TRANSFERRING DATA BETWEEN XML DOCUMENTS AND RELATION DATABASE

David Melkumyan
DESY Zeuthen

March 2005

Contents

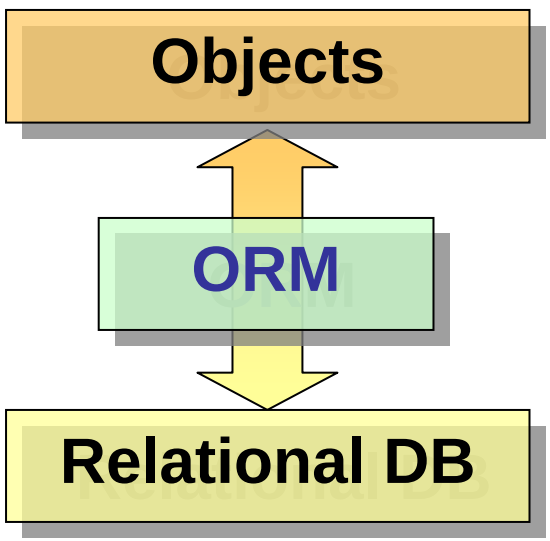
Introduction

- Technologies used at creation of information systems are:
  - *object-oriented programming*
  - *relational databases*
  - *eXtensible Markup Language* (*XML)*
- Combination of object- and not objects-oriented technologies
- Using specialized tools

Introduction

- Last years has been developed a lot of libraries of **object-relational mapping (ORM),** such as **Hibernate**, **JDO**, **iBatis** and some other.

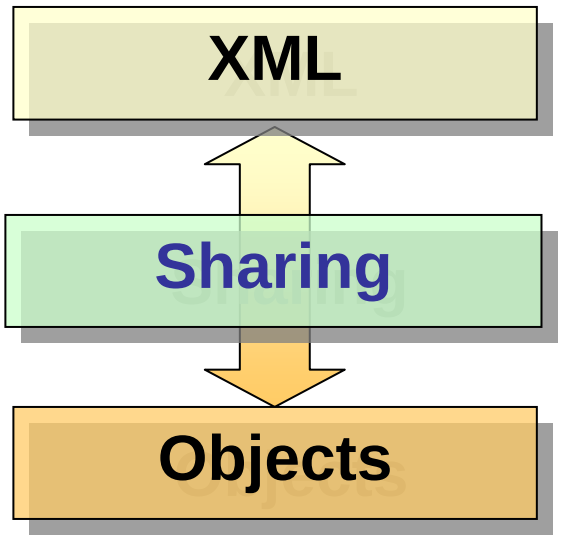
- Description of conformity between objects and database.



- ✓ Keep objects in RDB.
- ✓ Load objects from RDB.
- ✓ Execute queries to RDB (formulated in terms of objects).

Introduction

- Sharing of objects and XML is reduced to the following two tasks:

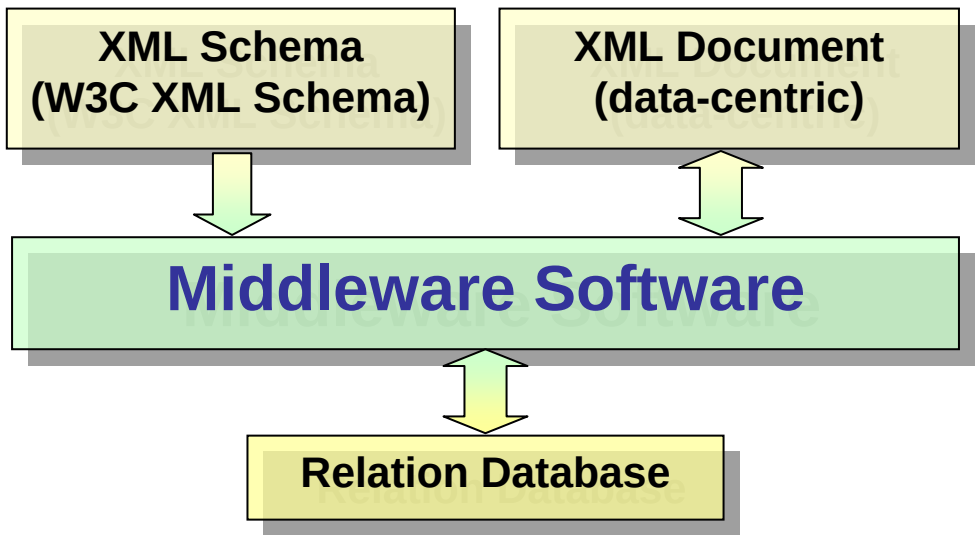


- ✓ Representation of objects as XML (marshalling).
- ✓ Transformation XML to structures of objects (unmarshalling).

- **JAXB**, **Castor**, **XGen**, **XMLBeans**, **Breeze XML Binder**, **JBind**, **JaxMe** and some other.

