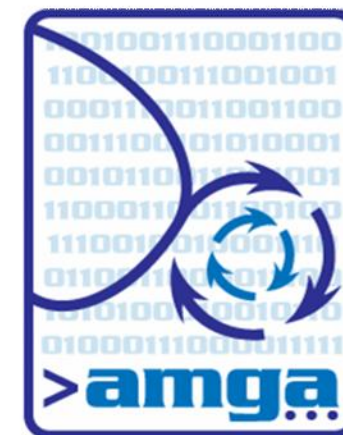


## Integration of WS-DAIR Interface in AMGA

*S. Ahn, S. Hwang, N. KIM (KISTI)*

*A. Calanducci (INFN)*

*B. Kobliz (CERN)*



e-infrastructure



**I. Overview of AMGA**

**II. Integration of WS-DAIR**

**III. Performance Study**

**IV. OGSA-DAI WS-DAIR vs. AMGA WS-DAIR**

**V. Issues**

**VI. Future Work**

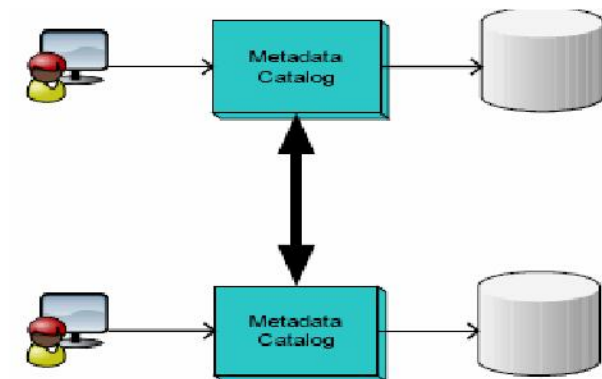
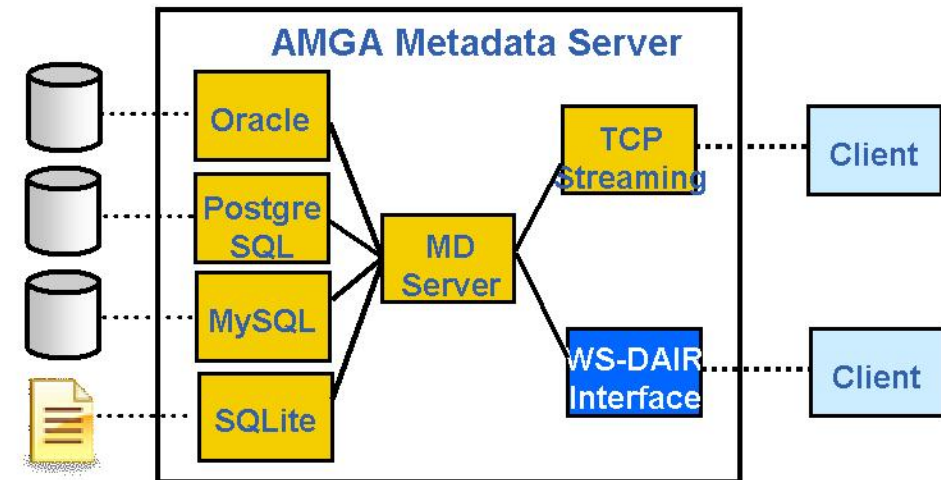
## What is AMGA ? (ARDA Metadata Grid Application)

- **Began as prototype to evaluate the Metadata Interface**
  - Evaluated by community since the beginning:
  - Matured quickly thanks to users feedback
- **Metadata Catalogue of EGEE's gLite 3.1 Middleware**
- **Requirements from HEP community**
  - Millions of files, 6000+ users, 200+ computing centres
  - Mainly (real-only) file metadata
  - **Main concerns : scalability, performance, fault-tolerance, Support for Hierarchical Collection**
- **Requirements from Biomed community**
  - Smaller scale than HEP
  - **Main concerns : Security**

