B.I.R.O.

Best Information through Regional Outcomes

A European Public Health Project, DG-SANCO, 2005-2008

B.I.R.O. Database Engine

Valentina Baglioni

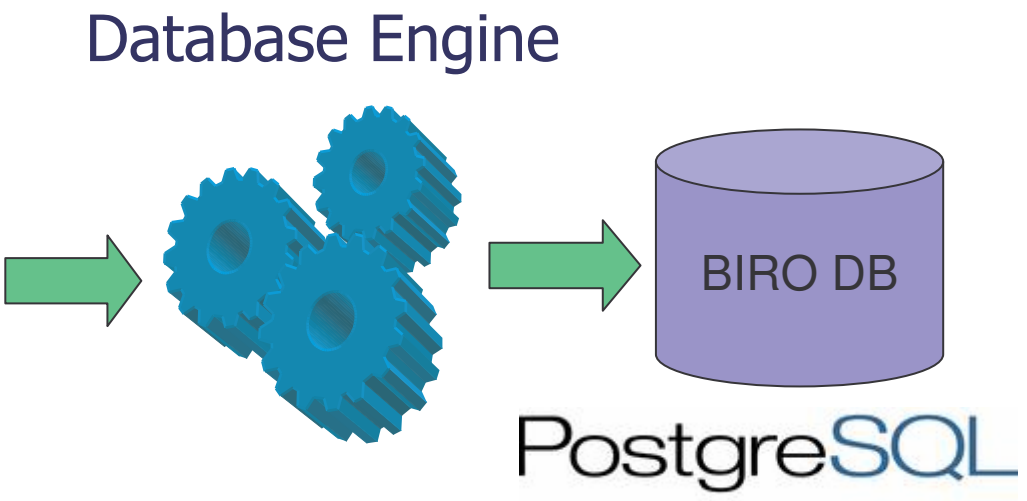
B.I.R.O Rome meeting

3rd December 2007

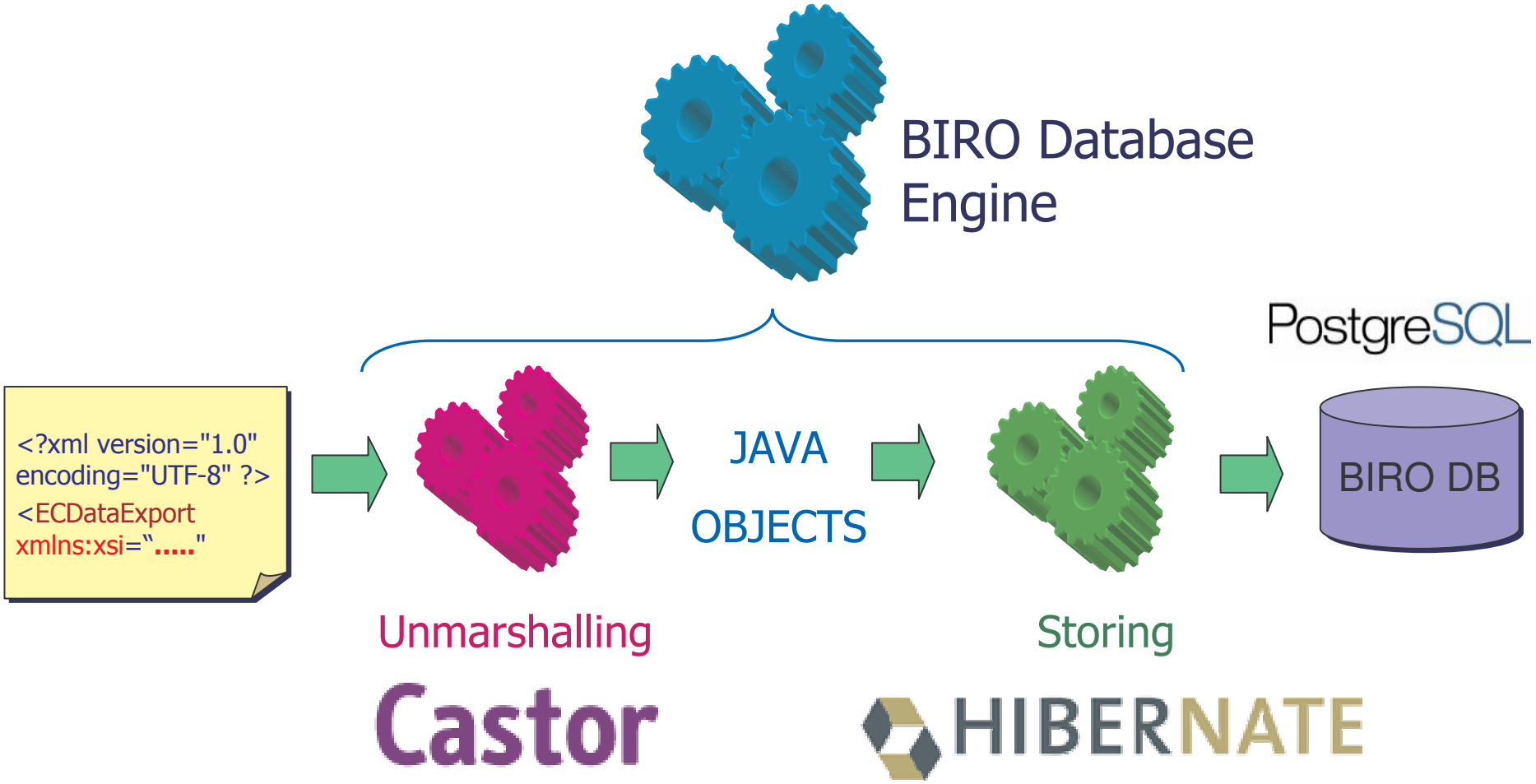
Database Engine task

- BIRO Database Engine parses XML files coming from each centre and stores data into BIRO database

```
<?xml version="1.0" encoding="UTF-8" ?>
<ECDataExport xmlns:xsi="....."
   xsi:noNamespaceSchemaLocation="...">
- <Patient>
- <Profile>
  <ProfileFieldName>TYPE_DM</ProfileFieldName>
  <ProfileFieldValue>2</ProfileFieldValue>
  </Profile>
- <Profile>
  <ProfileFieldName>DT_DIAG</ProfileFieldName>
  <ProfileFieldValue>01/01/1983</ProfileFieldValue>
  </Profile>
- <EpisodeData>
  <EpisodeDate>2004-07-06</EpisodeDate>
```

