



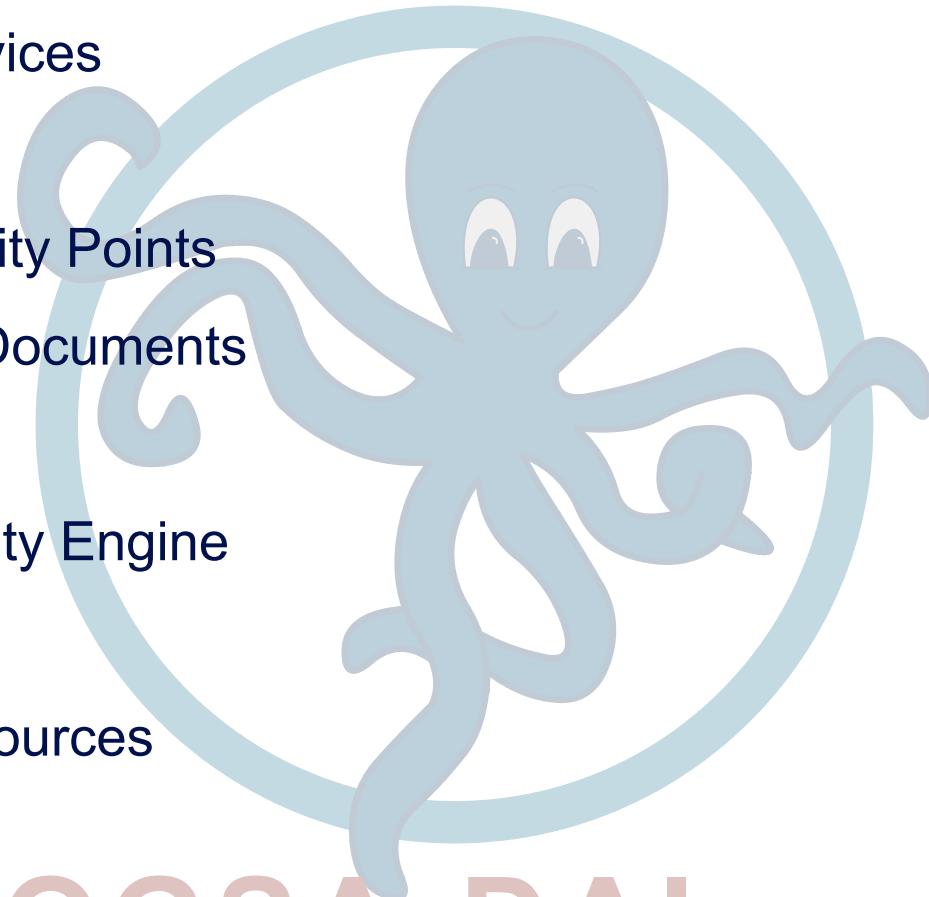
OGSA-DAI Architecture

OGSA-DAI Technology Update
GridWorld Community Activity
GGF15, Boston, MA (USA)

