OGSA-DAI for GT4 Developers Function, Platforms and Use

The OGSA-DAI Team

info@ogsadai.org.uk

Neil Chue Hong, Ally Hume, Mike Jackson

Outline

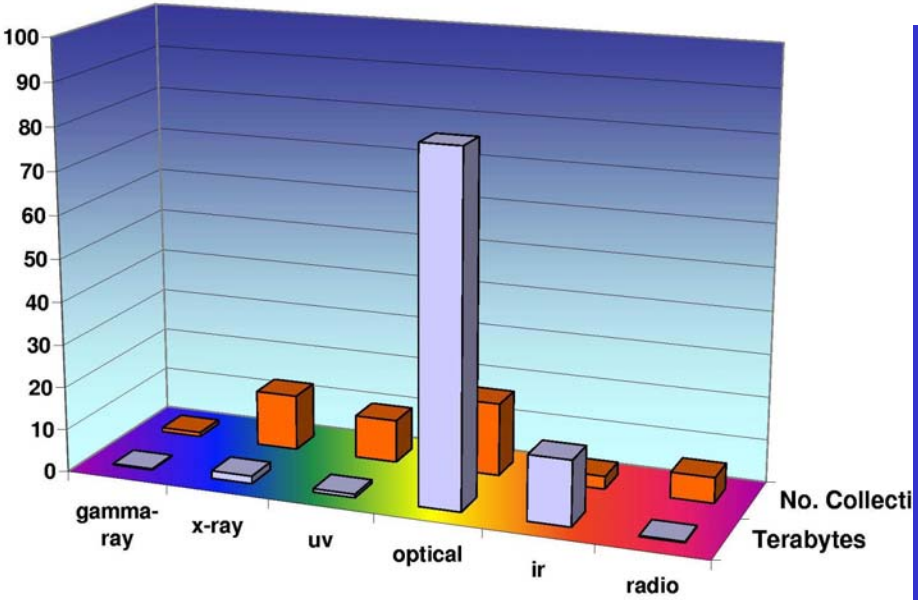
- Introduction to OGSA-DAI
- OGSA-DAI Architecture
  - Discussion of OGSI/WS-I/WS-RF issues
- Using the Client Toolkit
- Writing Activities
- Wrap Up

Introduction to OGSA-DAI

Motivation

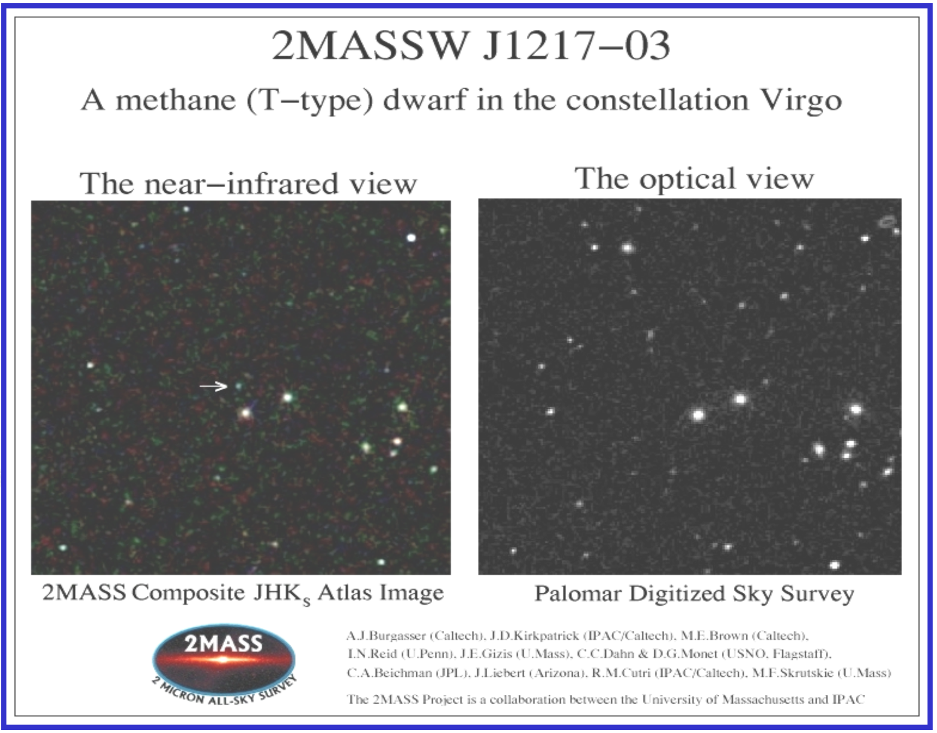
- Entering an age of data
  - Data Explosion
    - CERN: LHC will generate 1GB/s = 10PB/y
    - VLBA (NRAO) generates 1GB/s today
    - Pixar generate 100 TB/Movie
  - Storage getting cheaper
- Data stored in many different ways
  - Data resources
    - Relational databases
    - XML databases
    - Flat files
- Need ways to facilitate
  - Data discovery
  - Data access
  - Data integration
- Empower e-Business and e-Science
  - The Grid is a vehicle for achieving this

Composing Observations in Astronomy



No. & sizes of data sets as of mid-2002, grouped by wavelength

- 12 waveband coverage of large areas of the sky
- Total about 200 TB data
- Doubling every 12 months
- Largest catalogues near 1B objects



2MASSW J1217−03

A methane (T−type) dwarf in the constellation Virgo

The near−infrared view

The optical view

2MASS Composite JHK$_s$ Atlas Image

Palomar Digitized Sky Survey

A.J.Burgasser (Caltech), J.D.Kirkpatrick (IPAC/Caltech), M.E.Brown (Caltech), I.N.Reid (U.Penn), J.E.Gizis (U.Mass), C.C.Dahn & D.G.Monet (USNO, Flagstaff), C.A.Beichman (JPL), J.Liebert (Arizona), R.M.Cutri (IPAC/Caltech), M.F.Skrutskie (U.Mass)

The 2MASS Project is a collaboration between the University of Massachusetts and IPAC

THE NEW YORK TIMES NATIONAL TUESDAY, JUNE 1, 1999

Astronomers Detect New Category of Elusive 'Brown Dwarfs'

By JOHN NOBLE WILFORD

CHICAGO, May 31 — Ambitious

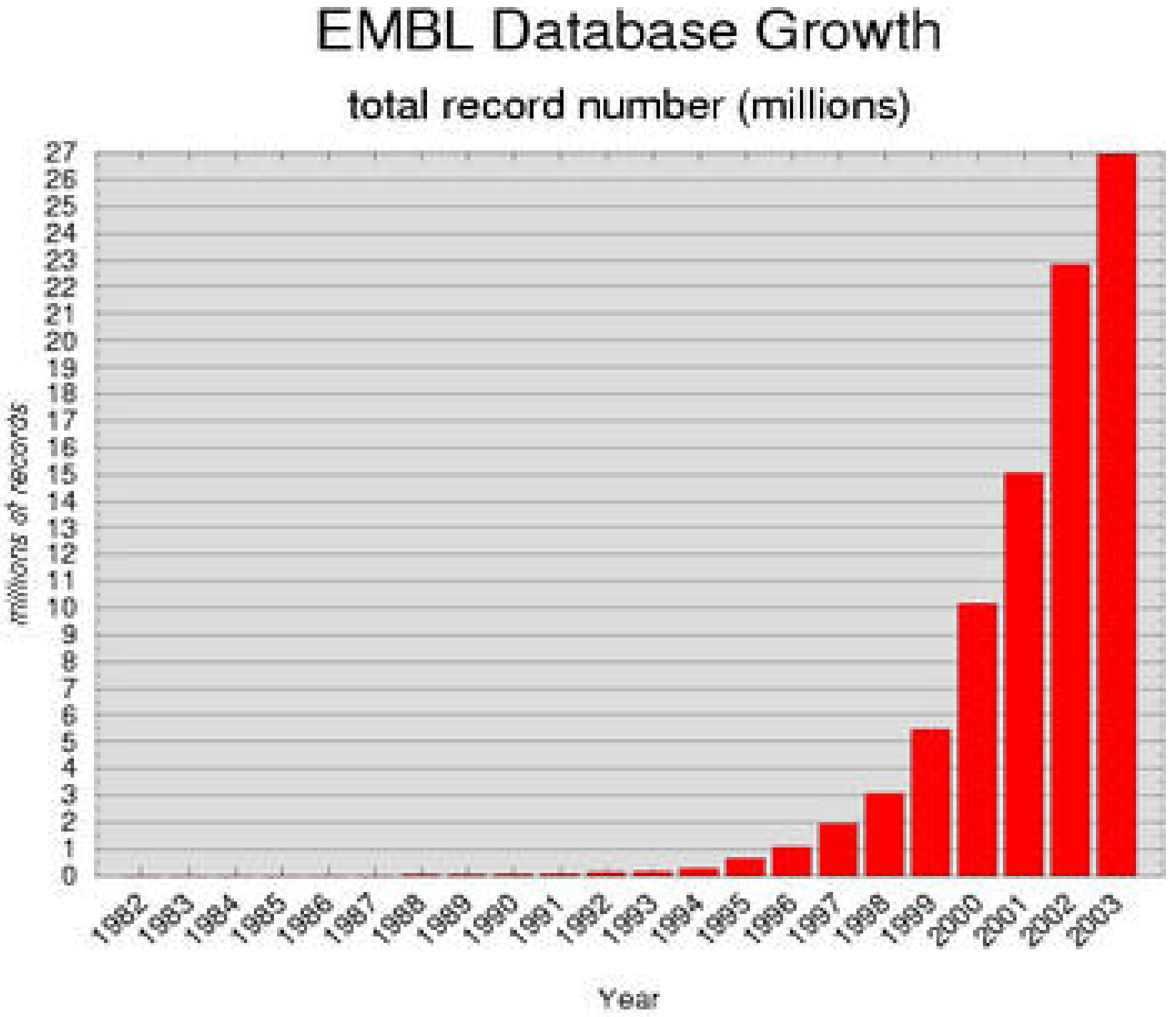
Apache Point, N. M. Dr. Michael Strauss and a graduate student,

was a brown dwarf, but was not associated with a star companion.

have been possible in hotter, younger objects. An estimate of their mass

Data and images courtesy Alex Szalay, John Hopkins

Database Growth

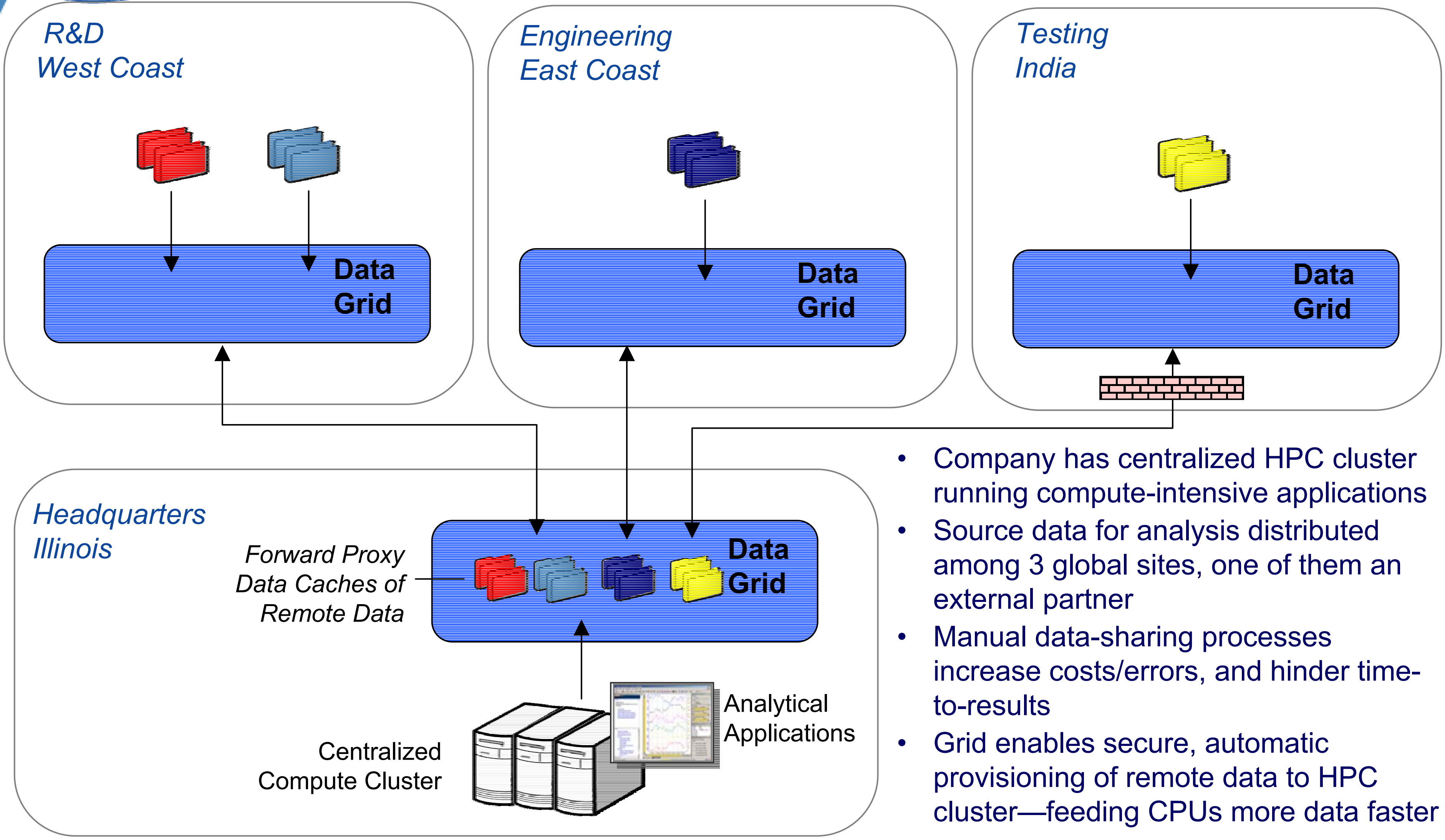


EMBL Database Growth

total record number (millions)

MRC HUMAN GENETICS UNIT • EDINBURGH •

Providing Data to Cluster-Based Analytical Application

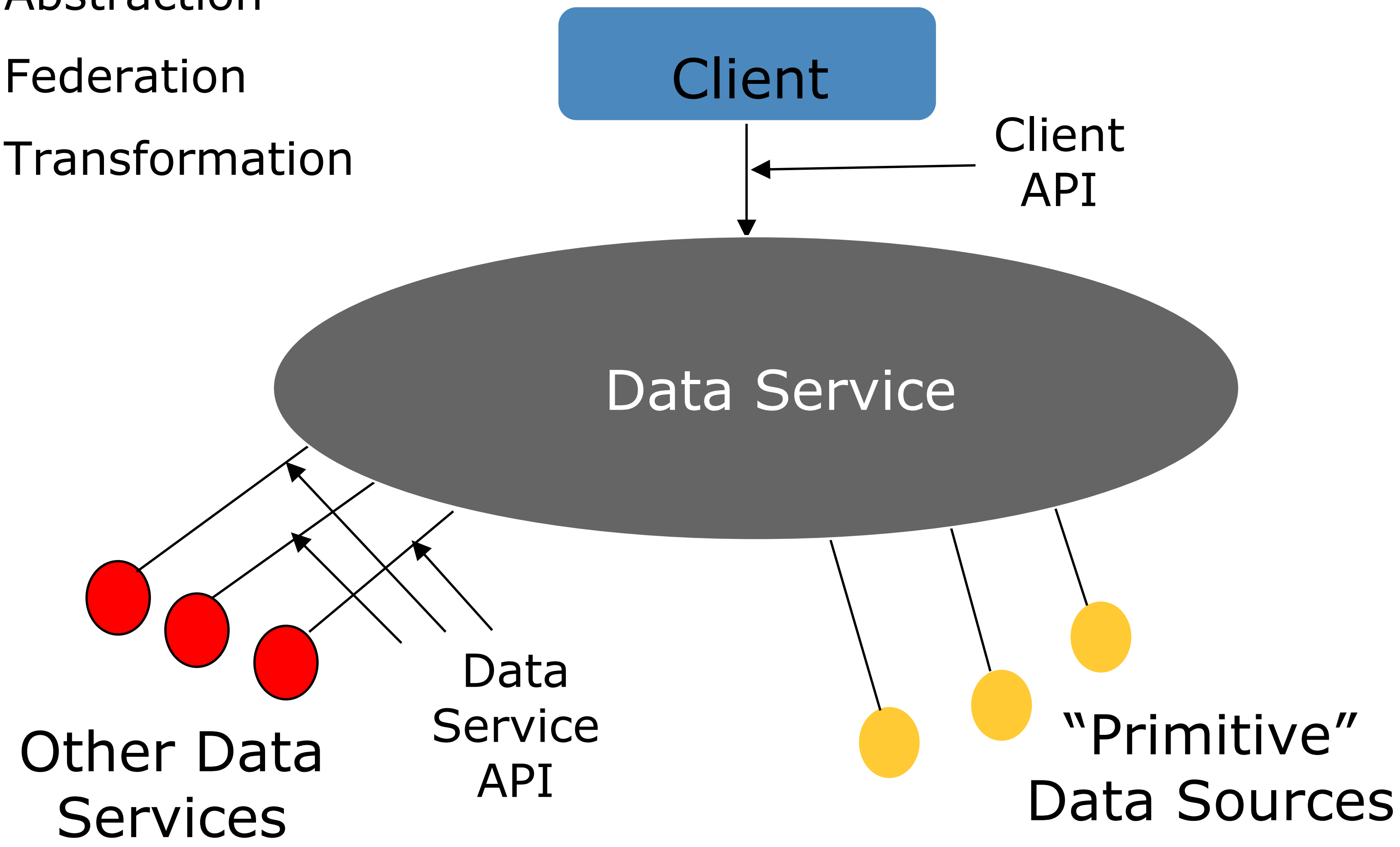


- Company has centralized HPC cluster running compute-intensive applications
- Source data for analysis distributed among 3 global sites, one of them an external partner
- Manual data-sharing processes increase costs/errors, and hinder time-to-results
- Grid enables secure, automatic provisioning of remote data to HPC cluster—feeding CPUs more data faster