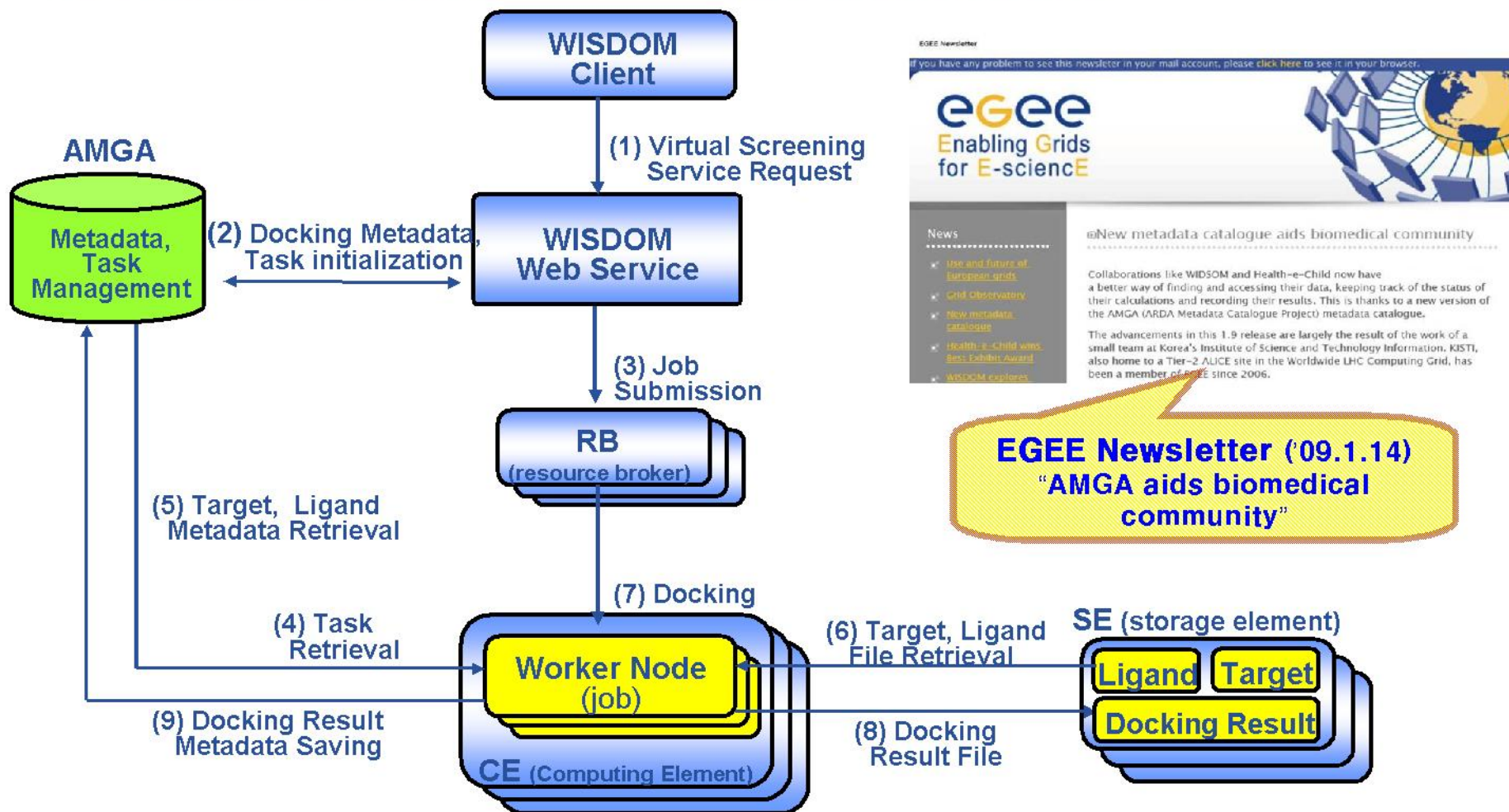


## Main Feature of AMGA

- **Modular back-end**
  - Oracle, PostgreSQL, MySQL, SQLite
- **Modular front-end**
  - TCP Streaming, **WS-DAIR**
- **Hierarchical Organisation**
  - Metadata organised in a tree-like structure
- **Security : SSL, GSI, VOMS, and ACL**
- **Replication : Master/Slave Model**
- **Import existing databases to AMGA**
- **Language**
  - AMGA Metadata Language, SQL-92 Direct Data Statement



