National e-Science Centre

OGSA-DAI TUTORIAL
Grid Computing
2005/2006

Femi Ajayi

Computing Science Department

University of Glasgow

National e-Science Centre

Overview

- The Need for Data
  - Challenges
  - Benefits
- Existing data technologies
  - Distributed RDBMS
  - Data Integration
  - OGSA-DAI
- Introduction to OGSA-DAI
  - Overview
  - Features
  - Uses
- Introduction to VOTES and its associated challenges
- Current data issues on the Grid

National e-Science Centre

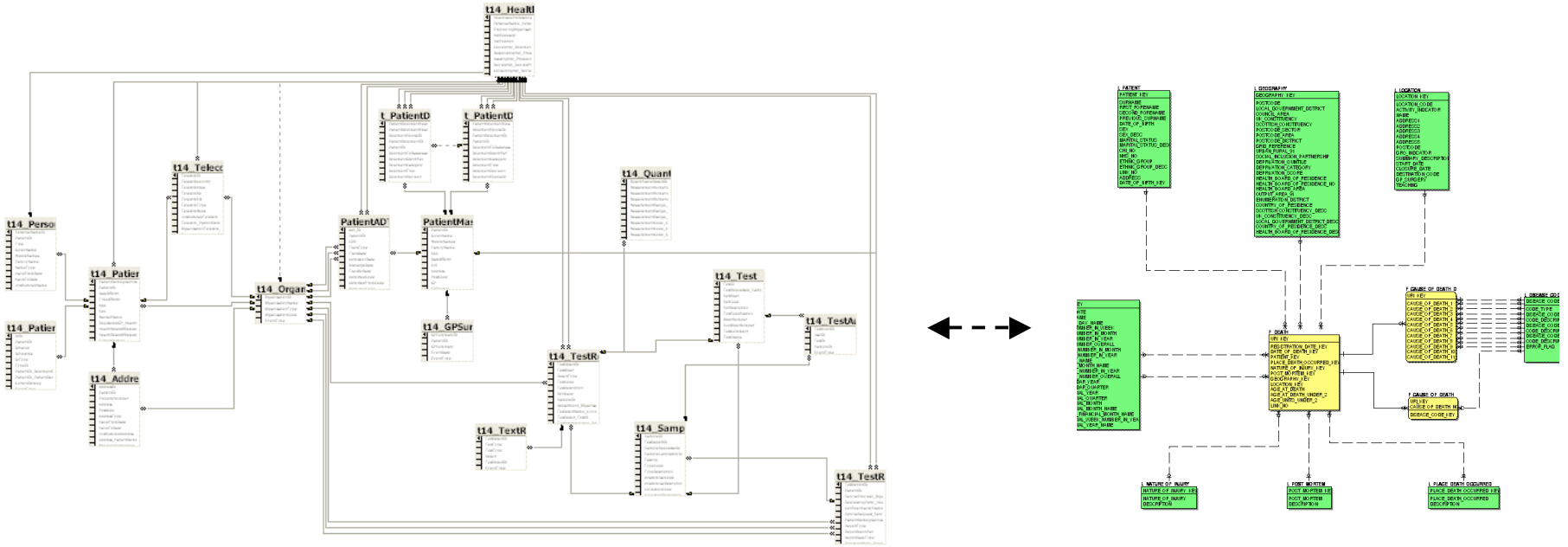
Challenges

- Data Discovery, Filtering and Translation
- Understanding metadata data and relationships
- Access to data, locally or geographically distributed
- Data growing in size, complexity and time
- Online analytical and transactional processing
- Examples
  - ATLAS: Particle physics experiment to investigate the origin of mass. 15PB of data a year, 2000 scientists, 150 universities, 34 countries.
  - BRIDGES: Integration of six geographically distributed research repositories to support cardiovascular functional genomics research.

National e-Science Centre

Data from Bio-domain

- Clinical Data: Clinical records, laboratory results, morbidity records, epidemiological studies, patient demographics, SCI, SMR, etc.
- Bioinformatics: Genomics data (microarray data, proteomic gel data, RNA, etc.)



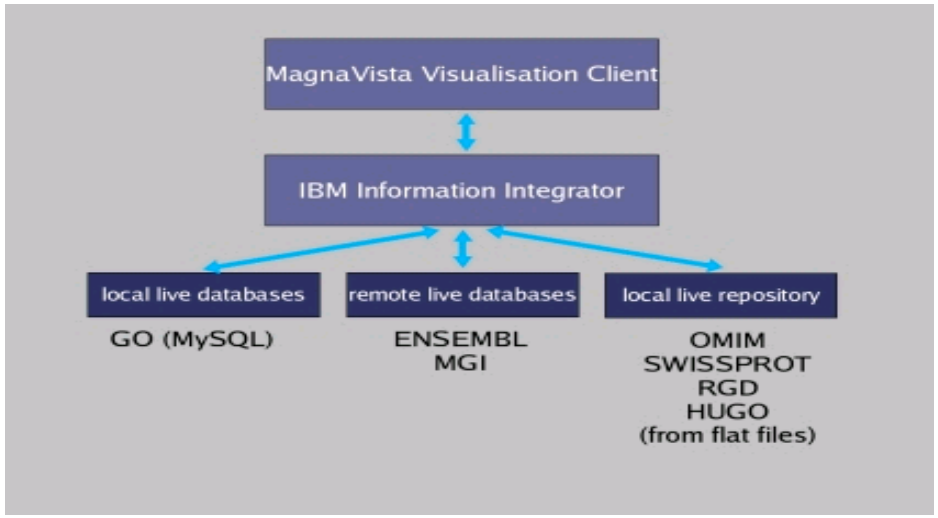
Integrating SCI and SMR data subsets

National e-Science Centre

Some Data Technologies

- Relational Database Management Systems
  - Oracle
  - MySQL
  - SQL Server
  - PostgreSQL
  - DB2
- Data Integrator
  - IBM Information Integrator
  - ACTUATE Enterprise Information Integrator (EII)
- Data Services
  - OGSA-DAI

BRIDGES Data Integration



National e-Science Centre

OGSA-DAI

- Goals
  - To provide service-based access to data resources based on OGSA architecture
  - To specify a selection of common interfaces for different data access including relational and xml data.
- Tutorial
  - Interacting with Data Service Resources
  - Client Toolkit API

National e-Science Centre

Why OGSA-DAI?

