

OGSA-DAI 3.0 – The Whats and the Whys

Mike Jackson

UK e-Science All Hands Meeting
10th–13th September 2007



Web: www.omii.ac.uk

Email: info@omii.ac.uk

Overview

- What is OGSA-DAI?
- From OGSA-DAI 2.2 to 3.0
 - Activities
 - Workflows
 - Extending the resource model
 - Extensibility
 - Implications
- What next?
- Conclusions

What is OGSA-DAI?

- An **extensible** framework
- accessed via **web services**
- that executes data-centric **workflows**
- involving **heterogeneous** data resources
- for the purposes of **data access, integration, transformation and delivery**
- within a **Grid**
- and is intended as a **toolkit** for building higher-level **application-specific** data services

OGSA-DAI 2.2

- OGSA-DAI has evolved constantly since February 2002
- Releases in 6 monthly cycles
- OGSA-DAI 2.2 was released in April 2006
- OGSA-DAI 3.0 was released in September 2007

From 2.2 to 3.0

- As the number of users grew so did the requirements
 - More effective data streaming
 - Standardisation of activity inputs and outputs
 - Use of multiple data resources in a single workflow
 - Supporting application-specific presentation layers
- Choice
 - EITHER continue with an ever more brittle and complex product
 - OR redesign and rewrite from scratch
- Stabilise the OGSA-DAI framework for developers, users (and ourselves)

Activities

- An activity is a named unit of functionality
 - A well-defined workflow unit
 - Pluggable
- Example activities include
 - Execute an SQL query
 - ZIP a batch of data
 - List the files in a directory
 - Execute an XSL transform on an XML document
 - Deliver data to an FTP server

Defining a standard activity library

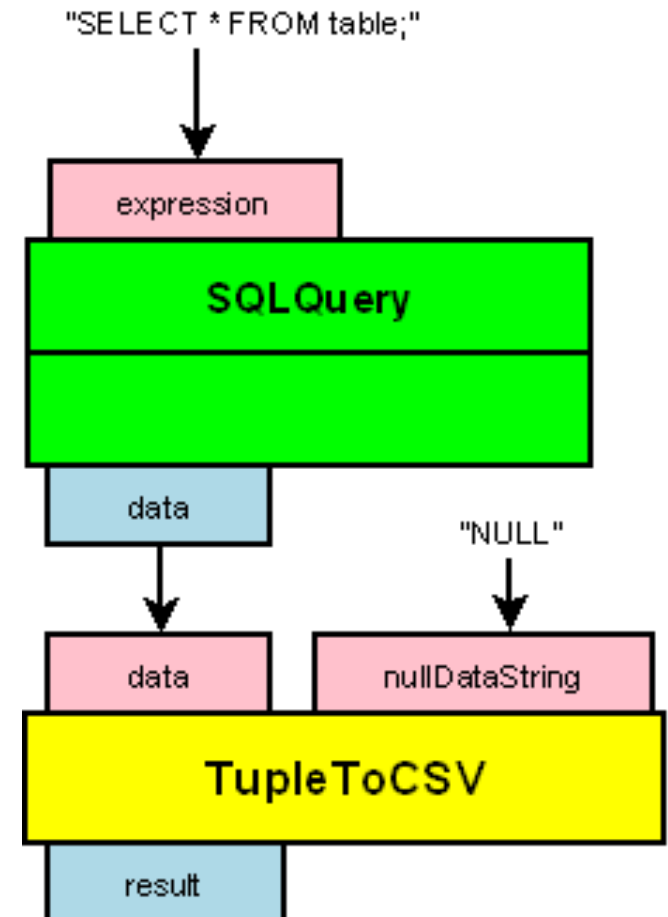
- Activities grew in an ad-hoc manner
 - Incomplete set
 - Lacked consistency
 - Did not compose easily
- Define a comprehensive and consistent standard activity set
 - Exploit understanding of data access and integration requirements
 - Karasavvas, K. Atkinson, M.P. and Hume, A.C. OGSA-DAI – Redesigned and New Activities
 - <http://www.ogsadai.org.uk/documentation/ogsadai3.0/RedesignedAndNewActivitiesVI.9.pdf>

Activity inputs, outputs and parameters in 2.2

- Inputs – output from another activity
- Outputs
- Parameters – specified once in a request
- Activity developer determined inputs and parameters
 - Client could not choose at run time
 - e.g. SQLQuery only accepted the SQL expression as a parameter
- Difficult to
 - Compose activities
 - Enforce connection semantics