Thanks to Dave Berry, Andrew Grimshaw – OGSA-WG

Data Virtualisation

- Abstraction
- Federation
- Transformation

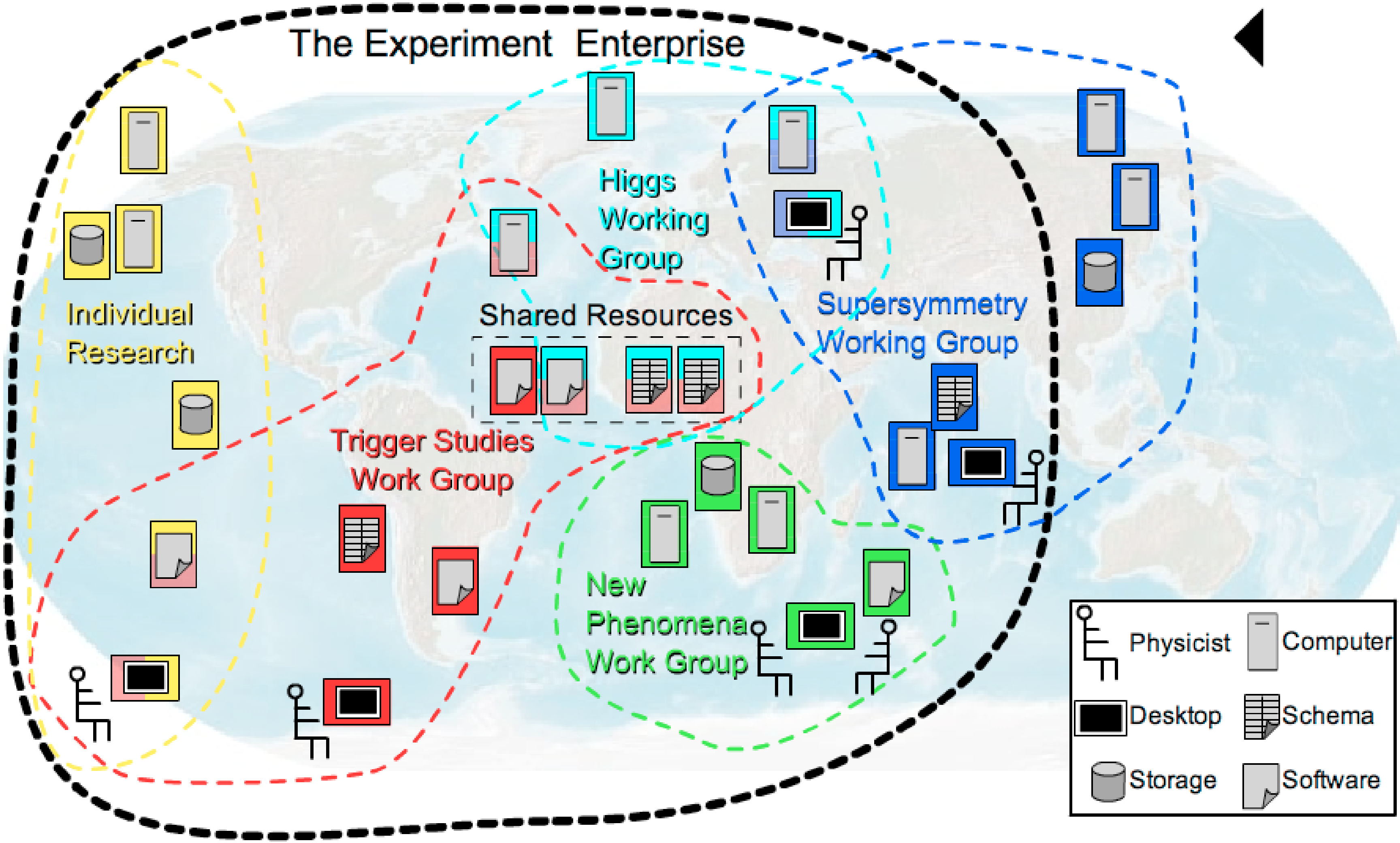


Thanks to Dave Berry, Andrew Grimshaw – OGSA-WG

Share and share alike!

- Many challenges:
  - Scalability, performance, heterogeneity, ownership, economics
  - Common schema, data description and semantics, data formats, process and procedure, provenance
- Can be solved only through collaboration and the sharing of:
  - Ideas
  - Efforts
  - Resources
- Perhaps most importantly: sharing of data
  - Beware of data huggers!
- Emerging Open Grid Infrastructures will
  - Allow global collaboration
  - Change the way that we can work

Emergence of Virtual Organisations



Data Requirements

- What do we need for effective sharing of data?
  - Structured, organised, annotated & curated data
  - Computable data models
  - Visualisation of data
  - Data provenance
  - Shared distributed systems
  - Networked workplaces, instruments, data sources
  - Metadata, ontologies, standards
  - Authentication, authorisation, accounting, policies

Terabyte → Petabyte

	Terabyte	Petabyte
RAM time to move	15 minutes	2 months
1GB WAN move time	10 hours ($1000)	14 months ($1 million)
Disk cost	7 disks = $5000 (SCSI)	6800 Disks + 490 units + 32 racks = $7 million
Disk power	100 Watts	100 Kilowatts
Disk weight	5.6 Kg	33 Tonnes
Disk footprint	Inside machine	60 m2

Approximately Correct in May 2003 *Distributed Computing Economics*
Jim Gray, Microsoft Research, MSR-TR-2003-24

Mohammed & Mountains

- Petabytes of Data cannot be moved
  - It stays where it is produced or curated
    - Hospitals, observatories, European Bioinformatics Institute
  - A few caches and a small proportion cached
- Distributed collaborating communities
  - Expertise in curation, simulation & analysis
- Diverse data collections
  - Discovery depends on insights
  - Unpredictable or unexpected use of data

Meta-data: describing data

- Choosing data sources
  - How do you find them?
  - How are they described and advertised?
  - Is the equivalent of Google possible?
- Meta-data is required describing:
  - Structure of data
  - Types of data
  - Operations supported/available
  - Access requirements
  - Quality of service?
- No established standards for heterogeneous data sources

Cultural Challenges

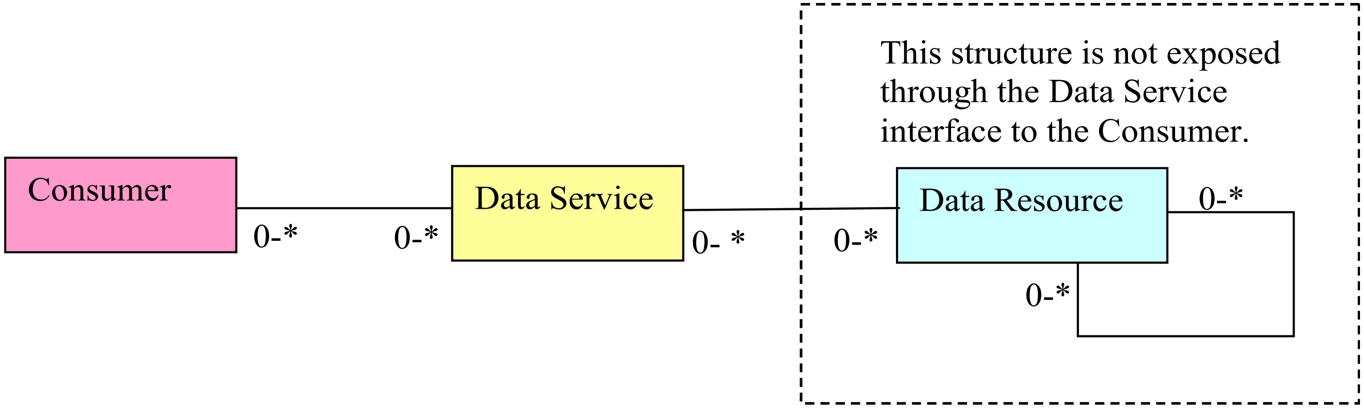
- Changing the way we work?
- Publication and sharing of results
  - Increased volume and diversity = increased opportunity?
  - Allows independent validation of methods and derivatives
  - Responsibility, ownership, credit, citation
- Many distributed data resources
  - Data collected from observation, simulation & experiment
  - Independently owned & managed
    - No common goals or design
    - Work hard for agreements on foundation types and ontologies
    - Autonomous decisions change data, structure, policy, etc
- Diversity
  - No “one size fits all” solutions will work

Security Challenges: Medical Imaging Data

- Diagnosing based on sensitive patient data
  - Users: a (group of) doctor(s)
  - Retrieve an image, run algorithm, examine result and write diagnosis, maybe re-run another algorithm.
- Secure Data Retrieval
  - Patient data is sensitive, needs to be stored anonymously at all times
  - Site admins are not trustworthy – strip or encrypt patient data from image
  - Replication of data not always allowed
- High security needs
  - Strong authorization
  - Fine-grained access control mechanisms
  - Leaking patient information results in prosecution.

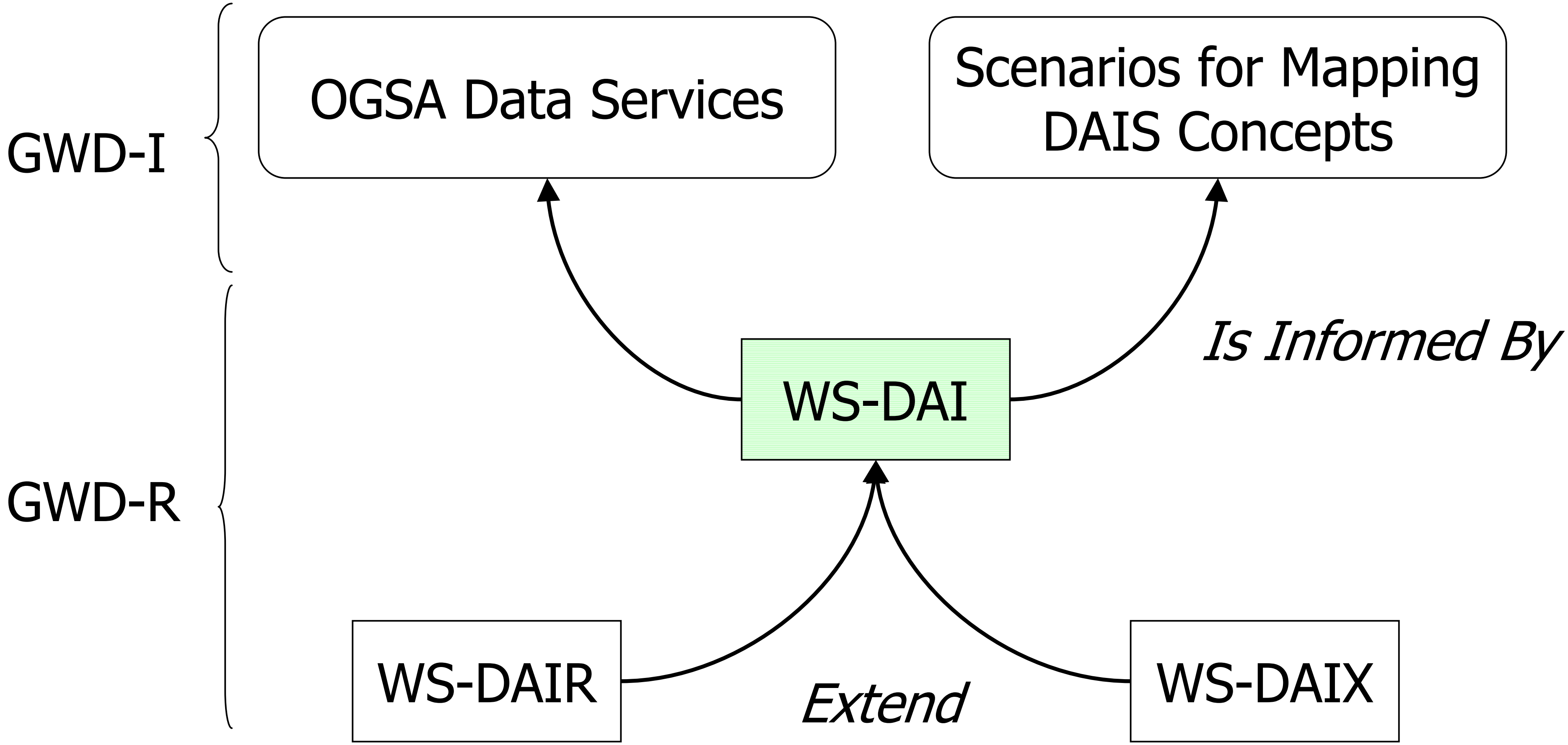
Thanks to Dave Berry, Andrew Grimshaw – OGSA-WG

DAIS View Of Data Services Model

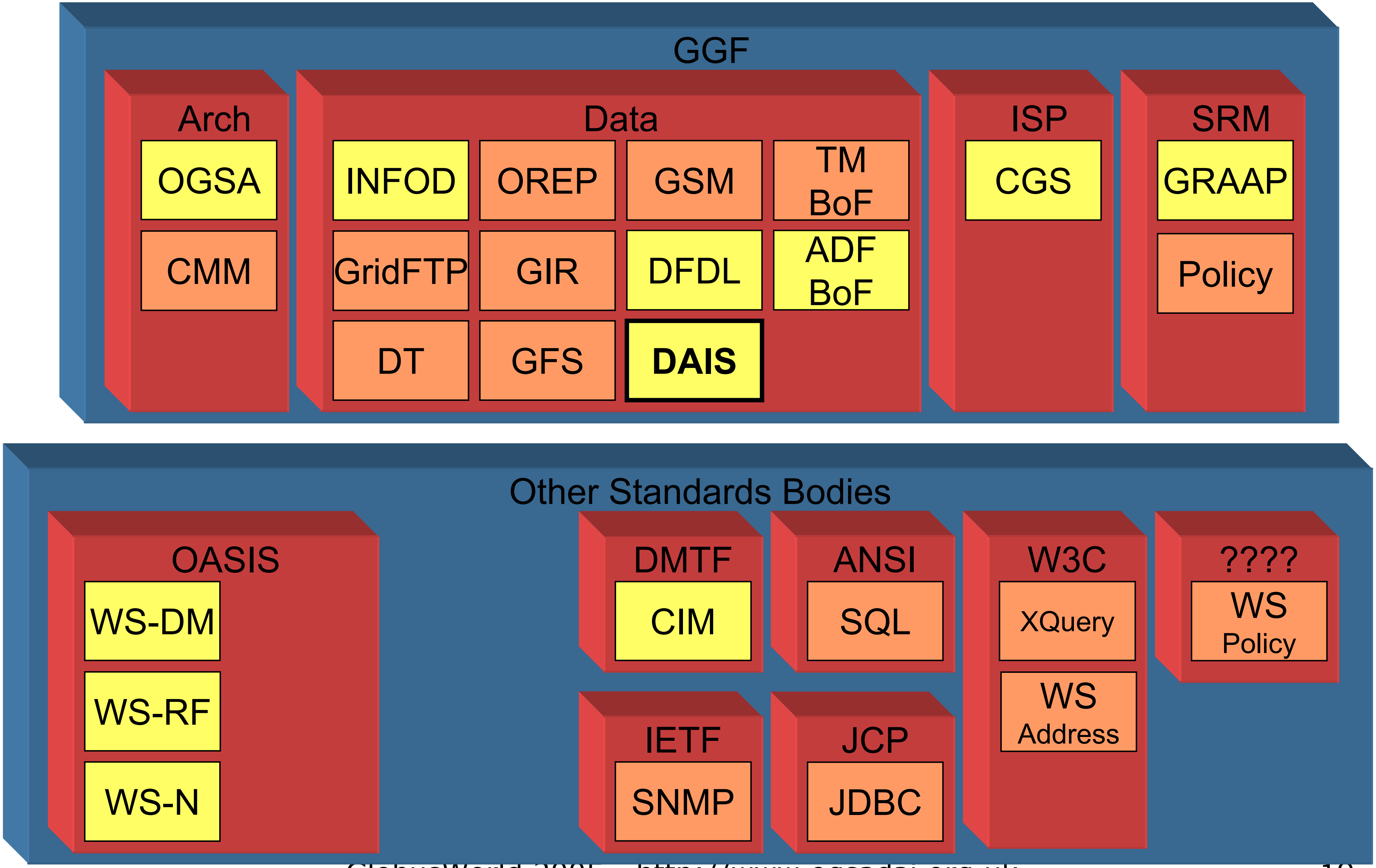

This structure is not exposed through the Data Service interface to the Consumer.
Consumer
0-*
0-*
Data Service
0- *
0-*
Data Resource
0-*
0-*

- A Data Service presents a Consumer with an interface to a Data Resource.
- A Data Resource can have arbitrary complexity, for example, a file on an NFS mounted file system or a federation of relational databases.
- A Consumer is not typically exposed to this complexity and operates within the bounds and semantics of the interface provided by the Data Service