- Unified service interface to different data resources
- Common API to all data resources
- Data transformation, workflows and delivery
- WSRF and WS-I compliance
  - Fits into the Grid vision
- Extensible model

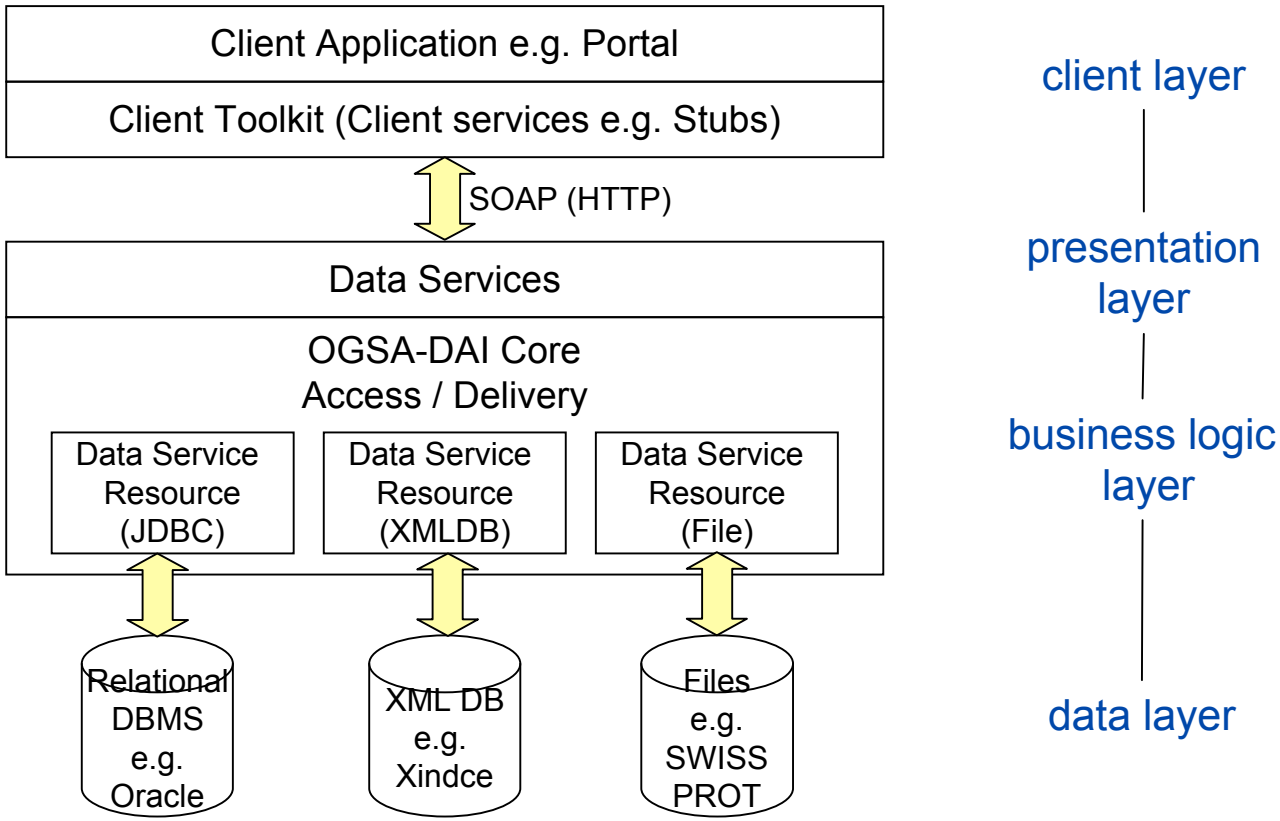
National e-Science Centre

Required for tutorial

- Understanding RDBMS
- Ability to use SQL
- XML basics
- Web Services basics (optional: WS-I and WSRF)
- Java Data API