Amy Krause
EPCC

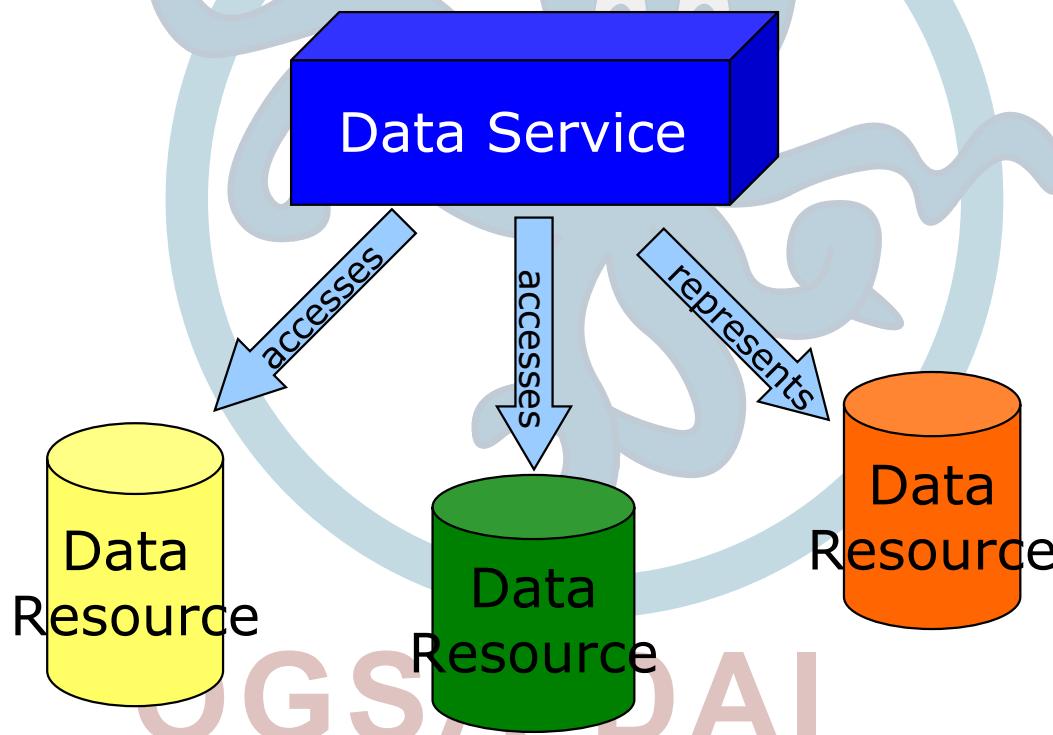
a.krause@epcc.ed.ac.uk

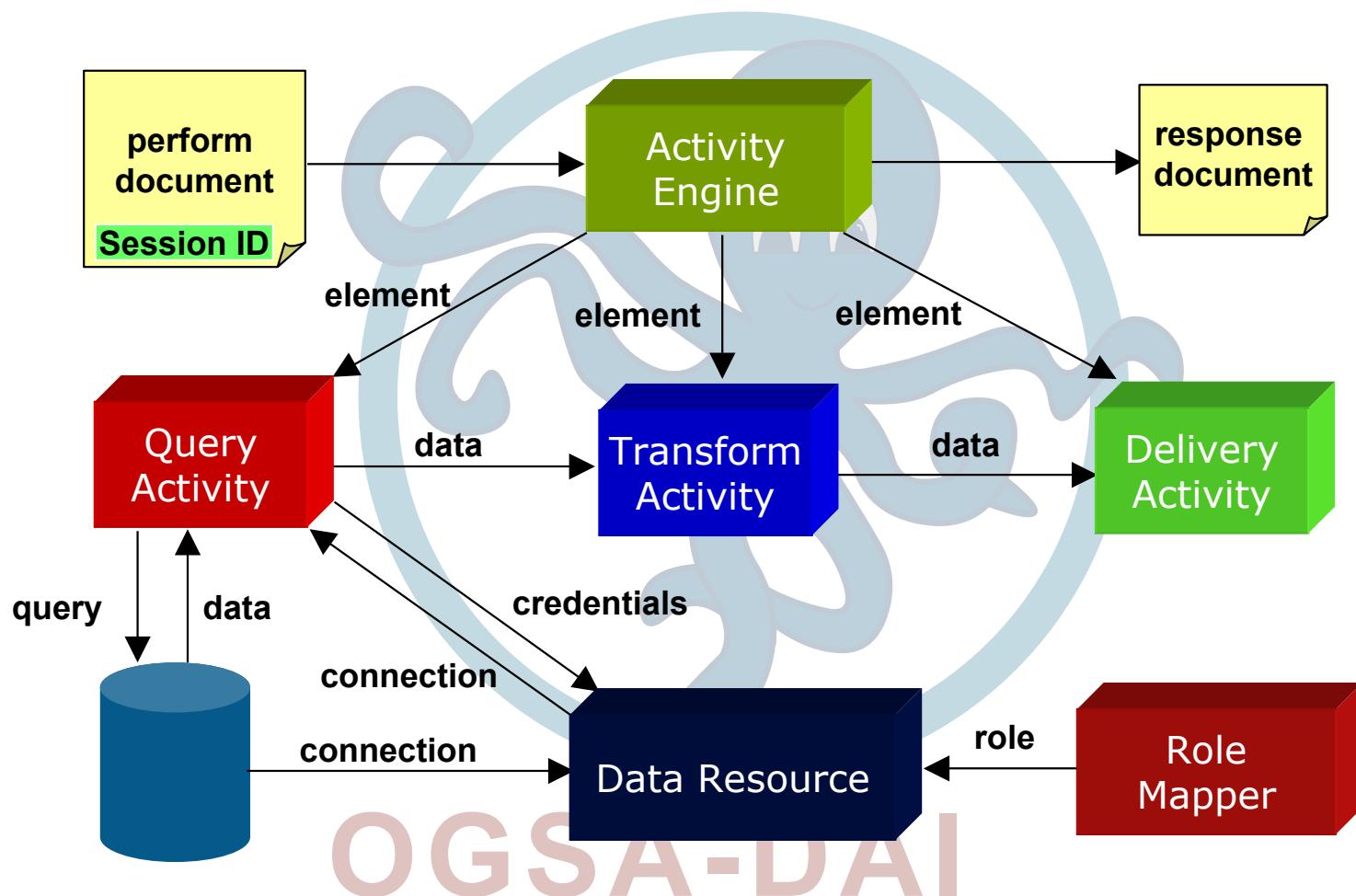
- Data Services
- Internals
- Extensibility Points
- Perform Documents
- Sessions
- The Activity Engine
- Activities
- Data Resources



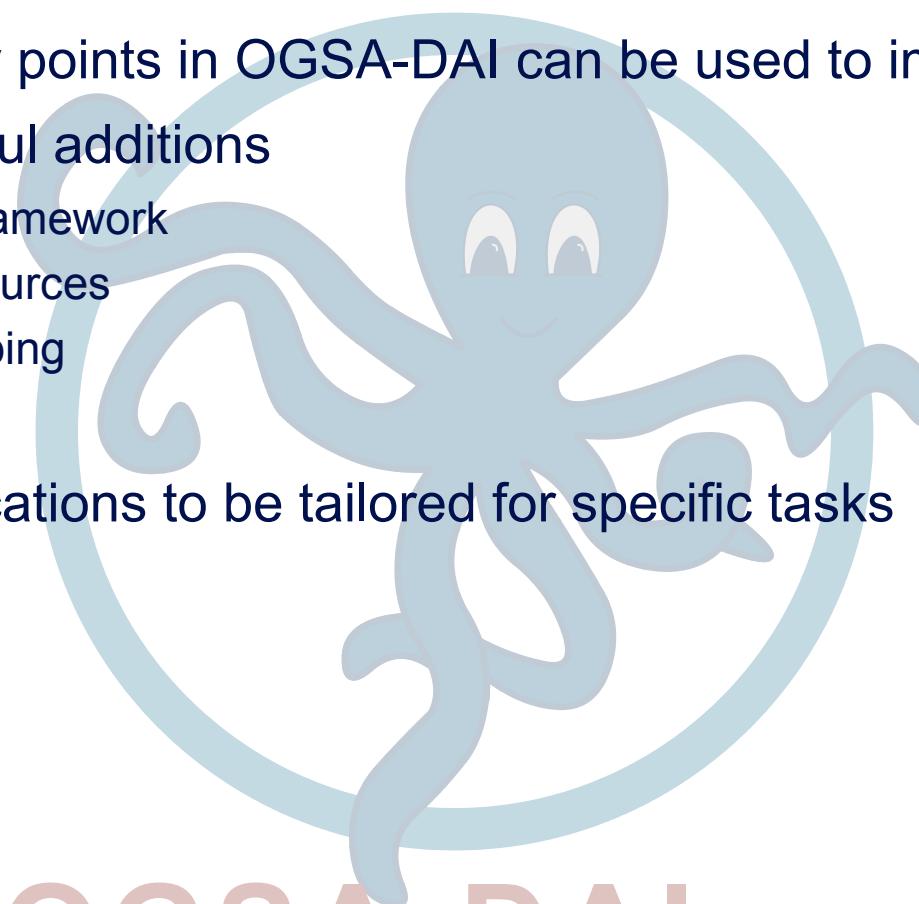
OGSA-DAI

- OGSA-DAI uses data services to represent and provide access to a number of data resources