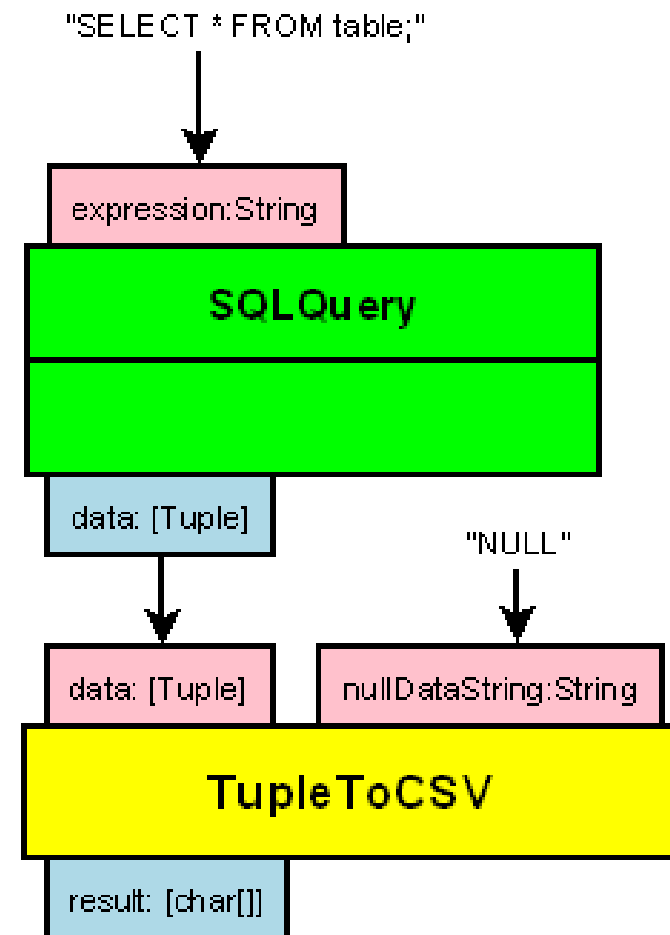
Activity inputs, outputs and parameters in 3.0

- Generic activity representation
- No distinction between inputs and parameters
- Input literal
 - Special kind of input
 - Value is provided by client
- Client chooses whether input value is provided
 - By them in a request
 - By the output of another activity



Activity input/output types and blocks

- Inputs expect blocks of specific types
- Outputs produce blocks of specific types



Activity input/output types and blocks in 2.2

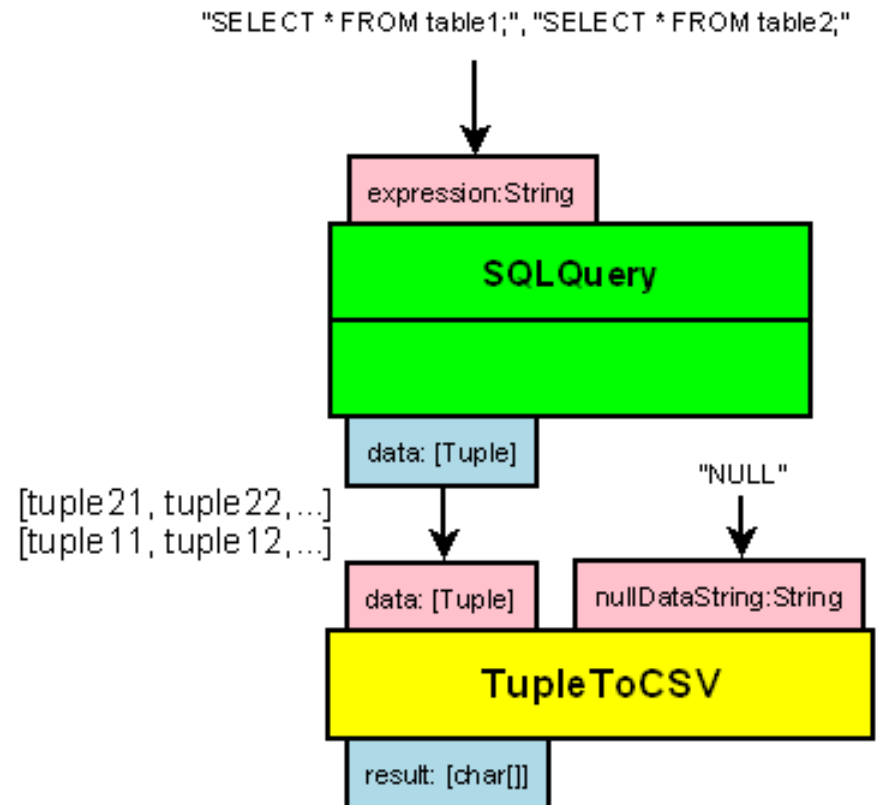
- No standard set of activity input and output block types
- No guidelines for activity developers to document or publish these
- Made construction of valid workflows problematic

Activity input/output types and blocks in 3.0

- Java's basic types
 - Object, String, Integer, Long, Double, Number, Boolean
- Binary types
 - char[], byte[], Clob, Blob
 - All binary data processing activities can handle both representations
- Tuple
 - OGSA-DAI representation of a row of relational data
 - One element per column
- MetadataWrapper
 - OGSA-DAI wrapper for any object to be treated as meta-data
 - Use application-specific meta-data within OGSA-DAI
 - Individual activities handle metadata blocks as they see fit
- Application-specific objects

Blocks and lists in OGSA-DAI 3.0

- A list groups related blocks together
 - Special blocks are used to mark the beginning and the end of a list
- For example SQLQuery can dynamically take any number of SQL query expressions as input
 - Lists allow differentiation between the results of query1 and those of query2
- Activities define the granularity of their inputs and outputs