## Working Scenario of WISDOM with AMGA

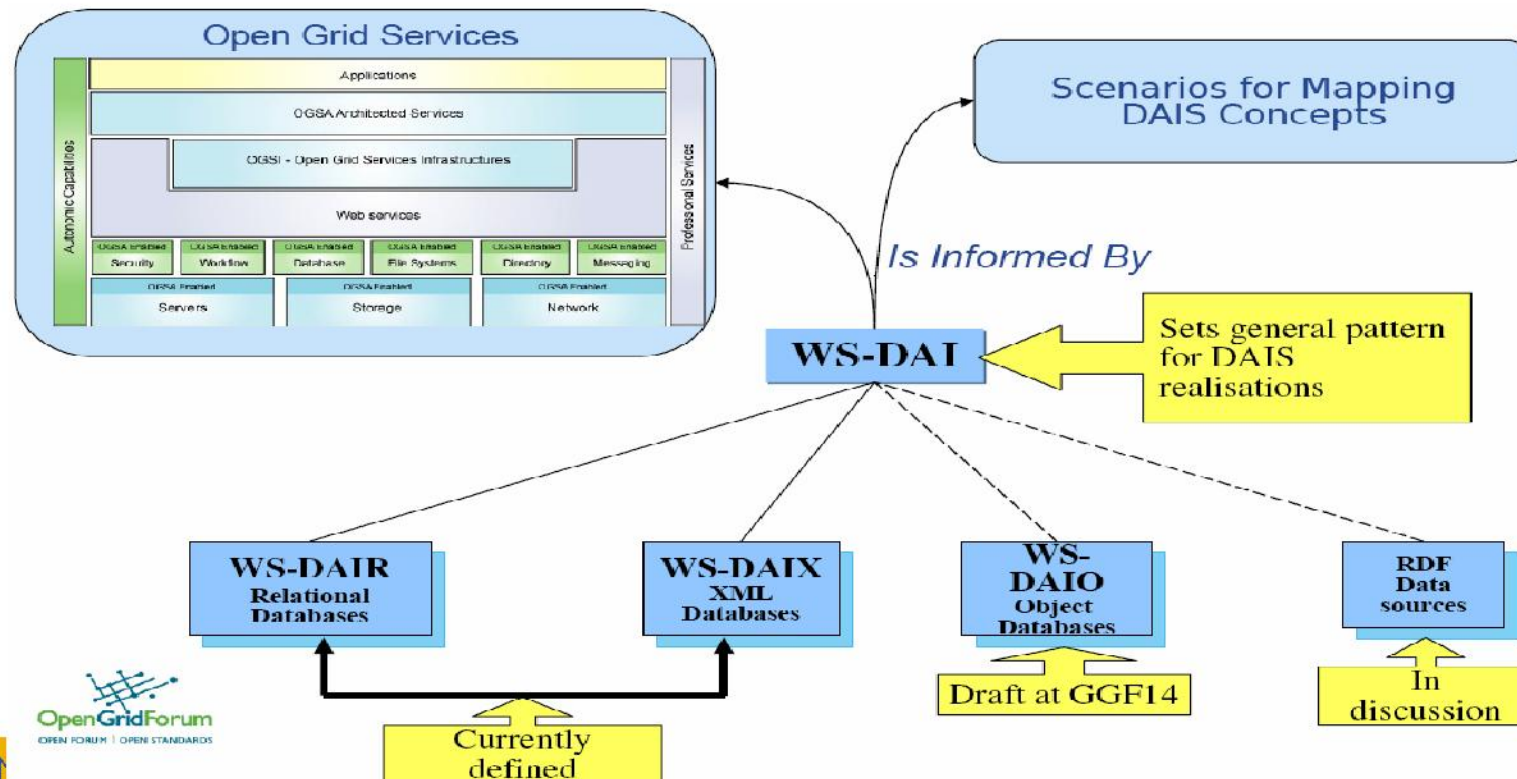


**EGEE Newsletter (09.1.14)**  
**"AMGA aids biomedical community"**

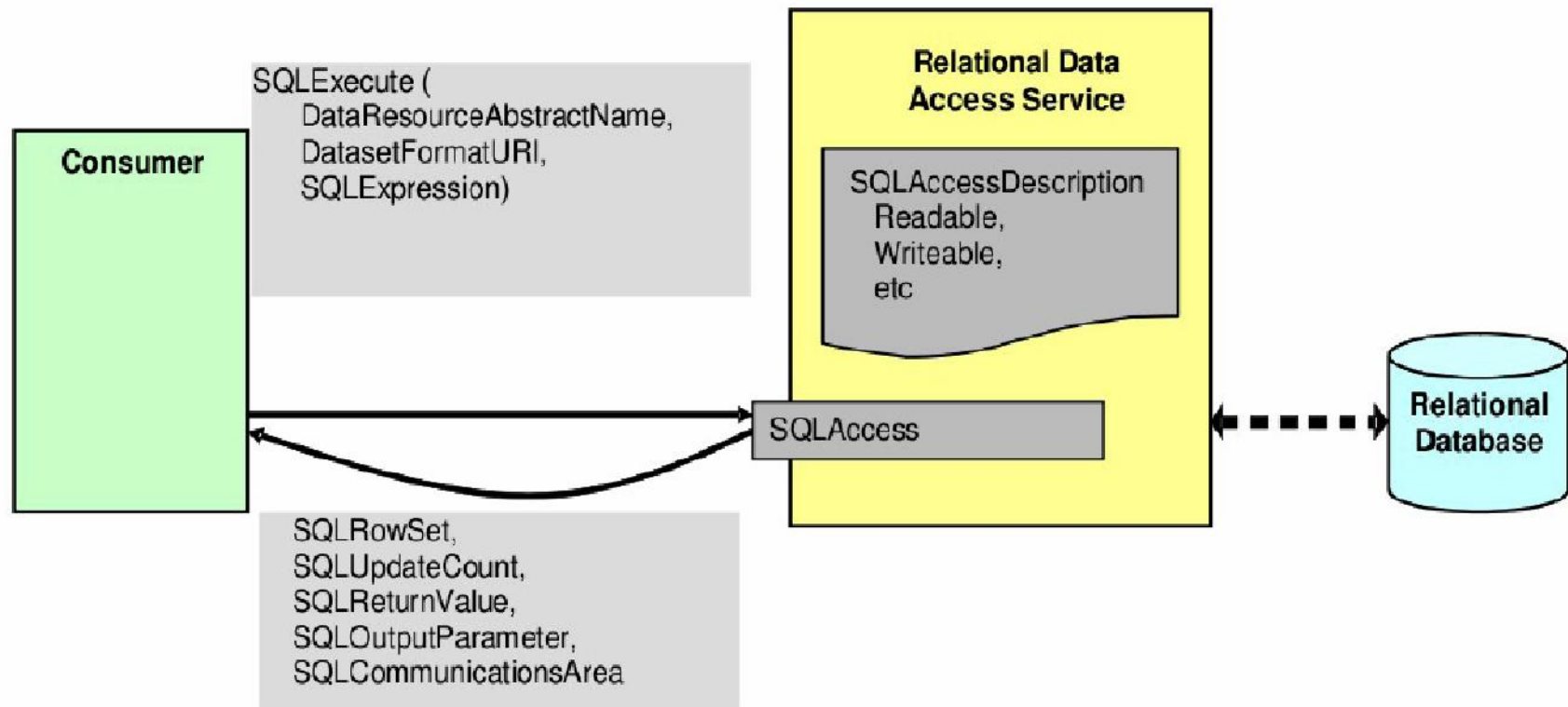
In Worker Nodes, step 4 ~ 9 are executed repeatedly until all docking tasks are finished

