

OGSA-DAI Architecture and Performance Analysis

Samatha Kottha

ZIH, TU Dresden

1. Introduction
2. Terminology
3. Architecture
4. Installation & Configuration
5. Data Access
6. Performance Analysis
7. Conclusion

Data Consumers vs Providers

Data Consumer



My application requires data from DB1, DB2 ...

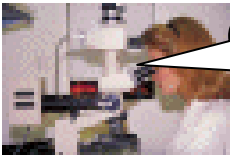
How can I access it?

Please provide an interface for my application?

I need several databases. I need to be a part time DBA?

Ohh! I have to change my application code?

Where can I find data? It is growing fast and complex. We will move shortly to Oracle?

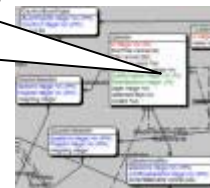
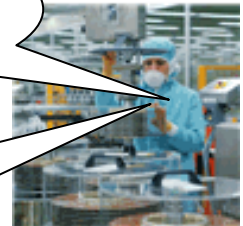


Data Providers

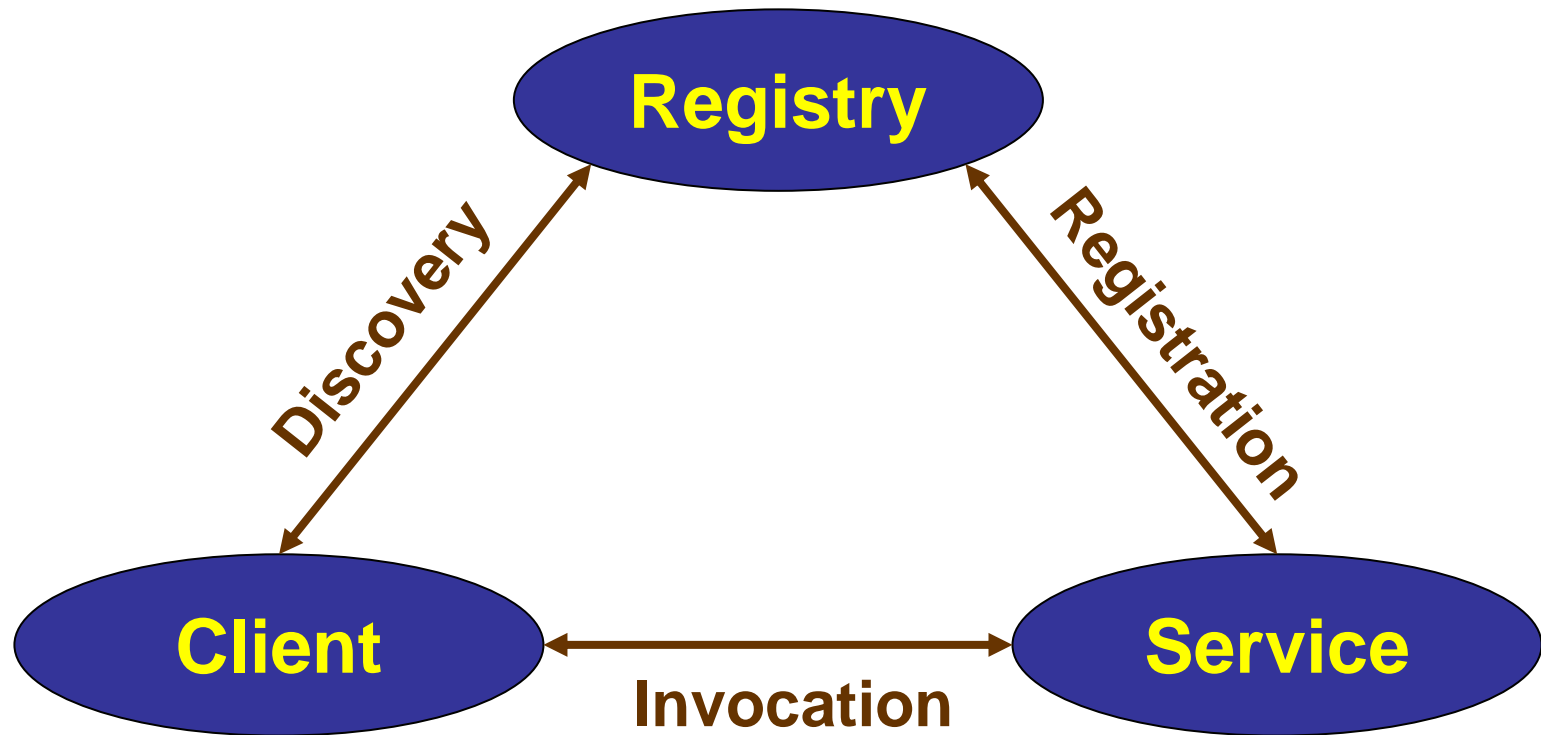
We have a database X!