National e-Science Centre

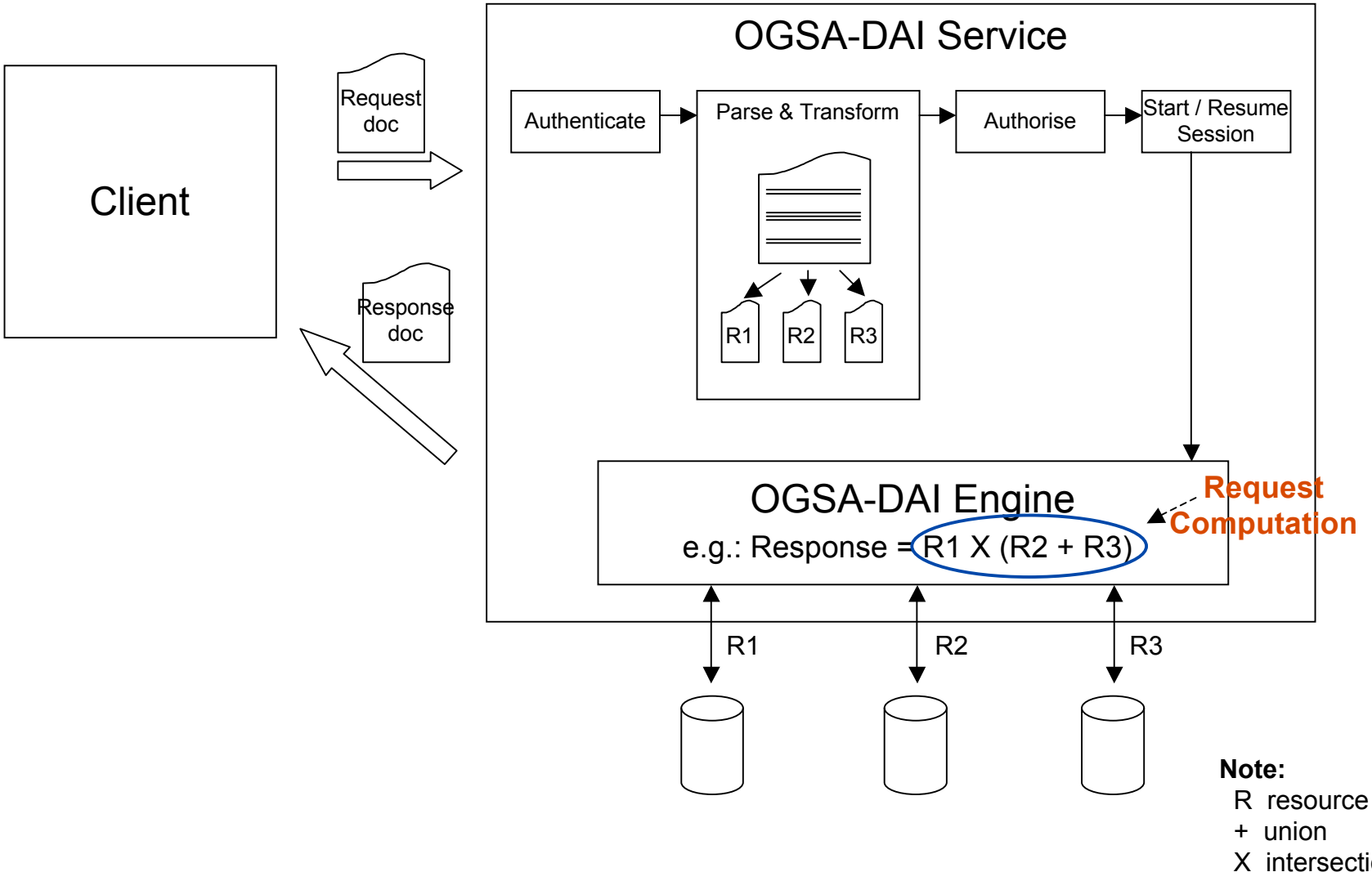
OGSA-DAI Architecture



- Client layer – interacts with presentation layer using WSRF/WSI standards or using OGSA-DAI client toolkit API
- Presentation layer – expose data services as a web service complaint with WSRF or WSI standards
- Business logic layer – provides core OGSA-DAI functionalities, perform documents, response documents, data resource access, data transport, session management and property management
- Data layer – currently supports RDBMS (Oracle, SQL Server, MySQL, DB2, PostgreSQL), Xindce, and Files formats (OMIM, SWISSPROT, EMBL)