- Extensibility points in OGSA-DAI can be used to implement very powerful additions
 - Activity Framework
 - Data Resources
 - Role Mapping
 - Engine
- Allow applications to be tailored for specific tasks



OGSA-DAI

- The Engine is the central OGSA-DAI component
- Dictates behaviour when perform documents are submitted
 - Manages concurrent requests and sessions
 - Parses and validates perform document
 - Identifies required activities implementations
 - Processes activities
 - Composes response document
 - Returns response document to the service layer

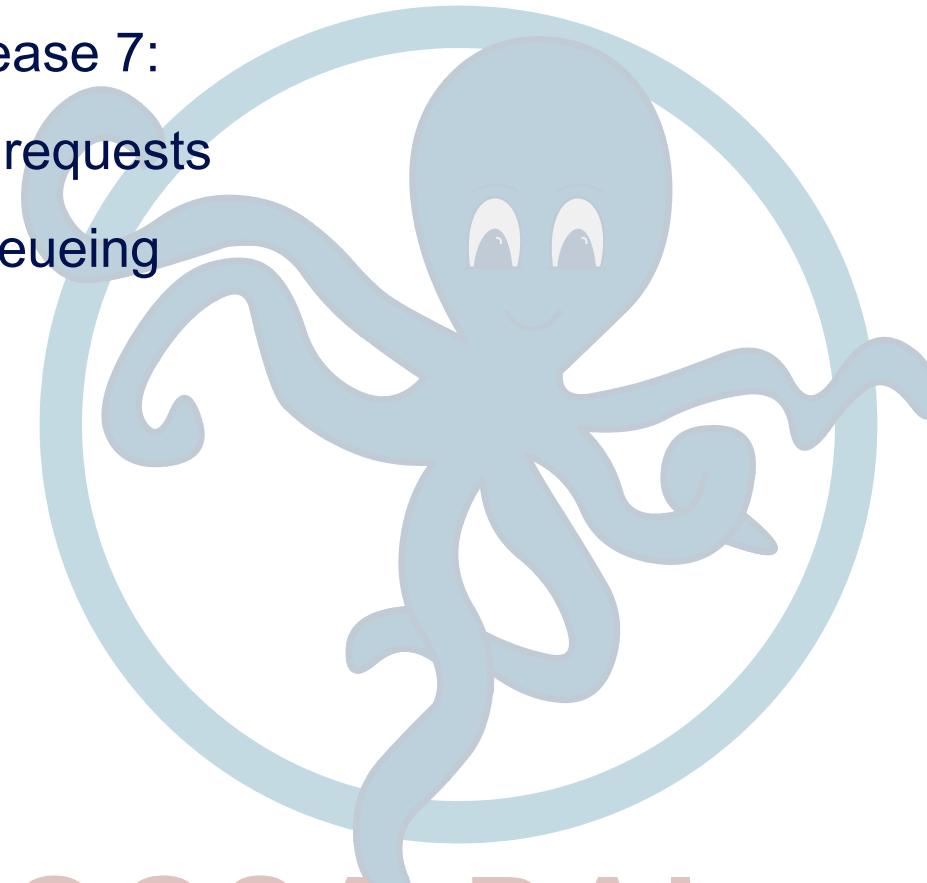




Concurrent Requests

epcc1

- New in Release 7:
- Concurrent requests
- Request queueing
- Sessions



OGSA-DAI



Perform Documents

epcc1

- Perform documents

- Encapsulate a serialisation of multiple interactions with a service into a single interaction
- Abstract each interaction into an “activity”
- Data can flow from one activity to another
- No control constructs present
 - no conditionals, loops or variables

- Not intended for human consumption
 - Generated and processed by client toolkit





OGSA-DAI

Example Perform Document

iepcci

```
<?xml version="1.0" encoding="UTF-8"?>

<perform xmlns="http://ogsadai.org.uk/namespaces/2005/03/types">

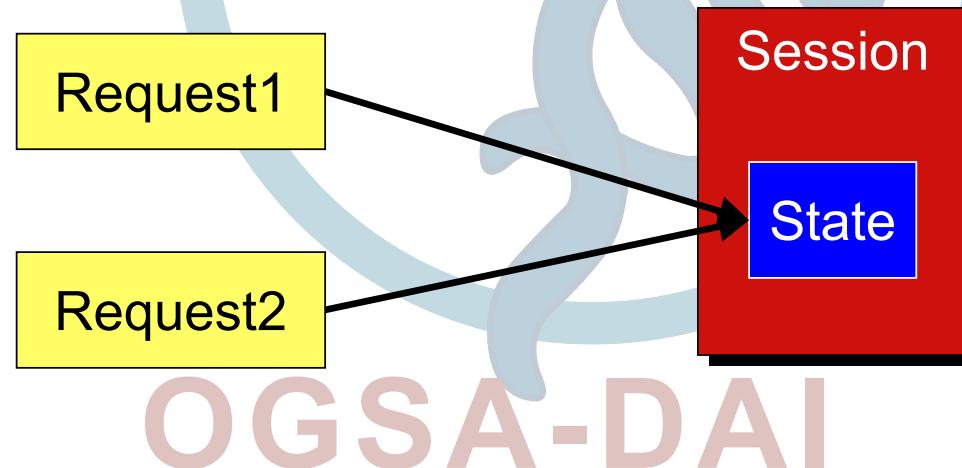
<documentation>
    This example performs a simple select statement to retrieve 100 rows from
    the test database. The results are delivered in the response document.
</documentation>

<sqlQueryStatement name="statement">
    <expression>select * from littleblackbook where id <= 100</expression>
    <webRowSetStream name="statementOutput"/>
</sqlQueryStatement>

</perform>
```