Do not have resources to such user support? Use dump files.

We can provide you files?



Service oriented architecture



Introduction (Cont.)

- Open Grid Service Architecture - Data Access and Integration (OGSA-DAI).
- Is a middleware.
- Transparent access to distributed and heterogeneous data management systems.
- <http://www.ogsadai.org.uk/>.
- Developed in UK by epcc, neresc, National e-science Center, IBM, and Oracle.

Introduction (Cont.)

- OGSA-DAI could be deployed on
 - Globus Container (GT4) - [OGSA-DAI WSRF](#)
 - Apache Tomcat Server - [OGSA-DAI WSI](#)
 - On Both - [OGSA-DAI WSRF](#)
- Tested release version is 2.1
- New release version is 2.2 (Released on 27th April 2006)

An extensive framework for building applications

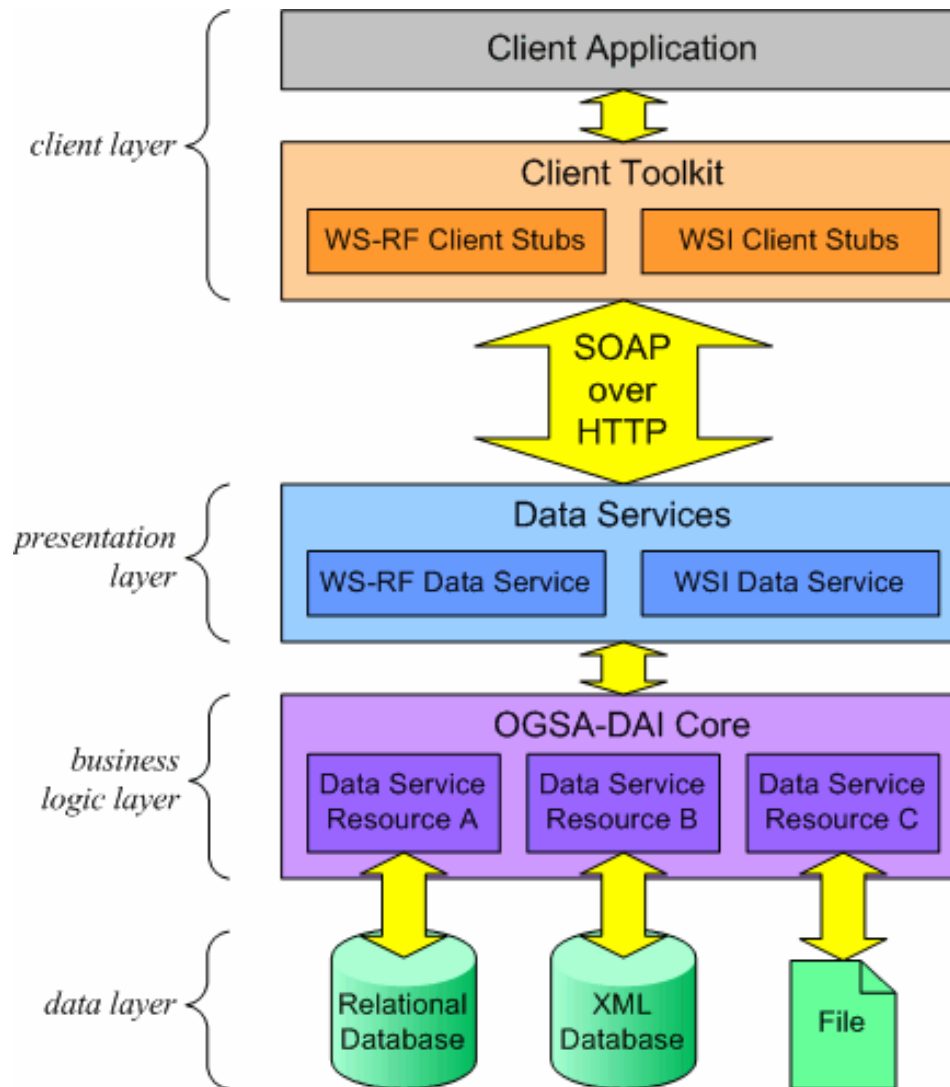
- Supports relational, XML, and some type of files
- Supports various delivery options
- Supports various transformations
- Supports message level security
- Client tool kit
- Documentation and tutorials