## What is WS-DAIR ?

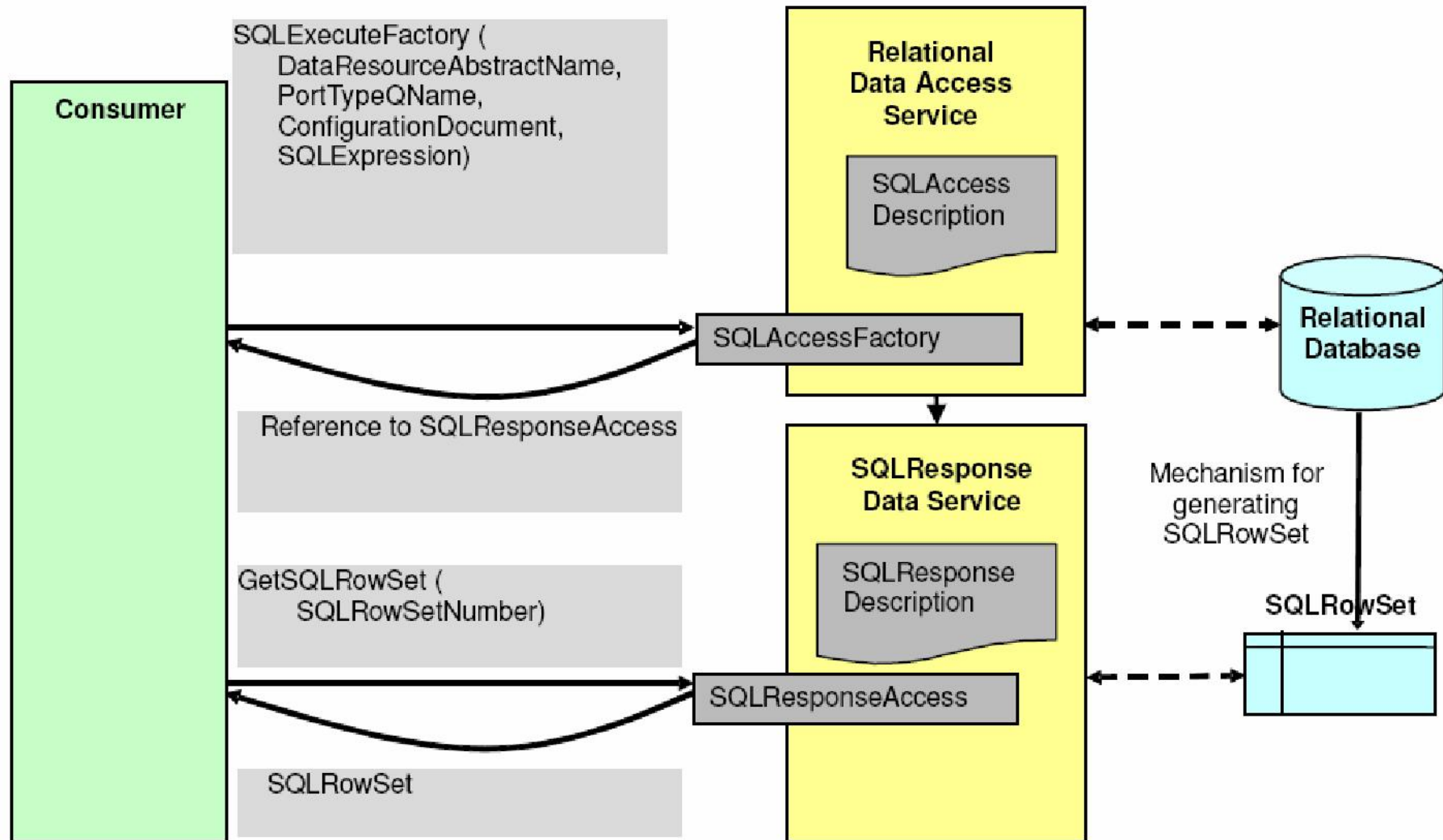
- Proposed OGF standards Recommendation for access to relational DB's on the Grid
  - Allows AMGA a seamless integration into the OGF standardized Grid Data Access Services