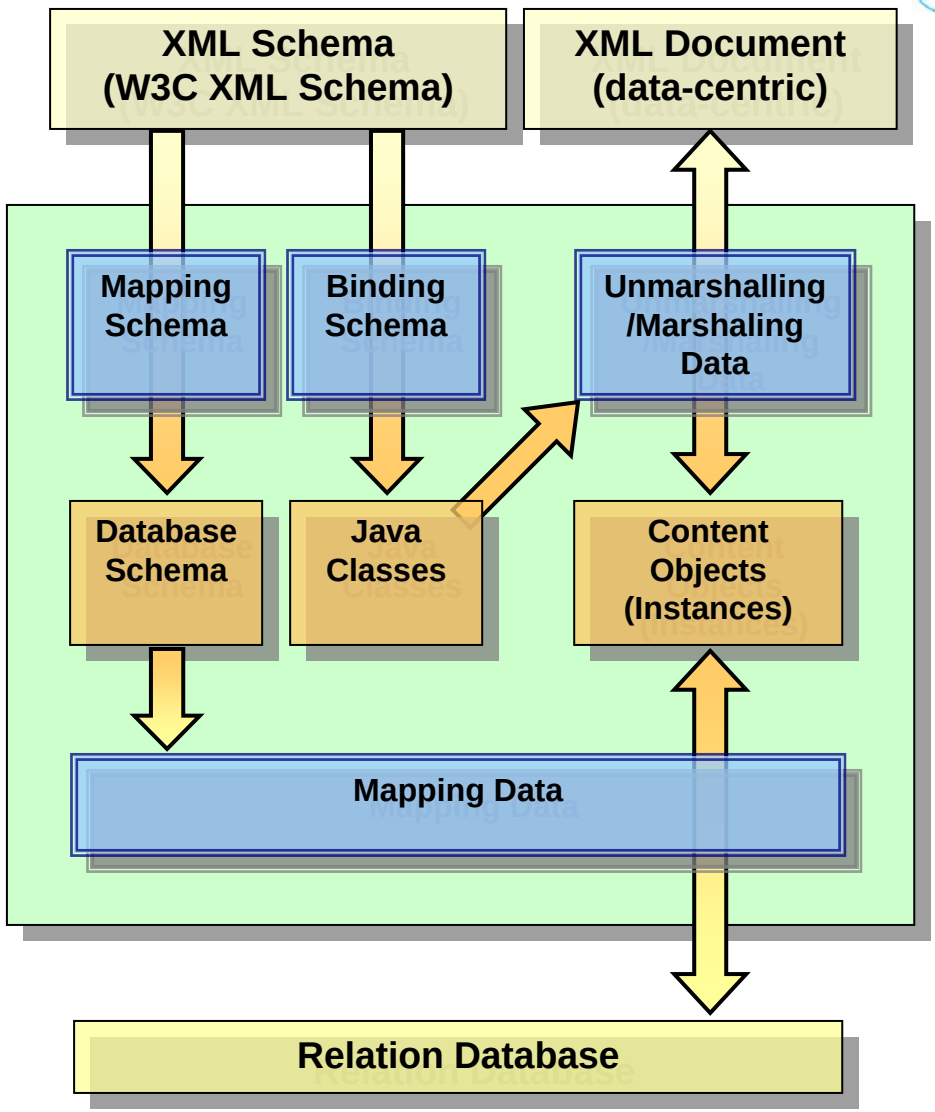
Task

- We shall consider the following scripts of use:
  - store the XML document in a relational database
  - load the XML document from a relational db



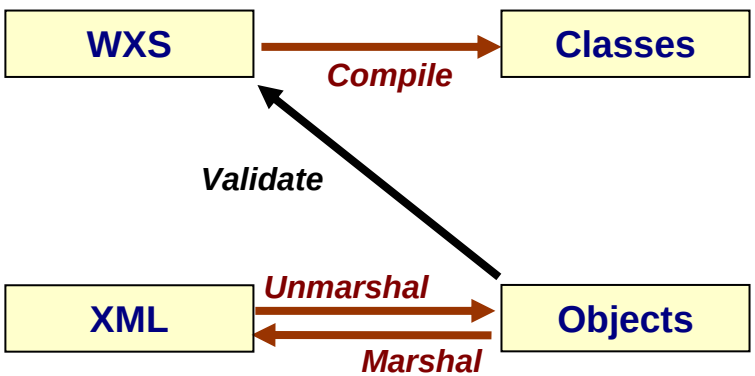
Solution

- Binding XML Schema
- Unmarshal (Marshal) Data
- Mapping Schema
- Mapping Data



Binding schema

- Generate Java code from **W3C XML Schema (WXS)**
- Primary operations of the binding
  - Unmarshalling of an XML into a tree of rellated instances
  - Marshalling of content trees into XML
  - Validatation of content trees
- Steps in binding process



- Generate Classes
- Compile Classes
- Unmarshal
- Validate (optional)
- Process Content tree
- Marshal

