Characteristics of the proposed approach

Hongtao Ren, Marek Makowski

Outline:

- Two illustrative use cases
- Scenarios (instances of the use cases)
- Data structure for interface
- Consuming Web-services

Simple use cases (energy supply: devices)

- Use Case: Upload_devices
- Description: upload devices data
- User: data manager
- Steps:
  1. user prepares data on devices
  2. user uploads the data
  3. user gets confirmation

- Use Case: Retrieve_devices
- Description: retrieve data on devices
- User: operator
- Steps:
  1. user sends a request for retrieving devices
  2. user gets data on devices

Scenarios (example of GUI)

Upload device(s):

Retrieve devices:

Show 20 entries Search:

Name	Investment cost	OM cost	Capacity	Total annual cost
Device1	222.0	212.0	122.1	342.1
Device2	121.0	112.0	23.9	134.2

Showing 1 to 2 of 2 entries

First Previous 1 Next Last

Scenarios (XML docs)

Upload_devices:

<storeDevicesRequest>
<device><name>device1</name><inv_cost>222.0</inv_cost><om_cost>212.0</om_cost><capacity>122.1</capacity></device>
<device><name>device2</name><inv_cost>121.0</inv_cost><om_cost>112.0</om_cost><capacity>23.9</capacity></device>
</ storeDevicesRequest>

<storeDevicesResponse> OK </ storeDevicesResponse >

Retrieve_devices:

<getDevicesRequest/>

<getDevicesResponse>
<device><name>device1</name><inv_cost>221.0</inv_cost><om_cost>212.0</om_cost><capacity>122.1</capacity>device>
<device><name>device2</name><inv_cost>121.0</inv_cost><om_cost>112.0</om_cost><capacity>23.9</capacity></device>
</getDevicesResponse>

XML schema

```
<xs:element name="storeDevicesRequest">
        <xs:complexType>
          <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="enrima:device"/>
          </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="storeDevicesResponse">
        <xs:complexType>
            <xs:all>
                <xs:element  name="result" type="xs:string" />
            </xs:all>
        </xs:complexType>
    </xs:element>
```

```
<xs:element name="getDevicesRequest" />
    <xs:element name="getDevicesResponse">
        <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" ref="enrima:device"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
```

```
<xs:element name="device">
        <xs:complexType>
          <xs:sequence>
                <xs:element name="name" type="xs:string" />
                <xs:element name="inv_cost" type="xs:double"/>
                <xs:element name="om_cost" type="xs:double"/>
                <xs:element name="capacity" type="xs:double"/>
                <!--  <xs:element name="tac" type="xs:double"/>     calculated by constructor-->
          </xs:sequence>
      </xs:complexType>
  </xs:element>
```