## Direct Data Access



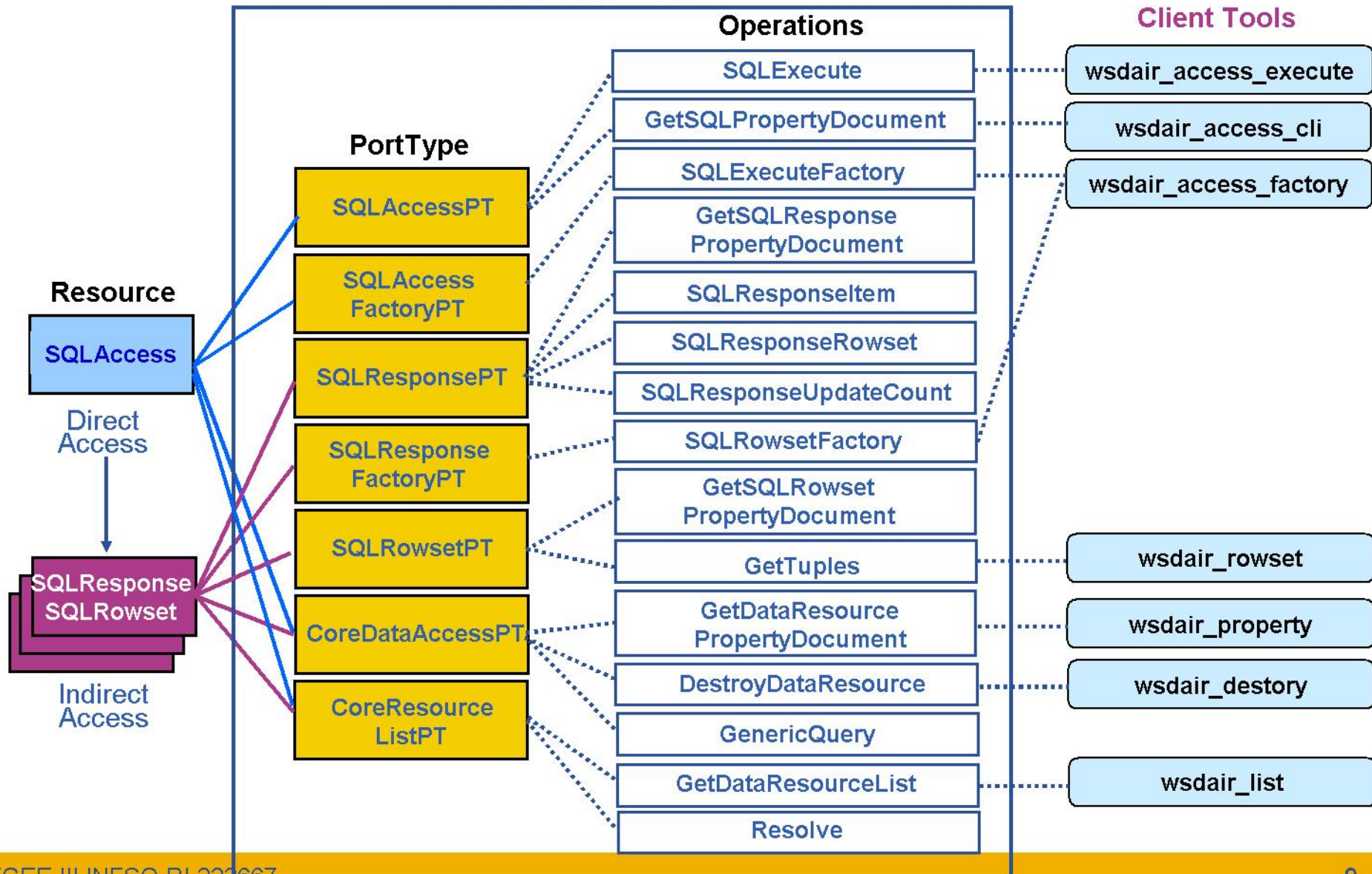
## Indirect Data Access





# II. Integration of WS-DAIR

## mdsoapsever



### AMGA WS-DAIR Interface Implementation

- **Written in C++ (gSOAP)**
  - SOAP Binding : document/literal
  - **Given WSDLs in WS-DAIR specification were used with few modification**
- **Features**
  - Supported Dataset Format : SUN JDBC WebRowSet (default)
  - Supported Language : SQL-92 Direct Data Statement, AMGA Metadata Language
- **Security : SSL, GSI, VOMS, and ACL**
- **Indirect Data Access Service**
  - **Data for a new indirect service is stored as a DB VIEW**

### Modified WSDLs in WS-DAIR specification

- **SQLExpression** in <WS-DAIR Appendix A.2 SQLAccess WSDL>

```

<!--xsd:element abstract="true" name="SQLExpression"
type="wsdair:SQLExpressionType"/ -->
<xsd:element abstract="false" name="SQLExpression"
type="wsdair:SQLExpressionType"/>