Activity API in 2.2

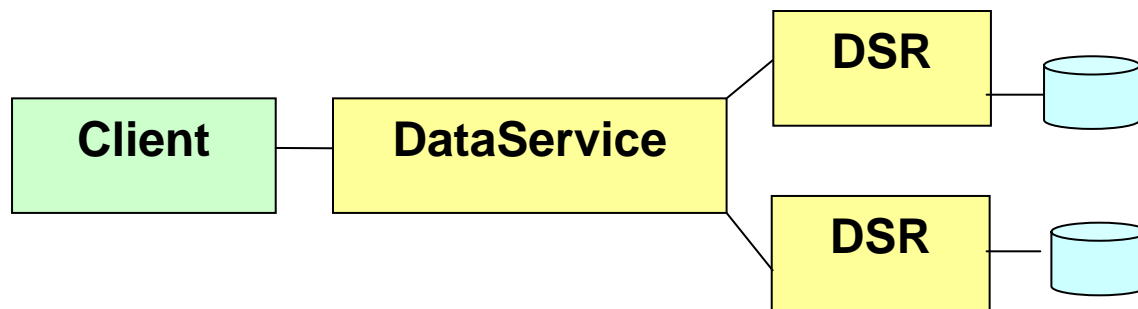
- Define an XML Schema specifying activity input, parameter and output names
- Write an activity in Java
 - Conform to OGSA-DAI activity API
 - Parse XML fragment containing activity input names, parameter values and output names
 - Get useful objects from an overloaded and monolithic request object
 - Data resource
 - Security context
 - Data resource factory
 - etc

Activity API in 3.0

- Activity input/output standardisation yields a simpler API
- No need to define XML Schema
- Write an activity in Java
 - Conform to OGSA-DAI activity API
 - No need to parse XML fragment
 - Activity framework provides input and output names as Java objects
 - Implement extension interfaces
 - Declares what the activity expects from the activity framework
 - Data resource
 - Security context
 - Data resource factory
 - Configuration values
 - Session
 - etc

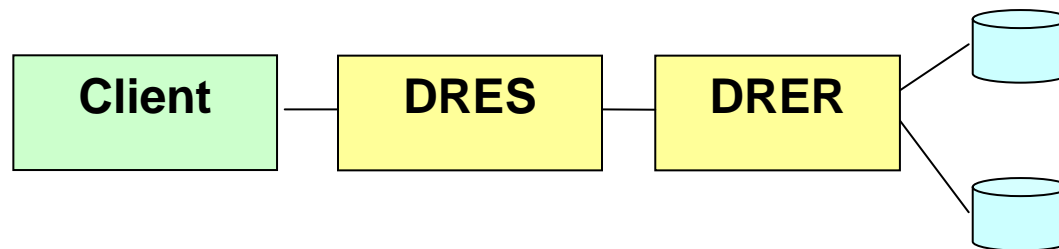
Executing workflows in 2.2

- Data service resource (DSR)
 - Exposed by data service
 - Executes workflows
 - Associated with 0..1 data resources
 - => One workflow can use 0..1 data resources

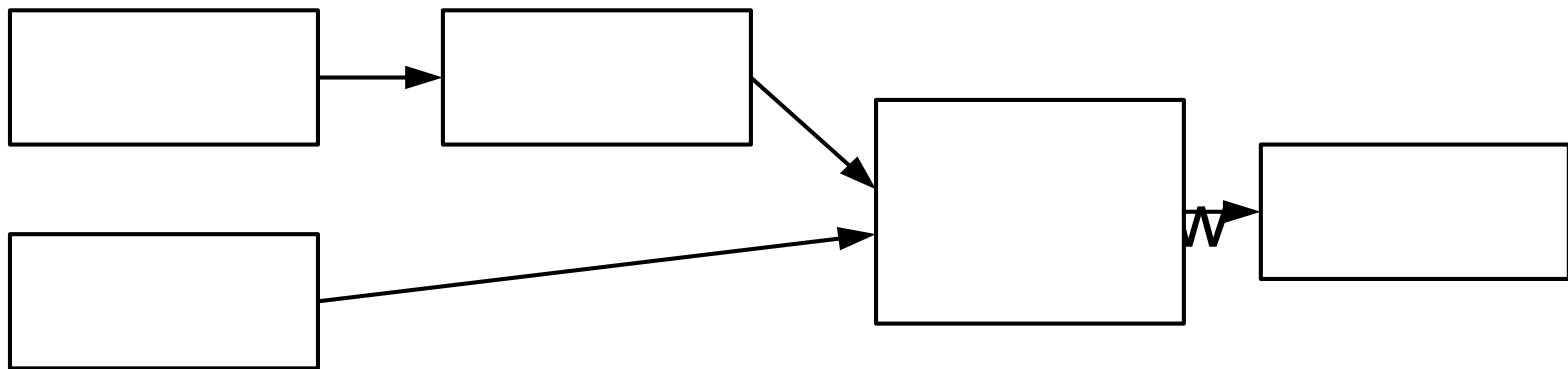
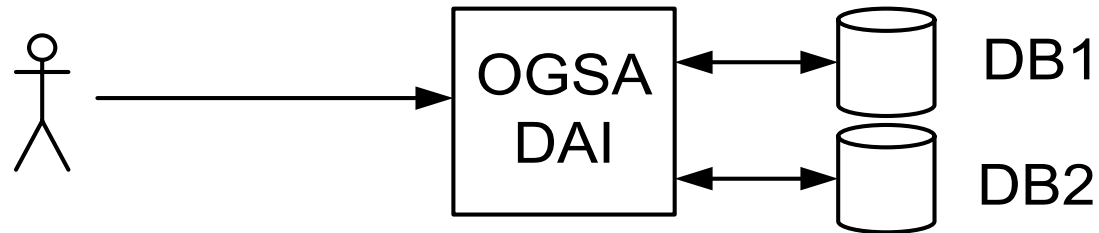