Comparing Java Binding Tools

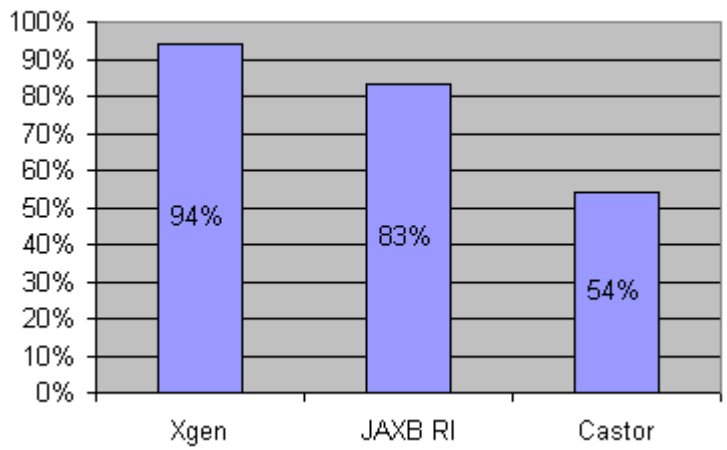
- Currently a many of interesting Binding tools available. The problem is that with so many choices, the technology being relatively new, and many XML Schema features are not supported.
- The binding tools that have been tested in this comparison are:
  - **Castor**
  - **JAXB**
  - **XGen**

Castor, JAXB RI and XGen

- **Castor**
  - Castor is an open source project tool under ExoLab and is available free of charge in binary or source form at http://www.castor.org.
    - The version tested in this comparison is 0.9.5. **JAXB**
- JAXB
  - JAXB is a standard mapping developed by **Sun**. Sun Java Web Services Developer Pack 1.2, which is available for free at http://java.sun.com/xml/jaxb.
    - The version tested in this comparison is 1.0.
- **XGen**
  - XGen is an open source tool produced by **Commerce One**. The tool has been developed with the Castor code as a basis, but with a substantial number of changes in the mapping and the functionality. The tool is included in the Commerce One Conductor DocSOAP XML Developer's Kit available free of charge at http://www.commerceone.com/developers/docsoapxdk.
    - The version tested in this comparison is 6.0.

Over-all test results

- The **code-generation test** exercises two main aspects of the code generation:
  - The XML Schema feature support. Any tool may have weaknesses in that features may be unsupported, or poorly supported.
  - Flaws in the generated Java Code. Once the code has been generated, it is compiled in a java compiler. Code must compile to be usable.
- The **runtime tests** consists of a "roundtrip", that is, converting the XML document to populated Java objects, converting those Java objects back to an XML instance. It also includes comparing the input and output. This is a good test for two important aspects of the ultimate functionality of the generated code.
- It should be noted that this comparison only compares the basic functionality, without any customization of the generated code.
- In spite of the fact that the greatest parameter has XGen tool, in the offered approach has been chosen **JAXB** since his features as much as possible correspond to requirements in realization of specialized schemes.

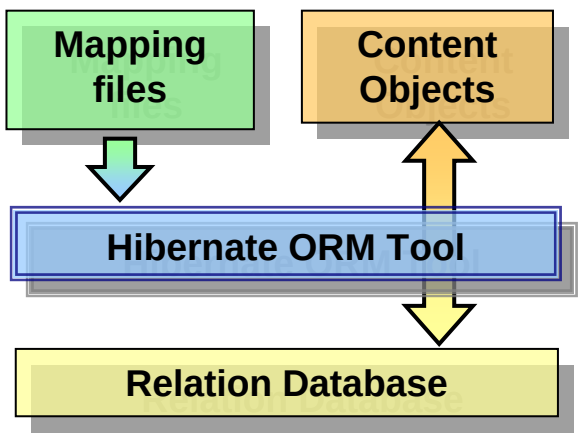


What is not supported in JAXB 1.0

- Schema component: wildcard (*any*)
  - JAXB implementations are not required to unmarshal or marshal XML content that does not conform to a schema that is registered with JAXBContext. However, wildcard content must be handled as detailed in Section 5.9.4 of the *JAXB Specification*.
- Schema component: attribute wildcard (*anyAttribute*)
- Notation declaration
  - Nothing is generated for notations.
- Redefinition of declaration
  - Since redefine is difficult to implement and not frequently used, it may be ignored by a conforming implementation until a future time when its use becomes common.
- Schema component: identity-constraint definition (*key, keyref, unique*)
  - Due to complexities surrounding supporting this feature, specify in a future version.
- Substitution group support:
  - Attributes: complexType.abstract, element.abstract, element.substitutionGroup
- **Note:** Type substitution support is experimental at present. JAXB 2.0 will specify the standard binding for type substitution. The next version of the spec (JAXB 2.0) is being developed as JSR-222.