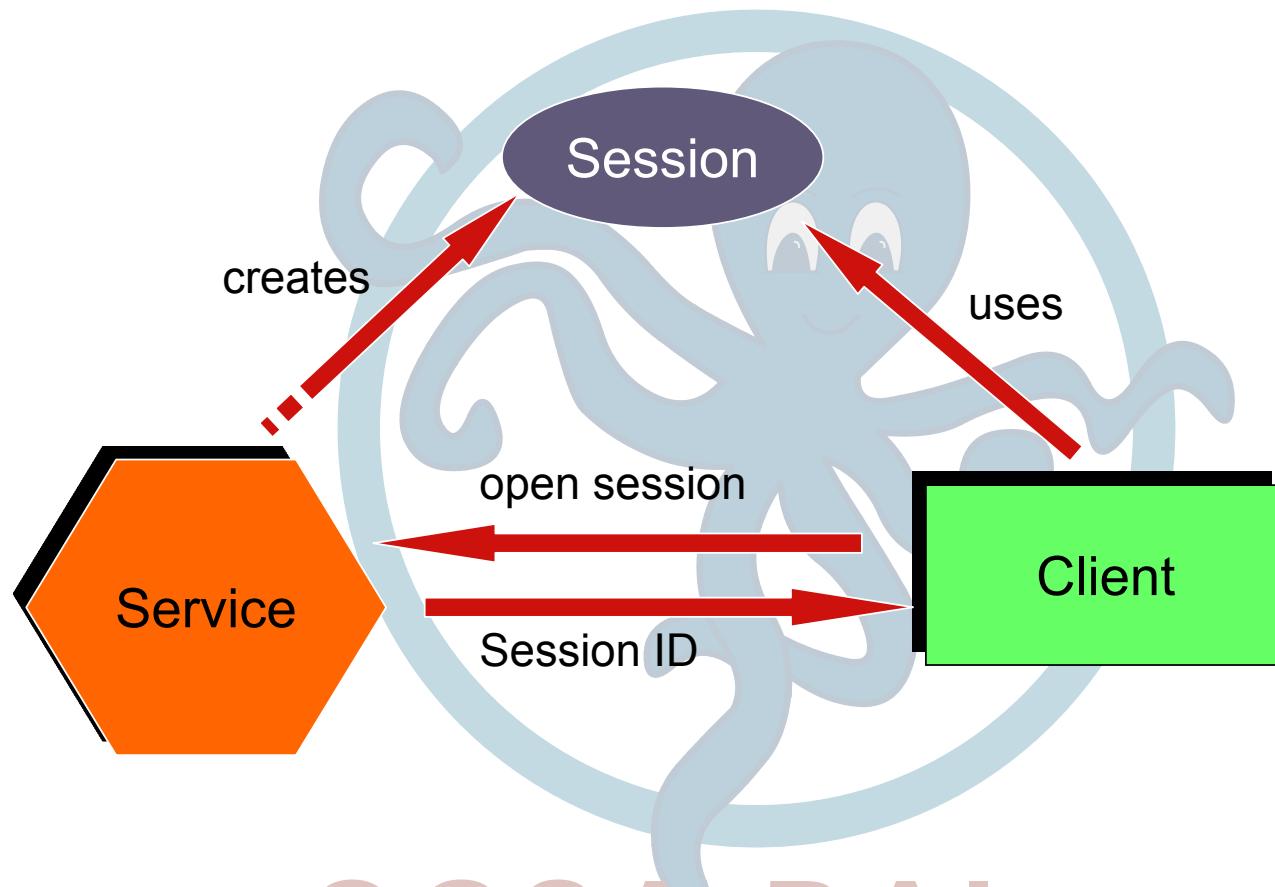
OGSA-DAI

- Sessions provide the ability to store state across multiple requests
 - e.g. a JDBC connection can be shared in order to process multiple requests in a transaction
- State is stored and modified in property objects which can be accessed by all activities within a session



- A new session for multiple request must be requested explicitly, by sending a *openSession* request
- The service provides a unique session ID
- Follow-on requests may join this session by referencing the session ID
- A session may be terminated explicitly or left to expire
- An implicit session is started automatically by any request that doesn't reference another session
 - This session is terminated after the request