DAIS Specification Landscape



DAIS and Other Standards/Specs

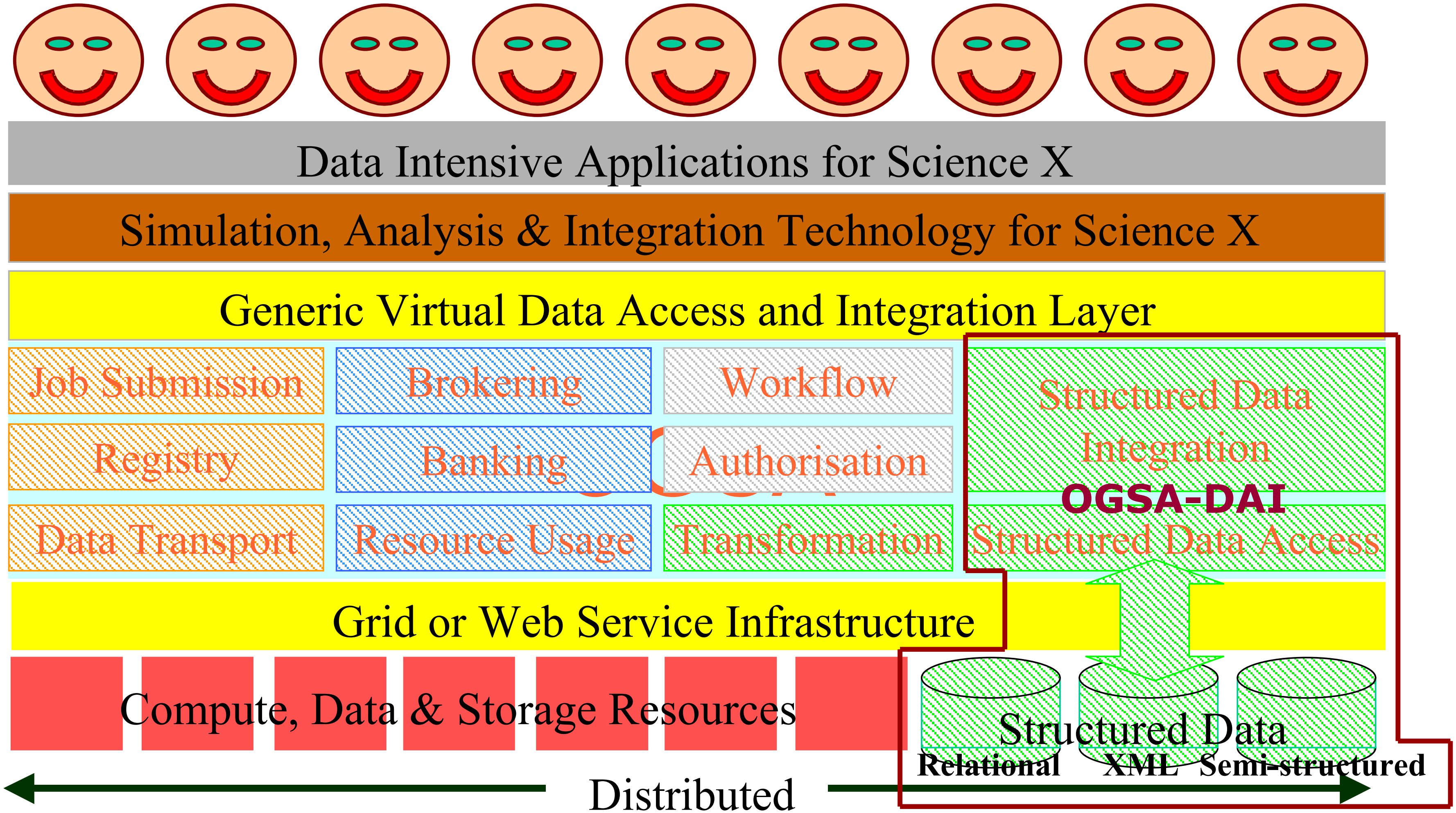


Goals for OGSA-DAI

- Aim to deliver application mechanisms that:
  - Meet the data requirements of Grid applications
    - Functionality, performance and reliability
    - Reduce development cost of data centric Grid applications
    - Provide consistent interfaces to data resources
  - Acceptable and supportable by database providers
    - Trustable, imposed demand is acceptable, etc.
    - Provide a standard framework that satisfies standard requirements
- A base for developing higher-level services
  - Data federation
  - Distributed query processing
  - Data mining
  - Data visualisation

Infrastructure Architecture

Data Intensive X Scientists



Data Intensive Applications for Science X

Simulation, Analysis & Integration Technology for Science X

Generic Virtual Data Access and Integration Layer

Job Submission

Brokering

Workflow

Structured Data Integration

Registry

Banking

Authorisation

OGSA-DAI

Data Transport

Resource Usage

Transformation

Structured Data Access

Grid or Web Service Infrastructure

Compute, Data & Storage Resources

Structured Data

Relational XML Semi-structured

Distributed

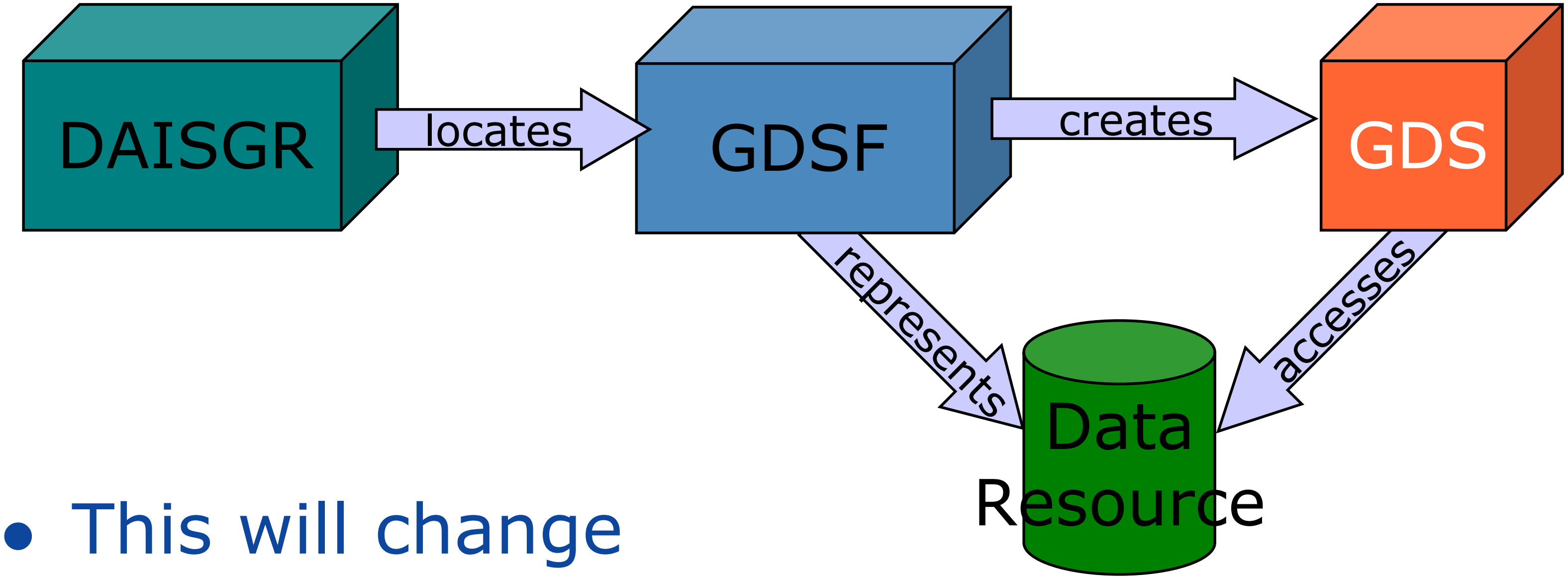
Virtual Integration Architecture

Core features

- An extensible framework for building applications
  - Supports relational, xml and some files
    - MySQL, Oracle, DB2, SQL Server, Postgres, XIndice, CSV, EMBL
  - Supports various delivery options
    - SOAP, FTP, GridFTP, HTTP, files, email, inter-service
  - Supports various transforms
    - XSLT, ZIP, GZip
  - Supports message level security using X509 certificates
  - Client Toolkit library for application developers
  - Comprehensive documentation and tutorials
- Third production release (R5) on 3 December 2004
  - OGSI/GT3 based
  - Also previews of WS-I/OMII and WS-RF/GT4 releases

OGSA-DAI Services

- OGSA-DAI uses three main service types
  - DAISGR (registry) for discovery
  - GDSF (factory) to represent a data resource
  - GDS (data service) to access a data resource

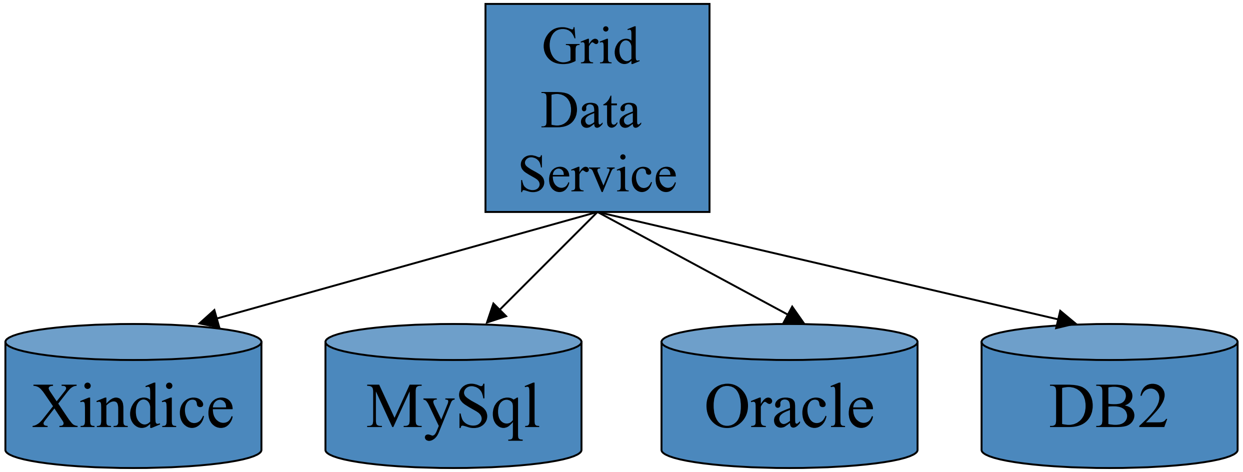


- This will change

GDSF and GDS

- Grid Data Service Factory (GDSF)
  - Represents a data resource
  - Persistent service
    - Currently static (no dynamic GDSFs)
      - Cannot instantiate new services to represent other/new databases
  - Exposes capabilities and metadata
  - May register with a DAISGR
- Grid Data Service (GDS)
  - Created by a GDSF
  - Generally transient service
  - Required to access data resource
  - Holds the client session

Heterogeneity

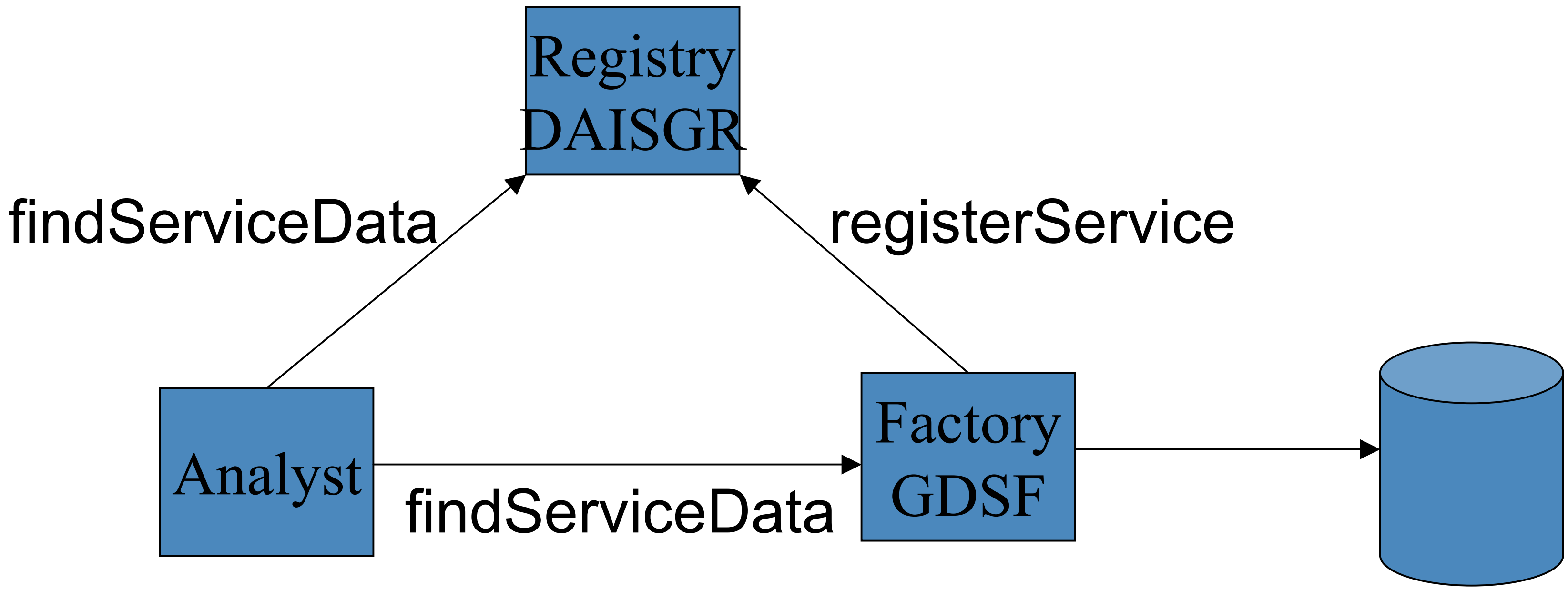


- Data source abstraction behind GDS instance
  - plug in “data resource implementations” for different data source technologies
  - does not mandate any particular query language or data format

DAISGR

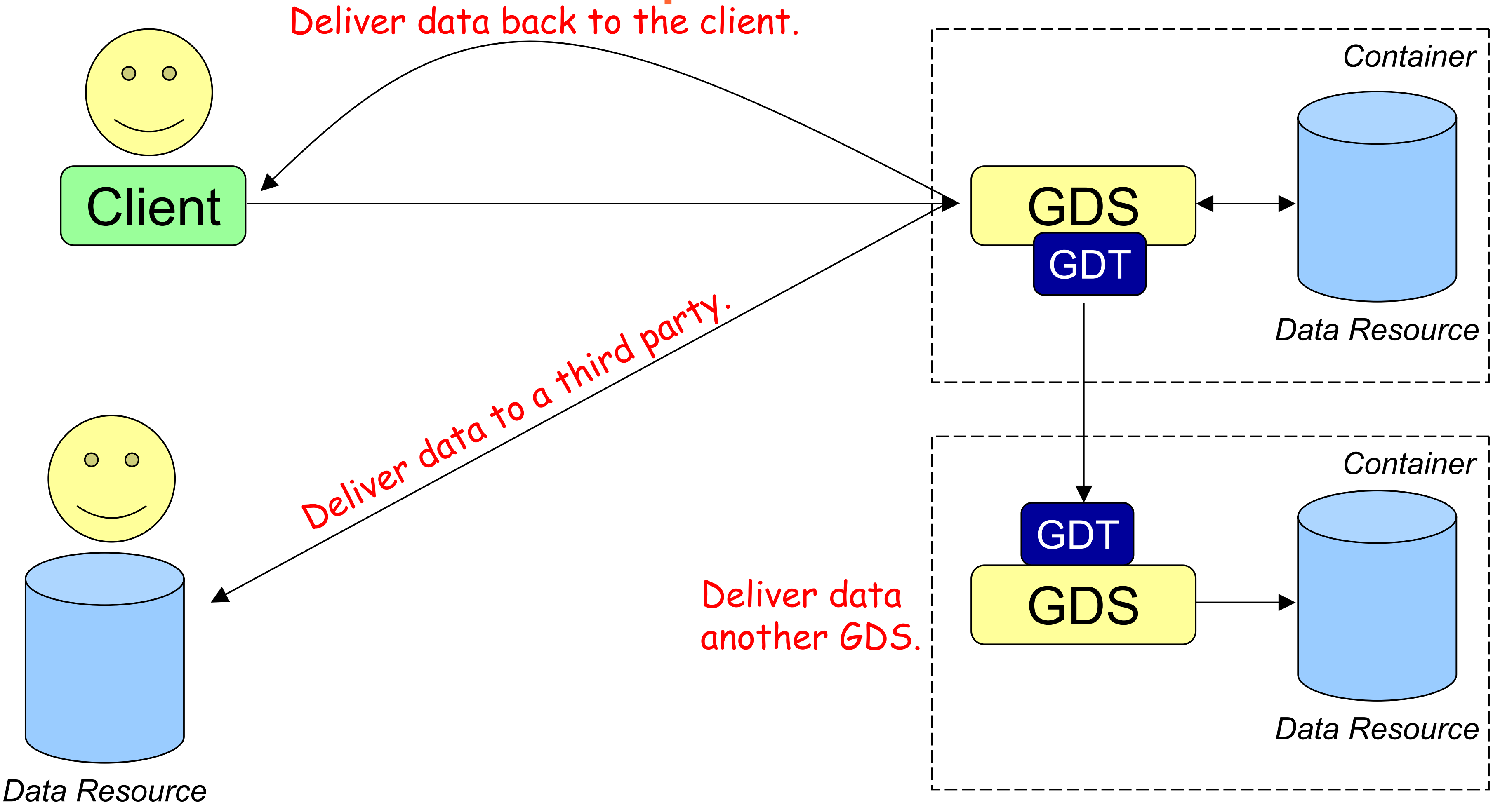
- DAI Service Group Registry (DAISGR)
  - Persistent service
  - Based on OGSI ServiceGroups
  - GDSFs may register with DAISGR
  - Clients access DAISGR to discover
    - Resources
    - Services (may need specific capabilities)
      - Support a given portType or activity

Location



- Data resource publication through registry
- Data location hidden by factory
- Data resource meta data available through Service Data Elements

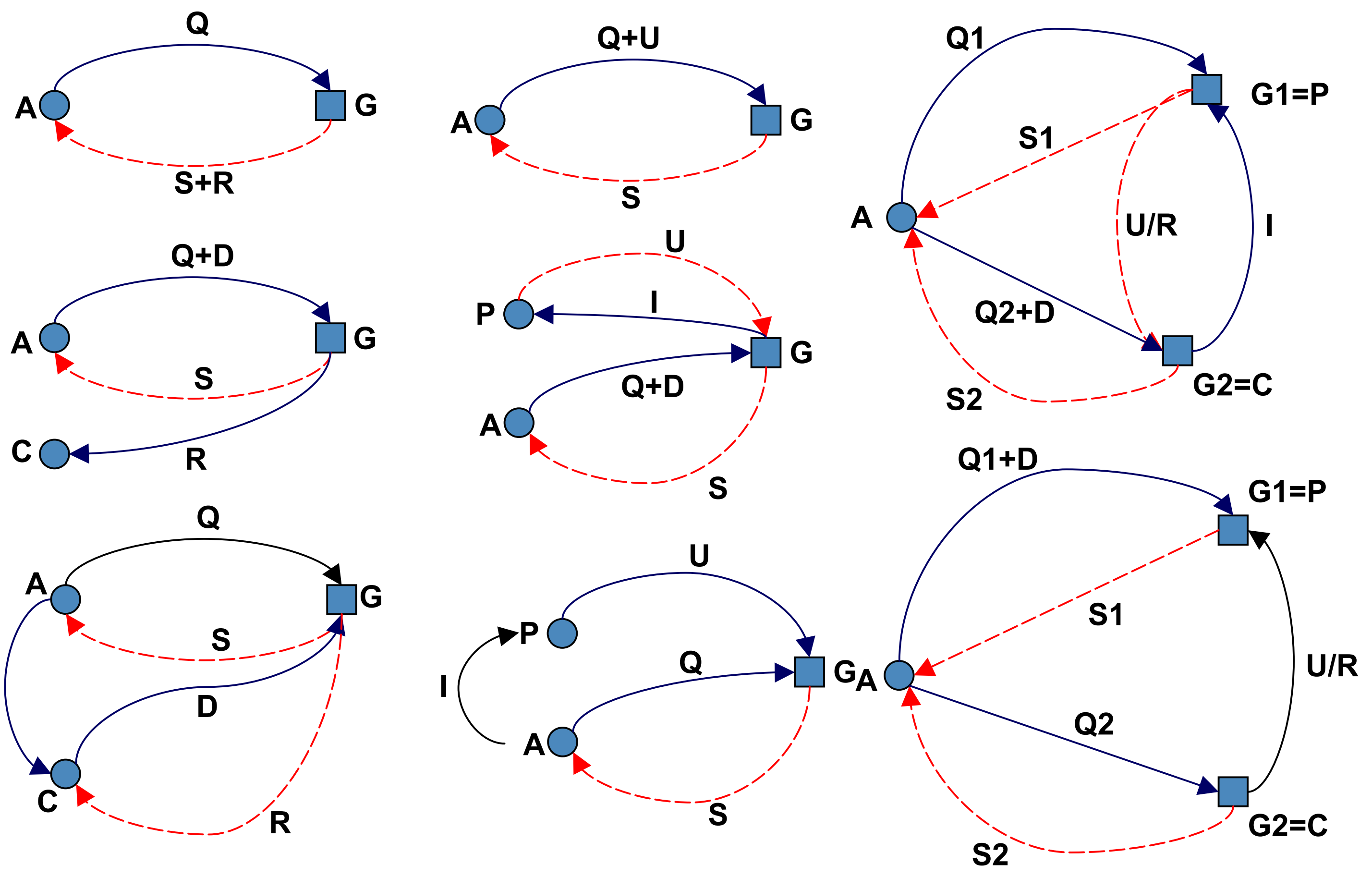
More Complex Behaviour



And there's a lot more that you can do …

Usage Patterns

Retrieve | Update/Insert | Pipeline



Data Flow

→ Call

⇢ Response

Actors

■ - OGSI process

● - Non-OGSI process

A - Analyst

C - Consumer

G - GDS

P - Producer

Data

Q - Query

D - Delivery

S - Status

R - Result

U - Update

I - Data id

Why OGSA-DAI?