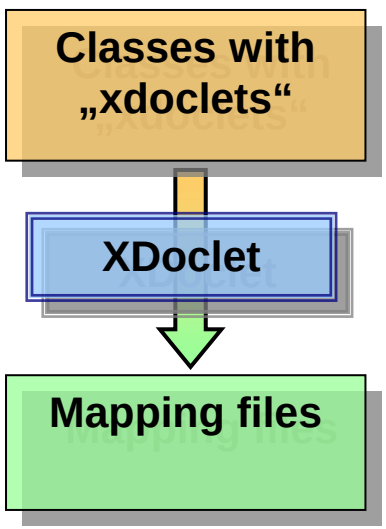
Hibernate ORM Tool

- Hibernate is a persistence service that stores Java objects in relational databases - or provides an object oriented view of existing relational data.
- Hibernate requires an explicit definition of mapping, usually in a form of XML file.
- Consequently, to persist JAXB objects with Hibernate, we need to produce object/relational mapping for these objects.



XDoclet Technology

- XDoclet technology allows to specify mapping properties directly in the source code of the object classes. Source code is annotated with special tags called “xdoclets”.
- XDoclet parses source code and generates required artifacts (like Hibernate mapping).
- XDoclet is a perfect point to bring JAXB and Hibernate together.

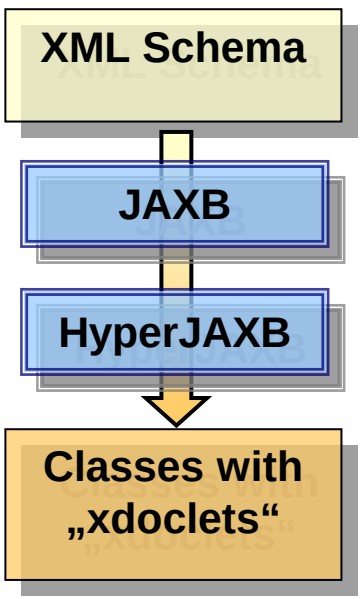


- The only missing fragment is automatic generation of “xdoclets” in JAXB objects.
- This missing fragment is implemented by the **HyperJAXB** package.

HyperJAXB Package

- **HyperJAXB** extends the code generated by Sun's reference implementation of JAXB with Hibernate “xdoclet” annotations.

- Annotated sources of JAXB objects are processed into object/relational mapping for Hibernate.



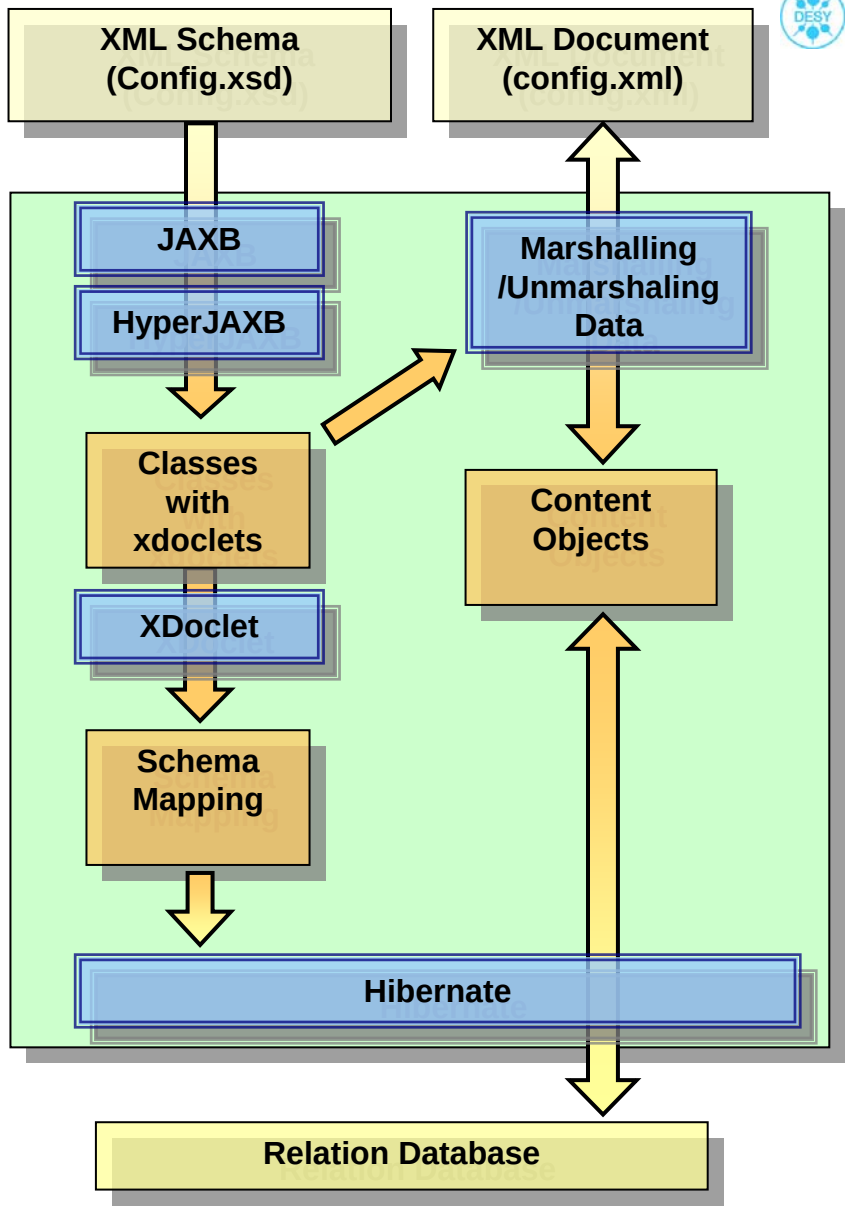
- Combination of JAXB RI, HyperJAXB and Hibernate tools allows automatically generating the following artifacts:
  - source code of JAXB objects with Hibernate XDoclet annotations
  - object/relational mapping for Hibernate
  - database schema for the target database