National e-Science Centre

OGSA-DAI Computational flow

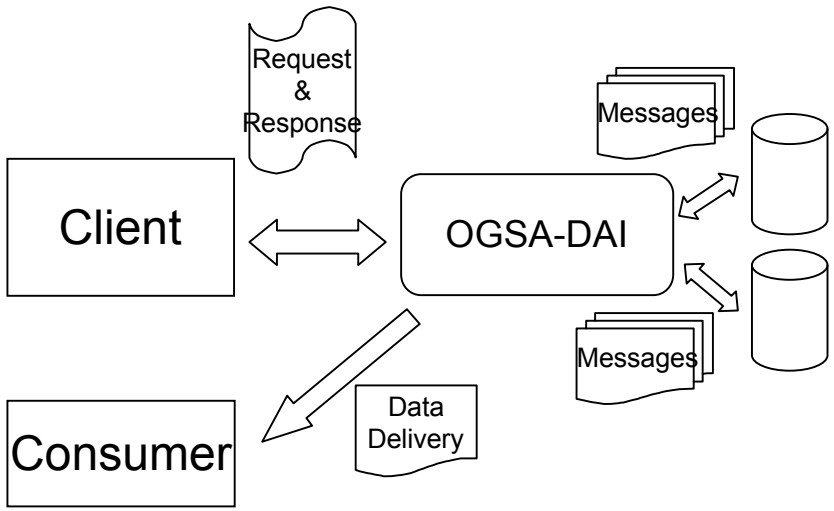


Note:
R resource request
\+ union
X intersection

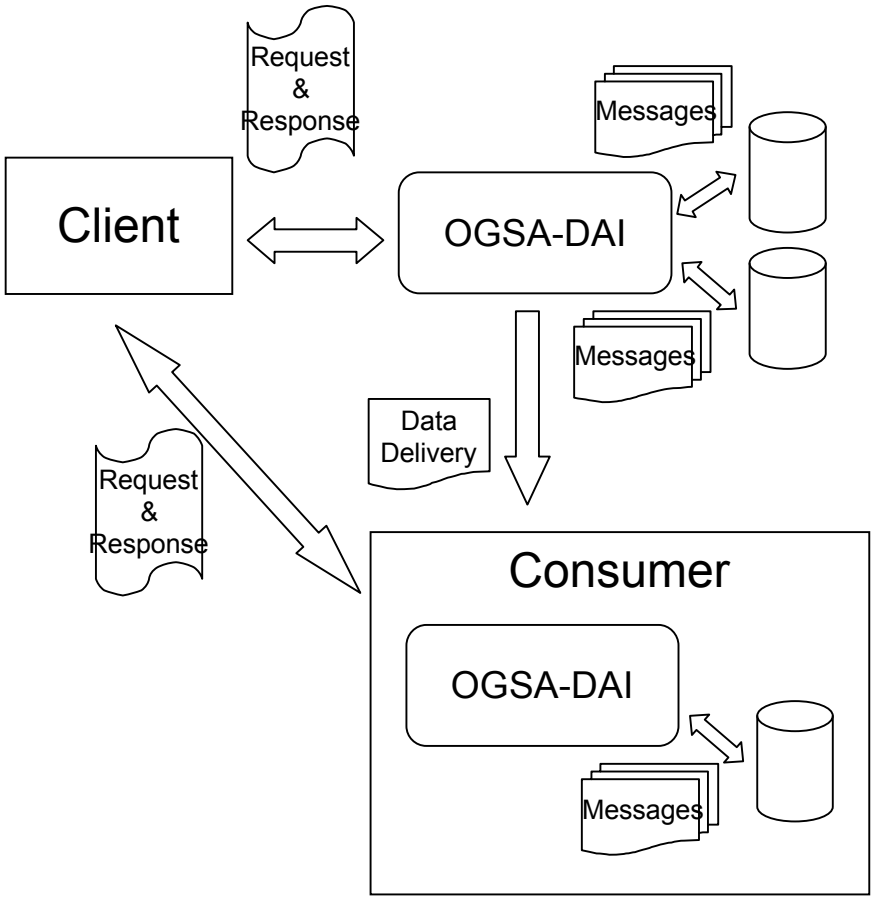
OGSA-DAI Data Service Scenarios

National e-Science Centre

Scenario 1: Single data service

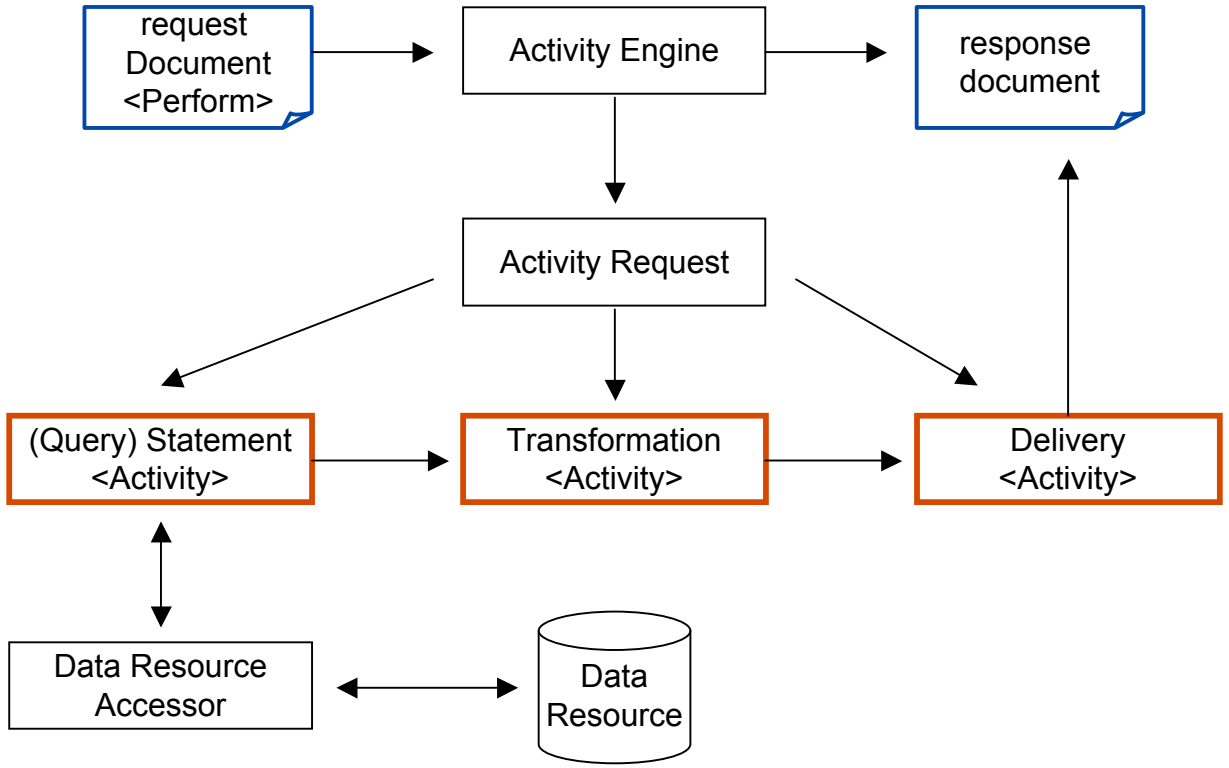


Scenario 2: Multiple data services



National e-Science Centre

Data Service Interactions



Interacting with Data Service Resources

National e-Science Centre

- Activities: actions to be performed
  - Data resource statement e.g. sqlQueryStatement, sqlUpdateStatement, sqlStoredProcedure, sqlResultsToXML
  - Data transformation e.g. xslTransform, gzipCompression
  - Data delivery to/from a consumer e.g. deliverFromURL, deliverToURL, deliverFromGFTP, deliverToGFTP
- Perform documents
  - Used by clients to define instruction sets for data service resources. Activities form instruction sets.
  - Data can flow from activity to activity
  - Better to be generated and processed by client toolkit.
- Response documents

Activities

- Workflow

Statement activity

```
<sqlQueryStatement name="listPatient">
    <!-- value of first parameter -->
    <sqlParameter position="1" from="minValue"/>
    <!-- value of second parameter -->
    <sqlParameter position="2">69</sqlParameter>
    <expression>
                select name, postcode from patientmaster
where age <= ? and id >= ?
    </expression>
    <resultStream name="queryResult"/>
</sqlQueryStatement>
```

Stream

Statement activity

```
<sqlResultsToXML name="results">
    <resultSet from="queryResult" />
    <webRowSet name="webrowset" />
</sqlResultsToXML>
```

Transform

Transform activity

```
<xslTransform name="listTransform">
  <inputXSLT>
    <xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
      <xsl:output method="text" indent="yes" />
      <xsl:template match="/">
        <xsl:value-of select="RowSet/data/row/col[1]"/>
      </xsl:template>
    </xsl:stylesheet>
  </inputXSLT>
  <inputXML from="queryResult"/>
  <output name="transformedResult"/>
</xslTransform>
```

Delivery

Delivery activity

```
<deliverToURL name="ftpDelivery">
  <fromLocal from="transformedResult"/>
  <toURL>ftp://test.nesc.gla.ac.uk/file/file.ext</toURL>
</deliverToURL>
```

Perform Document

National e-Science Centre

- Workflow control
  - Sequence: Executes activities sequentially. Used when dependences exist between activities.
  - Flow: Executes activities concurrently (default). Used when no dependences exist between activities.
- Client toolkit APIs could be used to build clients that can generate perform documents and interpret response documents.

```
<?xml version="1.0" encoding="UTF-8"?>
<perform xmlns="http://ogsadai.org.uk/namespaces/2005/10/types">
  <documentation>
    Perform a simple...
  </documentation>
  <sequence name="trialSequence">
     <sqlUpdateStatement name="updateCT">
       <expression>
         update clinicaltrial set minage='41' where id=1005
       </expression>
       <resultStream name="noOutput"/>
     </sqlUpdateStatement>
     <flow>
                  <sqlQueryStatement name="trialphase2"/>
                    <expression>select minage from clinicaltrial where trialid=1005 and
phase=2</expression>
                    <resultStream name="minValue"/>
                  </sqlQueryStatement>
                  <sqlQueryStatement name=listPatient">
                      <!-- value of first parameter -->
                      <sqlParameter position="1" from=minValue"/>
                      <!-- value of second parameter -->
                      <sqlParameter position="2">69</sqlParameter>
```