Executing workflows in 3.0

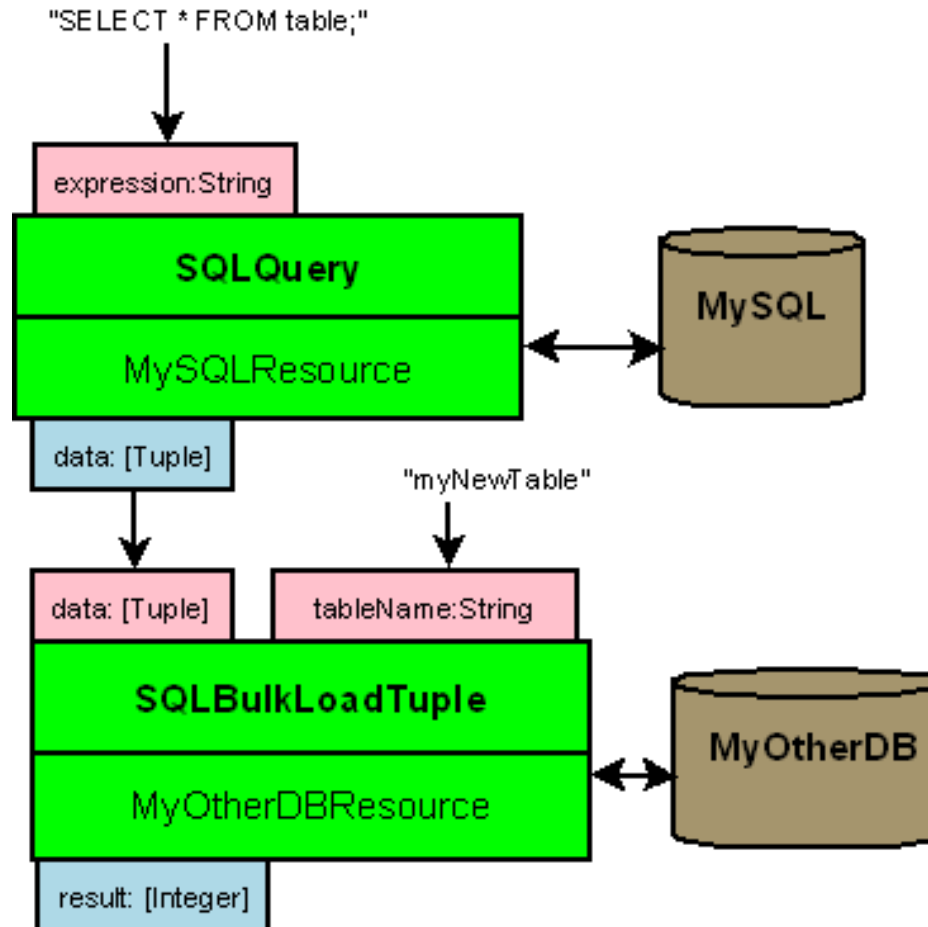
- Data request execution resource (DRER)
 - Exposed by data request execution service (DRES)
 - Executes workflows
 - Associated with 0..N data resources
 - => One workflow can use 0..N data resources



