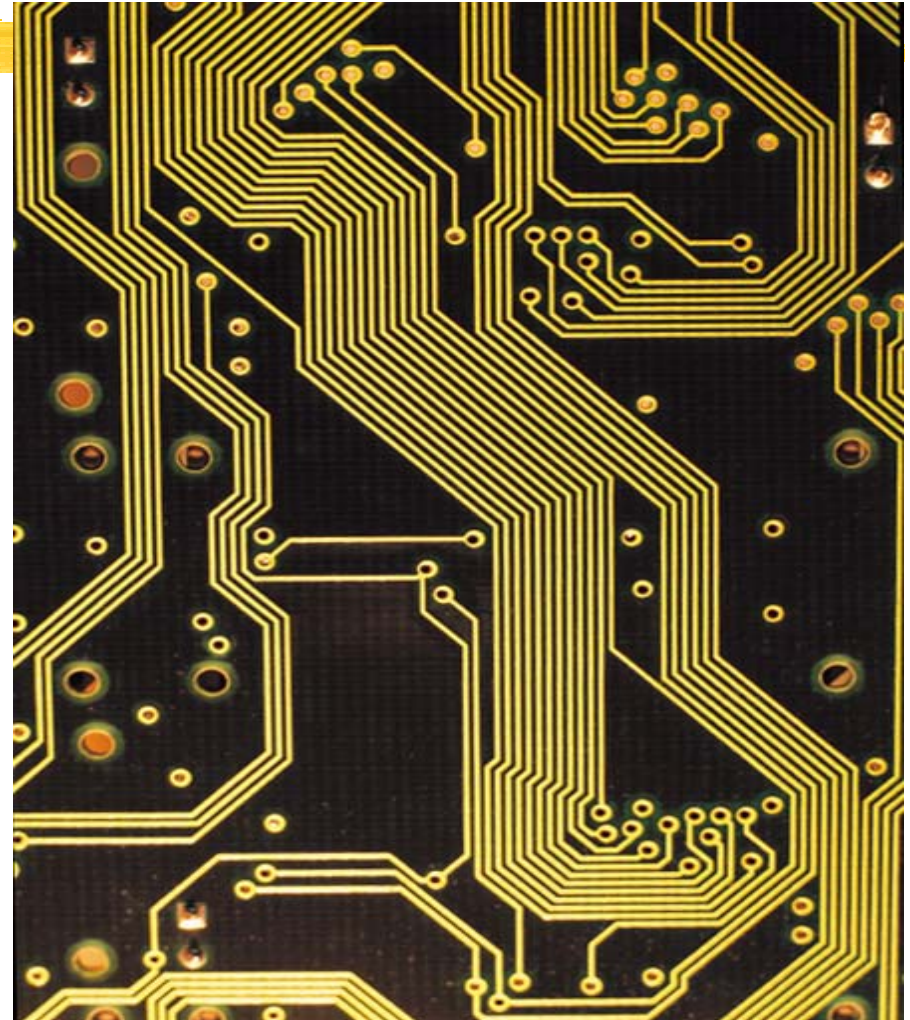
- ⌘ single LUW
- ⌘ abstraction layer
- ⌘ any language
- ⌘ reuse





# Writing Stored Procedures

- ⌘ Languages: Java, C,...
- ⌘ Generation with GUI:
  - ☑ Stored Procedure Builder (v7)
  - ☑ Development Center (v8)
- ⌘ SQL/PL



# Why are Web services important?



Because:

- ⊞ Implementation de-coupled from interface
  - ⊞ any language
  - ⊞ open standard transport 'distributed' technology
- ⊞ Optimized for Internet
  - ⊞ standard based
- ⊞ 'No' assumptions about technology
- ⊞ Backed by key software vendors