- **Data Resource:** Any database or a file system
- **Data Service Resource:** Exposes an underlying data resource and provide the access authorization
- **(Grid) Data Service (GDS):** Is an interface for data service resources
- **Activities:** The operations that a data service resource can perform

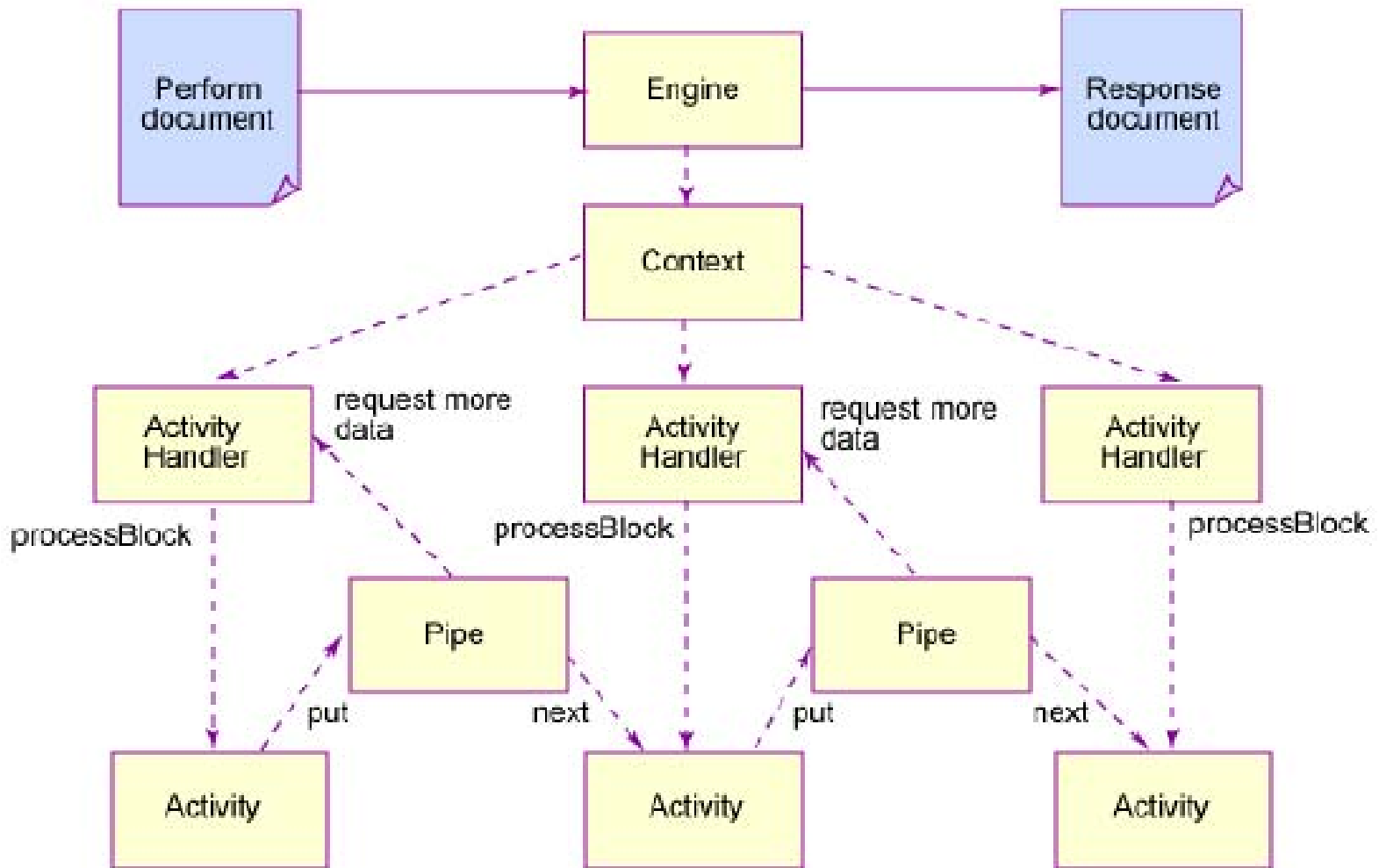
Terminology (Cont.)