Sample

- The sample demonstrates a full XML-objects-database roundtrip:
  - initial XML document is loaded and unmarshalled into a JAXB object
  - the object is saved into the database
  - the object is loaded from the database
  - loaded object is marshalled into an XML document (final document)
  - initial and final documents are compared for similary

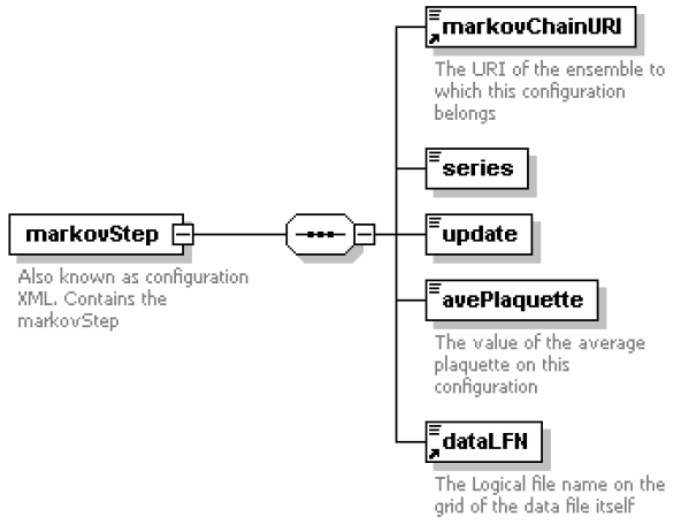
Sample

- Bind XML Schema
- Generate annotated source code (xdoclets)
- Specify mapping properties
- Create content objects
- Persist JAXB objects with Hibernate



Sample

- Part of XML Schema (config.xsd)



- Part of XML Document (config.xml)

```
<markovStep>
  <markovChainURI>
    www.lqcd.org/ildg/qcdsf/b5p40kp13610-24x48
  </markovChainURI>

  <series>561</series>
  <update>1500</update>

  <avePlaquette>0.5612510601</avePlaquette>

  <dataLFN>
    qcdsf_b5p40kp13610-24x48_bqcd.561.1.1.01500.tar
  </dataLFN>
</markovStep>
```

Sample

- Context of one of generated source files (MarkovStepTypeImpl.java) listed below

```
/*
  @hibernate.mapping auto-import="false"
  @hibernate.class table="MarkovStepType"
*/
public class MarkovStepTypeImpl implements
org.lqcd.ildg.qcdml.config1.MarkovStepType, com.sun.xml.bind.JAXBObject,
org.lqcd.ildg.qcdml.config1.impl.runtime.UnmarshallableObject,
org.lqcd.ildg.qcdml.config1.impl.runtime.XMLSerializable,
org.lqcd.ildg.qcdml.config1.impl.runtime.ValidatableObject
{
    protected String _MarkovChainURI;
    protected boolean has_AvePlaquette;
    protected double _AvePlaquette;
    protected String _Update;
    protected String _DataLFN;
    protected String _Series;

    private   String idInternal;
   . . .
}
```

Sample

- Context of one of Hibernate mapping files (MarkovStepTypeImpl.hbm.xml) :

```
<?xml version="1.0" encoding="UTF-8" ?>
- <hibernate-mapping auto-import="false">
  - <class name="org.lqcd.ildg.qcdml.config1.impl.MarkovStepTypeImpl" table="MarkovStepType" dynamic-update="false"
      dynamic-insert="false" select-before-update="false">
    - <id name="idInternal" column="idInternal" type="java.lang.String" length="32" unsaved-value="null">
        <generator class="uuid.hex" />
      </id>
      <property name="markovChainURI" type="java.lang.String" update="true" insert="true" access="property"
        column="ildg_markovChainURI" />
      <property name="avePlaquette" type="double" update="true" insert="true" access="property" column="ildg_avePlaquette" />
      <property name="update" type="java.lang.String" update="true" insert="true" access="property" column="ildg_update" />
      <property name="dataLFN" type="java.lang.String" update="true" insert="true" access="property" column="ildg_dataLFN" />
      <property name="series" type="java.lang.String" update="true" insert="true" access="property" column="ildg_series" />
    - <joined-subclass name="org.lqcd.ildg.qcdml.config1.impl.MarkovStepImpl" table="MarkovStep" dynamic-update="false"
        dynamic-insert="false">
        <key column="parentid" />
      </joined-subclass>
    </class>
  </hibernate-mapping>
```