```

- **DatasetDataType** in <WS-DAI Appendix A.1 Core XML Schema>

```

<xsd:complexType name="DatasetDataType" mixed="true">
  <xsd:sequence>
    <!--xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
    <xsd:element ref="wrs:webRowSet" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

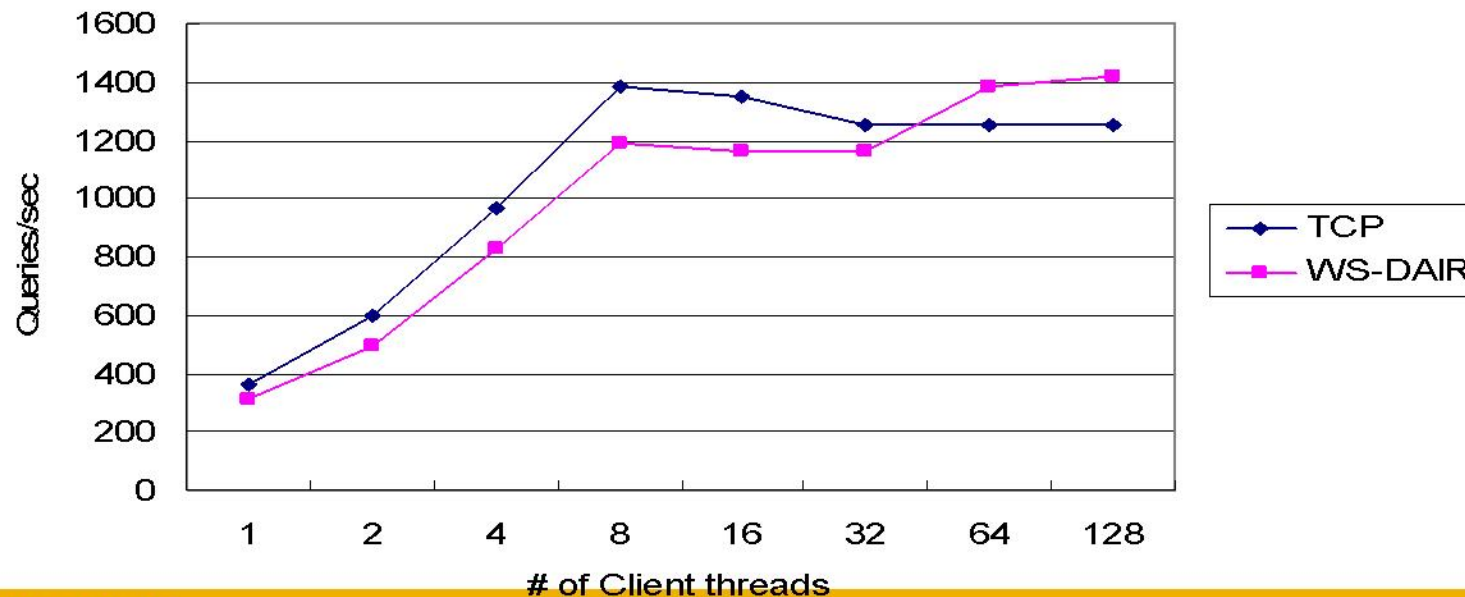
```

### Client Tools

- `$ wsdaire_access_cli "SELECT * FROM /simulation"`  
....
- `$ wsdaire_access_factory "SELECT * FROM /simulation"`  
[http://.....:8833/SQLResponse/responses487\\_19](http://.....:8833/SQLResponse/responses487_19)
- `$ wsdaire_rowset -P 0 -C 1 http://.....:8833/SQLResponse/responses487\_19`
- `$ wsdaire_list`  
[http://.....:8833/SQLResponse/responses487\\_19](http://.....:8833/SQLResponse/responses487_19)
- `$ wsdaire_property http://.....:8833/SQLResponse/responses487\_19`
- `$ wsdaire_destroy http://.....:8833/SQLResponse/responses487\_19`