- **Factory:** A service to create a GDS instance to access a specific data resource.
- **Service Group Registry:** A service to find the GDS you are looking for or a factory to create the GDS required.
- **Perform Document:** A XML document that specifies the activities to be executed on the GDS.
- **Response Document:** Used by OGSA-DAI services to inform clients as to the status of execution of their Perform documents.

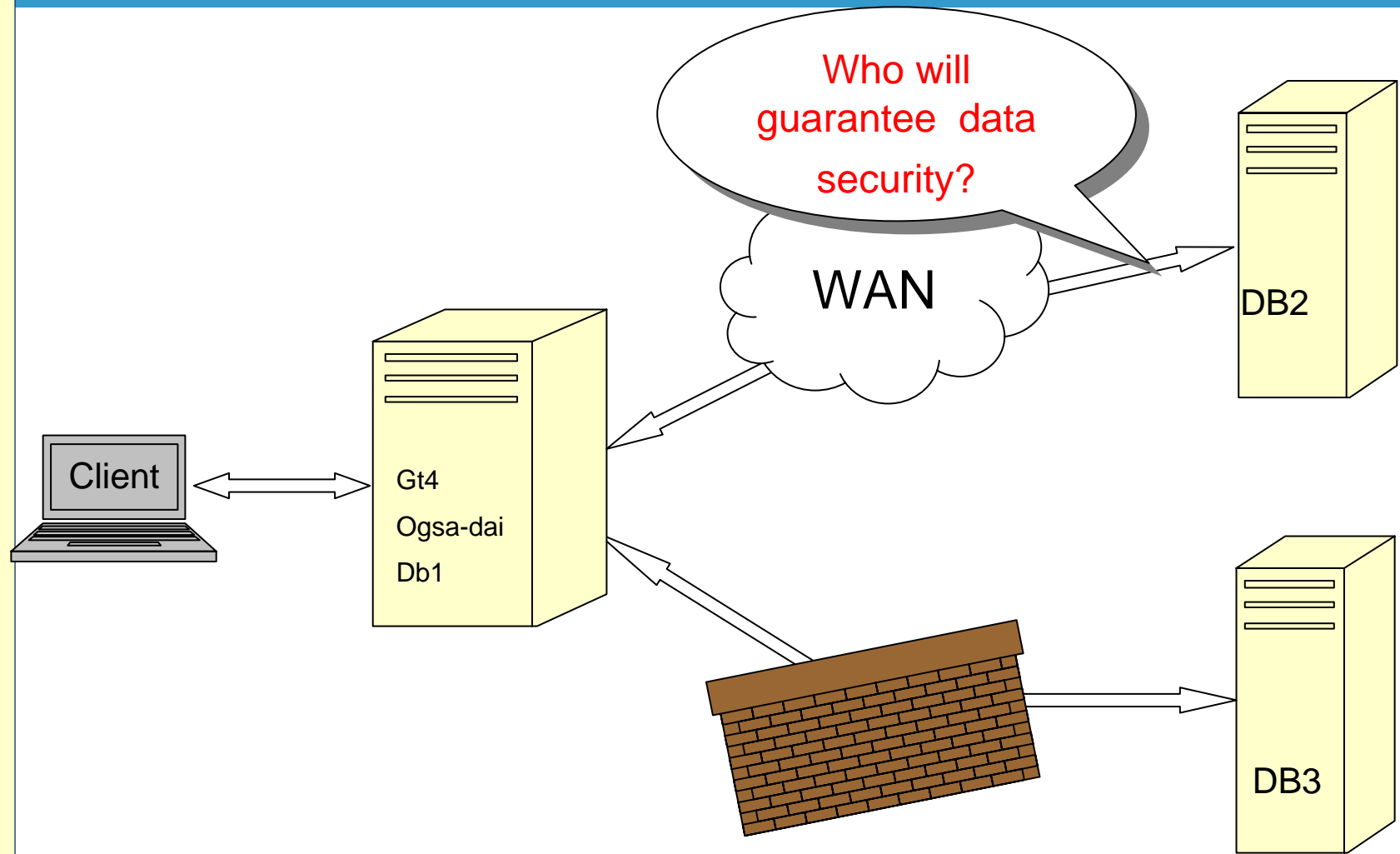
Architecture



Architecture (Cont.)