- Part of context of Hibernate properties file (hibernate.export.properties) :

  hibernate.dialect net.sf.hibernate.dialect.MySQLDialect

  hibernate.connection.driver_class org.gjt.mm.mysql.Driver

  hibernate.connection.driver_class com.mysql.jdbc.Driver

  hibernate.connection.url jdbc:mysql://mysqlsrv.ifh.de:3306/test_hibernate

  hibernate.connection.username mdavid

  hibernate.connection.password password

Sample

- Relation Table :

MarkovStepType					
Field	Type	Null	Key	Default	Extra
idInternal	varchar(32)		PRI		
ildg_markovChainURI	varchar(255)	YES		NULL	
ildg_avePlaquette	double	YES		NULL	
ildg_update	varchar(255)	YES		NULL	
ildg_dataLFN	varchar(255)	YES		NULL	
ildg_series	varchar(255)	YES		NULL	

Sample

- Unmarshall and saving

```
// Unmarshall the document
final MarkovStep ms = (GaugeConfiguration) unmarshaller.unmarshal(document);
// Open the session, save object into the database
final Session saveSession = sessionFactory.openSession();
// Save id for the later use
final String id = saveSession.save(ms);
saveSession.flush();
// Close the session
saveSession.close();
```

- Loading and marshalling

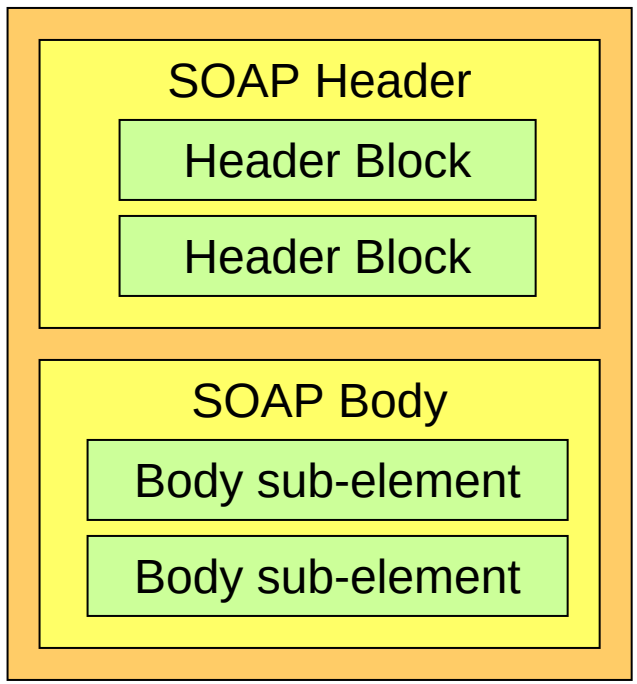
```
// Open the session, load the object
final Session loadSession = sessionFactory.openSession();
final MarkovStep loaded = (MarkovStep) loadSession.load(MarkovStepImpl.class, id);
loadSession.close();
// Marshall loaded object into the document
final Document loadedDocument = documentBuilder.newDocument();
marshaller.marshal(loaded, loadedDocument);
```

- Comparing XML Document

```
Document loadedDoc = (new DOMReader()).read( loadedDocument );
Document savedDoc = (new DOMReader()).read( savedDocument );
DetailedDiff d = new DetailedDiff(new Diff(savedDoc.asXML(), loadedDoc.asXML()));
System.out.println(d.similar() ? "XML Documents are similar!“ : "XML Documents are not similar!");
```

Working Application

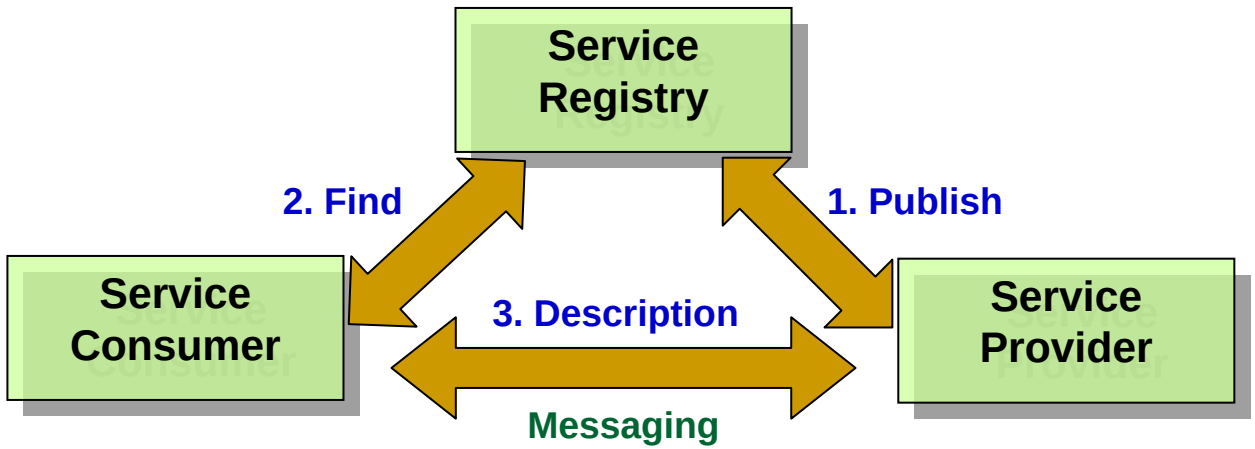
- SOAP ("Simple Object Access Protocol") is:
  - A XML based communication protocol.
  - A format for encoding datatype instances.
  - Used with XML Schema.
  - Transport independent.
- Apache Axis is an implementation of the SOAP submission to W3C.
- Axis has proven itself to be a reliable and stable base on which to implement Web Services.