Begin sequence

Statement Activity (A)

Begin flow

Statement Activity (B1)

Statement Activity (B2)

Perform Document cont.

```
                    <expression>
                        select name, postcode from patientmaster where age <= ? and id >= ?
                    </expression>
                    <resultStream name=queryResult"/>
                </sqlQueryStatement>
                <sqlResultsToXML name="results">
                    <resultSet from=queryResult" />
                    <webRowSet name="webrowset" />
                </sqlResultsToXML>
    </flow>
    <xslTransform name=listTransform">
      <inputXSLT>
        <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
          <xsl:output method="text" indent="yes" />
          <xsl:template match="/">
            <xsl:value-of select="RowSet/data/row/col[1]"/>
          </xsl:template>
        </xsl:stylesheet>
      </inputXSLT>
      <inputXML from=queryResult"/>
      <output name="transformedResult"/>
    </xslTransform>

    <deliverToURL name=ftpDelivery">
      <fromLocal from=transformedResult "/>
      <toURL>ftp://test.nesc.gla.ac.uk/file/file.ext</toURL>
    </deliverToURL >
  </sequence>
</perform>
```

Statement Activity (C)

Transformation Activity (D)

Delivery Activity (E)

Question: Show the workflow diagram for this perform document using labelled activities?

Response Document

National e-Science Centre

- A simple response document:
  - Returning query result: SELECT FamilyName, PostCode FROM PatientMaster WHERE Patientid LIKE '87%'

```
<?xml version="1.0" encoding="UTF-8"?>
<webRowSet xmlns="http://java.sun.com/xml/ns/jdbc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/jdbc
http://java.sun.com/xml/ns/jdbc/webrowset.xsd">
 <properties>
  <command><null/></command>
  <concurrency>1007</concurrency>
  <datasource><null/></datasource>
  <escape-processing>true</escape-processing>
  <fetch-direction>1002</fetch-direction>
  <fetch-size>0</fetch-size>
  <isolation-level>0</isolation-level>
  <key-columns>
  </key-columns>
  <map></map>
  <max-field-size>0</max-field-size>
  <max-rows>0</max-rows>
  <query-timeout>0</query-timeout>
  <read-only>true</read-only>
  <rowset-
type>ResultSet.TYPE_FORWARD_ONLY</rowset-
type>
  <show-deleted>false</show-deleted>
  <table-name><null/></table-name>
  <url><null/></url>
 </properties>
```

Cursor type

```
<metadata>
  <column-count>2</column-count>
  <column-definition>
   <column-index>1</column-index>
   <auto-increment>false</auto-increment>
   <case-sensitive>false</case-sensitive>
   <currency>false</currency>
   <nullable>1</nullable>
   <signed>false</signed>
   <searchable>true</searchable>
   <column-display-size>35</column-display-size>
   <column-label>FamilyName</column-label>
   <column-name>FamilyName</column-name>
   <schema-name></schema-name>
   <column-precision>35</column-precision>
   <column-scale>0</column-scale>
   <table-name></table-name>
   <catalog-name></catalog-name>
   <column-type>12</column-type>
   <column-type-name>varchar</column-type-name>
  </column-definition>
  <column-definition>
   <column-index>2</column-index>
   <auto-increment>false</auto-increment>
   <case-sensitive>false</case-sensitive>
   <currency>false</currency>
   <nullable>1</nullable>
   <signed>false</signed>
```

Metadata definition

Response Document cont.

National e-Science Centre

```
<searchable>true</searchable>
  <column-display-size>12</column-display-size>
  <column-label>PostCode</column-label>
  <column-name>PostCode</column-name>
  <schema-name></schema-name>
  <column-precision>12</column-precision>
  <column-scale>0</column-scale>
  <table-name></table-name>
  <catalog-name></catalog-name>
  <column-type>12</column-type>
  <column-type-name>varchar</column-type-name>
 </column-definition>
</metadata>
<data>
 <currentRow>
  ....
 <currentRow>
  <columnValue>BAKER</columnValue>
  <columnValue>G 133YR</columnValue>
 </currentRow>
 <currentRow>
  <columnValue>SILSTROM</columnValue>
  <columnValue>G 133YR</columnValue>
 </currentRow>
  ....
</data>
</webRowSet>
```

Requested Data

National e-Science Centre

Client Toolkit API

- Data services interaction
  - One API for both WSRF & WS-I services
  - Activities & Request (Queries, transformation & delivery)
  - Flow/sequence control
- Hidden service platform
- Data integration
- Extensibility

A Simple Client

National e-Science Centre

```
import java.sql.ResultSet;
import uk.org.ogsadai.client.toolkit.Response;
import uk.org.ogsadai.client.toolkit.activity.ActivityRequest;
import uk.org.ogsadai.client.toolkit.activity.ActivityOutput;
import uk.org.ogsadai.client.toolkit.activity.sql.SQLQuery;
import uk.org.ogsadai.client.toolkit.wsrf.WSRFServiceFetcher;
import uk.org.ogsadai.client.toolkit.service.DataService;

/**
* This example shows how to process the results of an SQL query.
*/
public class TutorialSampleQuery {

  public static void main(String[] args) throws Exception {

    // Locate a Grid Data Service Factory
    String handle =
      "http://localhost:8080/wsrf/services/ogsadai/DataService";
    // Factory handle may be provided as an argument
    if (args.length == 1) {
       handle = args[0];
    }

    String resourceID = "SQLServerResource";
    DataService service =
      WSRFServiceFetcher.getInstance().getDataService(
          handle, resourceID);
    System.out.println("Connected to data service at " + handle);

    // Perform a simple SQLQuery
    String sql = "select FamilyName, PostCode from PatientMaster
      where Patientid like'87%'";
    System.out.println("\nPerforming SQL query: " + sql);
    SQLQuery query = new SQLQuery(sql);

    ActivityRequest request = new ActivityRequest();
    request.add(query);
    Response response = service.perform(request);

    // Retrieve the output data as a string
    ActivityOutput output = query.getOutput();
    String data = query.getOutput().getData();
    System.out.println("Results:\n" + data);   //prints the
      webRowSet

    // Using JDBC as a resource accessor
    ResultSet result = query.getResultSet();
    while (result.next()) {
       String firstname = result.getString(2);
       String middlename = result.getString(3);
       String lastname = result.getString(4);
       String dob = result.getString(6);
       System.out.println(firstname);
       System.out.println(middlename);
       System.out.println(lastname);
       System.out.println("dob: " + dob);
       System.out.println();
      }
}
```