Consuming Web-services

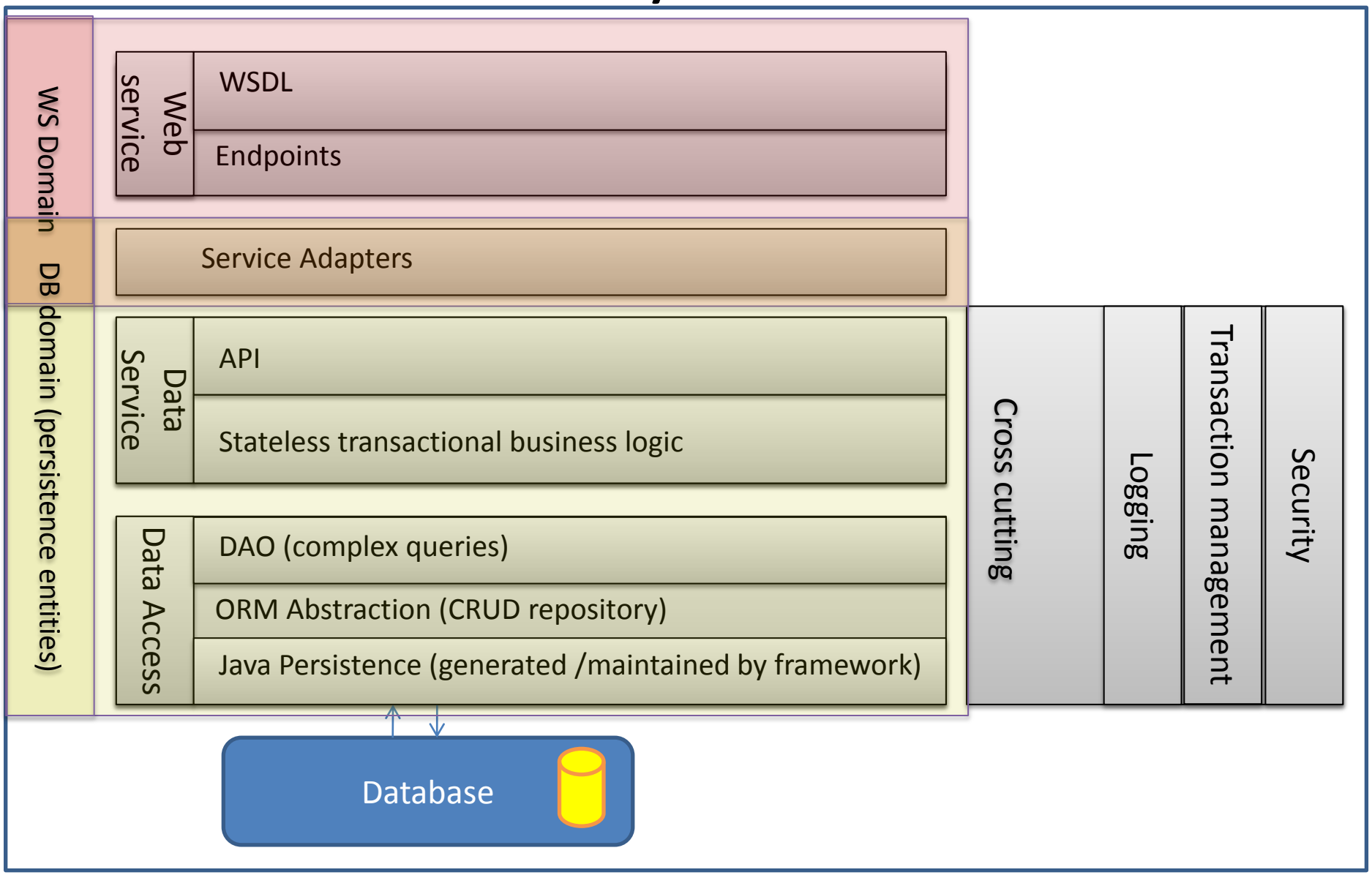
Steps:

1. Get definition URL :
   http://enrima.iiasa.ac.at/device-ws/deviceService.wsdl
2. Get available operations :
   getDevices, storeDevices
3. Get Endpoint :
   http://enrima.iiasa.ac.at/device-ws/deviceService/