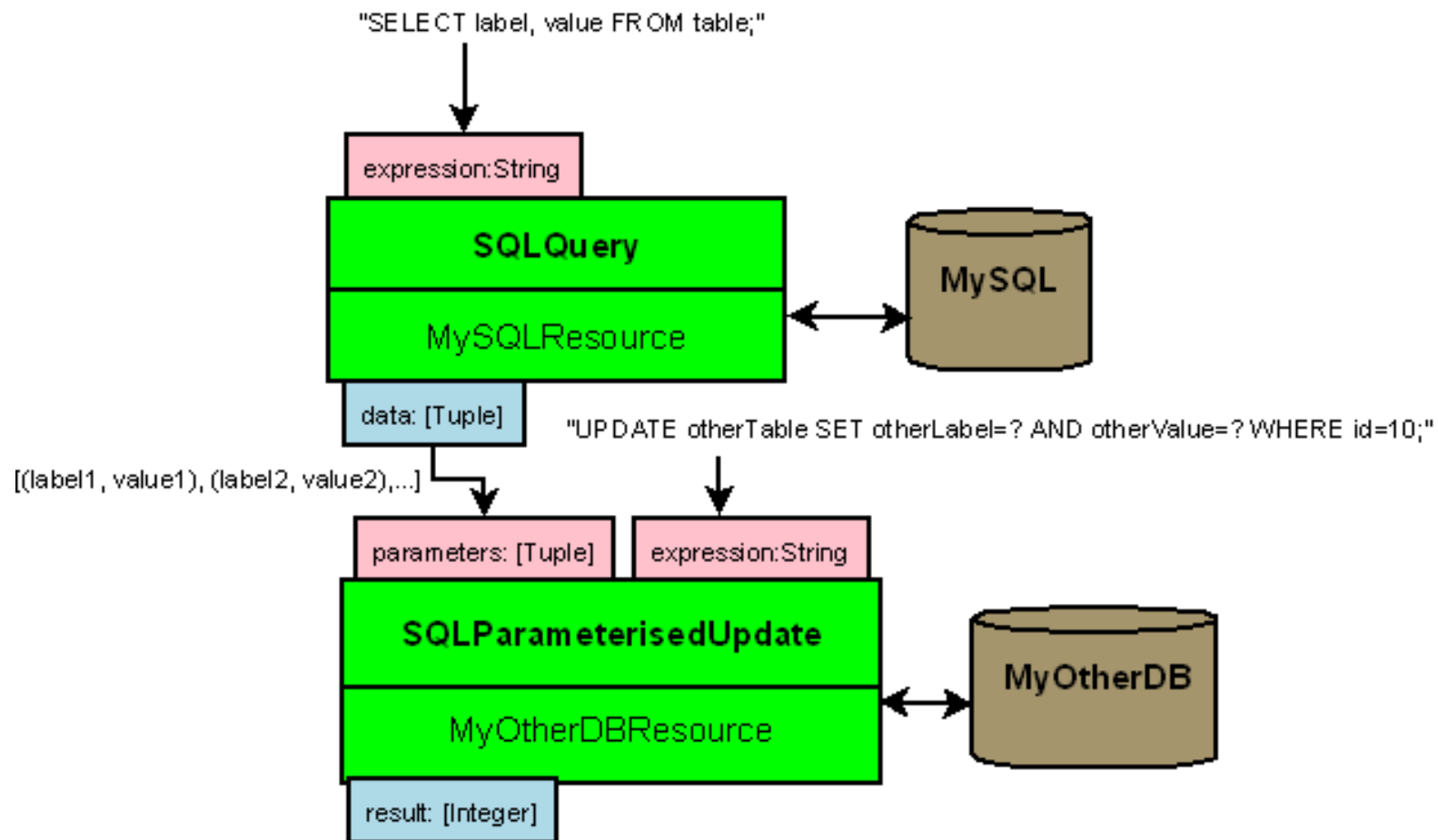
Multiple data resources in a workflow



Multiple data resources in a workflow



Multiple data resources in a workflow



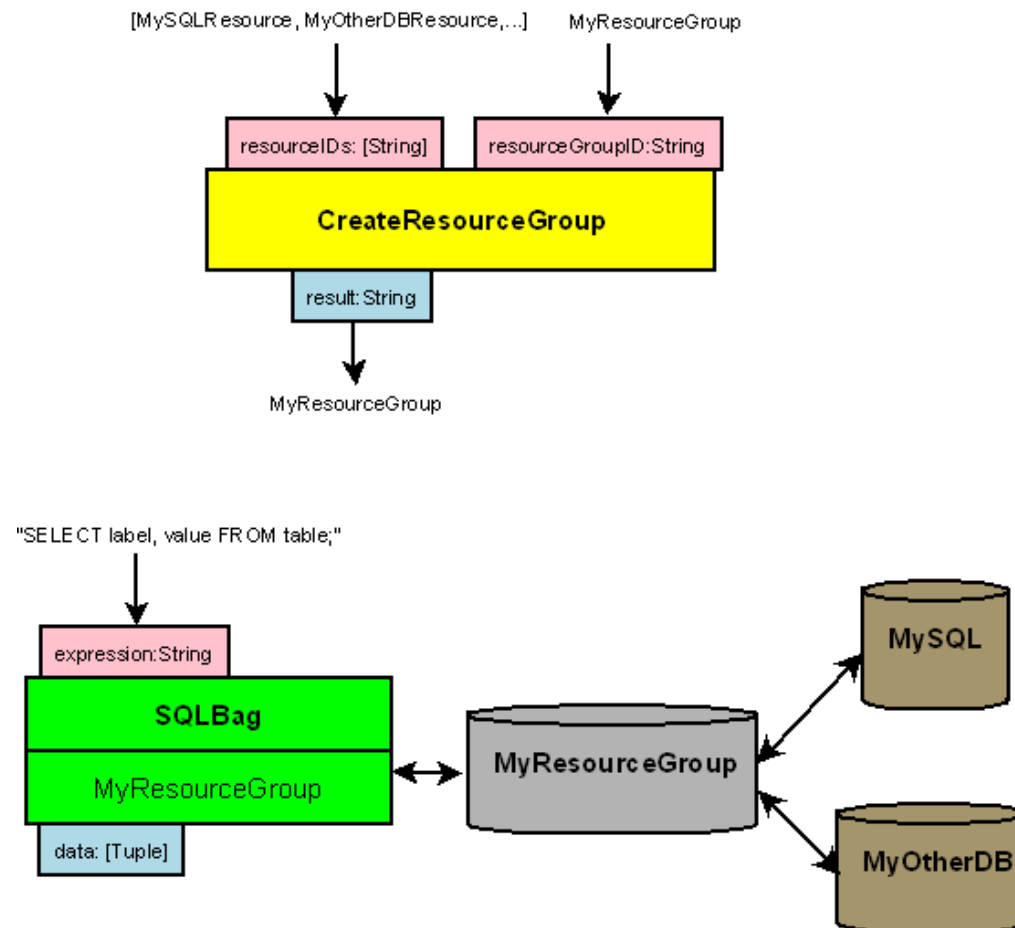
One approach to federation in 2.2

- Multi-resources
- Opaque federation of N data resources with the same database schema
- Set up by server administrator