Working Application

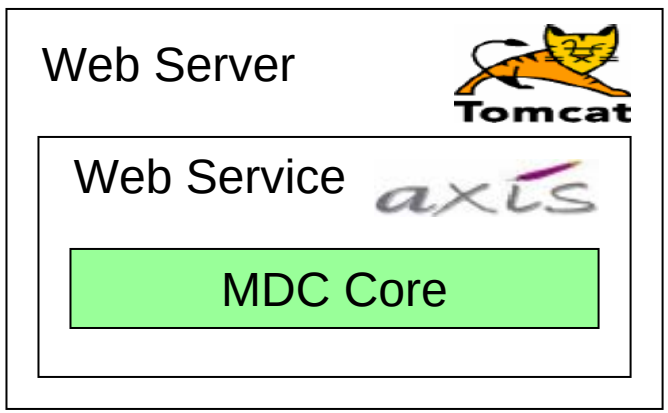
- Web Service as an application that:
  - Exposes its operations to other application via open interoperable standards
  - Communicates with other applications over the Internet
  - Can be implemented in any programming language or platform

- Web Service Model



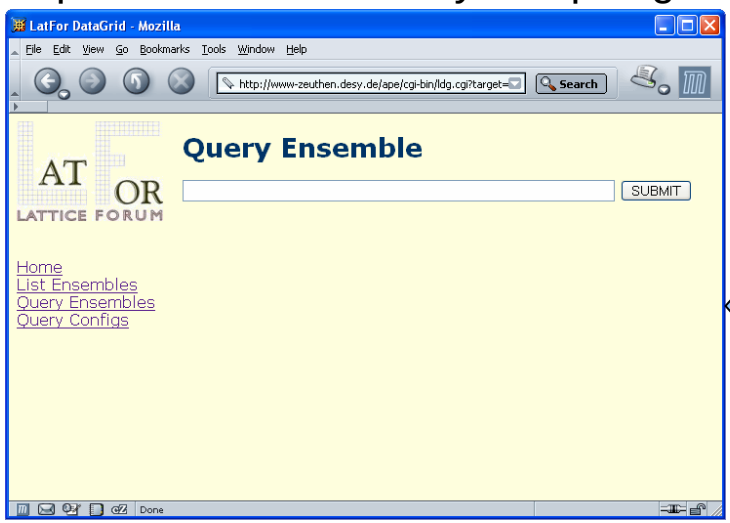
Working Application

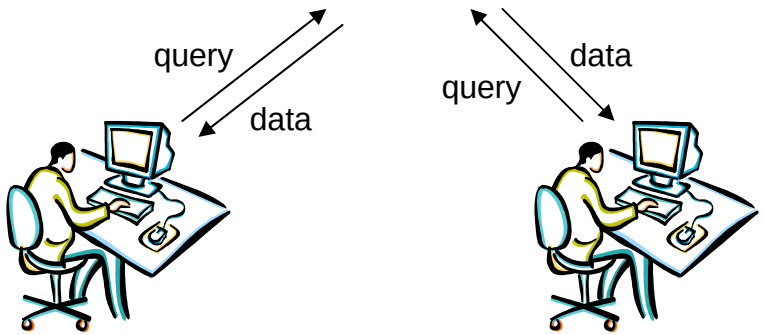
- MDC Web Service uses:
  - Jakarta **Tomcat** as the java “web server”.
  - Apache **Axis** is the implementation of the XML-based SOAP protocol for exchange of information in a distributed environment.