OGSA-DAI



OGSA-DAI

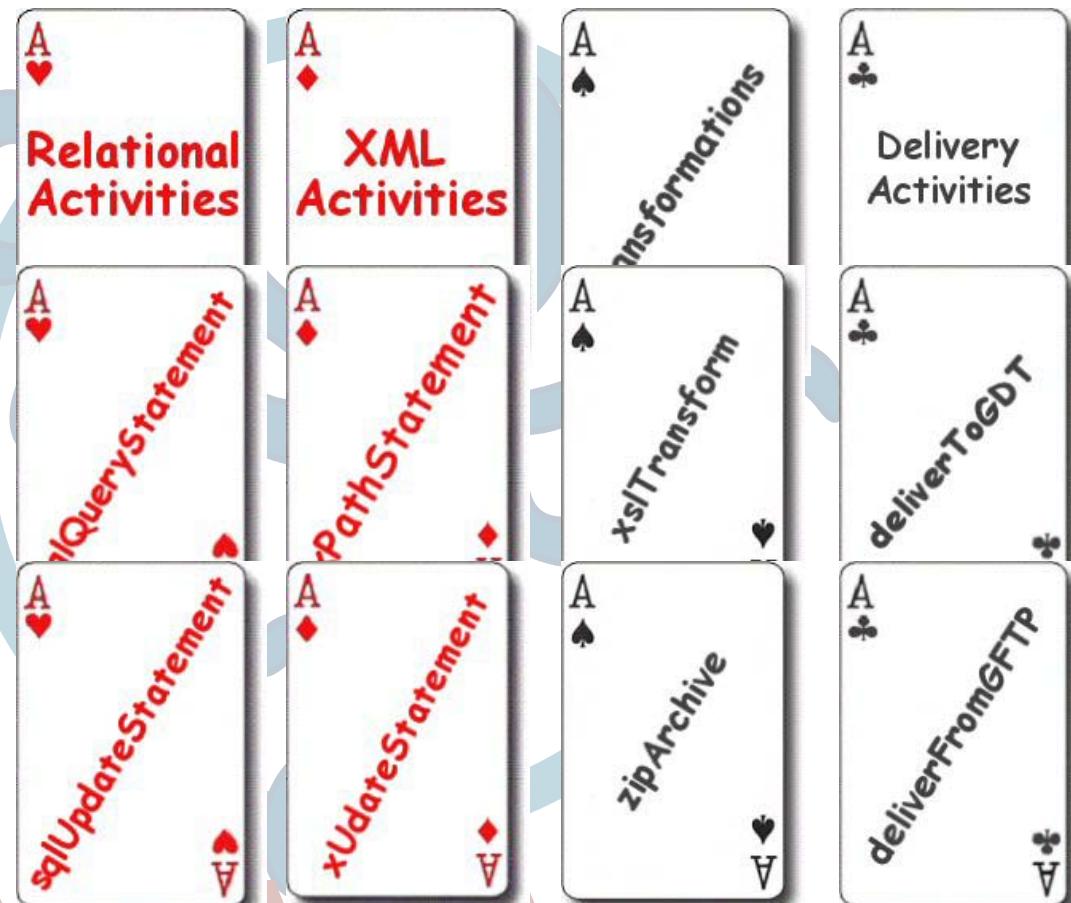
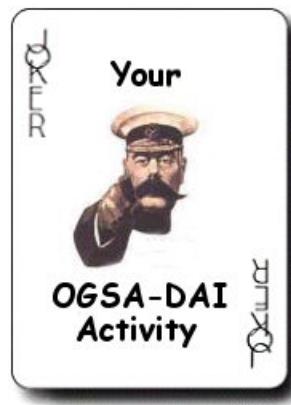
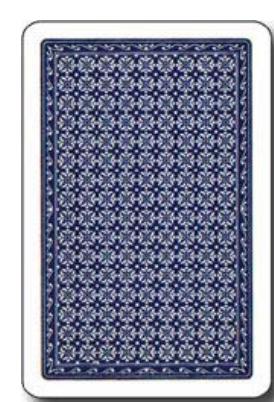
- An Activity dictates an action to be performed
 - Query a data resource
 - Transform data
 - Deliver results
- Engine processes a sequence of activities
- Subset of activities available to a Data Resource
 - Specified in configuration files
- Data can flow between activities