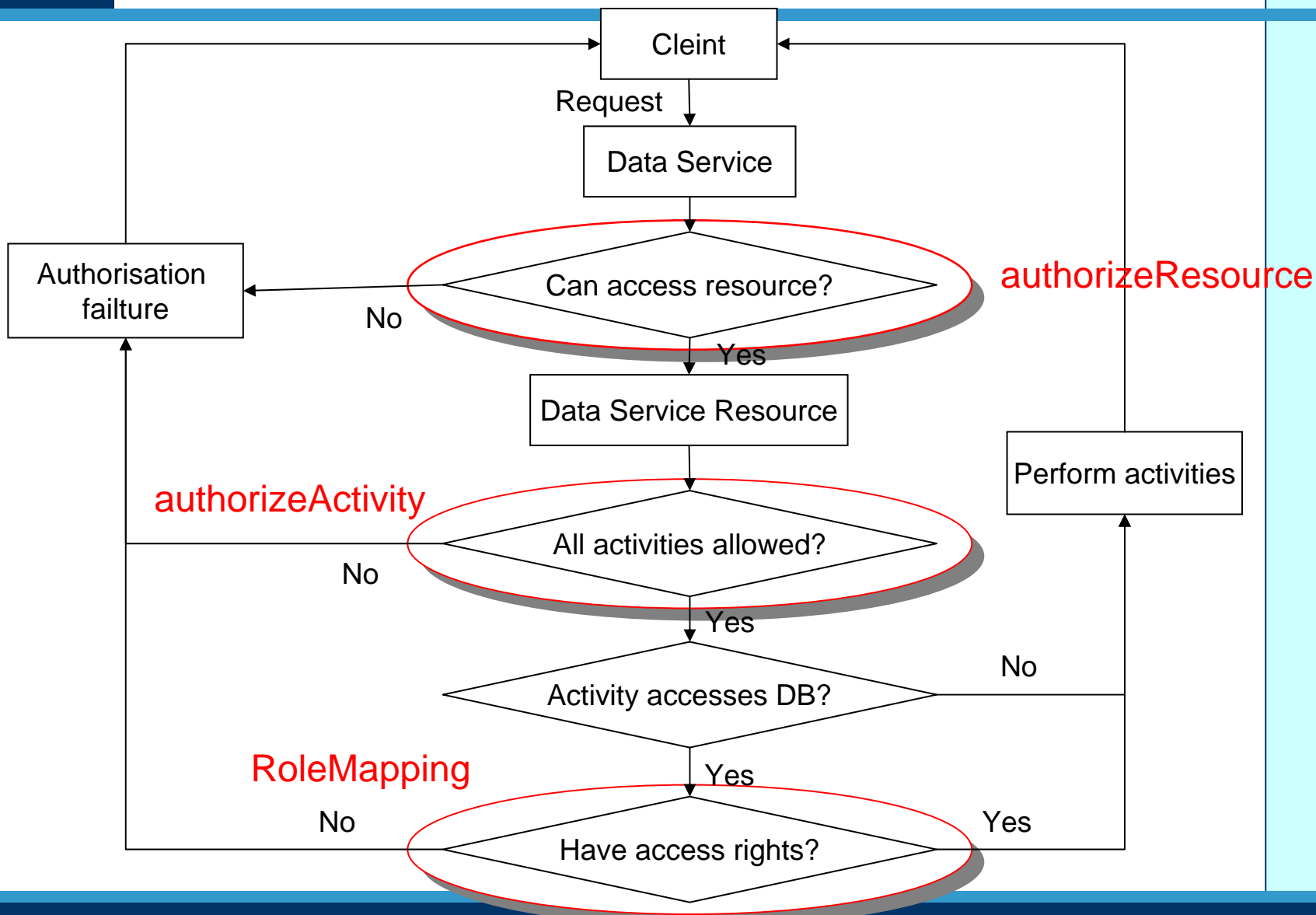


- OGSA-DAI installation is straight forward and binaries are available
- Command line as well GUI installer and uninstaller
- Deploy a new data service
- Listresources client
- Deploy a new data service resource
- Expose the data service resource



- Using DB user and password during the data service resource creation.
 - Not possible to provide user authorization.
- Using GSI credentials.
 - We can provide user authorization using role mapping.
 - Authorizeresource (WSRF 2.2).
 - Authorizeactivity (WSRF 2.2).