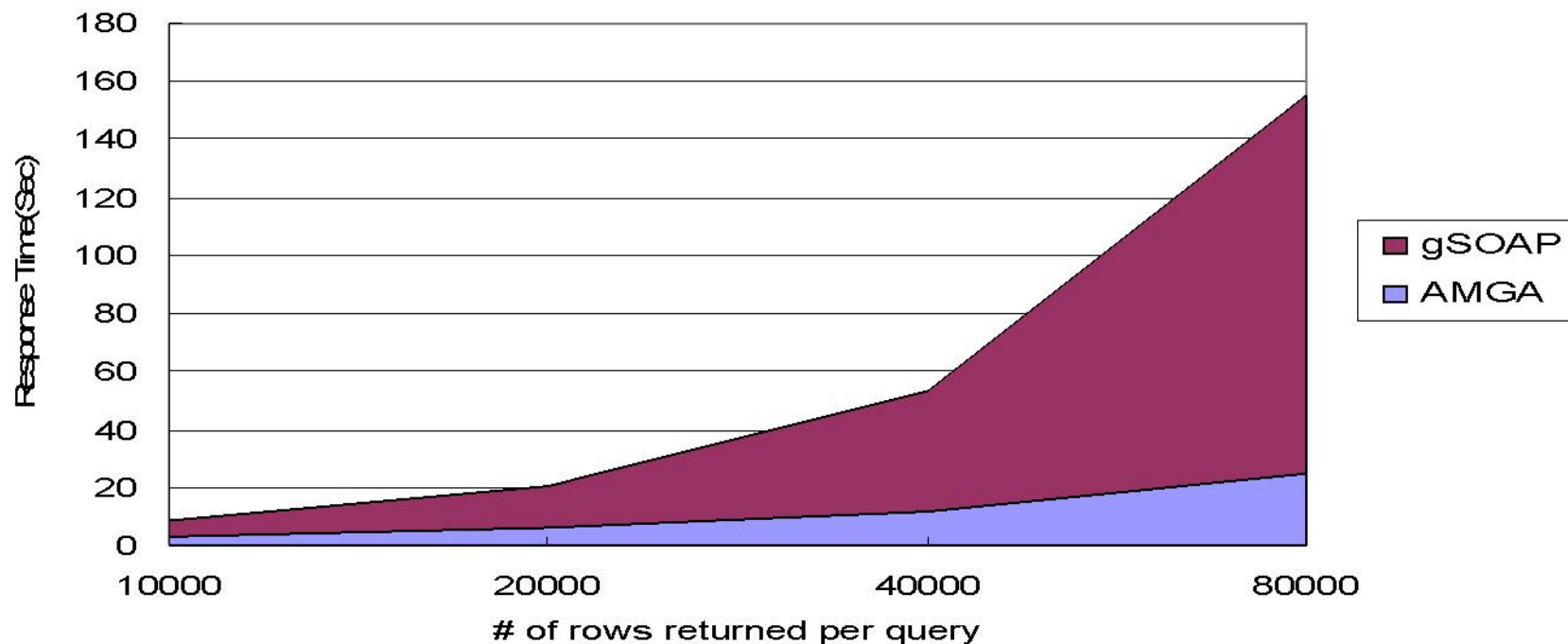
## Performance Study : Throughput

- **Testing Environment (Direct Data Access)**
  - DB : the simulation table used for the WISDOM project
  - One row with two attributes was retrieved by each query
- **Results**
  - Little difference between TCP/IP streaming and WS-DAIR SOAP
  - Maximum Throughput : about 1,400 queries / sec



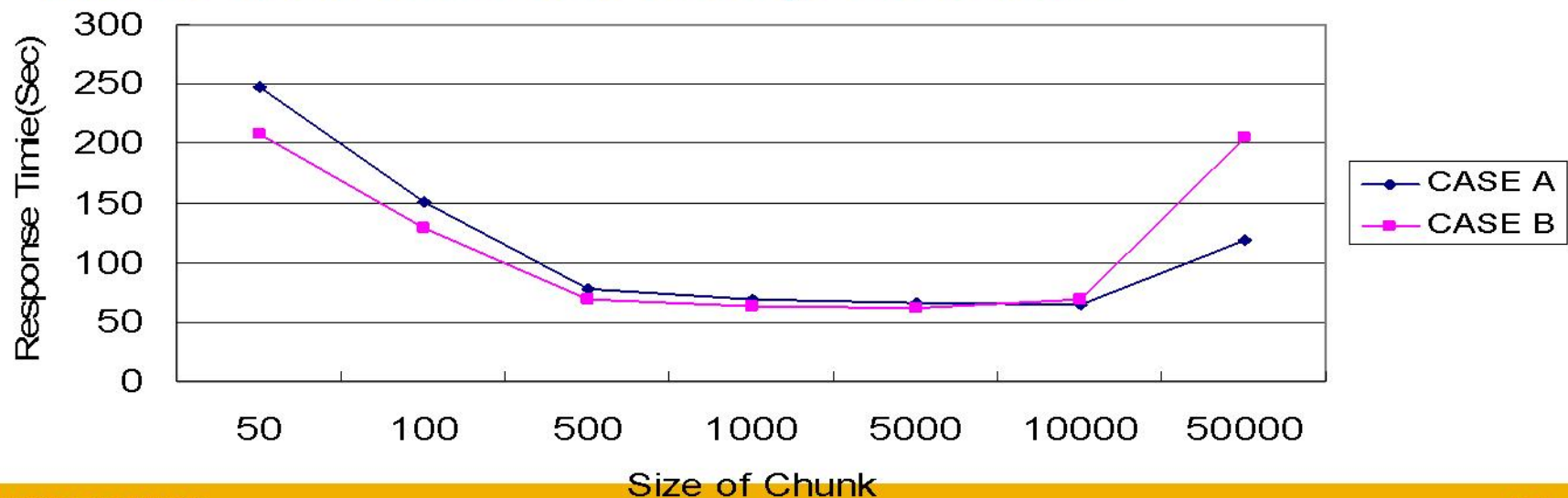
## Performance Study : Response Time

- Testing on response time taken to retrieve various # of rows per query
  - DB : average 4.5Kbyte per row, 19 columns (ligand table in WISDOM)
- Results
  - Response time increases exponentially when # of rows increases