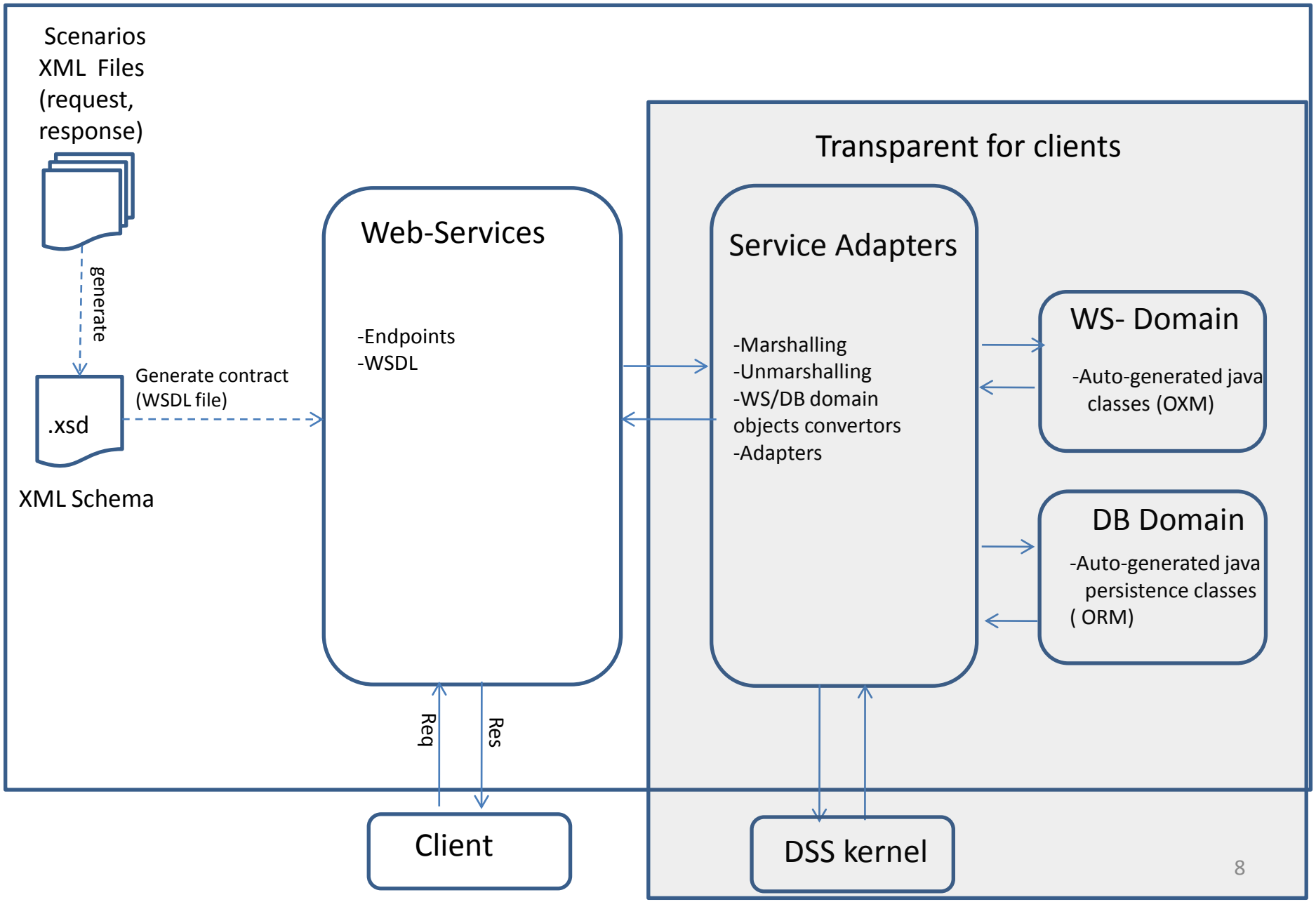
Clients

- GUI : java, javascript (jquery, dojo, etc), flex
- Solvers: C++, Java, Matlab
- SOAP testing tools: SOAPUI, Web king, XML spy, etc.
- …