- Why use OGSA-DAI over JDBC?
  - Can embed additional functionality at the service end
    - Transformations, compressions
    - Third party delivery
    - The extensible activity framework
  - Avoiding unnecessary data movement
  - Common interface to heterogeneous data resources
    - Relational, XML databases, and files
  - Usefulness of the Registry for service discovery
    - Dynamic service binding process
    - Provision of good meta-data is necessary
  - Language independence at the client end
    - Do not need to use Java
  - Platform independence
    - Do not have to worry about connection technology, drivers, et

OGSA-DAI Architecture

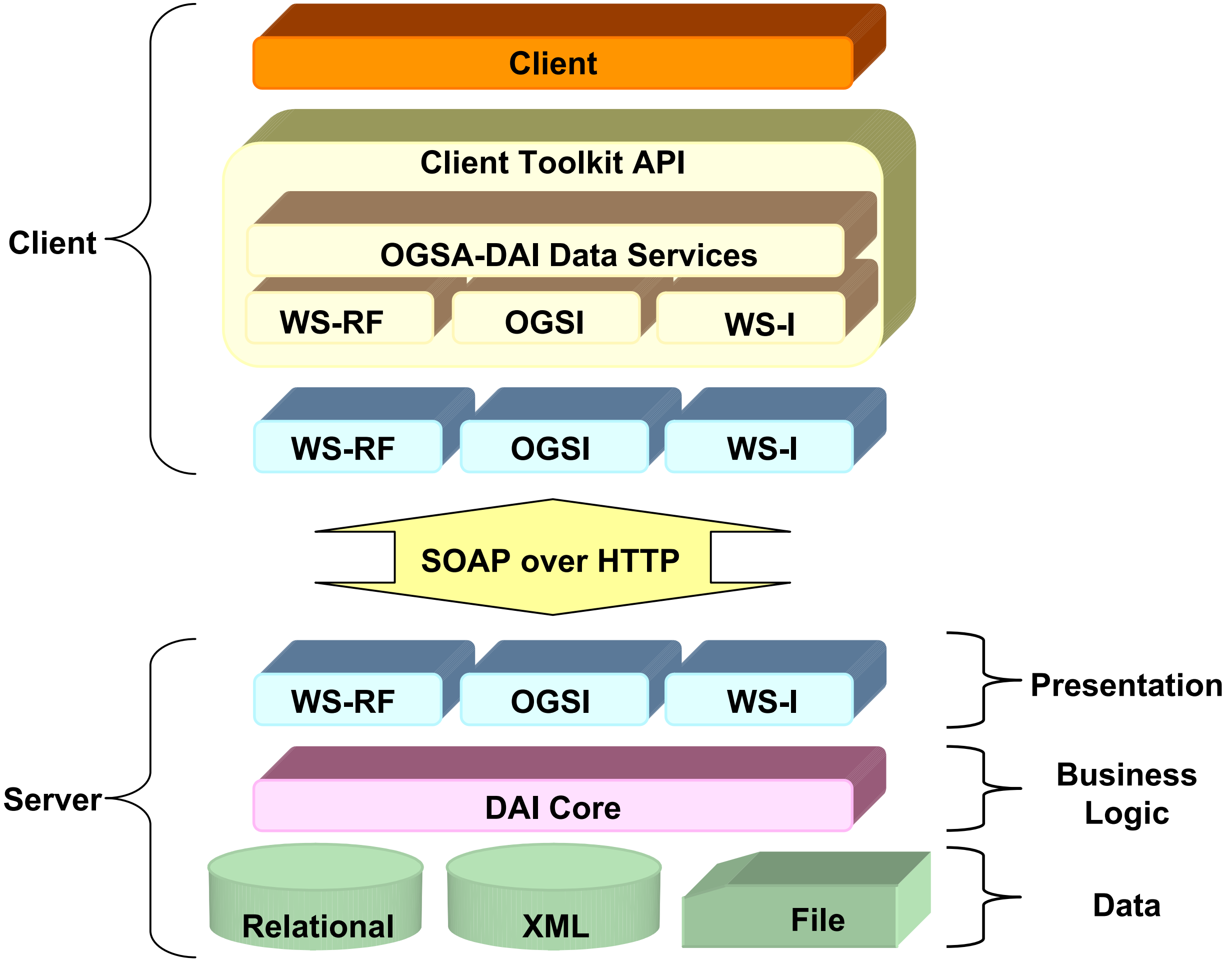
The OGSA-DAI Team
info@ogsadai.org.uk

OGSA-DAI Architecture

Address Multiple Interfaces

- OGSI:
  - OGSI 1.0
  - Globus Toolkit 3
- WS-I:
  - WSDL, SOAP, UDDI, WS-I 1.0, WS-Security
  - Axis 1.2 / Tomcat
- WS-RF:
  - WS-Addressing, WS-RF specifications.
  - Globus Toolkit 4
- DAIS

High Level Design



Data Layer

- Need to support heterogenous data resources:
  - Relational: MySQL, SQL Server, Oracle, DB2, HSQL.
  - XML:Xindice, eXist.
  - Files: files, BinX files.
  - RowSet, SQLResponse, XMLSequence, XMLDocument:
    - Views onto Relational / XML resources or explicit XML files.

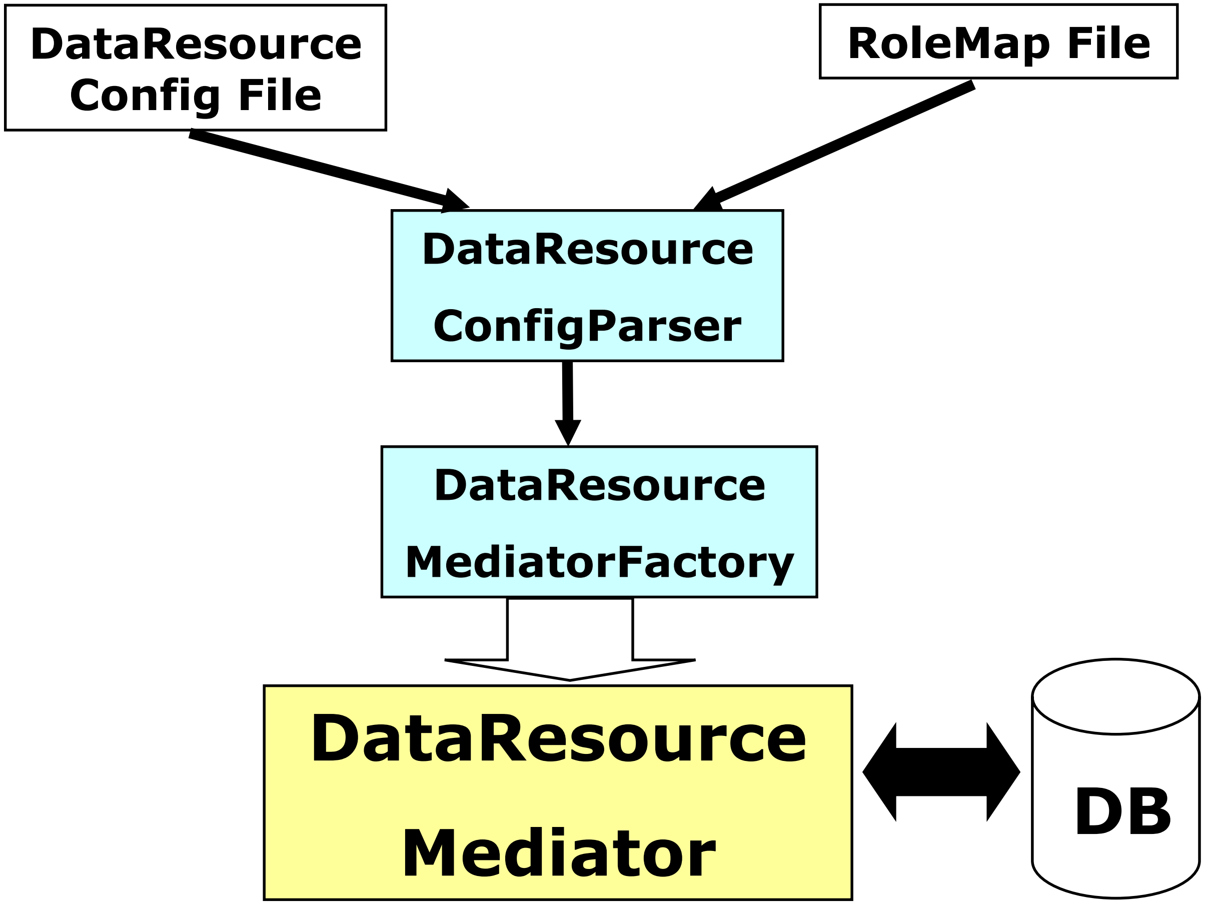
Data Layer

- Other resources:
  - Lists of (data) resources currently in existence.
  - Mappings of logical data resource names to database URIs and database names, collection names,….
  - Mappings of logical resource names to actual resource names.
  - Contexts for sessions / transactions:
  - Cached / transformed data.
  - Data awaiting delivery / collection.
  - Registry.
- Resource persistence

Business Logic Layer – DAI-Core

- Connection to, management of and interaction with data resources.
- Engine:
  - Execution of individual activities.
  - Execution of Perform documents specifying sequences of activities.
- Data transformation and delivery.
- Full DAI resource management:
  - Activity / Perform status resources.
  - Data cache / asynchronous delivery-related resources.
  - Session and transaction resources.
  - …

Data – DAI-Core Interface – DataResourceMediator



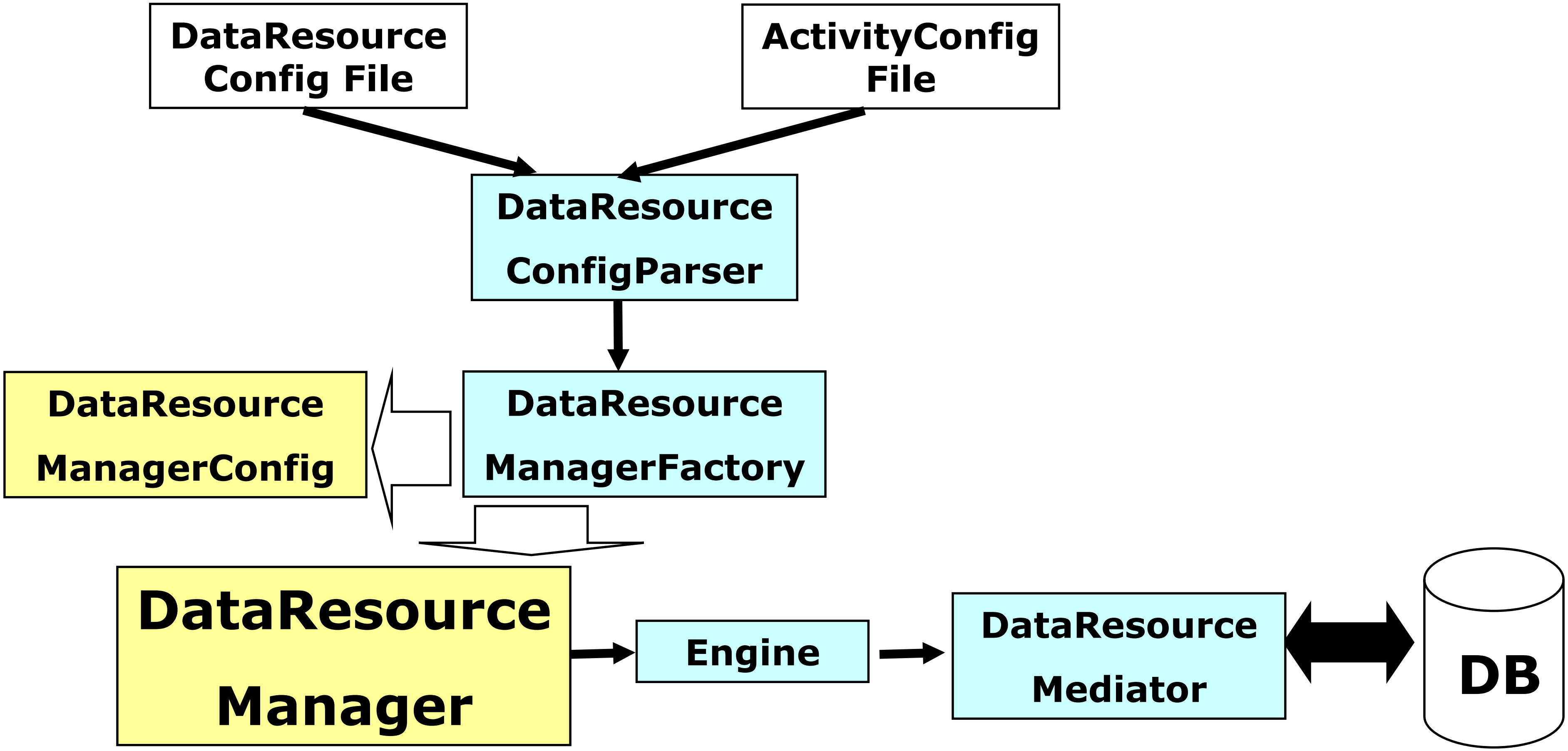
Presentation Layer

- OGSI:
  - Globus Toolkit 3.2 / Tomcat
- WS-I:
  - Apache Axis 1.2 / Tomcat
- WS-RF:
  - Globus Toolkit 3.9.x (4.0) / Tomcat
- DAIS

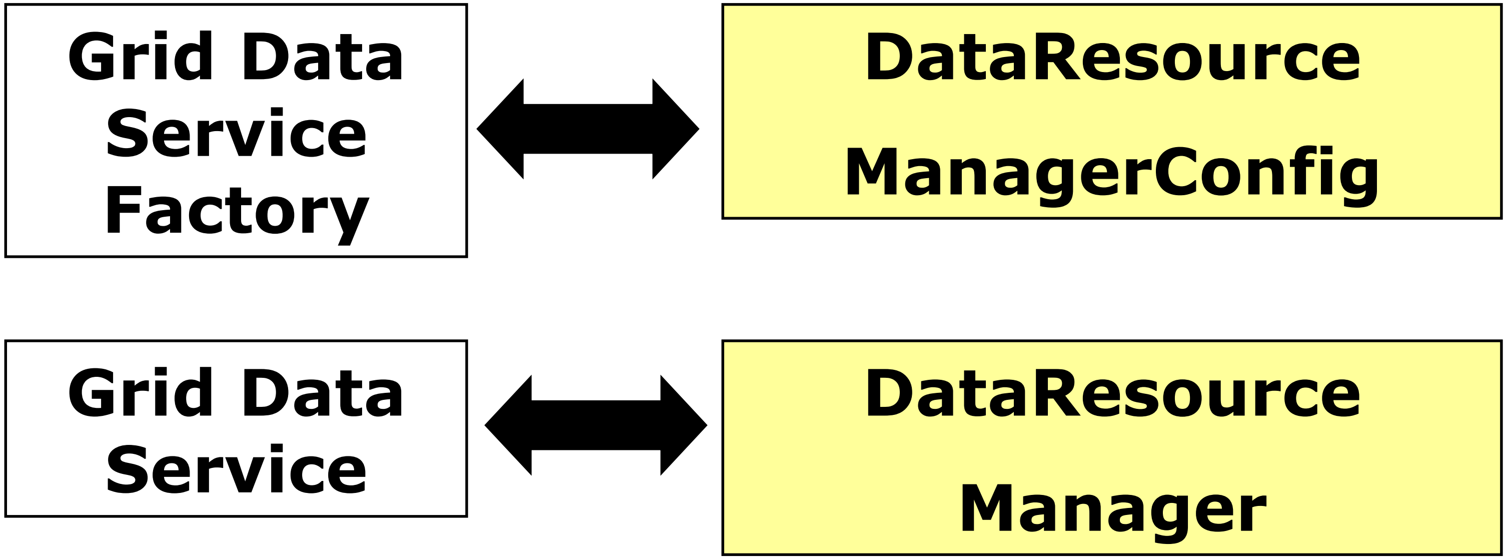
Interface – Information Flow

- Service=>DAI Core
  - Data resource configuration information.
  - (Data) resource property names.
  - Perform documents.
  - Security context.
- DAI Core=>Service
  - Response documents.
  - (Data) resource properties:
    - Request status.
    - Database schema.
    - Supported activities.
    - Perform document schema.