## Performance Study : Response Time

- Testing on response time taken to retrieve 100,000 rows with various chunk size (Indirect Data Access)
  - Case A : 4.5Kbyte for each row, 19 columns (ligand table)
  - Case B : 250byte for each row, 32 columns (simulation table)
- Results
  - Minimum response time : in around a chunk of 1,000 rows
  - Response time depends on # of elements in a SOAP message rather than the size of data in case of a large size of chunk



	OGSA-DAI WS-DAIR	AMGA WS-DAIR
Dev. Tools	Axis, Java	gSoap, C++
<b>SOAP binding</b>	<b>rpc/literal</b>	<b>document/literal</b>
Supported DBMS	MySQL, (tested) Oracle, DB2, PostgreSQL ..	PostgreSQL, MySQL, SQLite Oracle (partially)
<b>Security Feature</b>	<b>NO</b>	<b>SSL, GSI, VOMS, and ACL</b>
# of databases supported in a service	Many	1 (AMGA metadata DB)
Supported Language	SQL-92	- SQL-92 Entry Level Direct Data Statement - AMGA Metadata Language
Supported Dataset Format	WRS, CSV	WRS
<b>Throughput</b>	<b>Tens of queries per sec</b>	<b>Hundreds of queries</b>



## Interoperability Issues : SOAP Binding

AMGA WS-DAIR

```

<wsdair:SQLExecuteRequest>
  <wsdai:DataResourceAbstractName>Metadata
</wsdai:DataResourceAbstractName>
  <wsdair:SQLExpression xsi:type="wsdair:SQLExpressionType">
    <Expression>SELECT ..... </Expression>
  </wsdair:SQLExpression>
</wsdair:SQLExecuteRequest>

```

OGSA-DAI WS-DAIR

```

<SQLExecute xmlns="">
  <ns1:SQLExecuteRequest xmlns:ns1="http://www.ggf.org/.. >
    <ns2:DataResourceAbstractName xmlns:ns2="http://....">
      wsdai:PGSQLDataResource </ns2:DataResourceAbstractName>
    <ns1:SQLExpression>
      <ns1:Expression>SELECT ..... </ns1:Expression>
    </ns1:SQLExpression>
  </ns1:SQLExecuteRequest>
</SQLExecute>

```

For testing purpose only,

- We have changed AMGA WS-DAIR to use rpc / literal binding
  - Modified the given WSDL in the spec a lot
- **OGSA-DAI WS-DAIR client to AMGA WS-DAIR server (Fail)**
  - Namespace Problem of SQLExecute
  - When `<SQLExecute xmlns="">` is changed to `<SQLExecute xmlns=http://www.ggf.org/namespaces/2005/12/WS-DAIR>`, it works fine

`<SQLExecute xmlns="">`

OGSA-DAI WS-DAIR

```

<ns1:SQLExecuteRequest xmlns:ns1="http://www.ggf.org/.. >
  <ns2:DataResourceAbstractName xmlns:ns2="http://....">
    wsdai:PGSQLDataResource </ns2:DataResourceAbstractName>
  <ns1:SQLExpression>
    <ns1:Expression>SELECT ..... </ns1:Expression>
  </ns1:SQLExpression>
</ns1:SQLExecuteRequest>
</SQLExecute>
  
```

- **AMGA WS-DAIR client to OGSA WS-DAIR server (Fail)**
  - **Namespace Problem of Expression**
  - **When <Expression> is changed to <wsdair:Expression>, It works fine**

<pre> &lt;wsdair:SQLExecute&gt;   &lt;wsdair:SQLExecuteRequest xsi:type="wsdair:SQLExecuteRequest" &gt;   &lt;wsdai:DataResourceAbstractName&gt;wsdair:PGSQLDataResource   &lt;/wsdai:DataResourceAbstractName&gt;   &lt;wsdair:SQLExpression xsi:type="wsdair:SQLExpressionType"&gt;     &lt;Expression&gt;SELECT ..... &lt;/Expression&gt;   &lt;/wsdair:SQLExpression&gt; &lt;/wsdair:SQLExecuteRequest&gt; &lt;/wsdair:SQLExecute&gt;                 </pre>	Modified AMGA WS-DAIR
--	-----------------------

## 1. Handling Large Size of data

- Requesting very large size of data may cause the service crash.
- Possible Solutions
  - returns error message on Requesting too large size of data
  - Streaming : not defined in SOAP 1.1 & WSDL 1.1
    - MTOM (SOAP 1.2)
    - DIME (extension of WSDL 1.1)

## 2. Response time increases exponentially when # of elements in a SOAP message increases

- Other very simple Dataset Formats other than WebRowSet will be examined (e.g. csv)

## Future Work

- Handling very large size data
- Interoperability Testing against OGSA-DAI WS-DAIR

## Release

- 1.9 : Native SQL Support, Multi-Thread Multi-Process Model, DB Connection Sharing (Nov. 2008)
- 2.0 : WS-DAIR support (Spring 2009)