

---

# AstroGrid-D

Deliverable



## Distributed Database Access and Data Stream Management

Access to Persistent Data<sup>1</sup>

Deliverable	D4.3
Authors	Working Group Distributed Database Access and Data Stream Management
Editors	Tobias Scholl
Date	November 22, 2006
Document Version	1.0.0
Current Version	1.0.0
Previous Versions	0.1.0, 0.2.0

### A: Status of this Document

Deliverable D3 of working group 4.

### B: Reference to project plan

Third deliverable of working group *Distributed Database Access and Data Stream Management*.

---

<sup>1</sup>This work is part of the AstroGrid-D project and D-Grid. The project is funded by the German Federal Ministry of Education and Research (BMBF).

**C: Abstract**

Databases play an integral role for storing scientific data sets. Depending on the application and the data characteristics, databases are used to store data itself or its metadata. The astrophysics community and Grid Computing projects aim at providing access to heterogeneous data sources to share their results with colleagues throughout the world.

We describe three aspects, namely access on persistent data by means of *OGSA-DAI*, the integration of databases in our data stream management, and the *Data Management Component (DMC)* developed at the MPA.

**D: Change History**

<b>Version</b>	<b>Date</b>	<b>Name</b>	<b>Brief summary</b>
0.1.0	08.10.2006	Tobias Scholl	Initial draft.
0.2.0	30.10.2006	Tobias Scholl	added corrections by Hans-Martin Adorf.
1.0.0	21.11.2006	Tobias Scholl	incorporated comments and corrections from the review within the project.

**E:****Contents**

<b>Abstract</b>	<b>2</b>
<b>Change History</b>	<b>3</b>
<b>1 Introduction</b>	<b>5</b>
<b>2 OGSA-Data Access and Integration (OGSA-DAI)</b>	<b>5</b>
2.1 Installation . . . . .	5
2.2 Securing Database Access by Certificates . . . . .	6
2.2.1 OGSA-DAI as proxy . . . . .	6
2.2.2 OGSA-DAI with credential delegation . . . . .	8
2.2.3 File Transfer with GridFTP . . . . .	8
<b>3 Streaming Database Access</b>	<b>8</b>
<b>4 Data Management Component (DMC)</b>	<b>10</b>
<b>References</b>	<b>11</b>

## 1 Introduction

Several data sets in astrophysics are stored in databases. In databases, data needs to adhere to predefined schemas and can be accessed using a query language (like SQL). With the introduction of structure comes scalability which is very important to cope with the data deluge of forthcoming years. Out of the variety of relational, semi-structured and object-oriented databases, we concentrate on the relational ones. We do not distinguish between data and metadata in the following discussion. It often depends on the perspective of the application whether a piece of information is regarded as data or metadata.

## 2 OGSA-Data Access and Integration (OGSA-DAI)

The *Open Grid Services Architecture - Data Access and Integration (OGSA-DAI)* project [1, 2] is currently developed and maintained at the e-Science Center in Edinburgh. It is closely integrated with the Globus Toolkit and is active in the standardization process within the *Data Access and Integration Services (DAIS)* working group of the Open Grid Forum (OGF), formerly GGF.

The motivation behind OGSA-DAI is to provide a unified way of accessing resources using web services or Grid services. The interface to integrate resources is very flexible and supports files, RDF,<sup>2</sup> relational or XML databases. The main focus of the OGSA-DAI support are relational and XML databases.

There are two different flavors of OGSA-DAI available: *Web Services Resource Framework (WSRF)* and *Web Services Interoperability (WS-I)*. We concentrate on the WSRF implementation as the Globus Toolkit 4 is also based on this technology.

The DGI-project provides OGSA-DAI as data access interface and has currently started to offer several sites with OGSA-DAI access. Samatha Kottha et al. [3] have conducted a performance evaluation of OGSA-DAI within the MediGrid project for medical applications.

### 2.1 Installation

The installation process is very well documented on the OGSA-DAI website. A basic configuration lasts approximately one hour, provided all prerequisites are fulfilled.

Figure 1 shows the anticipated installation of OGSA-DAI: it allows web services and grid services to access persistent databases. In many situations, these databases are behind institutional firewalls and therefore not directly accessible via Java Database connectivity (JDBC).