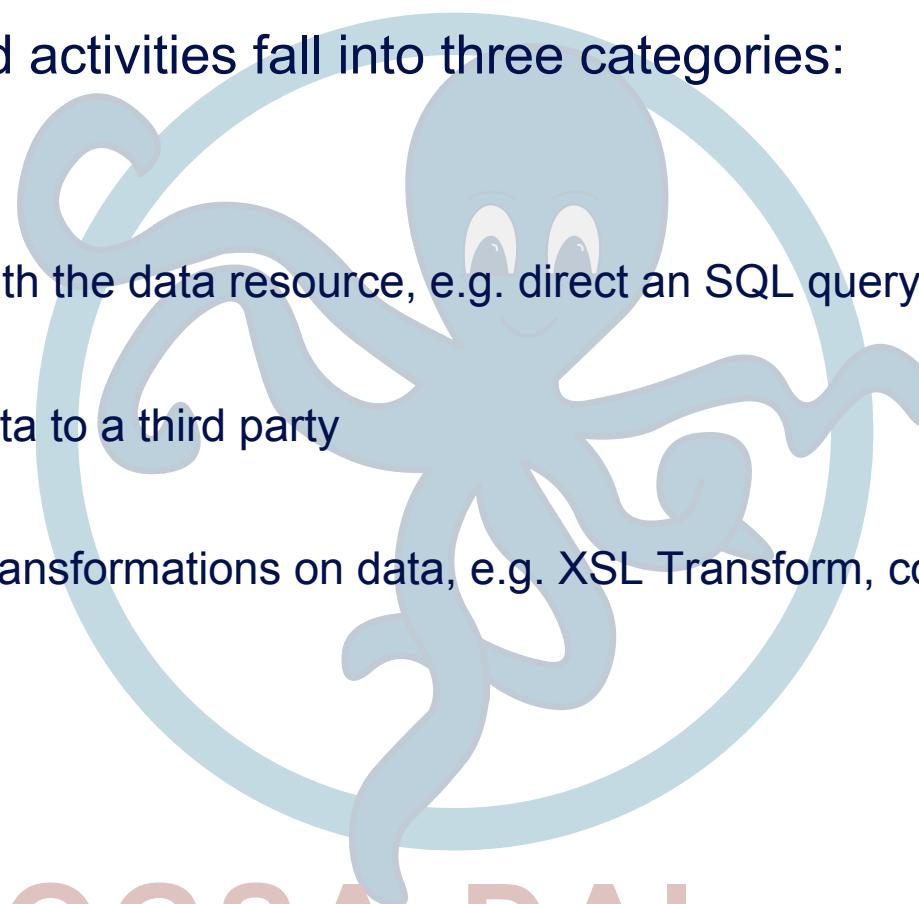
OGSA-DAI Deck of Activities

epcc1

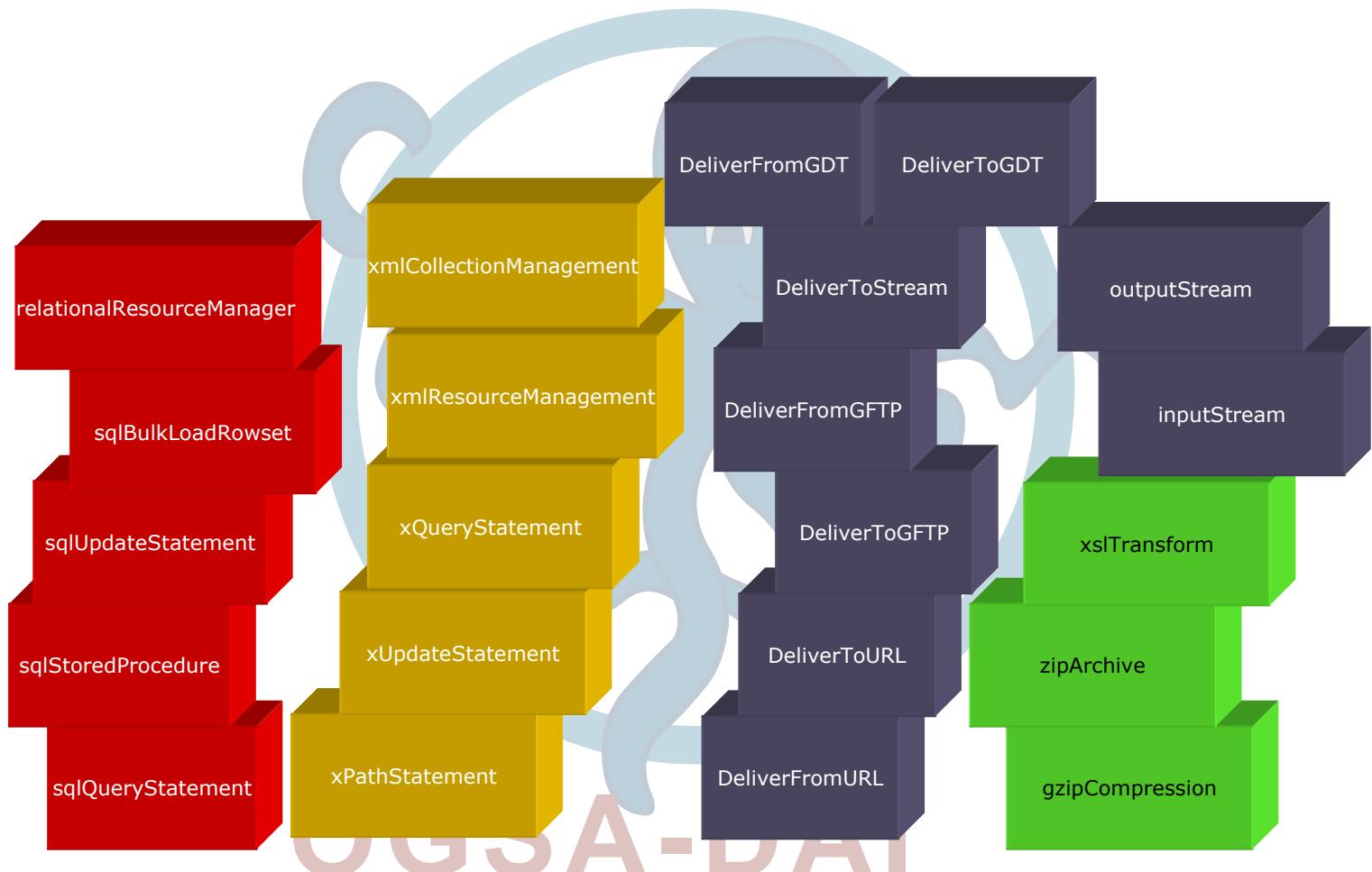


OGSA-DAI

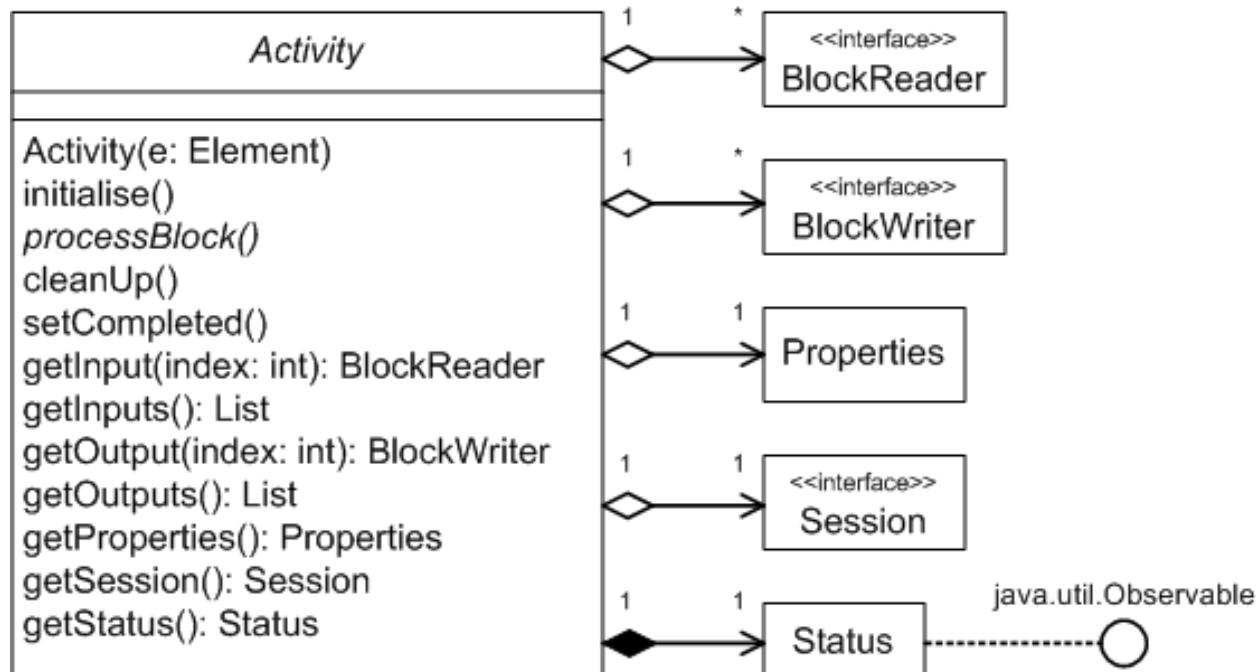
- Predefined activities fall into three categories:
- Statement
 - Interact with the data resource, e.g. direct an SQL query to a DBMS
- Delivery
 - Deliver data to a third party
- Transform
 - Perform transformations on data, e.g. XSL Transform, compression