Virtual Organisation for Trials and Epidemiological Studies (VOTES)

Objective:

- To develop Grid infrastructure framework that would support both clinical trials and epidemiological studies.
  - Patient recruitment
  - Data collection
  - Study management

  - ➢ Not about submitting or running jobs
  - ➢ It's about accessing & integrating data
  - ➢ About distributed queries – accessing federated (or distributed) resources

National e-Science Centre

Clinical Trials

- Reliable assessment of moderate effects of treatment of important diseases (e.g. cancer) requires studies that guarantee
  - strict control of bias (requiring randomization and analysis)
  - strict control of random error
- Up to date information on patterns of disease/frequency of clinical procedures can help inform the design of a clinical trial
  - enable appropriate strategies for rapid and efficient recruitment to be devised
- Generating this evidence typically requires the collection of data on many thousands of people over a period of several years.
  - essential to avoid misleading results caused by the play of chance
- Clinical trials usually require collaborative effort
  - data are collected from individuals and from existing health records
  - from multiple investigative sites
- Study progress, data quality, analysis of results managed by one or more coordinating centres

National e-Science Centre

Clinical Trials - Example

- Drug: Metformin
  - Current use: Diabetes treatment
  - Possible use: Could double the chances of success for women undergoing IVF
- Trial
  - Recruit women with Polycystic Ovarian Syndrome (PCOS) seeking IVF
    - Ages between 25 -40
    - Some overweight women
    - Some pregnant women
  - Questions
    - Any fertility improvements
    - Any ongoing pregnancy improvement beyond 12 weeks gestation
    - Any associated risks by the use of this drug

Clinical Trials stages

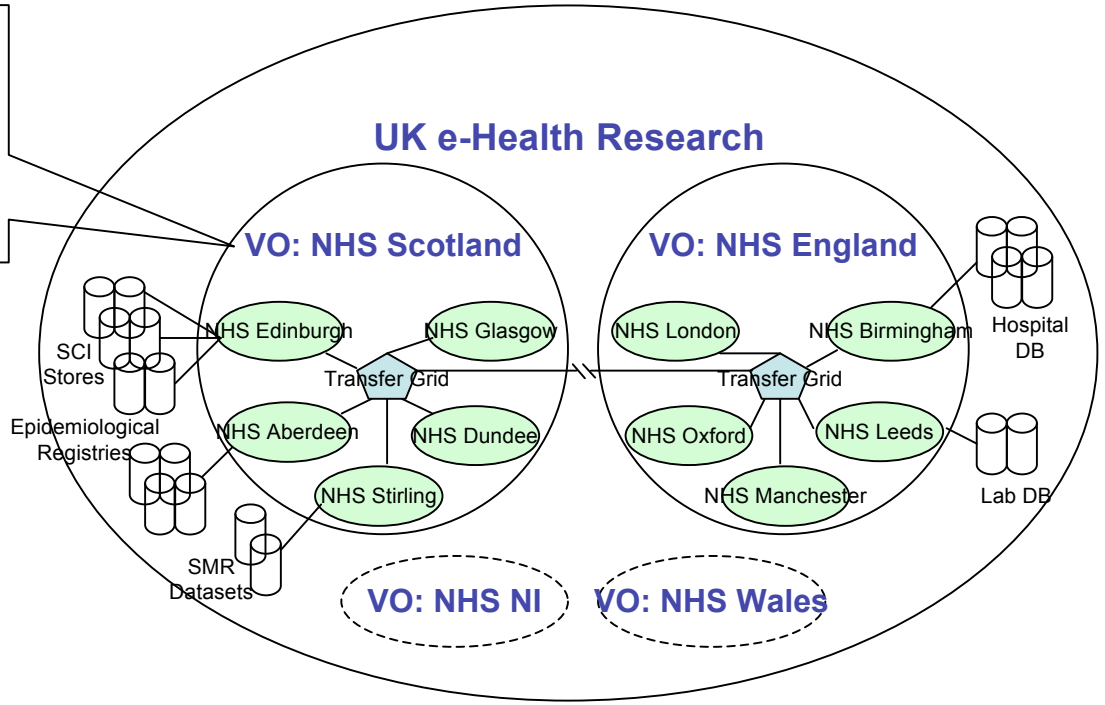
	Phase	Phase II	Phase III	Phase IV
Question	Is it a safe treatment?	Does the treatment work?	What are the long-time results in lots of people?	What is the long-term safety information?
Volunteers	Usually healthy volunteers / without comparison groups	Volunteers with the disease or condition	Both healthy & non-healthy volunteers	
Goal	Determine best doses	Get safety information	More information on safety and effectiveness	Get information about very rare side effects
Risk	High risk	Moderate risk	Low risk	Very low risk
Length	Few weeks/months	About a year	Up to three years	May last for years
Size	Few participants	About a hundred participants	Some hundreds participants	Some thousands participants
Time	Before drug approval			After drug has been approved