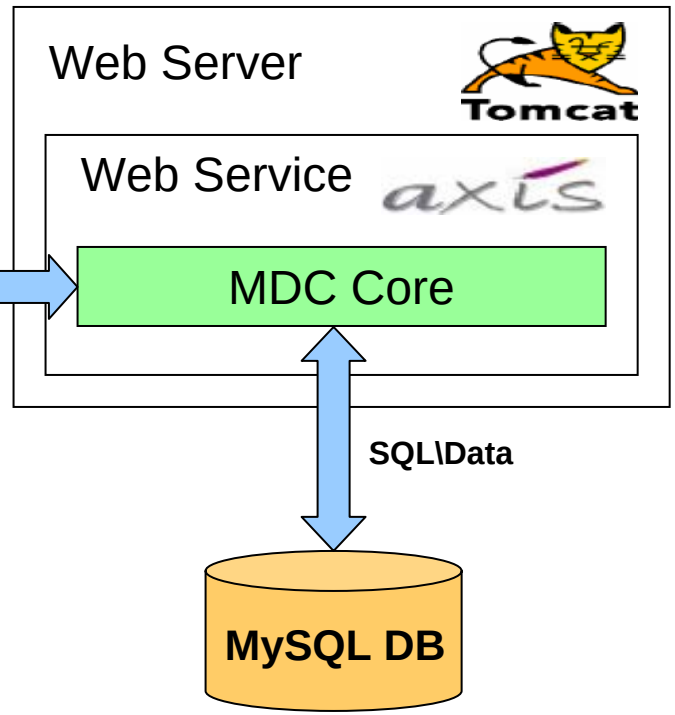


Working Application

http://www-zeuthen.desy.de/ape/cgi-bin/ldg.cgi?target=index







59 tables

~ 10,000 records

Next steps

- Grid Security Infrastructure (GSI)
- Secure web services using **GSI**
  - Based on Generic Security Services API (GSS-API)
  - Uses an extension to X509 certificates
- Provides a mechanism to:
  - Authentication based on certicates.
  - Authorization: establishing rights.

Acknowledgements

- HyperJAXB and the sample application use or include the following products, libraries and tools:
- JAXB and XJC; Java Architecture for XML Binding (JAXB) provides a convenient way to bind an XML schema to a representation in Java code. You can find Jaxb at http://java.sun.com/xml/jaxb/.
- Hibernate; Hibernate is a high performance object/relational persistence and query service for Java. You can find Hibernate at http://www.hibernate.org
- Jakarta Commons libraries (Collections, Lang, Logging, BeanUtils, JXPath); The Jakarta Commons libraries are dedicated to creating and maintaining reusable Java components. The libraries are:
- Collections - provides a suite of classes which extend the Java Collections Framework.
- Lang - This library provides extra funtionality for classes in java.lang.
- Logging - This library is a wrapper around a variety of logging API implementations.
- BeanUtils - provides easy-to-use wrappers around the Java reflection and introspection APIs.
- JXPath - provides utilities for manipulating Java classes that conform to the JavaBeans naming conventions using the XPath syntax. The library also supports maps, DOM and other object models. Jakarta Commons can be found at http://jakarta.apache.org
- DOM and SAX; DOM (Document Object Model) is a language neutral programming interface for XML-processing. Accessing the XML is done via the object tree - rather than running through the document sequentially as done in SAX (Simple API for XML). DOM is well suited for smaller documents since the access is more comfortable while SAX is better suited for bigger documents. While DOM can create XML documents SAX can not. You can find DOM at http://www.w3.org/DOM/ and SAX at http://www.saxproject.org
- Dom4j; Dom4j is a open source library for working with XML, XPath and XSLT for the Java platform using the Java Collenctions Framework. It has full support for DOM, SAX and JAXP. You can find Dom4j at http://www.dom4j.org/
- Xalan and Xerces; Xalan and Xerces are both part of the Apache XML project (http://xml.apache.org). Xalan is a XSLT stylesheet processor and Xerces is a XML parser. Yet, Xerces can not only parse XML files but also generate them. Xalan for Java can be found at http://xml.apache.org/xalan-j/index.html, Xerces for Java at http://xml.apache.org/xerces2-j/index.html.
- CGLib; CGLib, the Java Code Generation Library, can be used to extend Java classes and implement Java interfaces at runtime. You can find CGLib at http://cglib.sourceforge.net
- HSQL database; The Hypersonic SQL database is a Java database engine with a standard SQL and JDBC interface. Through it are compactness, it is well suited for usage as a server in applets and applications. You can find the Hypersonic SQL database at http://hsql.sourceforge.net
- Programmer's Friend components; The Programmer's Friend class library contains a large collection of Java utility classes which can simplify programming compared to just using the JDK. Programmer's Friend can be found at http://www.programmers-friend.org/
- JUnit; JUnit is a regression testing framework for developers who implement unit test cases. JUnit has a console user interface as well as a GUI. You can find JUnit at: http://www.junit.org
- XML Unit; XML Unit extends JUnit (and NUnit) to enable unit testing of XML. It can compare a control XML document to a test document, validate documents and compare results of XPath expressions. XML Unit can be found at http://xmlunit.sourceforge.net/
- HyperJAXB; HyperJAXB is and add-on for Sun's reference implementation of JAXB (Java Architecture for XML Binding). HyperJAXB automatically generates Hibernate mappings for classes produced by JAXB effectively providing JAXB objects with relational persistence capabilities. HyperJAXB can be found at https://hyperjaxb.dev.java.net/

The End

- Thank you…