OGSA-DAI



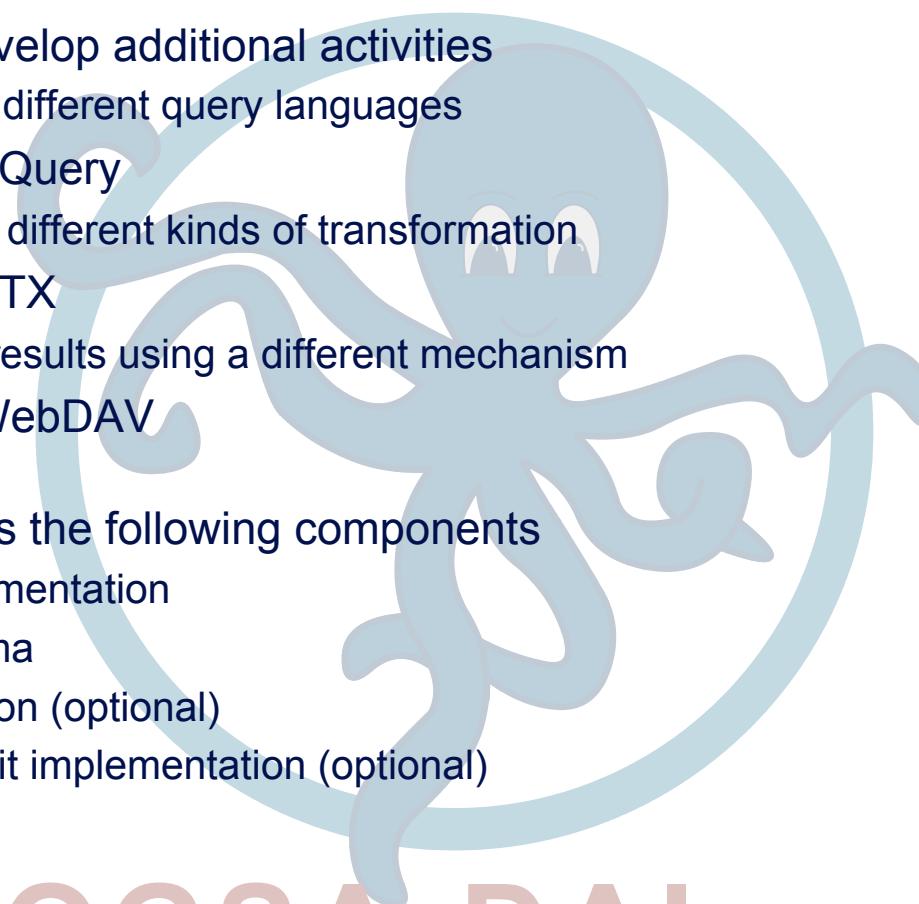
- Extensibility point
- All Activity implementations extend the abstract Activity class



- Activities read and write blocks of data
 - Allows efficient streaming between activities
 - Reduces memory overhead
- A block is a Java Object
 - Untyped but usually a String or byte array
- Interfaces for reading and writing
 - BlockReader and BlockWriter



- Users can develop additional activities
 - To support different query languages
 - e.g. XQuery
 - To perform different kinds of transformation
 - e.g. STX
 - To deliver results using a different mechanism
 - e.g. WebDAV
- An activity has the following components
 - Java implementation
 - XML schema
 - Configuration (optional)
 - Client toolkit implementation (optional)