DAI-Core – Presentation Interface – DataResourceManager



OGSI



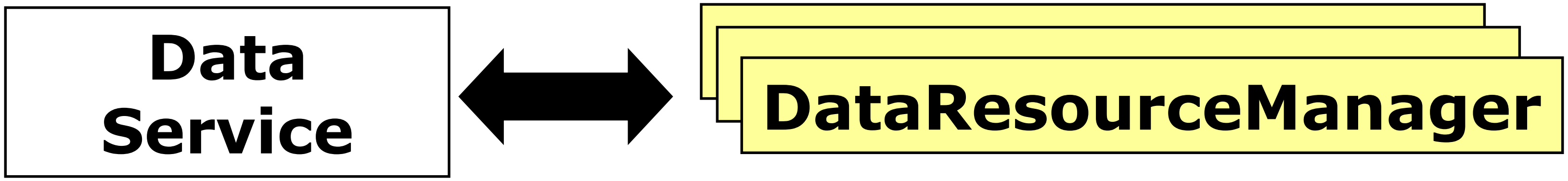
Grid Data Service Factory

DataResource ManagerConfig

Grid Data Service

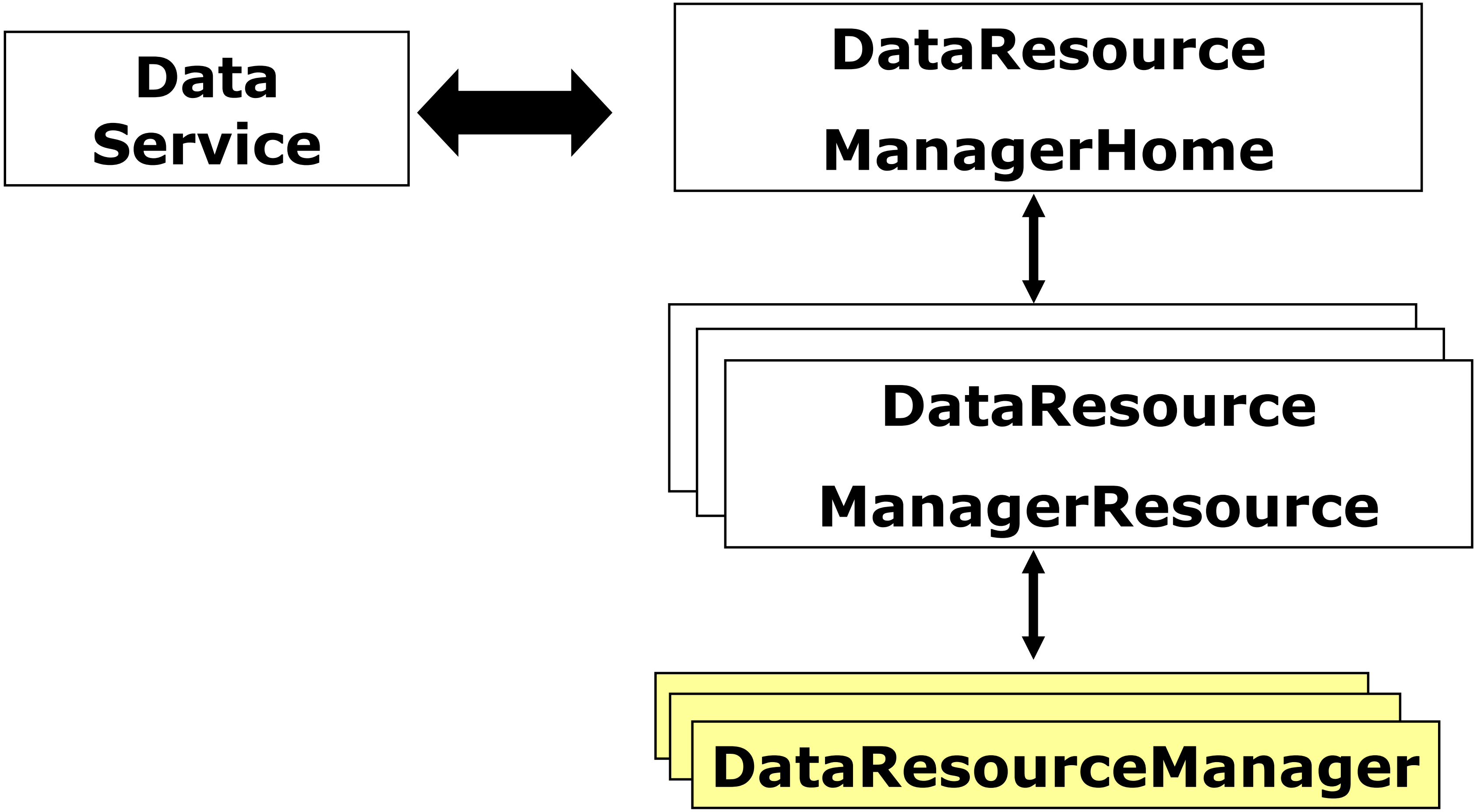
DataResource Manager

WS-I



- Service maintains:
  - List of data resources it exposes.
  - Indexed collections:
    - DataResourceManagerConfigs.
    - DataResourceManagers.
- Issues:
  - Exposing properties – callbacks.
  - Dynamic DataResourceManager creation.
  - Persisting information on current DataResourceManagers, request status…

WS-RF


Data Service
DataResource ManagerHome
DataResource ManagerResource
DataResourceManager

WS-RF

- Service maintains:
  - List of data resources it exposes.
  - Indexed collection - DataResourceManagerHome
    - DataResourceManagerResources – wrapper.
- Issues:
  - Exposing properties – callbacks.
  - Dynamic DataResourceManager creation.
  - Persisting information on current DataResourceManagers, request status…
  - Rely on GT4 infrastructure or provide our own?
    - Needed for WS-I anyway.

Other Issues

- Supporting GridFTP and GDT-related activities in WS-I and WS-RF:
  - Without dependence on GT3 code.
- Pluggable architecture:
  - Logging.
  - Auditing and accounting.
  - Performance and benchmarking.
  - Security.
  - …

Things that must be changed

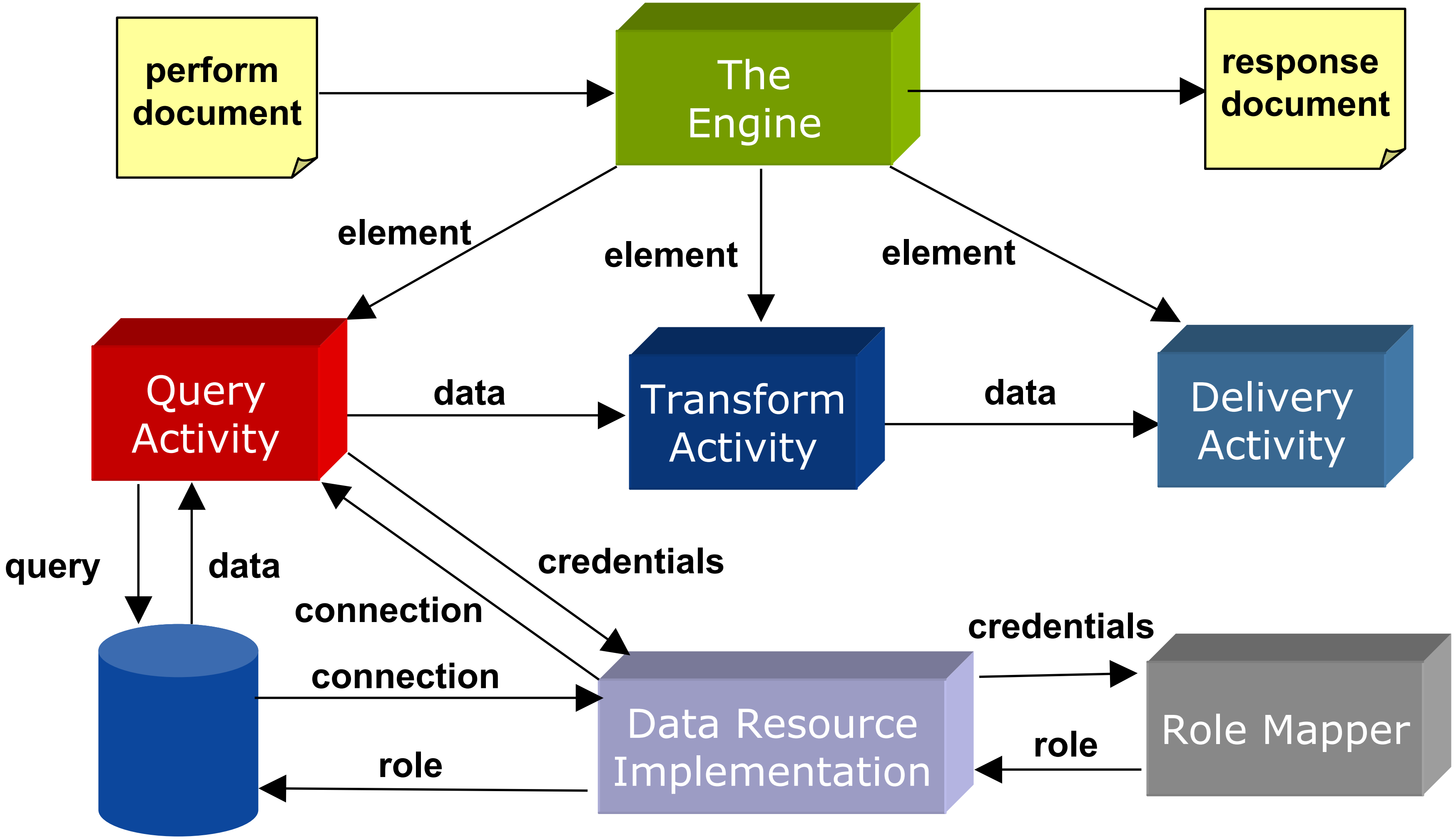
- Java names
- Qualified Names in metadata
- Metadata / Service Data
- XML Namespaces
- XML Schemas

OGSA-DAI GDS Overview

- Low-level components of a Grid Data Service
  - Engine
  - Activities
  - Data Resource Implementation
  - Role Mapper

- Extensibility of OGSA-DAI architecture
  - Interfaces and abstract classes
  - Design Patterns

GDS Internals



Grid Data Service

- GDS has a document based interface
  - Consumes perform documents
  - Produces response documents
  - Additional operations for 3rd party data delivery
- Motivation for using a document interface
  - Change in behaviour ≠> interface change
  - Reduce number of operation calls
  - Extensible

The GDS Engine

- Engine is the central GDS component
- Dictates behaviour when perform documents are submitted
    - Parses and validates perform document
    - Identifies required activities
    - Processes activities
    - Composes response document
    - Returns response document to GDS

Perform Documents

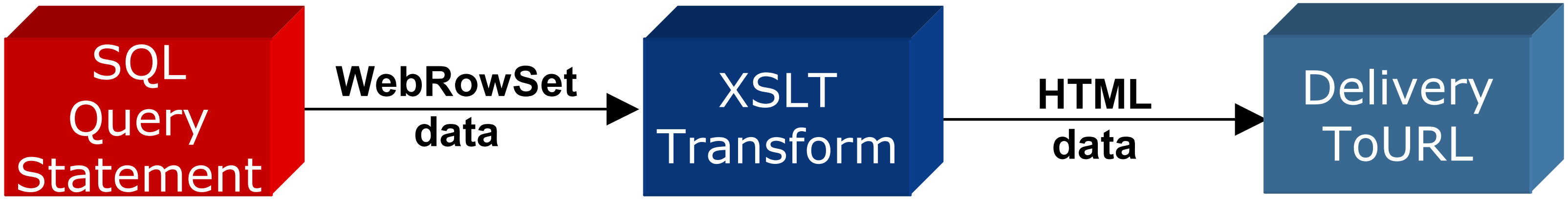
- Perform documents
  - Encapsulate multiple interactions with a service into a single interaction
  - Abstract each interaction into an “activity”
  - Data can flow from one activity to another

Query → Transformation → Delivery

  - Not quite workflow
    - No control constructs present (conditionals, loops, variables)

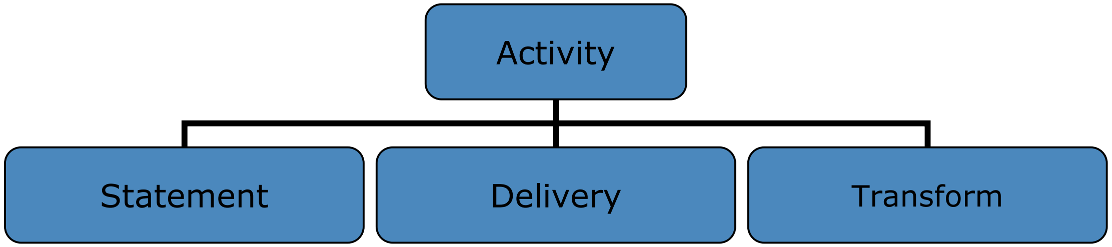
Activities

- 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 GDS
  - Specified in a configuration file
- Data can flow between activities (chained)



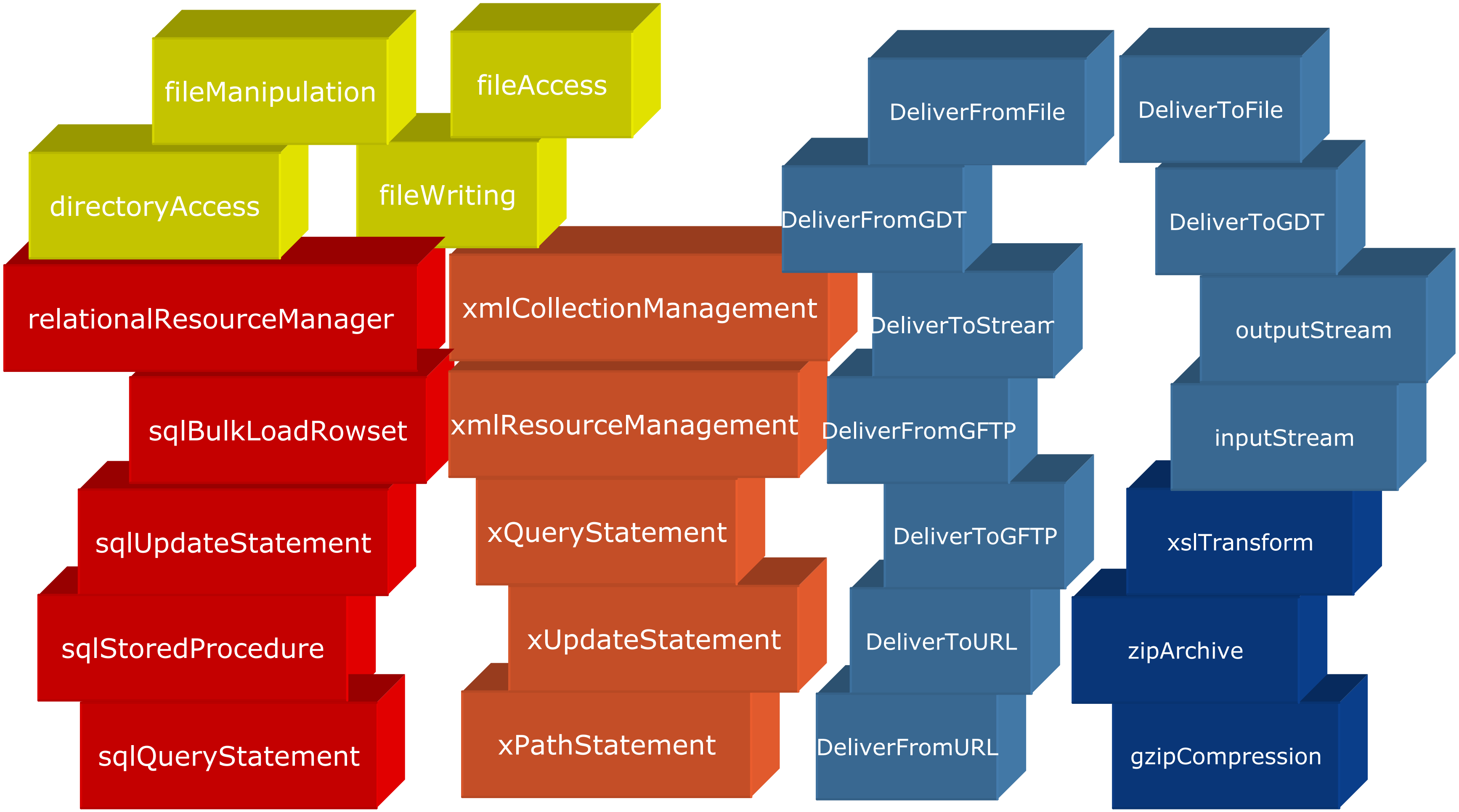
Activity Taxonomy

- Activities fall into three main functional groups



- Statement
  - Interact with the data resource
- Delivery
  - Deliver data to and from 3rd parties
- Transform
  - Perform transformations on data

Predefined Activities



fileManipulation
fileAccess
directoryAccess
fileWriting

relationalResourceManager
sqlBulkLoadRowset
sqlUpdateStatement
sqlStoredProcedure
sqlQueryStatement

xmlCollectionManagement
xmlResourceManagement
xQueryStatement
xUpdateStatement
xPathStatement

DeliverFromFile
DeliverFromGDT
DeliverToStream
DeliverFromGFTP
DeliverToGFTP
DeliverToURL
DeliverFromURL

DeliverToFile
DeliverToGDT
outputStream
inputStream
xslTransform
zipArchive
gzipCompression