National e-Science Centre

Virtual Organisations

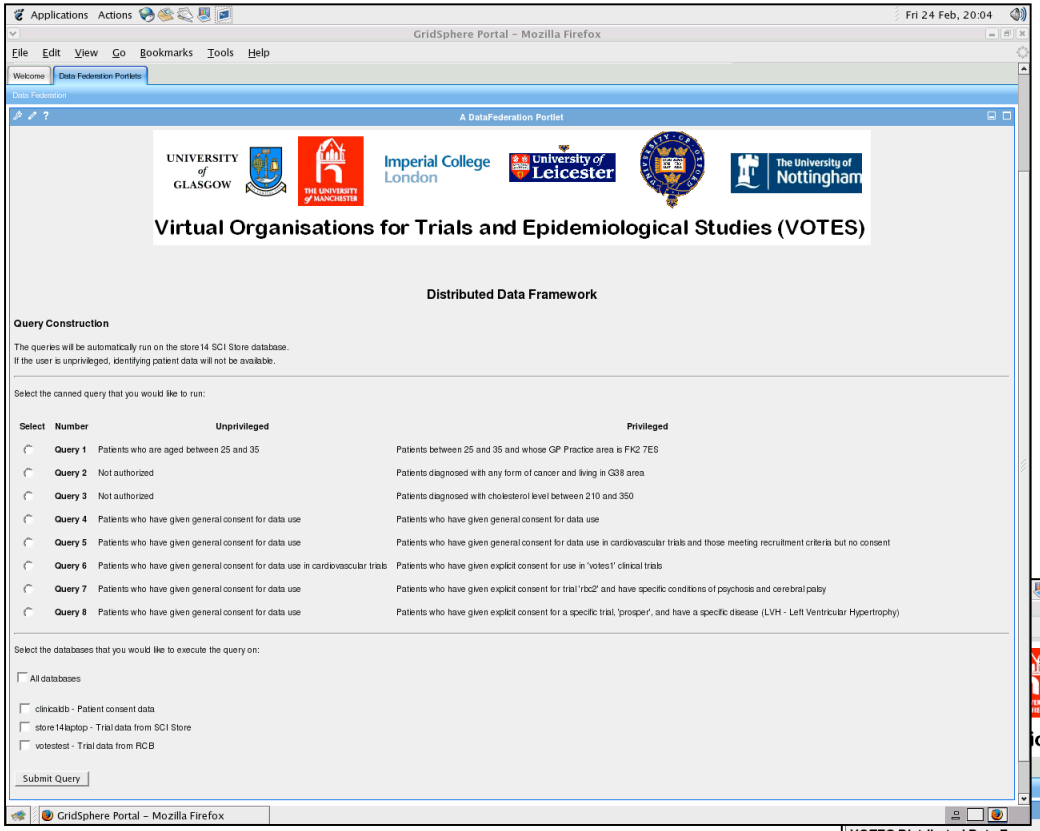
- A coordinated use and sharing of dynamic and distributed resources by a set of individuals or institutions or domains.
- Allows members to collaborate to achieve shared goals.
- No centralised point of control; Diverse administrative domains
- Single sign-on; Delegation; Various local security integration

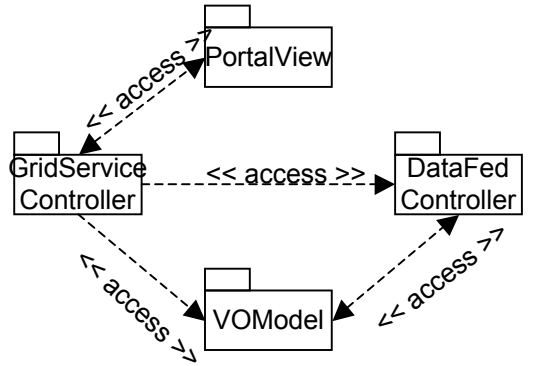
- 19 SCI-Stores in Scotland
  - each store is a repository of about 12 different sources
- SMR datasets
- GPASS datasets

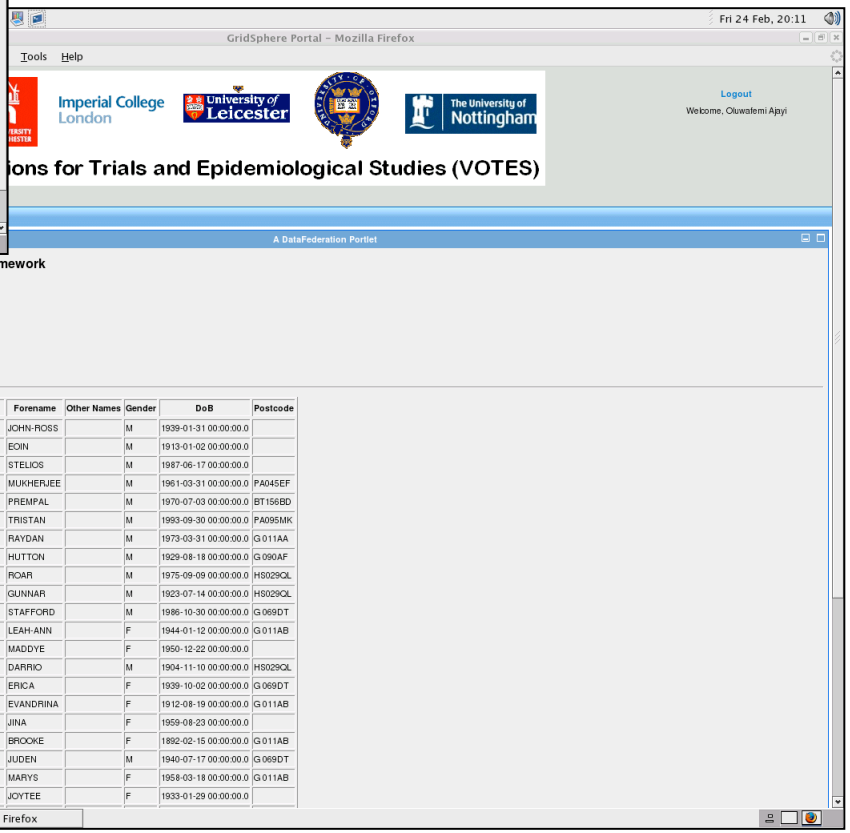


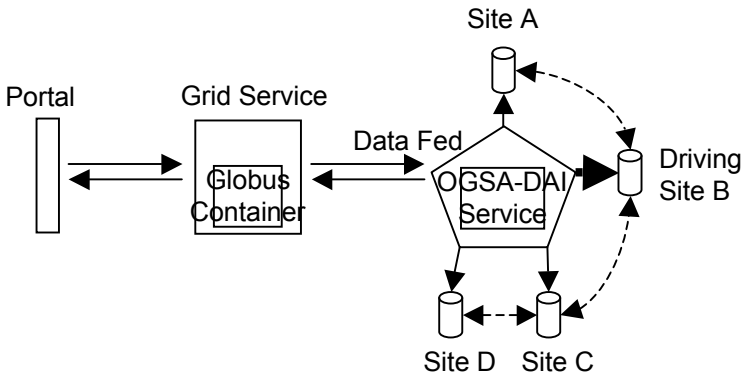
VOTES Data Federation Portal v1

National e-Science Centre









VOTES Challenges with OGSA-DAI

- Data federation
  - Many autonomous and geographically distributed data resources
  - Changing data models
  - Different data models – no common model
  - Distributed queries – query composition and decomposition
  - Data aggregation
  - Replication
- Data access and security
  - Access control (inc. federation), Confidentiality, Privacy, Aggregation & Inference
- Provenance & Persistence
- Changing requirements
  - Use cases
  - Static vs dynamic queries
- Dynamic discovery