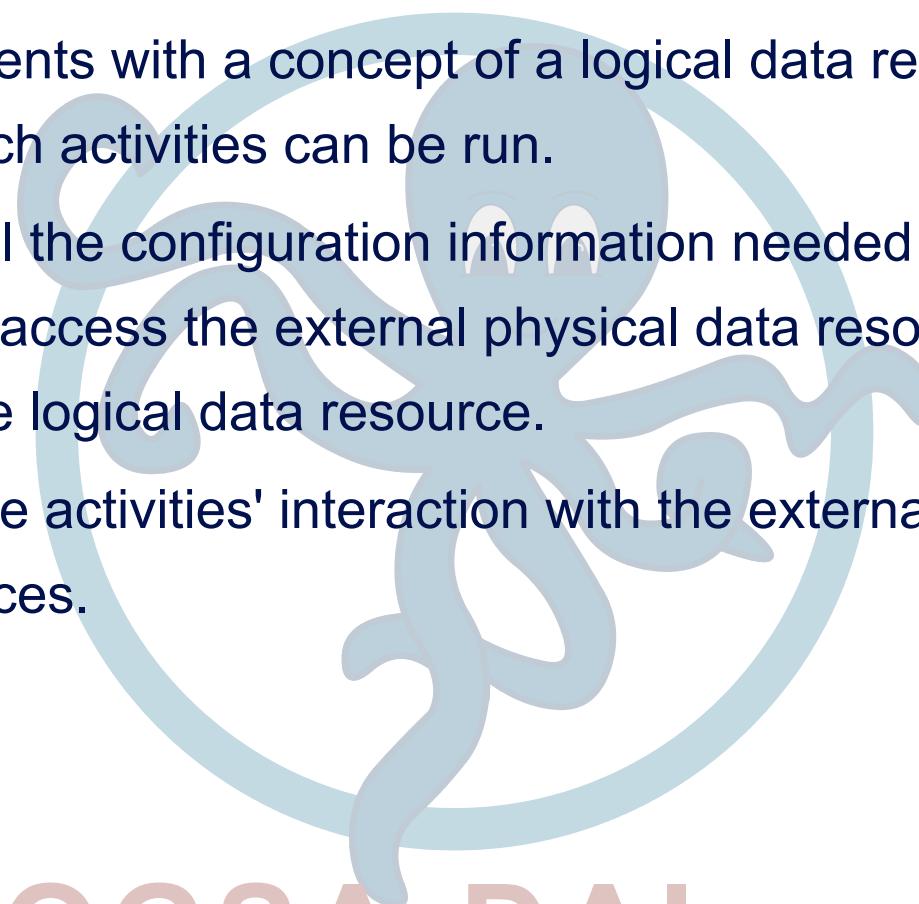
OGSA-DAI



- Avoid multiple message exchanges
- Extensible
 - Developers can add functionality
 - Could import third party trusted activities
- Allows for optimisation
 - Engine can optimise workflow

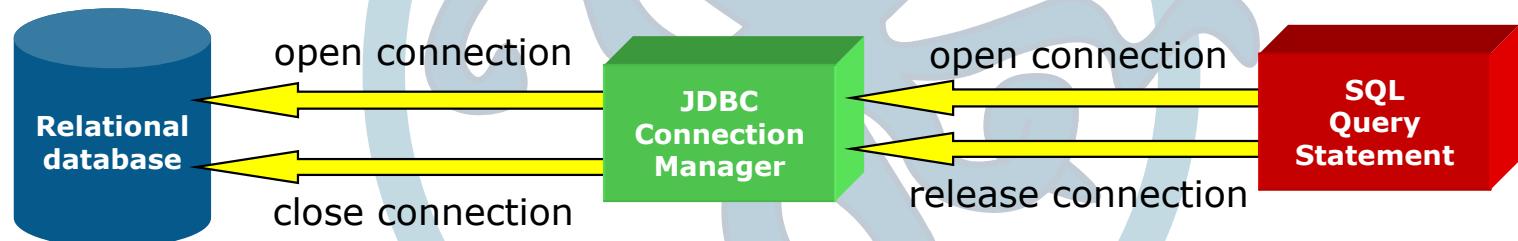
OGSA-DAI

- Provides clients with a concept of a logical data resource against which activities can be run.
- Holds all the configuration information needed by activities to access the external physical data resources that make up the logical data resource.
- Manages the activities' interaction with the external physical data resources.



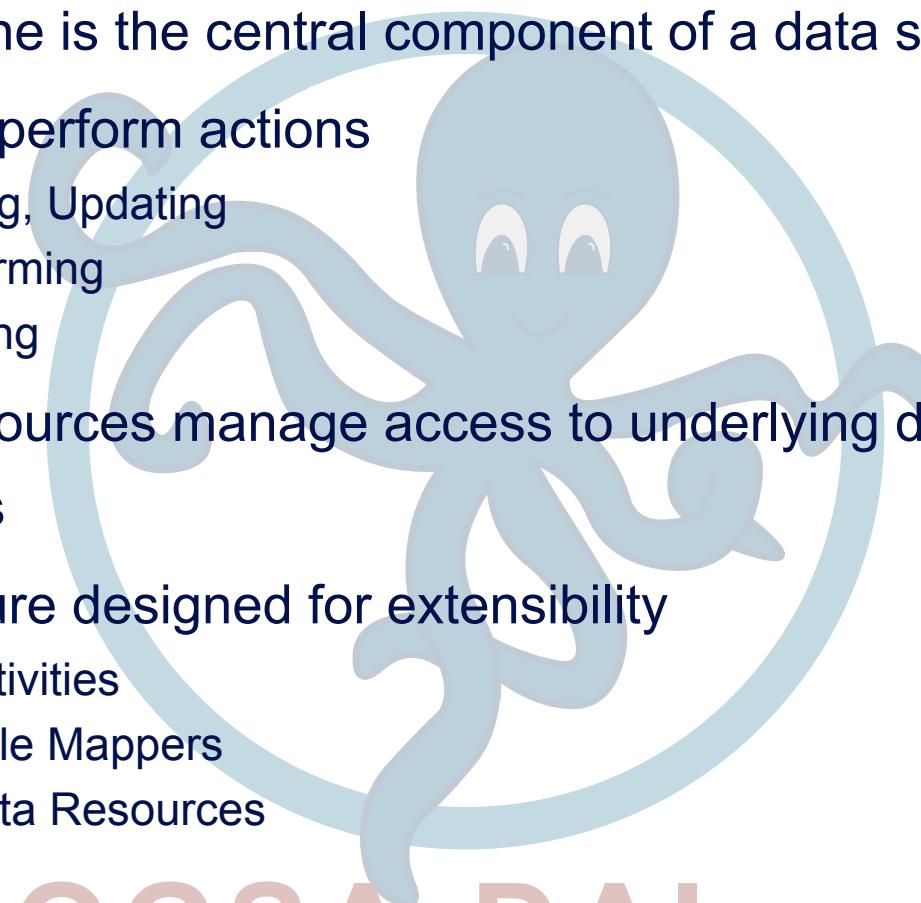
OGSA-DAI

- Provided for JDBC, XML:DB and file systems
- Can be extended for any other data source
 - e.g. web server, WebDAV server, streams



OGSA-DAI

- The Engine is the central component of a data service
- Activities perform actions
 - Querying, Updating
 - Transforming
 - Delivering
- Data Resources manage access to underlying data resources
- Architecture designed for extensibility
 - New Activities
 - New Role Mappers
 - New Data Resources



OGSA-DAI