The Activity Framework

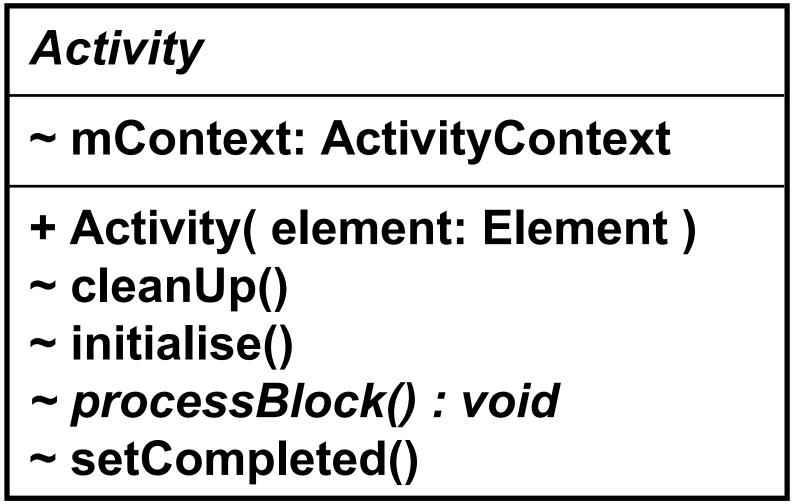
- Extensibility point
- Users can develop additional activities
  - To support different query languages
    - XQuery
  - To perform different kinds of transformation
    - STX
  - To deliver results using a different mechanism
    - WebDAV

- An activity requires

XSD schema	sql_query_statement.xsd
Java implementation	SQLQueryStatementActivity

The Activity Class

- All Activity implementations extend the abstract Activity class

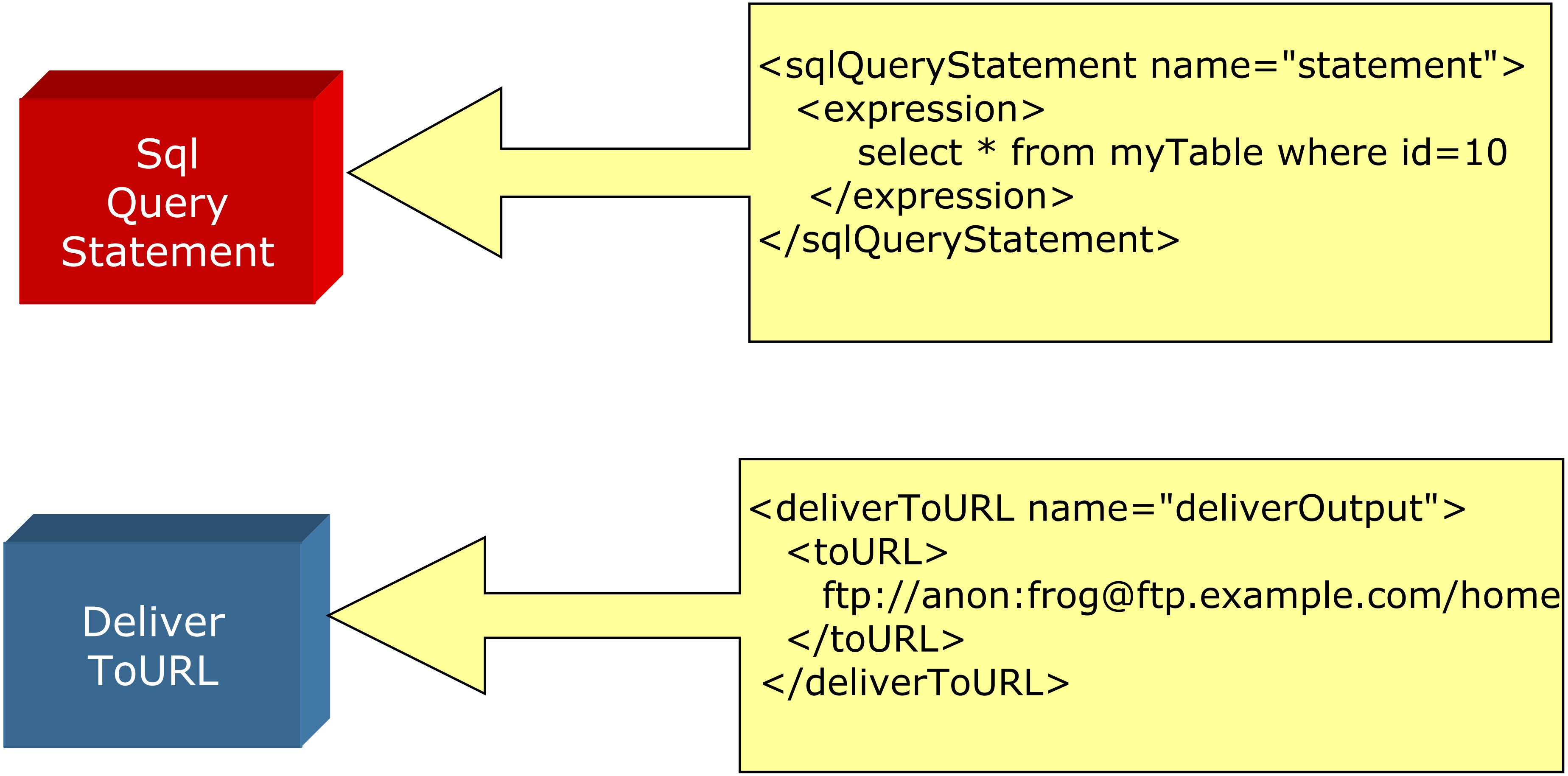


Activity

~ mContext: ActivityContext

+ Activity(element: Element)
~ cleanUp()
~ initialise()
~ *processBlock() : void*
~ setCompleted()

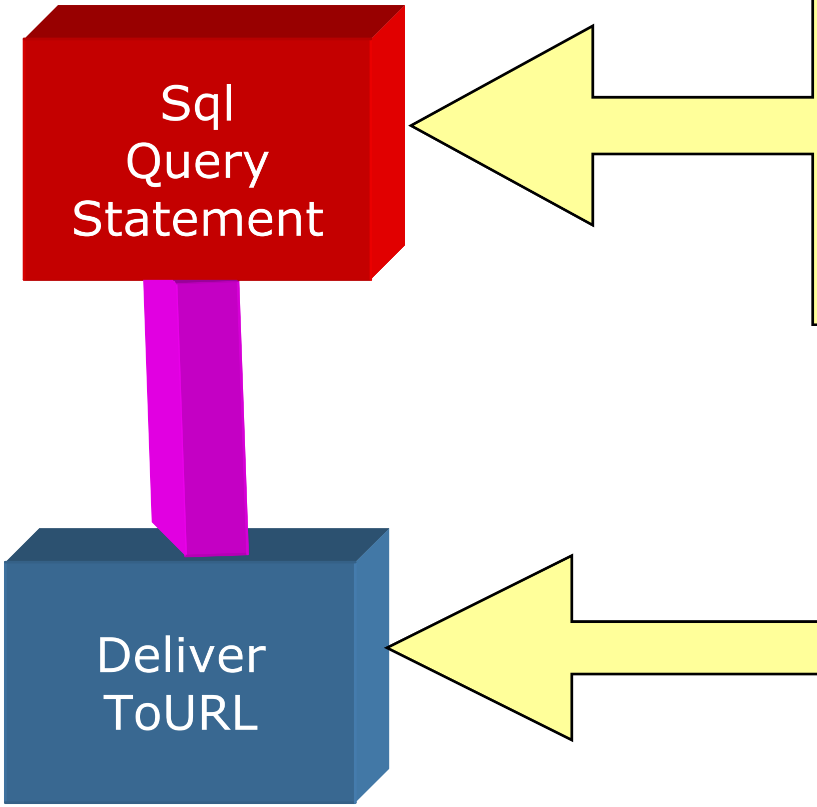
Connected Activities



```
<sqlQueryStatement name="statement">
   <expression>
        select * from myTable where id=10
    </expression>
</sqlQueryStatement>
```

```
<deliverToURL name="deliverOutput">
   <toURL>
      ftp://anon:frog@ftp.example.com/home
   </toURL>
 </deliverToURL>
```

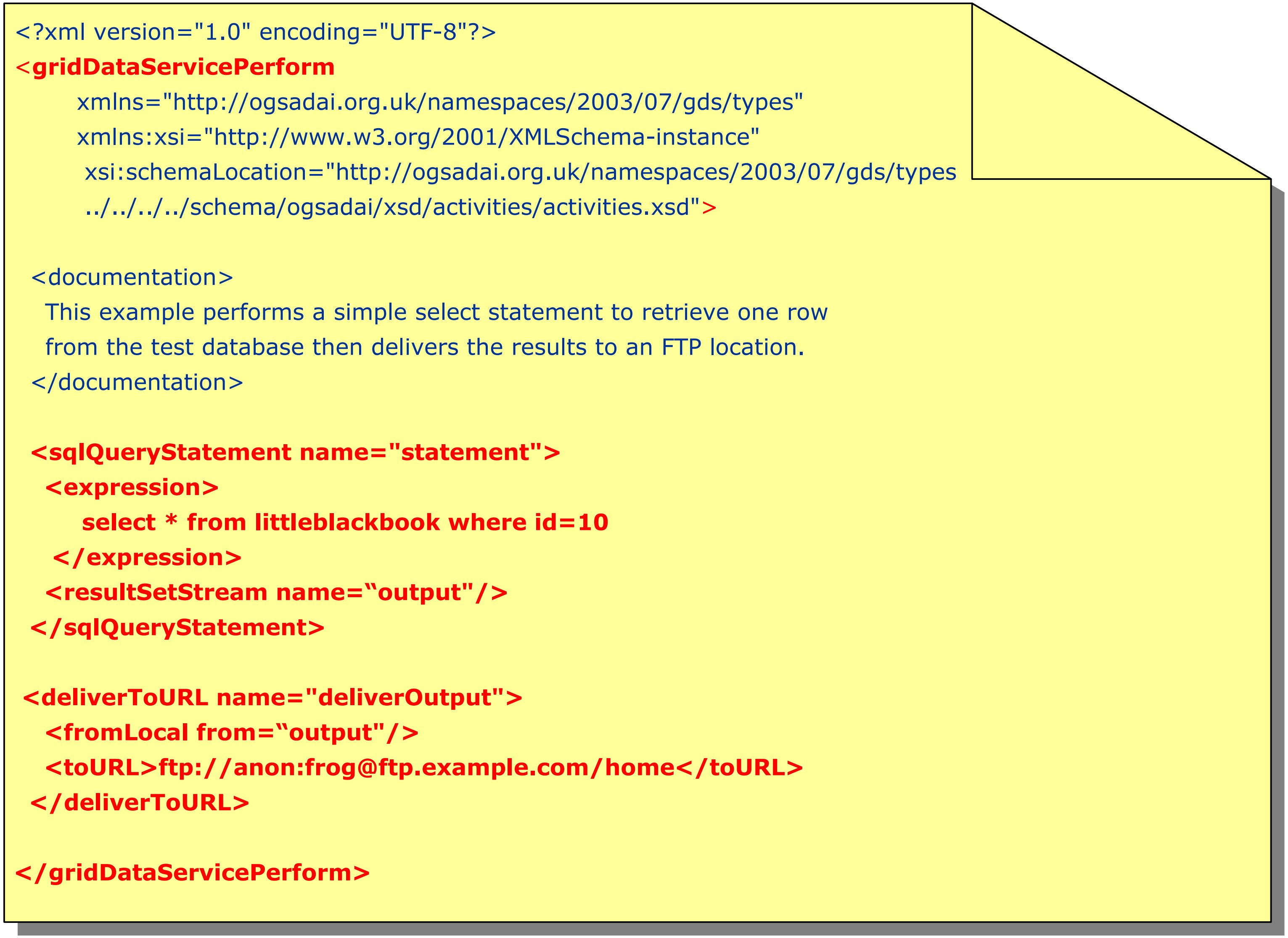
Connected Activities cont.



```
<sqlQueryStatement name="statement">
   <expression>
        select * from myTable where id=10
    </expression>
    <resultSetStream name="MyOutput"/>
</sqlQueryStatement>
```

```
<deliverToURL name="deliverOutput">
   <fromLocal from="MyOutput"/>
   <toURL>
      ftp://anon:frog@ftp.example.com/home
   </toURL>
</deliverToURL>
```

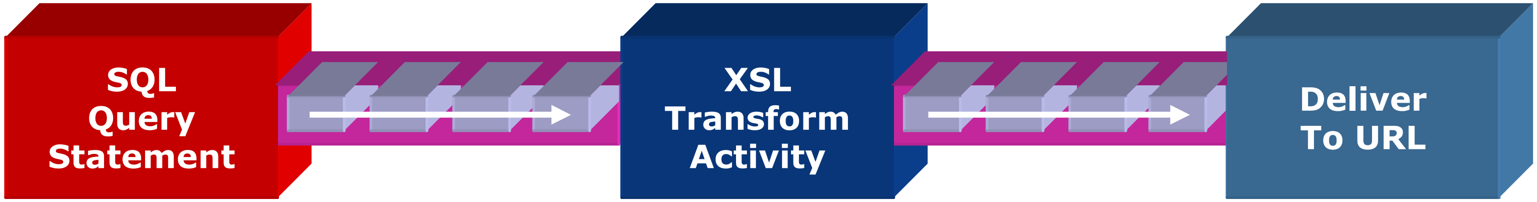
The Perform Document



```
<?xml version="1.0" encoding="UTF-8"?>
<gridDataServicePerform
       xmlns="http://ogsadai.org.uk/namespaces/2003/07/gds/types"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://ogsadai.org.uk/namespaces/2003/07/gds/types
        ../../../../schema/ogsadai/xsd/activities/activities.xsd">

  <documentation>
   This example performs a simple select statement to retrieve one row
   from the test database then delivers the results to an FTP location.
  </documentation>

  <sqlQueryStatement name="statement">
   <expression>
        select * from littleblackbook where id=10
    </expression>
   <resultSetStream name=“output"/>
  </sqlQueryStatement>

 <deliverToURL name="deliverOutput">
   <fromLocal from=“output"/>
   <toURL>ftp://anon:frog@ftp.example.com/home</toURL>
  </deliverToURL>

</gridDataServicePerform>
```

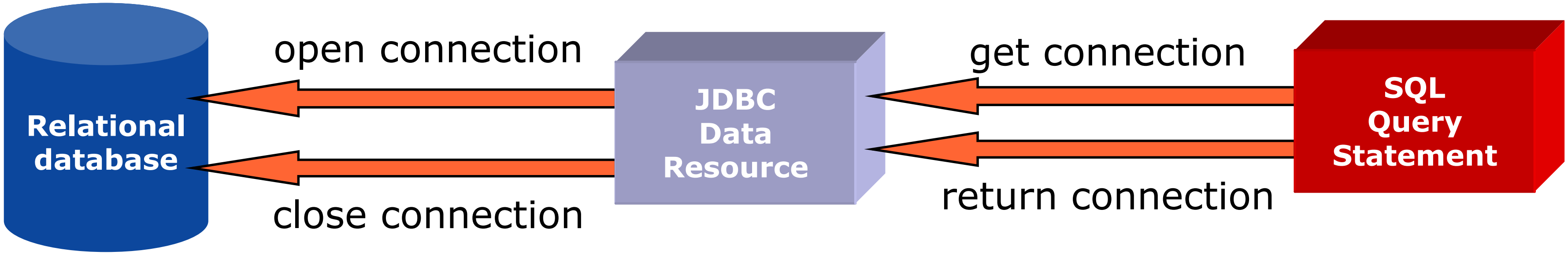
Activity Inputs and Outputs

- 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

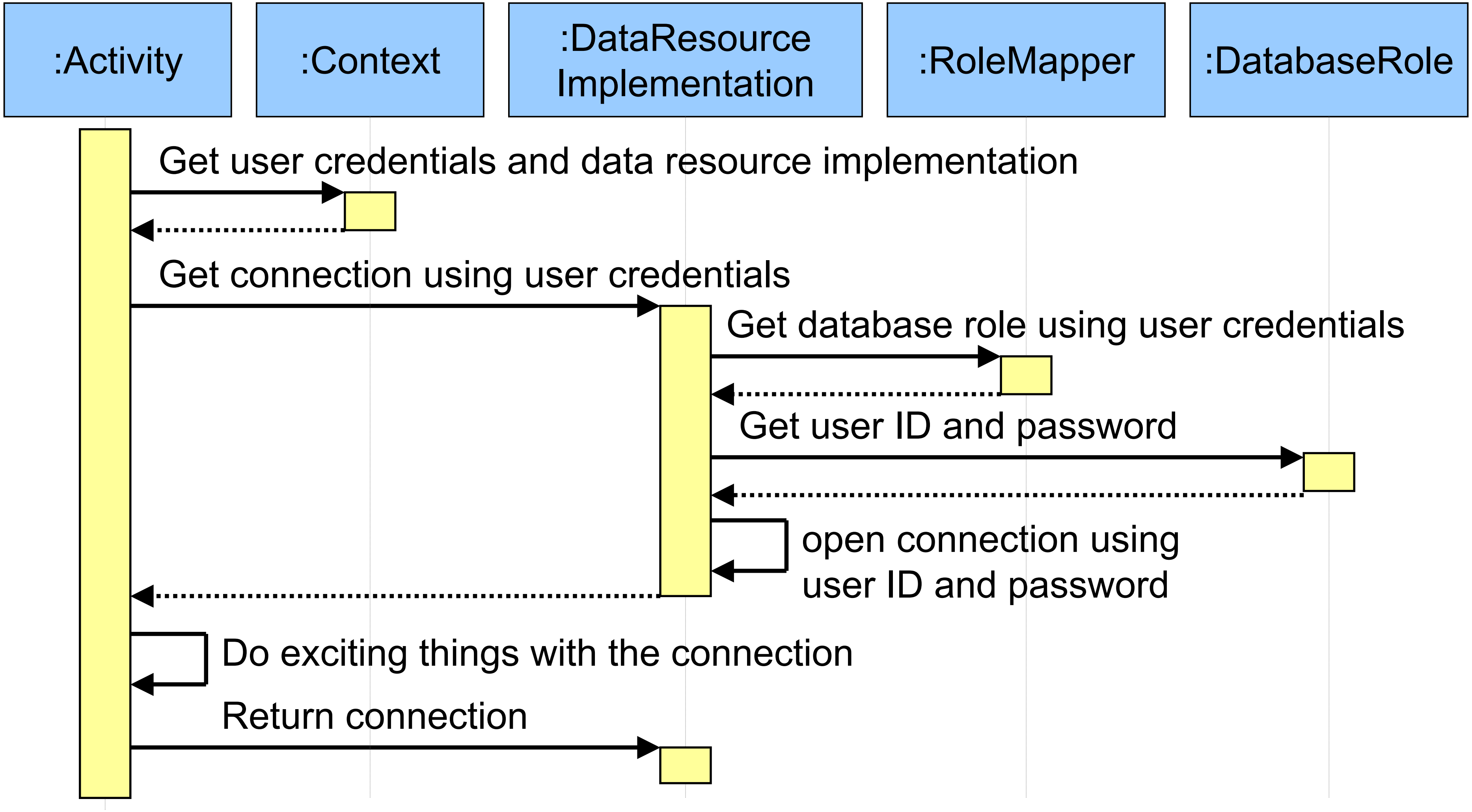


Data Resource Implementations

- Governs access to a data resource
  - Open/close connections
  - Validate user credentials using a RoleMapper
  - Facilitate connection pooling
- Provided for JDBC, XML:DB, and Files