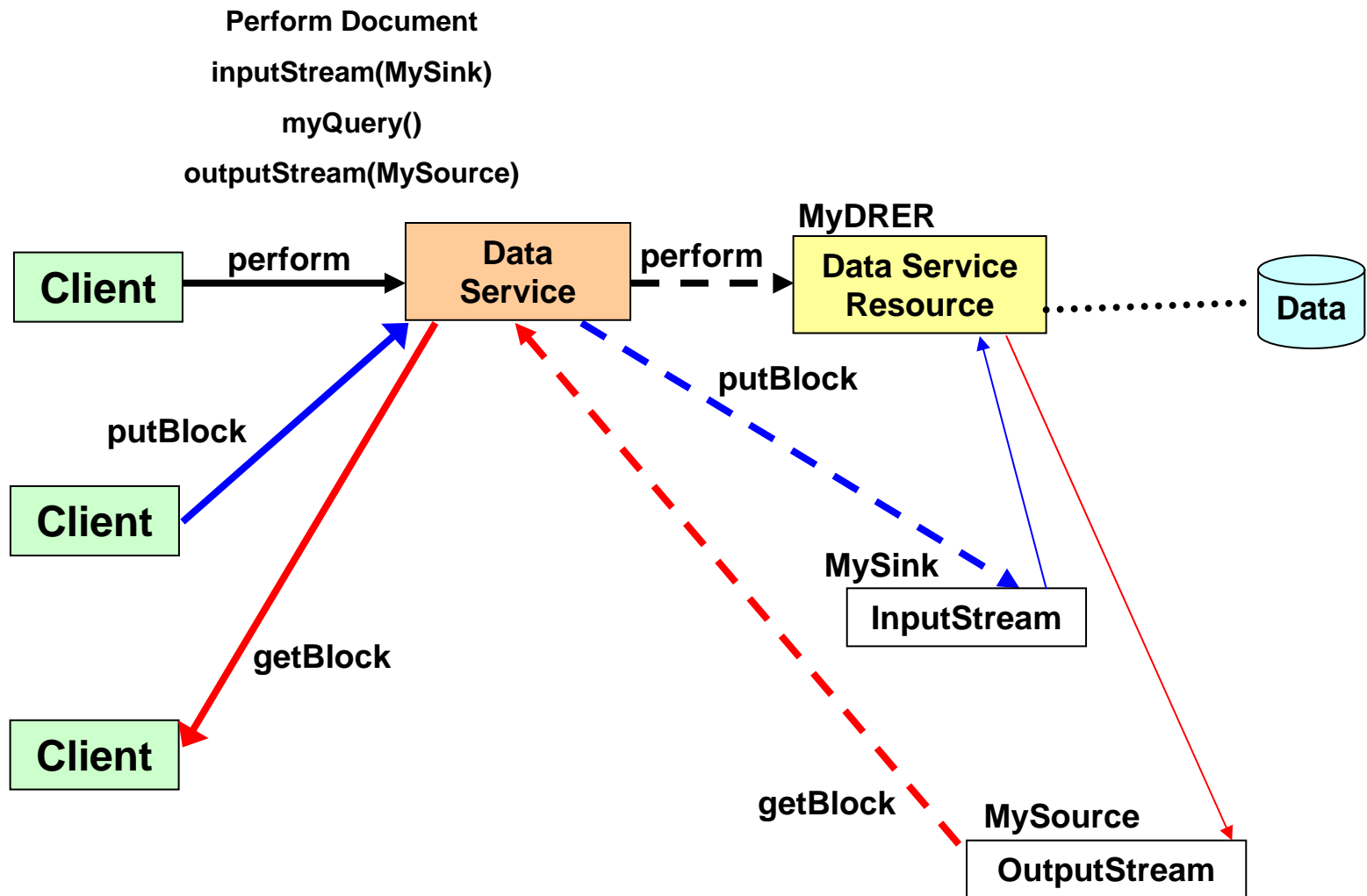
One approach to federation in 3.0

- Resource groups
- Group any set of data resources
- Set up by server administrator
- AND/OR
- Set up by client using `CreateResourceGroup` activity
- Opaque or transparent federation