National e-Science Centre

Current Issues/Challenges

- Data visualisation
- Transactions implementation
- Data aggregation from disparate models
- Conversion between metadata
- Changing schemas
- Semantic Web, Ontologies
- Auto discovery of data service
  - Queries are directed to specific targets
- Fine grain security model
- Performance bottlenecks
- Provenance information
- Data access != Data integration

National e-Science Centre

Appendix

- Installation: To better understand OGSA-DAI components
- Exposing a Data Resource
  - Deploy an OGSA-DAI data service
  - Deploy a data service resource
  - Add the data service resource to the data service

National e-Science Centre

Installing OGSA-DAI

- Requirements
  - WSRF: GT4.x or Tomcat
    - » OR
  - WS-I: Apache Axis on Tomcat or OMII
  - Java 1.4.x, ANT 1.5/1.6
  - Prerequisite JARs: jakarta-oro-2.0.8.jar, xmldb.jar, lucene-1.4.3.jar
  - DBMS jars e.g. pg73jdbc3.jar, mysql-connector-java-3.1.8-bin.jar
- Registration of OGSA-DAI from www.ogsadai.org.uk/downloads
- Download OGSA-DAI WSRF (our focus is on WSRF on GT4)
- Building the binary distribution from source distribution
  - Change directory to the downloaded source distribution directory
  - Copy all JARs to the lib directory
  - # export GLOBUS_LOCATION=/usr/local/globus4.0.1 //change to exact Globus path
  - # ant createBinaryDistribution //this will create a binary directory
  - # cd binary

National e-Science Centre

Installing OGSA-DAI II

- Deploying OGSA-DAI WSRF onto GT4
  - Shutdown GT4 container if already started:

    # $GLOBUS_LOCATION/bin/globus-stop-container

    //or ctrl + c from the container console
  - # ant deployGT4

    //from OGSA-DAI binary distribution directory
  - May get copy errors but can safely be ignored
- Undeploying OGSA-DAI WSRF
  - Make sure GLOBUS_LOCATION is set
  - # ant undeployGT4

National e-Science Centre

Expose a Data Resource I

- Deploy an OGSA-DAI data service
  - Change to OGSA-DAI WSRF binary distribution directory
  - # ant deployDataServiceGT4 -Dgt.dir=/path/to/GT4/directory -Ddai.service.path =service/path
    - dai.service.path specifies the local URL of the service. For example -Ddai.service.path=ogsadai/DataService specifies a data service whose URL will be http://HOST:PORT/wsrf/services/ogsadai/DataService.
    - gt.dir specifies the location of GT4 when deploying. If this argument is omitted then the GT4 location specified within GLOBUS_LOCATION is used.
    - Providing the flag -Ddai.service.config=y requests that the data service support dynamic service configuration.
  - Shutdown GT4 container
    - # $GLOBUS_LOCATION/bin/globus-stop-container
      //or ctrl + c from the container console
  - Restart GT4 container *(in a new console)*
    - # $GLOBUS_LOCATION/bin/globus-start-container –nosec
      //OGSA-DAI Data Service interface should be listed
  - Alternatively you could check if service has deployed under GT4 by:
    - ant listResourcesClient -Ddai.url=http://HOST:8080/wsrf/services/ogsadai/MyDataService

National e-Science Centre

Expose a Data Resource II

- Deploy a data service resource
  - Write or edit a data service resource file

Data federation security issues here!

SAMPLE: dt.svr.rsc.prop.postgres
dai.resource.id= testResourceID
dai.data.resource.type=[Relational | XML | Files]
dai.product.name= ***PostgreSQL***
dai.product.vendor= ***Postgres***
dai.product.version= ***7.3.10***
dai.data.resource.uri= **jdbc:postgresql://localhost:5432/testDb**
dai.driver.class= **org.postgresql.Driver** *(optional only if dai.data.resource.type=Files)*
dai.credential= [NO Certificate Provided] or leave blank
dai.user.name= ***testuser***
dai.password= ***testpwd***

  - Deploy using the command-line
    - Change to OGSA-DAI WSRF binary distribution directory
    - #ant deployResourceGT4 -Dgt.dir=**/path/to/GT4/directory** -Ddai.service.resource.file= **dt.svr.rsc.prop.postgres**
    - Ignore any copy warnings

National e-Science Centre

Expose a Data Resource III

- Add a data service resource to a data service
  - Using the command-line
    - Change to OGSA-DAI WSRF binary distribution directory
    - # ant addResourceGT4 -Ddai.service.path=ogsadai/DataService -Ddai.resource.id= testResourceID -Dgt.dir=**/path/to/GT4/directory**
    - Restart GT4 container

      OR
    - To dynamically add a data service resource without the need for restarting GT4:

      # ant dataServiceClient -Ddai.url=http://HOST:PORT/wsrf/services/ogsadai/DataService -Ddai.resource.id= testResourceID -Ddai.action=deploy
  - Testing the new data service resource
    - # ant listResourceClient –Ddai.url=http://HOST:PORT/wsrf/services/ogsadai/DataService

National e-Science Centre

OGSA-DAI WSRF Clients

- ListResources Client
  - List the data service resources exposed by a data service. Example on previous slide
- GetProperty Client
  - Print the properties of a data service resource e.g. a database schema
    - First ensure GT4 container is running and Globus Location is set
    - Execute from OGSA-DAI WSRF binary distribution directory

      # ant getPropertyClient Ddai.url=http://HOST:PORT/wsrf/services/ogsadai/DataService
      –Ddai.resource.id=testResourceID
      –Ddai.property={http://ogsadai.org.uk/namespaces/2005/10/properties}**databaseSchema**
      - This shows the database schema of the data service resource
    - Another property is activityTypes, which shows valid activities for perform documents on the data service resource.
- End-to-End Client
  - Shows a list of data resources
  - Submit a perform document to a data service resource & prints the response
    - # ant dataServiceClient –Ddai.url=SERVICE-URL –Ddai.resource.id=RESOURCE-ID –Ddai.action=PERFORM-DOC-LOCATION