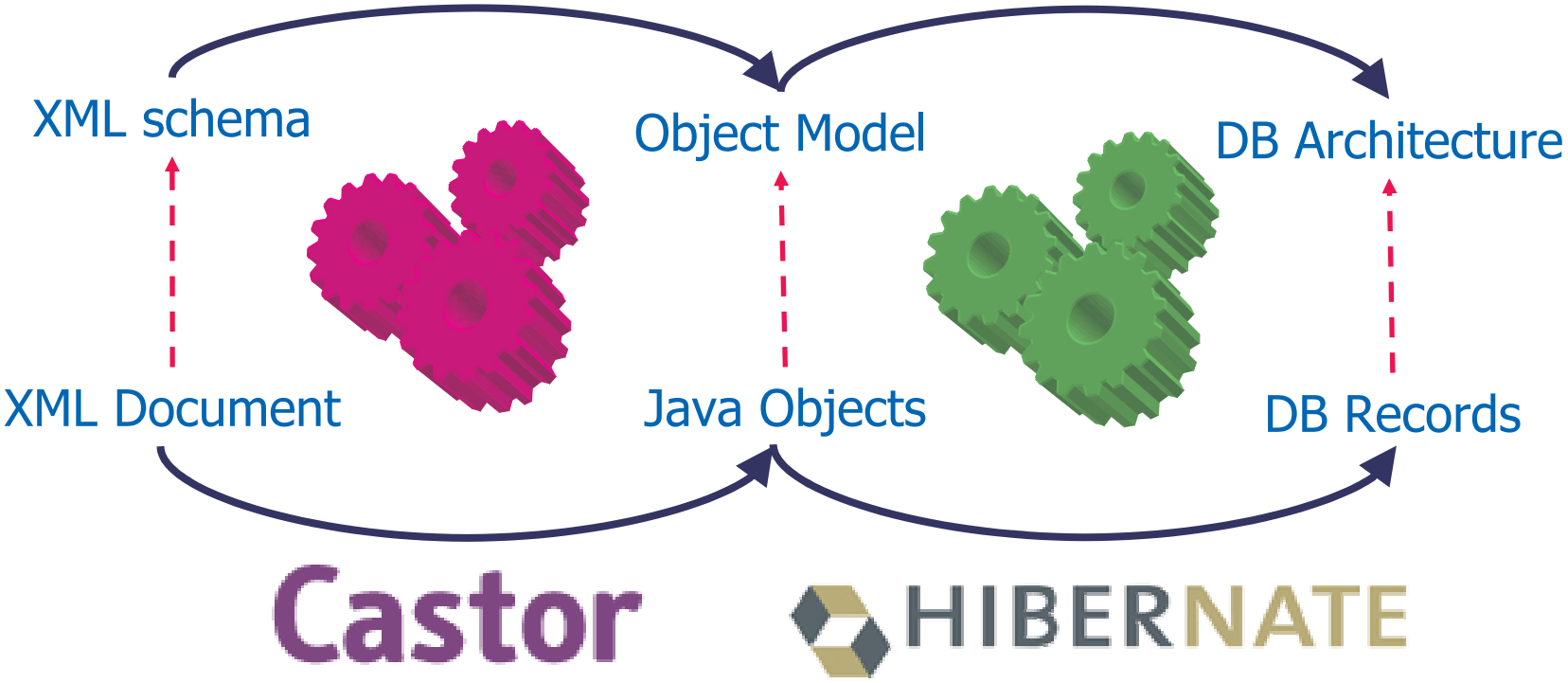


Inside BIRO Database Engine (1)



Inside BIRO Database Engine (2)

- The complete process



An alternative approach

- standard ways to access XML files make use of W3C DOM API and SAX API
- they allow to manipulate XML at low level and extracting pieces of data
- they deal directly with the structural components (elements, attributes) of XML files
- we don't need to insert the intermediate step of Java Objects Model
- we can use standard API to extract strings from XML files and then execute queries to directly store them into DB

Comparison between approaches (1)

using standard API	using frameworks
↓ Write custom utilities to access the data is a laborious task, error-prone, and tricky to test and debug	↑ A very little effort is required to use frameworks, code is highly readable
↓ API deal with structure of XML file	↑ Creating Java Objects allow developers to deal with the logical model of data instead of their representation
↓ API parse the XML files and return simply strings which need to be converted into proper format to be manipulated	↑ Frameworks automatically convert string into desired format and also validate the data

Comparison between approaches (2)

using standard API	using frameworks
↓ Code is very difficult to mantain and modify if changes to XML schema are necessary. Difficulties grow if XML schema is large.	↑ It is very easy to maintain and modify the DB engine using frameworks even if XML schema is large because all changes at low level are transparent to the user
↑ Working on XML files at low level allow to obtain high performance regarding both time and memory usage.	↓ Creation of Java Objects requires long time and much memory. Performance is worse.

BIRO DB Engine maintainability

- If the XML schema changes, the unmarshalling section automatically creates the Java Object Model from the new schema

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
        "-//Hibernate/Hibernate Mapping DTD
3.0//EN"
        "...">
<hibernate-mapping>
    <class name="export.Data" table="DATA">
        <id name="ID" column="DATA_ID">
        <generator class="native"/> </id>
        <property name="EpisodeFieldName"
type="export.types.BIRODataSetUserType"
column="EPISODE_FIELD_NAME"/>
        <property name="EpisodeFieldValue"
type="string"
column="EPISODE_FIELD_VALUE"/>
   </class>
</hibernate-mapping>
```

- Developer has got to change the file which contains the mapping betweeen the classes of the Model and the tables of the Database.
- The storing section automatically changes the database structure according to the new mapping file and stores data again

BIRO Database Engine Performance

- In order to improve BIRO DB engine performance:
  - we chose to import data of each patient separately (BIRO Adaptor produces many little XML files intead of a big one which contains all patients)
  - we forced Castor to use a different parsing tool named Piccolo that is faster than the native one (Xerces)
- During the last test on Umbria Data, BIRO DB Engine stored data of about 60.000 patients in 20 minutes
- In the next weeks we are going to work hard to improve BIRO DB Engine performance