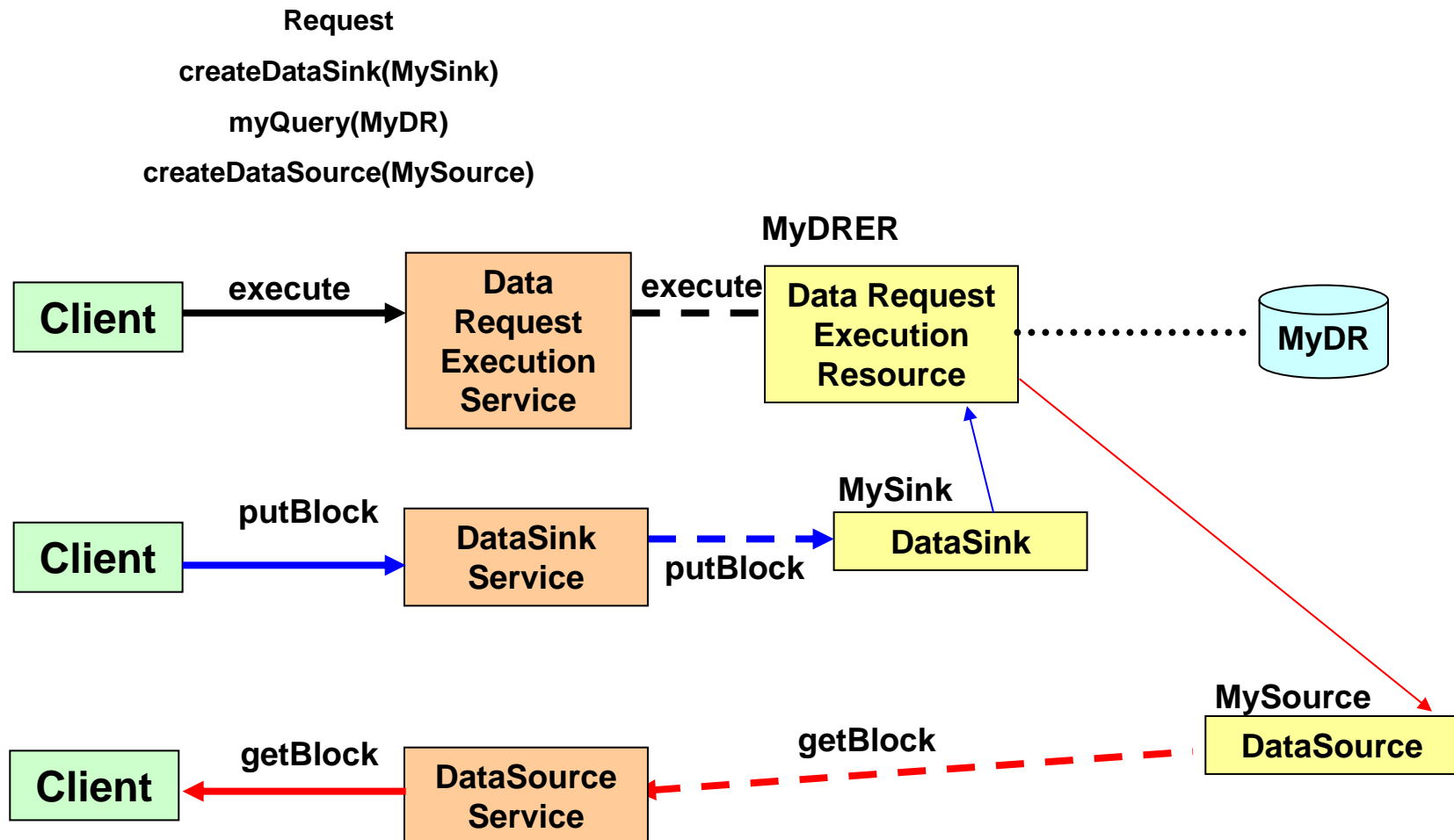
Federation of resources via resource groups