Accessing Data Resource Sequence Diagram


:Activity
:Context
:DataResource Implementation
:RoleMapper
:DatabaseRole
Get user credentials and data resource implementation
Get connection using user credentials
Get database role using user credentials
Get user ID and password
open connection using user ID and password
Do exciting things with the connection
Return connection

Advantages of the Activity Model

- Avoid multiple message exchanges
  - Multiple activities within a single request
- Extensible
  - Developers can add functionality
  - Could import third party trusted activities
- Simplicity
  - Internal classes manage data flow, access to databases, etc
- Allows for optimisation
  - GDS engine can optimise internals

Issues with current Activity Model

- Incomplete syntax
  - No typing of inputs and outputs
    - How do you determine the data types that can be accepted?
- Incomplete semantics
  - What does it mean to be a FilterActivity?
- Keeping implementation and XML Schema fragment in synch
- Puts workload on the server
  - May need dynamic job placement

Summary (Architecture)

- Supporting different interfaces is difficult
  - lowest common denominator means loss of functionality or increase in workload
  - want interoperability as no platform will dominate (just now)
  - Globus Toolkit provides a lot of useful functionality

Summary (GDS Design)

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

The OGSA-DAI Client Toolkit

Overview

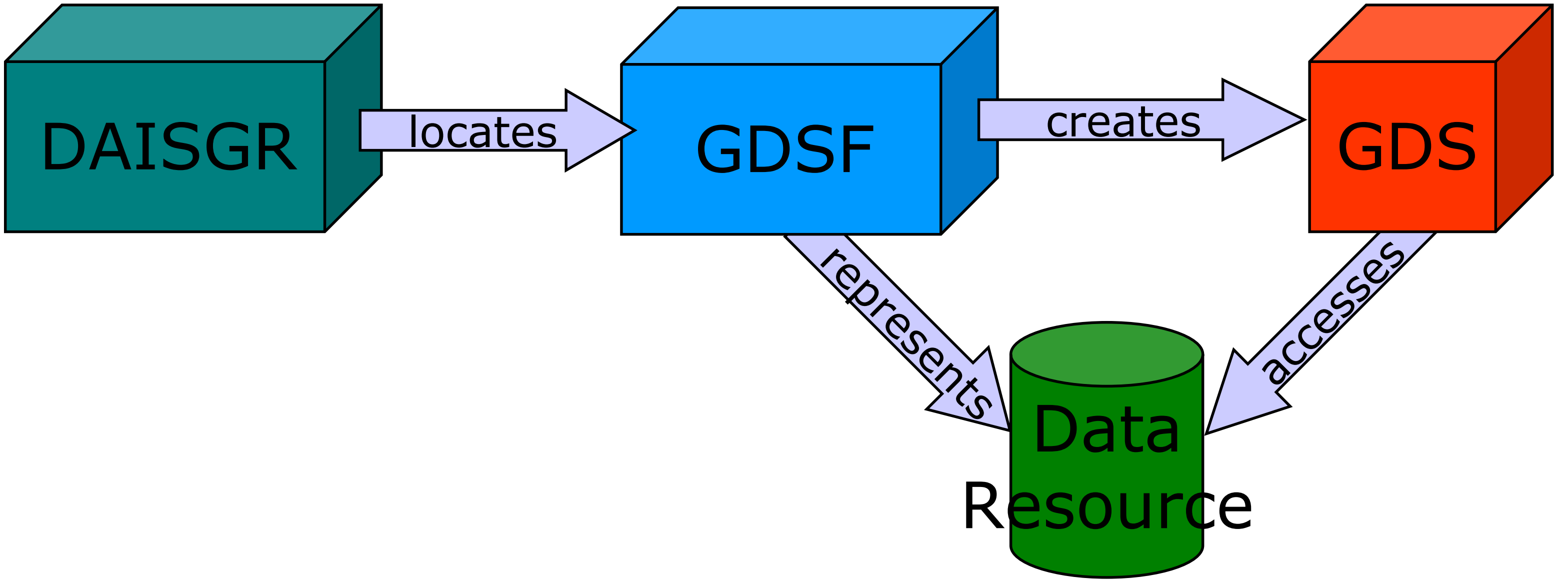
- The Client Toolkit
- OGSA-DAI Service Types
- Locating and Creating Data Services
- Requests and Results
- Delivery of Data
- Data Integration

Why use a Client Toolkit?

- Nobody wants to write XML!
- Users aren’t concerned about the connection mechanism
- Protects developer from
  - Changes in activity schema
  - Changes in service interfaces
  - Low-level APIs
  - DOM manipulation

OGSA-DAI Services

- OGSA-DAI uses three main service types
  - DAISGR (registry) for discovery
  - GDSF (factory) to represent a data resource
  - GDS (data service) to access a data resource



ServiceFetcher

- The ServiceFetcher class creates service objects from a URL

```
ServiceGroupRegistry registry =
       ServiceFetcher.getRegistry( registryHandle );
GridDataServiceFactory factory =
       ServiceFetcher.getFactory( factoryHandle );
GridDataService service =
       ServiceFetcher.getGridDataService( handle );
```

Registry

- A registry holds a list of service handles and associated metadata
- For example, clients can query a registry for all registered Grid Data Factory Services

```
GridServiceMetaData[] services =
    registry.listServices(
          OGSADAIConstants.GDSF_PORT_TYPE );
```

- The *GridServiceMetaData* object contains the handle and the port types that the factory implements

```
String handle = services[0].getHandle();
QName[] portTypes = services[0].getPortTypes();
```

Creating Data Services

- A factory object can create a new Grid Data Service.

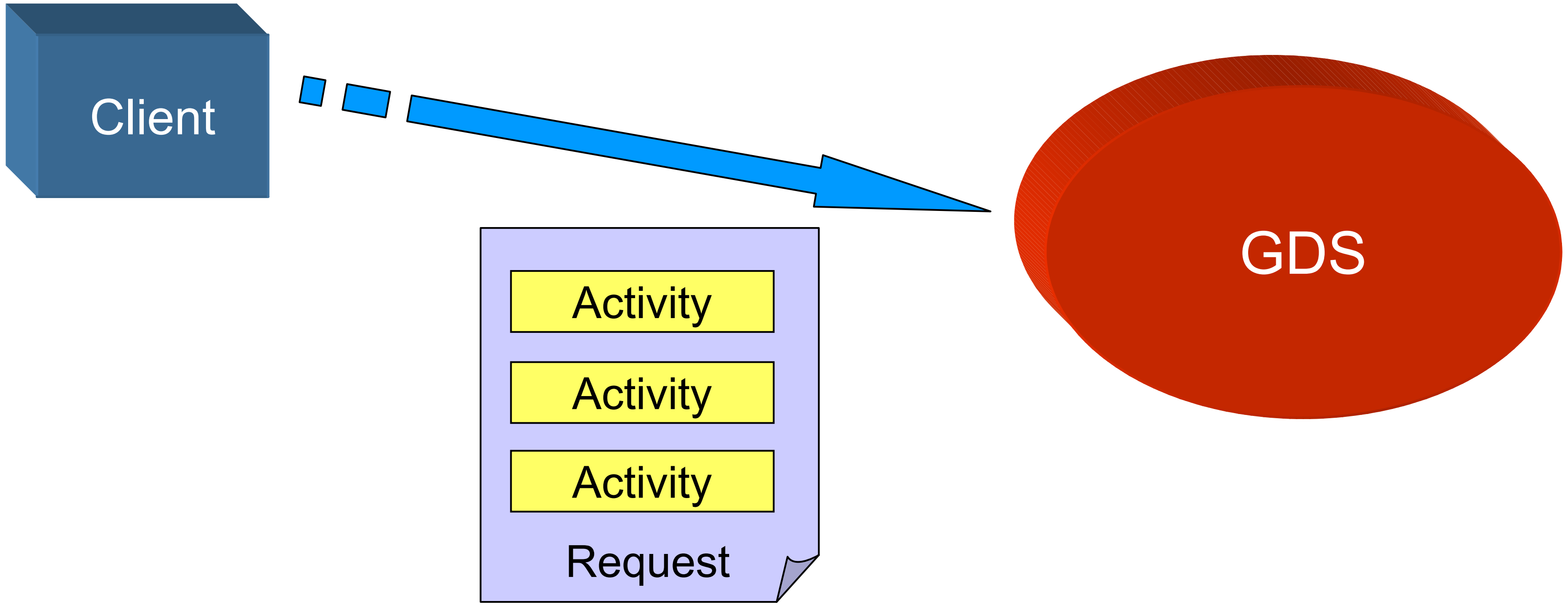
```
GridDataService service =
                factory.createGridDataService();
```

- Grid Data Services are transient (i.e. have finite lifetime) so they can be destroyed by the user.

```
service.destroy();
```

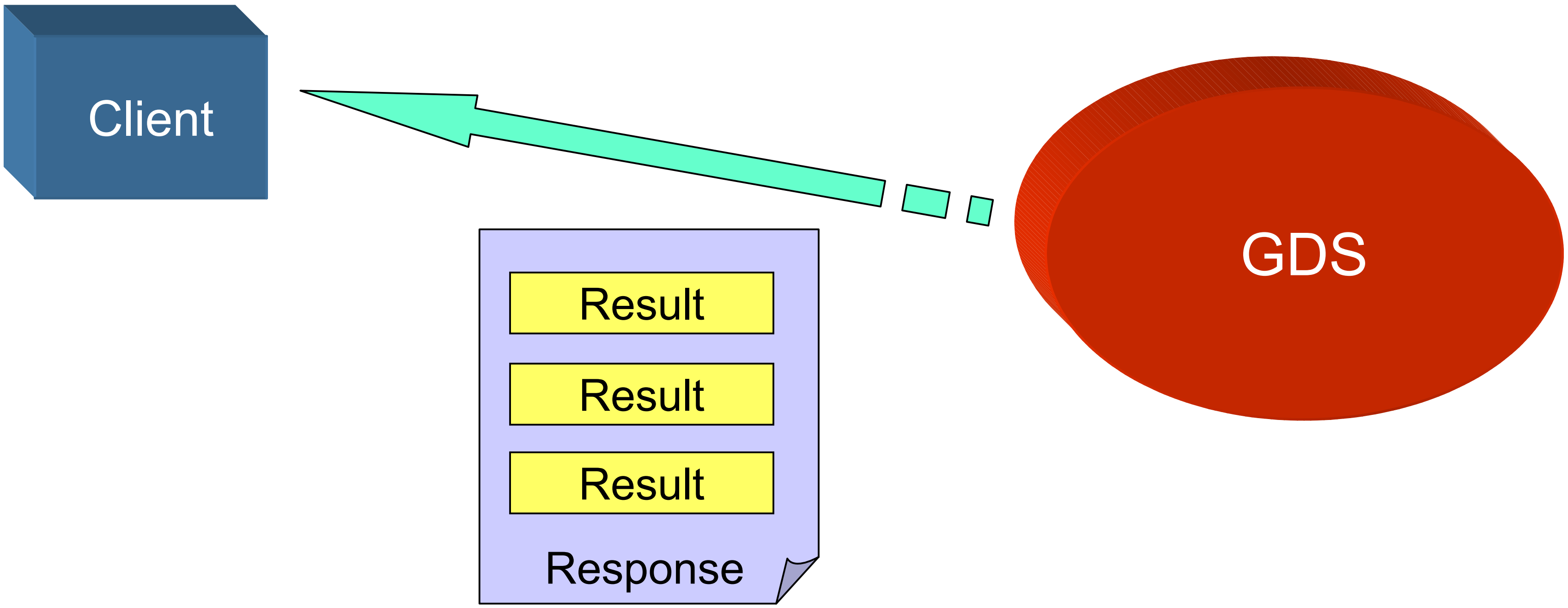
Interaction with a GDS

- Client sends a request to a data service
- A request contains a set of activities



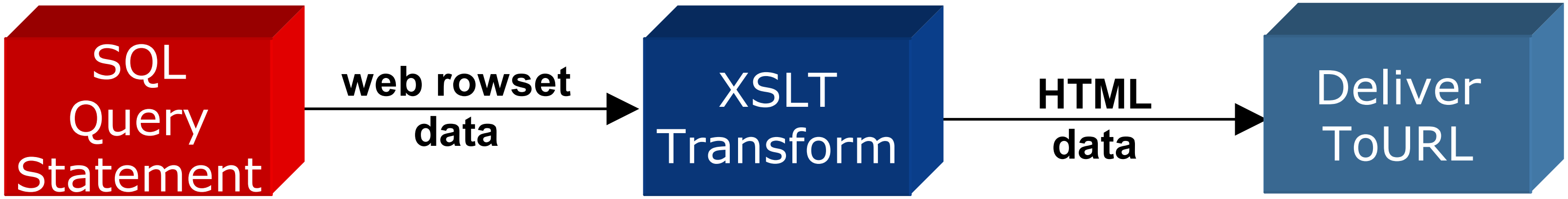
Interaction with a GDS

- The Data service processes the request
- Returns a response document with a result for each activity



Activities and Requests

- A request contains a set of activities
- An activity dictates an action to be performed
  - Query a data resource
  - Transform data
  - Deliver results
- Data can flow between activities



Examples of Activities

- SQLQuery

```
SQLQuery query = new SQLQuery(
   "select * from littleblackbook where id='3475'");
```

- XPathQuery

```
XPathQuery query = new XPathQuery( "/entry[@id<10]" );
```

- XSLTransform

```
XSLTransform transform = new XSLTransform();
```

- DeliverToGFTP

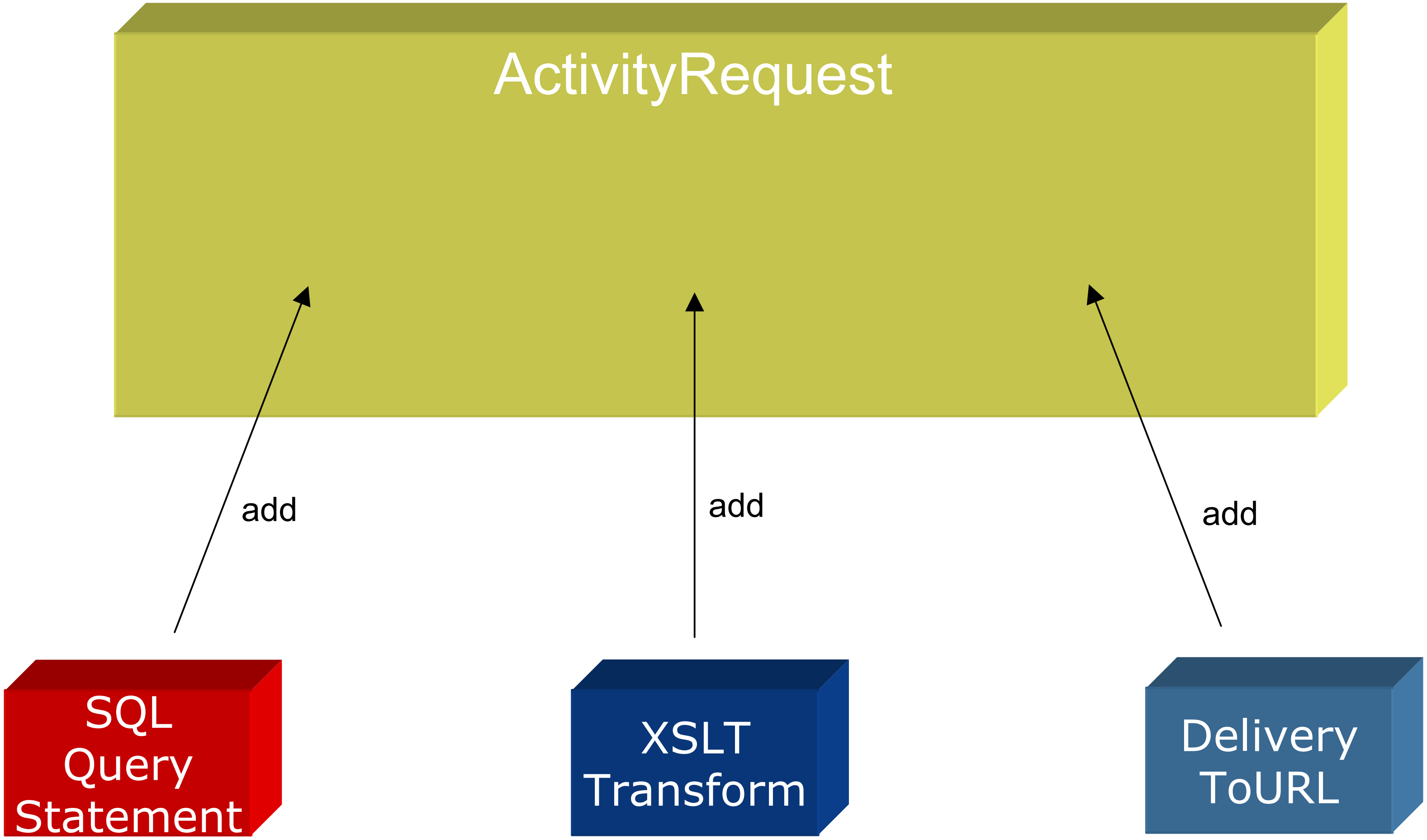
```
DeliverToGFTP deliver = new DeliverToGFTP(
       "ogsadai.org.uk", 8080, "myresults.txt" );
```

Simple Requests

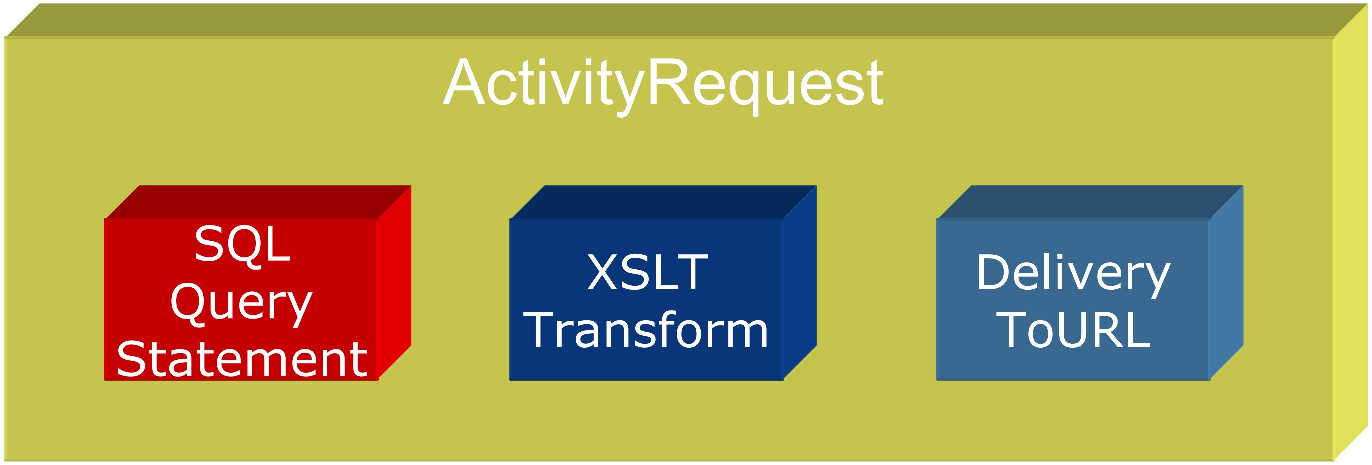
- Simple requests consist of only one activity
- Send the activity directly to the perform method

```
SQLQuery query = new SQLQuery(
   "select * from littleblackbook where id='3475'");

Response response = service.perform( query );
```

Constructing an ActivityRequest



Constructing a Request cont.