In the following the integration of a relational database is in our focus. The current supported databases range from commercial to open source database systems (DB2,

<sup>2</sup><http://projects.gtrc.aist.go.jp/dbwiki/index.php?OGSA-DAI-RDF%20Overview>

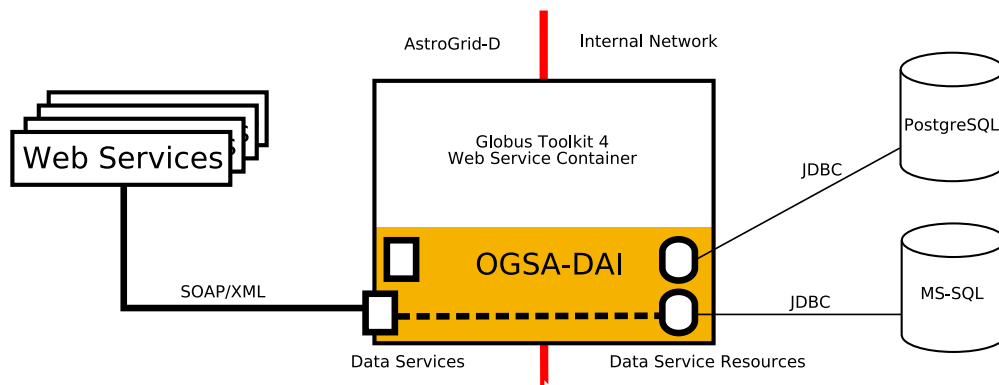


Figure 1: Overview of the OGSA-DAI deployment

MySQL, Microsoft SQL Server, Oracle, PostgreSQL).

Following the documentation<sup>3</sup>, OGSA-DAI is set up correctly within four steps:

1. Install OGSA-DAI into the container (i. e., GT4).
2. Install a *data service*, the interface from the Grid to your OGSA-DAI installation.
3. Deploy *data service resources* which are the connectors to the databases. An example configuration for a PostgreSQL database is shown in Figure 2.
4. After *exposing* the data resources, Grid users can connect to the resources via the data service.

Exposed data resources can be accessed by using *activities*. A relational database offers several activities such as query-, update-, or bulk-insertion-activities. Detailed descriptions of additional activities for other resource types or other tasks (e. g., XML transformation, data delivery using GridFTP) are available from the OGSA-DAI web site.<sup>4</sup>

## 2.2 Securing Database Access by Certificates

OGSA-DAI uses the security components of the Globus Toolkit and therefore is interesting for three scenarios which are described next.

### 2.2.1 OGSA-DAI as proxy

In this scenario, a user (with grid-credentials) contacts the OGSA-DAI service in the Globus container of the data provider. If Globus verifies that the credentials of

<sup>3</sup><http://www.ogsadai.org.uk/documentation/ogsadai-wsrf-2.2/doc/wsrf/install/install.html#Install>

<sup>4</sup><http://www.ogsadai.org.uk/documentation/ogsadai-wsrf-2.2/doc/interaction/Activities.html>

```
## (c) International Business Machines Corporation, 2002 - 2005.
## (c) University of Edinburgh, 2002 - 2005.
## See OGSAs-DAI-Licence.txt for licencing information.

## 1-Define the identifier of the resource
dai.resource.id=DataServiceResource

## 2-Specify type of resource
dai.data.resource.type=Relational

## 3-Enter metadata and connection information to the data resource
dai.product.name=PostgreSQL
dai.product.vendor=Postgres
dai.product.version=8.1
dai.data.resource.uri=jdbc:postgresql://localhost:5432/ogsadai
dai.driver.class=org.postgresql.Driver

## 4-(optional) Give credentials to access the data resource
dai.credential=

## 5-database user name and password to access the database
dai.user.name=user
dai.password=password

## 6-Put any JAR files containing your database drivers into
## the "drivers directory" NOW.

## 7-To deploy your data service resource:
## A-Save this file
## B-To deploy on your container, run:
## "ant deployResource -Ddai.container=PATH_TO_WEB_SERVICES_CONTAINER"
```

Figure 2: Exemplary data service resource configuration for a PostgreSQL database

the user are valid, the user is able to access the database. There are several benefits for clients and administrators. On the one hand, users do not need to apply for a database accounts as they are only using the Grid service interface and their credentials. On the other hand, administrators only need to create and maintain one account which is used by the OGSA-DAI service in order to interface with the database. For database users from the Grid, the OGSA-DAI service acts as a proxy.

### 2.2.2 OGSA-DAI with credential delegation

This scenario is important, when data providers want to differentiate between users with varying access rights. The subject of a user certificate (its *distinguished name*) can be mapped onto a database account with appropriate rights.<sup>5</sup> How this can be realized based on Virtual Organizations is outside the scope of this working group.

### 2.2.3 File Transfer with GridFTP

In some use cases (e.g. Clusterfinder) subsets of data which is stored in a database need to be transferred to compute nodes. OGSA-DAI allows the result of an SQL query to be transformed into a *comma separated values (CSV)* file on a GridFTP server. The result is directly piped to the GridFTP server, that means there is no file created on the OGSA-DAI server before transmitting the data. A perform-document for such a file transfer is shown in Figure 3.

## 3 Streaming Database Access

Currently, our prototype of the data stream management component does not yet access data from databases directly and does not store result streams into a database. Therefore we want to extend our data stream management system to access data in databases directly, and to support storing stream results in a database.