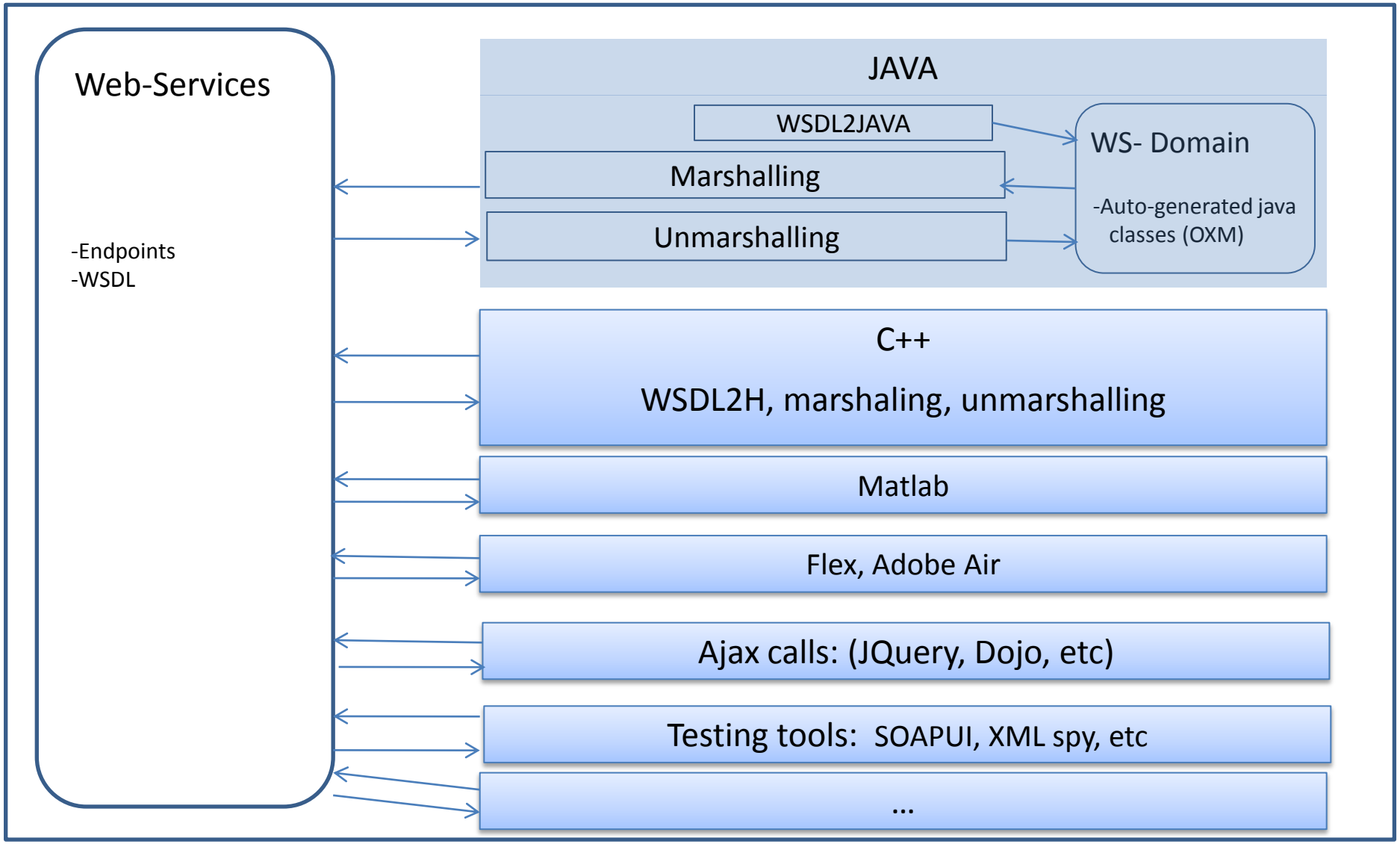
DSS layers


WS Domain
DB domain (persistence entities)
Web service
WSDL
Endpoints
Service Adapters
Data Service
API
Stateless transactional business logic
Data Access
DAO (complex queries)
ORM Abstraction (CRUD repository)
Java Persistence (generated /maintained by framework)
Cross cutting
Logging
Transaction management
Security
Database