Security breach: all the information is stored in plain XML files.



OGSA-DAI Landscape at ZIH



4-way Itanium @ 1.4 GHz
8 GB RAM , 750 GB Disk
Suse 9.3 64-bit
GT 4.0.1, OGSA-DAI WSRF 2.1
Oracle, MySQL DBs of
BioInformatik



1-way AMD Athlon 64 3800+ @ 1 GHz
2 GB RAM , 250 GB Disk
Suse 10.0 64-bit
GT 4.0.2, OGSA-DAI WSRF 2.2
MySQL DB



1-way AMD Athlon XP 1800+ @ 1.5 GHz
750 MB RAM , 50 GB Disk
Suse 9.3 32-bit
GT 4.0.1, OGSA-DAI WSRF 2.1
MySQL DB

Short Summary

We know now

- What is OGSA-DAI?
- What are prerequisites?
- How to install?
- How to configure?
- How to use?

- ListResources client
- GetProperty Client
- End-to-end Client
- DataBrowser

Client Tool Kit (JAVA API)

An activity can pipe its output to another activity.

- **sqlQueryStatement:** Queries relational databases with a JDBC connection using an SQL SELECT expression. Returns WebRowSet XML document.
- **sqlUpdateStatement:** Updates relational databases using SQL CREATE, UPDATE, INSERT, or DELETE expression.
- **sqlBulkLoadRowSet:** To load WebRowSet XML document into a relational database.

Delivery Activities

- **deliverFromURL & deliverToURL**: It delivers data from or to an URL. HTTP, HTTPS, and FTP protocols are supported.
- **deliverFromFile & deliverToFile**: It delivers data from or to a file on the file system of GT4 web service container.
- **deliverFromGFTP & deliverToGFTP (only WSRF)**: It delivers data from or to a file on GridFTP Server.
- **outputStream**: Can deliver as single block or multiple block objects.

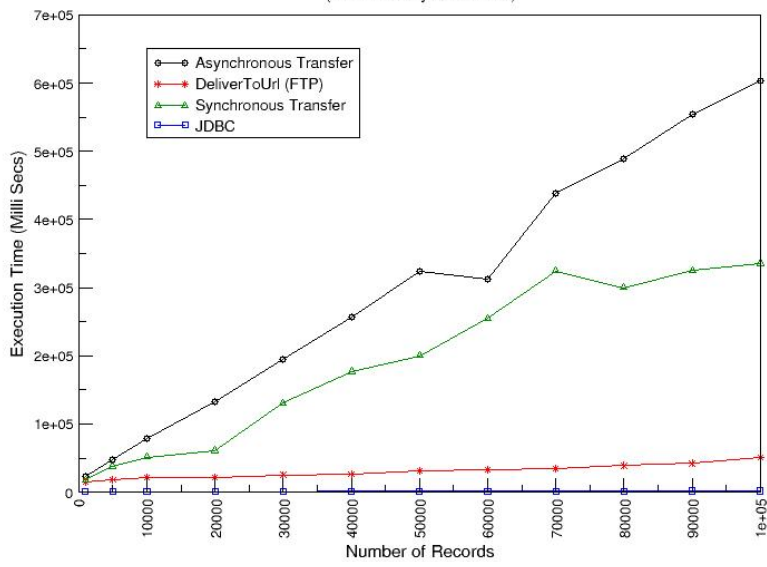
- **zipArchive**
- **gzipCompression**
- **gzipDecompression**
- **xslTransform**
- **stringTokenizer**