The database will be accessed using OGSA-DAI. The received data will be transformed into a WebRowSet (an XML format defined by Sun Microsystems). Using an XSLT Stylesheet the WebRowSet gets transformed into a stream which conforms to a given DTD and this will be published by the ContentProvider. Using the bulk-insert functionality of OGSA-DAI, we can feed data streams back into databases. Figure 4 illustrates the described scenario.

<sup>5</sup><http://www.ogsadai.org.uk/documentation/ogsadai-wsrf-2.2/doc/reference/config/RoleMap.html>

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- (c) International Business Machines Corporation, 2002 - 2005.-->
<!-- (c) University of Edinburgh, 2002 - 2005.-->
<!-- See OGSA-DAI-Licence.txt for licencing information.-->
<perform
  xmlns="http://ogsadai.org.uk/namespaces/2005/10/types">

  <documentation>
    This perform document demonstrates how to deliver
    data from a data service by Grid FTP.
  </documentation>

  <sqlQueryStatement name="statement">
    <expression>
      select * from mytable
    </expression>
    <resultStream name="statementOutputRS"/>
  </sqlQueryStatement>

  <sqlResultsToCSV name="results">
    <resultSet from="statementOutputRS" />
    <delimiter value="," />
    <lineBreak value="LFCR" />
    <nullDataStr value="NULL" />
    <includeHeader value="true" />
    <escapeFields value="true" />
    <csvOutput name="CSV" />
  </sqlResultsToCSV>

  <deliverToGFTP name="delivery">
    <fromLocal from="CSV"/>
    <toGFTP host="astrogrid.aei.mpg.de" port="2811" file="/tmp/scholl.xml" append="false" />
  </deliverToGFTP>

</perform>

```

Figure 3: Perform-document for transferring the result of an SQL query to a GridFTP server

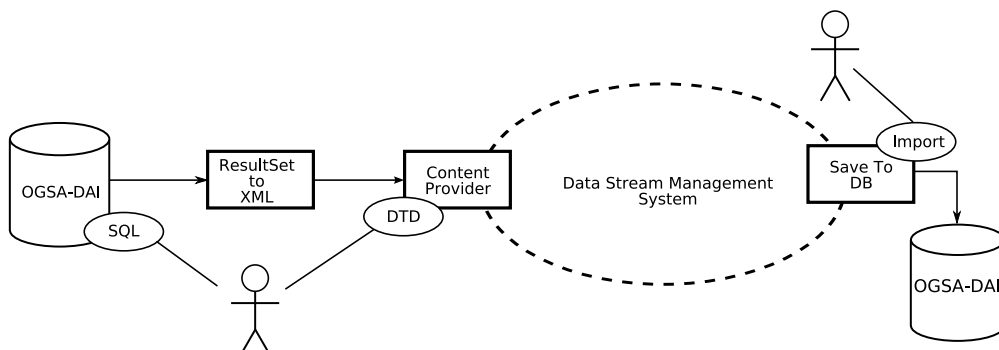


Figure 4: Integration of persistent data with the data stream management.

## 4 Data Management Component (DMC)

An alternative component to store metadata and data within a database independently of a specific database system is the *Data Management Component (DMC)* developed by the Planck group at MPA. Having started with an object-oriented database as backend, they now added support for relational databases and currently use Oracle 9i.

The DMC is written in Java and uses the *Java Data Object (JDO)* API-model developed by Sun Microsystems to directly store its Java Objects into the database. For further details on JDO we refer to the SUN-webpage.

Besides the Java-API, the DMC offers libraries for C and Fortran. The astrophysicists then can use these "science-ready" wrappers.

The developers currently envision two usage scenarios for the DMC: as logistics engine or as data storage. The ProC, for example, uses the DMC for keeping track of the metadata and parameter files. The database enables the users to describe more complex queries on the metadata. Used as data storage, the DMC offers partial data access (defined on a semantic level) using the query language.

**F: References / Bibliography****References**

- [1] M. Antonioletti, M.P. Atkinson, R. Baxter, A. Borley, N.P. Chue Hong, B. Collins, N. Hardman, A. Hume, A. Knox, M. Jackson, A. Krause, S. Laws, J. Magowan, N.W. Paton, D. Pearson, T. Sugden, P. Watson, and M. Westhead. The design and implementation of Grid database services in OGSA-DAI. *Concurrency and Computation: Practice and Experience*, 17(2-4):357–376, 2005.
- [2] K. Karasavvas, M. Antonioletti, M.P. Atkinson, N.P. Chue Hong, T. Sugden, A.C. Hume, M. Jackson, A. Krause, and C. Palansuriya. Introduction to OGSA-DAI Services. *Lecture Notes in Computer Science*, 3458:1–12, 2005.
- [3] S. Kottha, K. Abhinav, R. Müller-Pfefferkorn, and H. Mix. Accessing Bio-Databases with OGSA-DAI – A Performance Analysis. In *Proc. of the Intl. Workshop on Distributed, High-Performance and Grid Computing in Computational Biology*, Eilat, Israel, 2006.