Asynchronous data delivery in 2.2



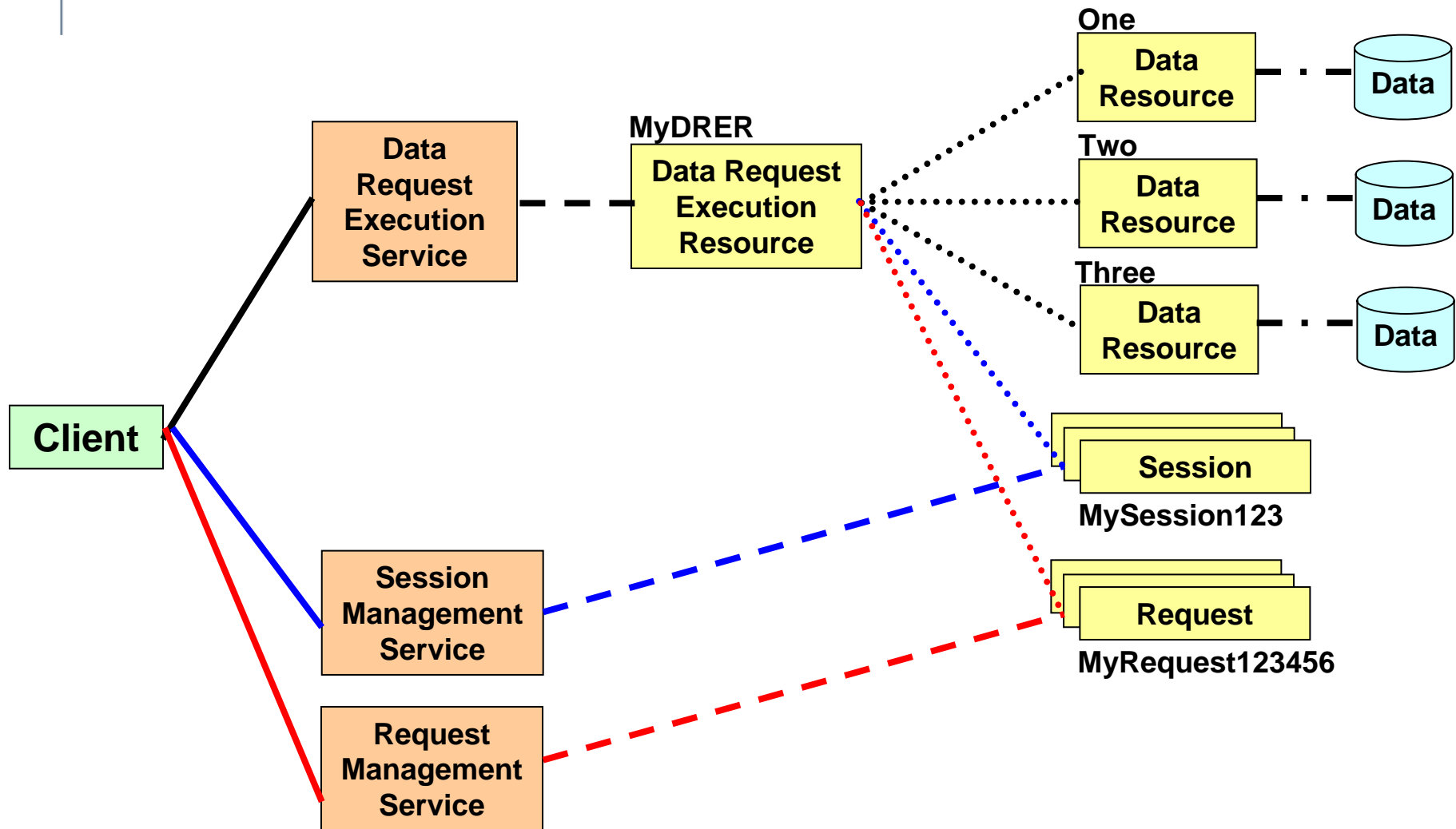
Asynchronous data delivery in 3.0



Requests and sessions

- Request
 - Workflow submitted from client to server
 - Provide information on status of asynchronous requests
- Session
 - Container for state which can be accessed across multiple workflows
- OGSA-DAI 2.2
 - As for data sources and data sinks – special OGSA-DAI objects
 - An overloaded OGSA-DAI data service

Requests and sessions in 3.0



Web services and workflows in 2.2

- OGSA-DAI 2.2 data service
 - AnyType Perform(AnyType)
 - Data service did not publish what “AnyType” should be
 - i.e. OGSA-DAI perform document schema
- Perform document schema was not known until OGSA-DAI started
 - Due to activity-specific schema
- Problems for workflow engines e.g. Taverna

Web services and workflows in 3.0

- OGSA-DAI 3.0 data request execution service
 - RequestStatus execute(Request)
 - Generic request/workflow and request status schema
 - Benefit of a generic activity schema
- Can now run client auto-generation tools
 - e.g. Axis WSDL2Java
 - Get a set of Java objects which can be used to construct workflows and represent request status
- Facilitates easier integration into workflow engines

Extensibility – 2.2

- Activities
- Data resources
- Role mappers
- Authorizers
 - Resource
 - Activity
- Services
 - With difficulty – some core functionality in presentation layer

Extensibility – 3.0

- Activities
- Data resources
- Login providers
- Authorizers
 - Resource
 - Activity
- Security contexts
- Workflow execution monitor
- Services
- Persistence/configuration layer
- Resource and activity managers

Implications for OGSA-DAI users

- Developers of clients
 - Changes to client toolkit APIs – activities, workflows, resources, services
 - Basic model is unchanged – build a workflow and submit to proxy
 - Proxy handles the low level client-server communications
- Developers of application-specific activities
 - Major changes to activity APIs
 - Activity input and output standardisation
 - Support for activity iteration
 - Extension interfaces
 - Configuration – specification of activities to server and association of activities with resources
- Developers of application-specific services
 - Changes to APIs to the OGSA-DAI core
 - Boundary between services and the core OGSA-DAI functionality redrawn
 - Only application-specific code need reside in the service layer

What now?

- Align activities with 2.2 and the activity specification
- Performance and reliability
- Visualisation tools
- New functionality e.g. transactions
- WS-DAI
- OGSA-DQP
- Component interaction
 - Taverna, ActiveBPEL, VOMS, GRIMOIRES,.....
- External project-driven development
 - Data integration
 - Security

Conclusions

- OGSA-DAI 3.0 includes a number of major changes
 - Simplification and unification of activities
 - Easier workflow composition
 - Targeting multiple resources within a single workflow
 - Modular services and resources
 - Increased modularity and extensibility

Further information

- Come and chat to us at the booth
- Grab anyone wearing our T-shirts
- OGSA-DAI 3.0 tutorial – Thursday 09:00
- OGSA-DAI
 - WWW site – <http://www.ogsadai.org.uk>
 - Info – info@ogsadai.org.uk
 - Users e-mail list – users@ogsadai.org.uk
- OMII-UK
 - WWW site – <http://www.omii.ac.uk>
 - Info – info@omii.ac.uk