Performance of Delivery Activities

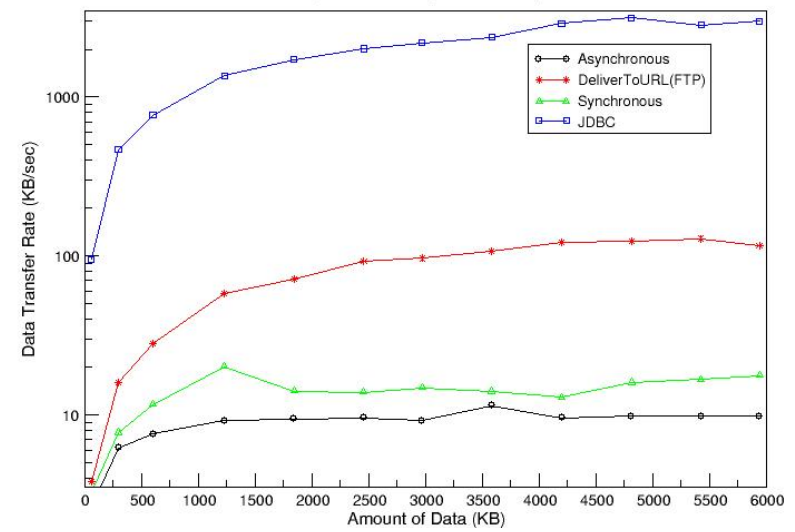
Data Transfer in Secure Mode

(A Select Query on Local DB)



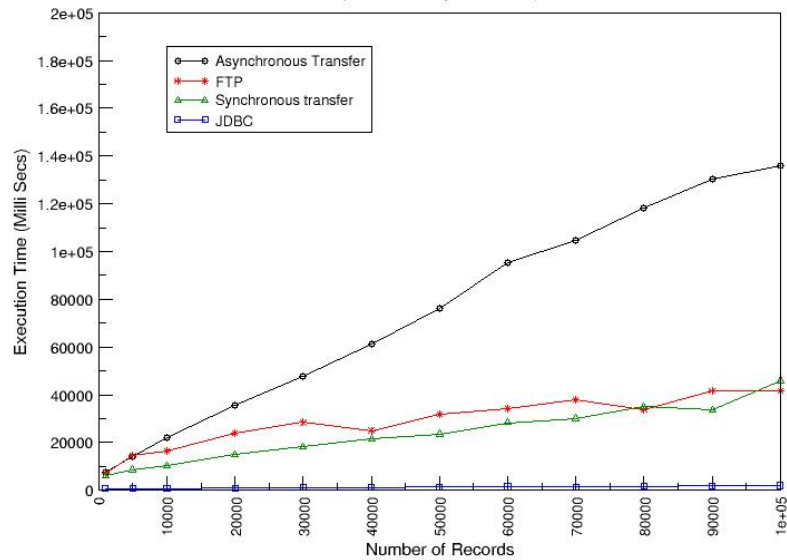
Data Transfer in Secure Mode

(A Select Query on Local DB)



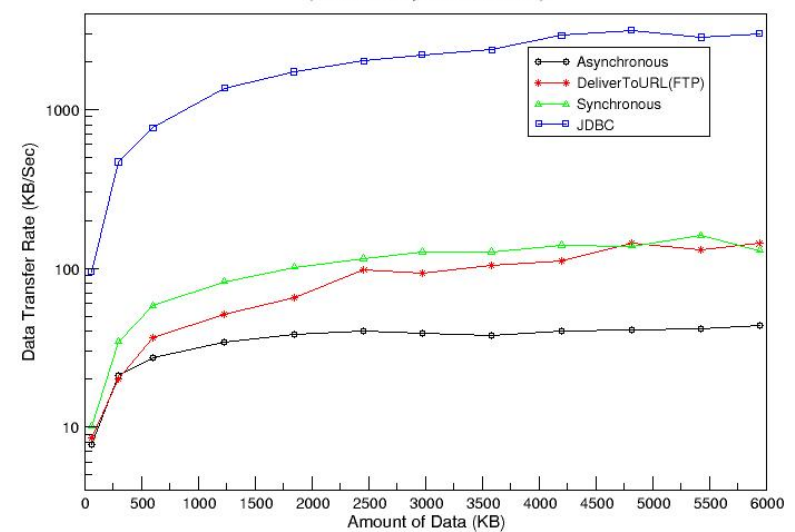
Data Transfer in Non-Secure Mode

(A Select Query on Local DB)



Data Transfer in Non-Secure Mode

(A SQL Query on a Local DB)



Observed performance degradation: Factor of 1.5 to 8 and 2.5 to 15 times for non-secure and secure loading respectively and 11 to 20 and 25 to 26 times for non-secure and secure retrieving respectively.

Then why we need it?

- Provides transparent interface to heterogeneous data resources.
- Data Providers <-> Data Resource Providers <-> Users.
- Provides message and transport level security.
- Of course, it is for GRID computing.
- Scalable, flexible, and location independent.
- Provides additional functionality than JDBC like transformation and compressions.
- Language independence at the client end – Need not be in Java.
- Platform independence – Need not bother about connection technology or drivers etc.

Thank you!!

Questions?