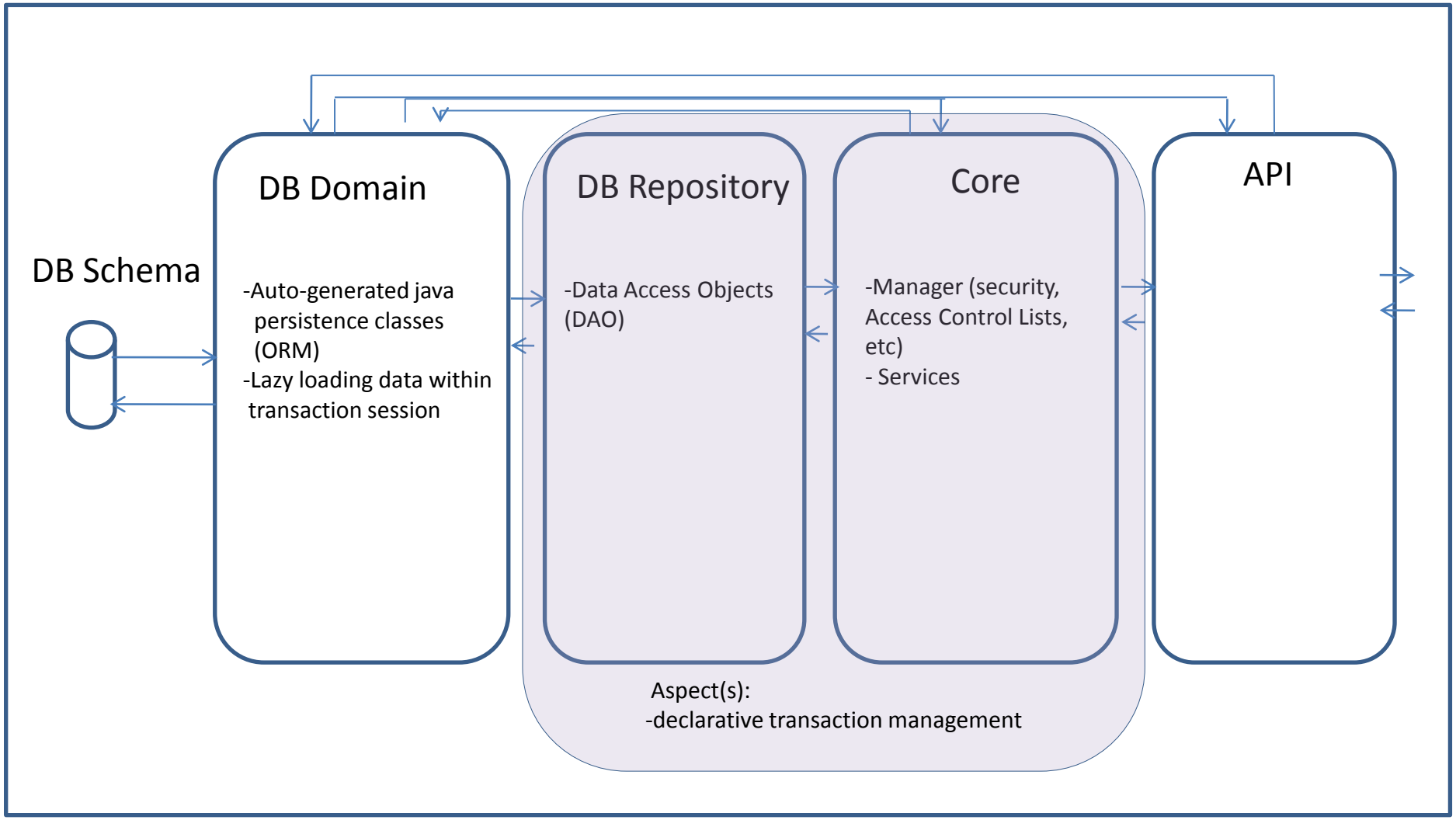
DSS Web-services



DSS clients


Web-Services
-Endpoints
-WSDL
JAVA
WSDL2JAVA
Marshalling
Unmarshalling
WS- Domain
-Auto-generated java classes (OXM)
C++
WSDL2H, marshaling, unmarshalling
Matlab
Flex, Adobe Air
Ajax calls: (JQuery, Dojo, etc)
Testing tools: SOAPUI, XML spy, etc
...

DSS-kernel (transparent for clients)



Summary

- Modularity
- Multi-layer
- Interface of client through web-services
- DB structure changes hidden for clients
- XML schema changes decoupled from DB structure