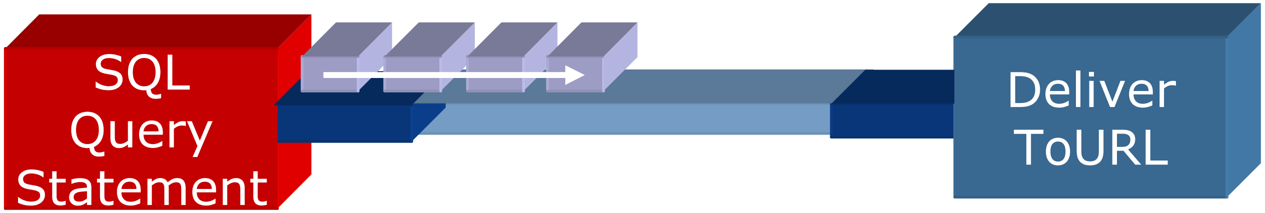


```
ActivityRequest request = new ActivityRequest();
request.add( query );
request.add( transform );
request.add( delivery );
```

Data Flow

- Connecting activities

```
SQLQuery query = new SQLQuery(
   "select * from littleblackbook where id<=1000");
DeliverToURL deliver = new DeliverToURL( url );

deliver.setInput( query.getOutput() );
```



Performing Requests

- Finally… perform the request!

```
Response response = service.perform( request );
```

- The response contains status and results of each activity in the request.

```
System.out.println( response.getAsString() );
```

Processing Results

- Varying formats of output data
  - SQLQuery
    - JDBC ResultSet:

      ```
      ResultSet rs = query.getResultSet();
      ```
  - SQLUpdate
    - Integer:

      ```
      int rows = update.getModifiedRows();
      ```
  - XPathQuery
    - XML:DB ResourceSet:

      ```
      ResourceSet results = query.getResourceSet();
      ```
- Output can always be retrieved as a String

  ```
  String output = myactivity.getOutput().getData();
  ```

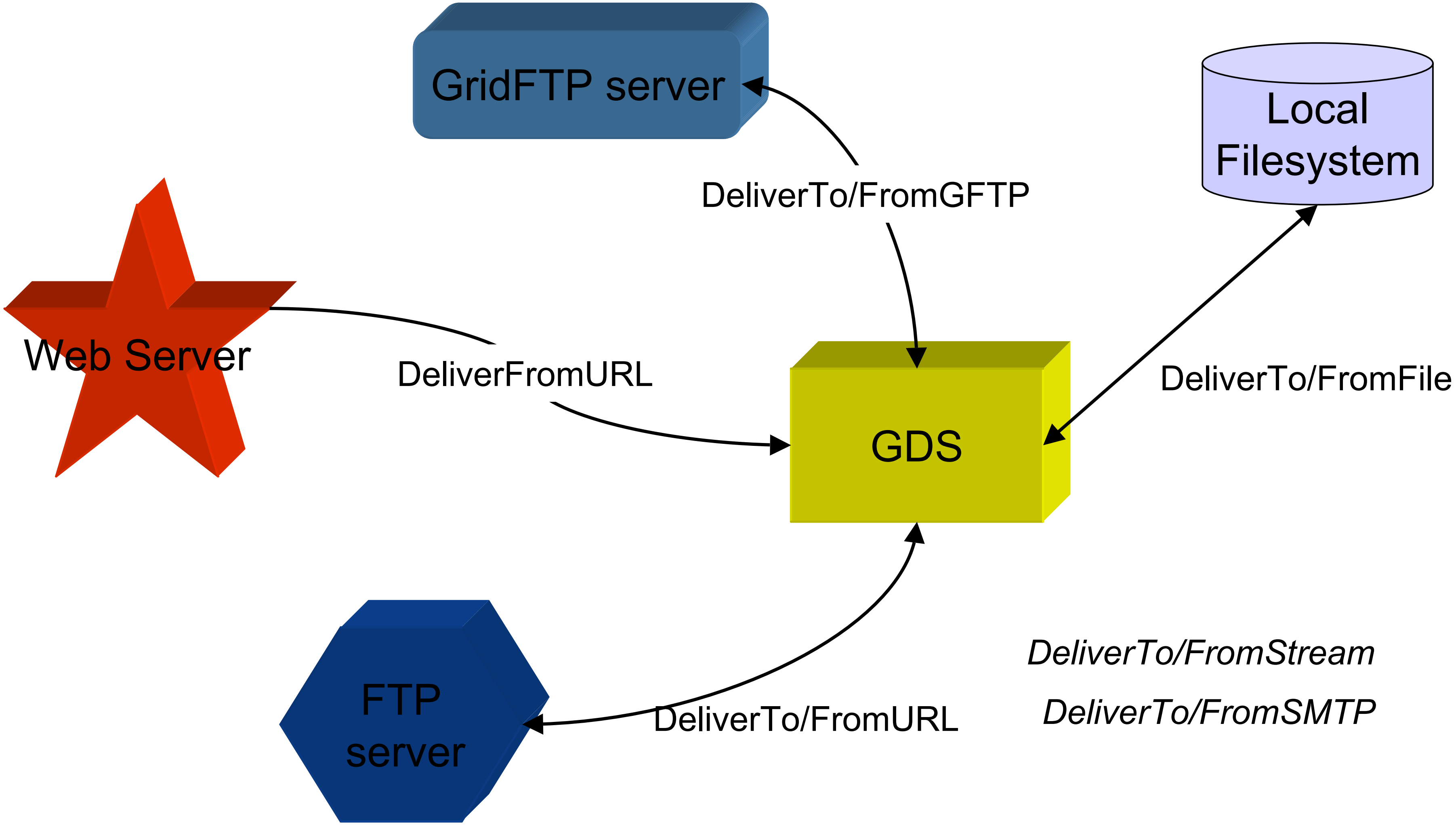
Delivery

- Data can be pulled from or pushed to a remote location.
- OGSA-DAI supports third-party transfer using *FTP*, *HTTP*, or *GridFTP* protocols.

```
DeliverToURL deliver = new DeliverToURL( url );
deliver.setInput( myactivity.getOutput() );
```

```
DeliverToGFTP deliver = new DeliverToGFTP(
    “ogsadai.org.uk”, 8080, “tmp/data.out” );
deliver.setInput( myactivity.getOutput() );
```

Delivery Methods



Delivery Activities

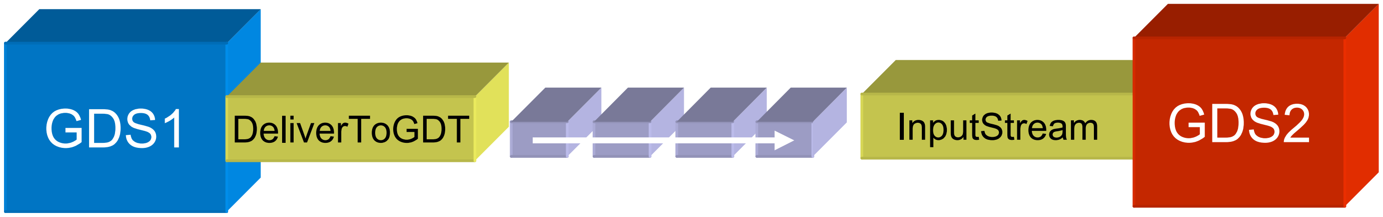
- The *DeliverFromURL* and *DeliverToURL* activities transfer data to/from a remote location.

```
DeliverFromURL deliver = new DeliverFromURL( url );

myactivity.setInput( deliver.getOutput() );
```

- Supported protocols are *http*, *ftp*, and *file*.
- Other delivery activities:
  - DeliverFromGFTP/DeliverToGFTP
  - DeliverToStream

Delivering data to another GDS

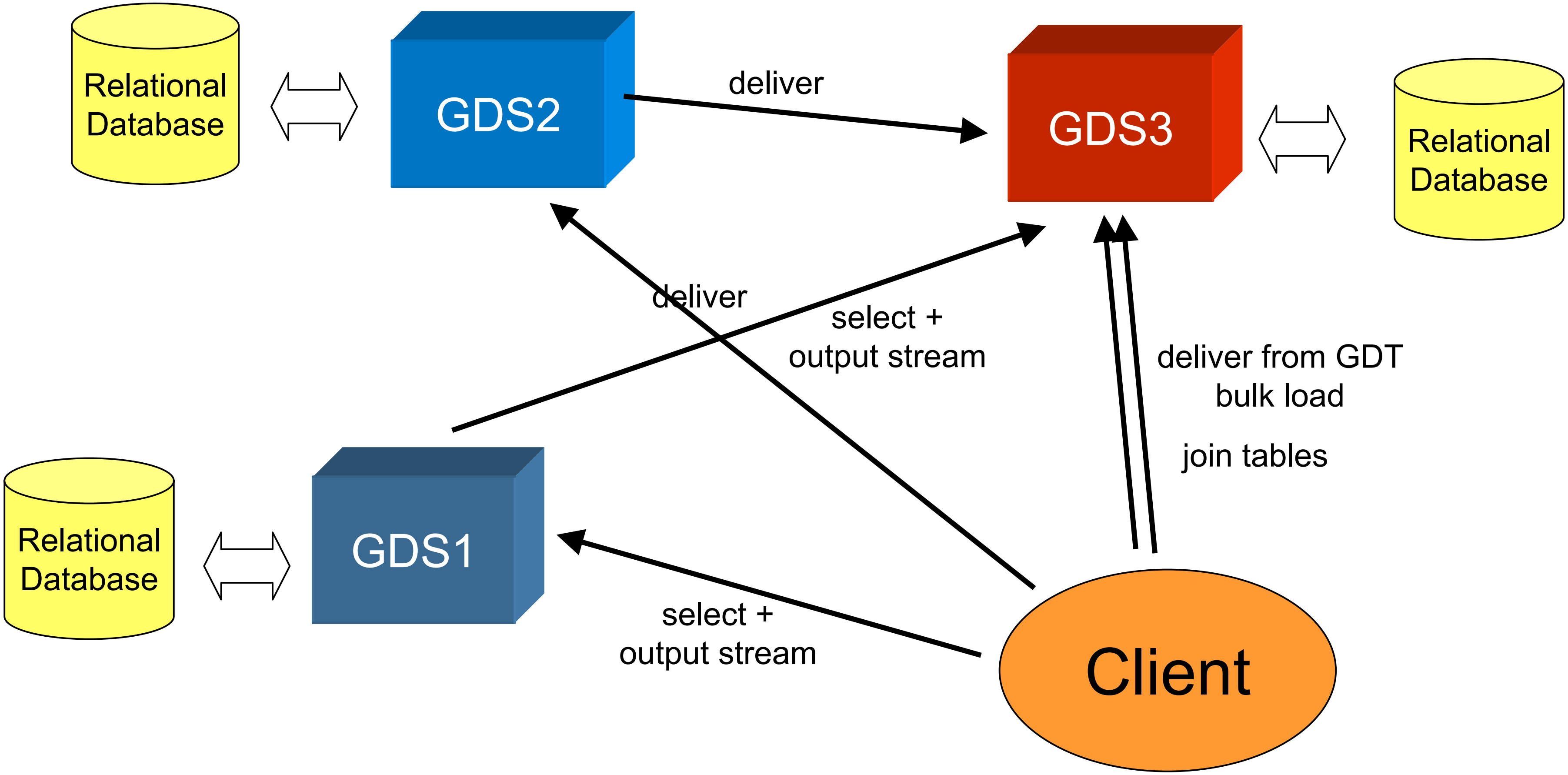
- The GDT port type allows to transfer data from one data service to another.
- Push: A *DeliverToGDT* activity of GDS1 connects to an *InputStream* activity of GDS2
- Pull: Alternatively, an *OutputStream* activity can be connected to a *DeliverFromGDT* activity



Delivering Data

- Transfer in blocks or in full
- *InputStream* activities wait for data to arrive at their input
  - Therefore, the *InputStream* activity at the sink has to be started before the *DeliverToGDT* activity at the source
- *OutputStream* activity waits for data to be read from its output
  - *OutputStream* activity at the source must be started before *DeliverFromGDT* at the sink

Data Integration Scenario



Conclusion

- Easier to use than the perform document
  - Higher-level APIs
  - Improves usability and shortens learning curve for client development
- Protects developer
  - Shielded from schema changes, protocols
  - Deprecation policy needed
- Limitations
  - Metadata and service-data not yet addressed adequately
  - Higher-level abstraction possible (no factory)

OGSA-DAI Future Work and Wrap Up

Roadmap / Workplan

- Roadmap document available for comment:
  - http://www.ogsadai.org.uk/docs/OtherDocs/OGSA-DAIRoadmapV2.0.pdf
  - User feedback required to drive this document
- Integrate parts of DQP into OGSA-DAI core
  - Addressing platform dependencies
  - Want to include XML data resources
- Move Computation to Data
  - Java mobile code?

WS-I Technical Preview

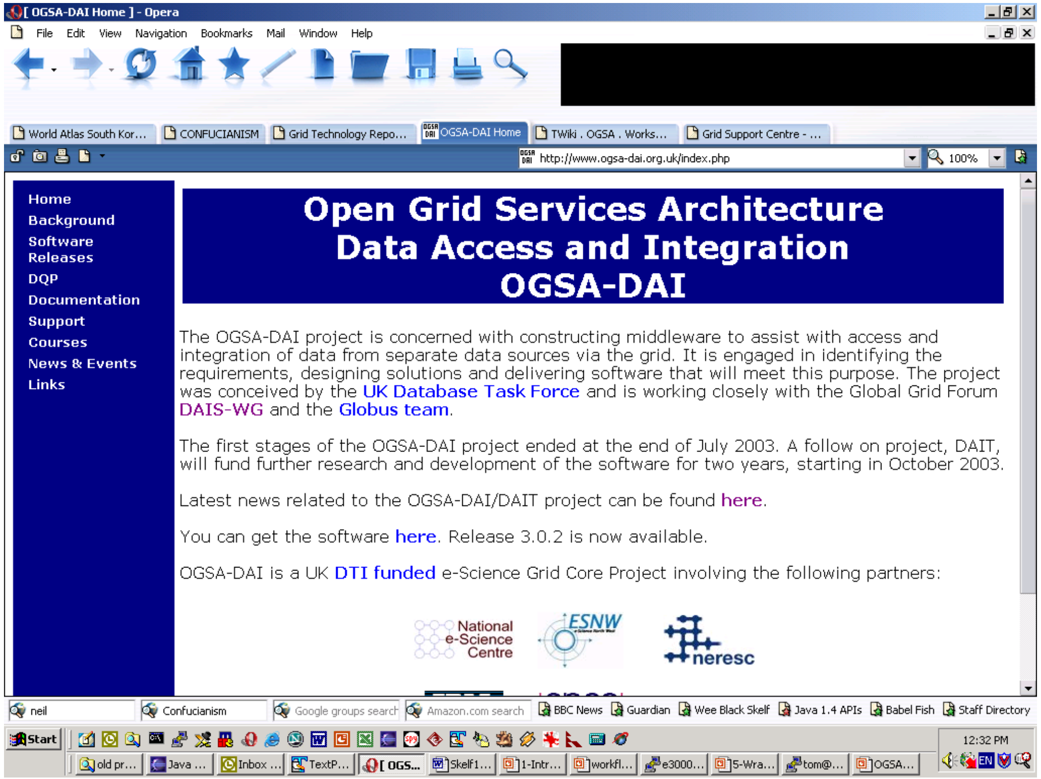
- A limited functionality evaluation version
  - An OGSA-DAI “Data Service” combining the metadata, configuration and perform document capabilities of the OGSI-based GDSF and GDS services.
  - Access to service metadata provided by a partial implementation of the WS-ResourceProperties specification.
  - Example clients are for testing and coding reference.
- Caveats/Issues:
  - No registry component, no support for 3rd party delivery.
  - Security soon (based on OMII WS-Security plug-in).
  - Document schema and interfaces WILL change.
  - The WSDL is based on the OGSI-based WSDL from OGSA-DAI
- Also works with OMII middleware distribution

WS-RF Technical Preview

- An evaluation version OGSA-DAI based on the Globus Toolkit 4.0 beta implementation of WSRF.
  - Provides an amalgamation of the capabilities of the OGSI-based GDSF and GDS services
  - Access to multiple data resources from a single service provided by data resource identifiers specified by a client within the WS-Addressing endpoint reference to a data service.
  - Access to service metadata (database schemas, request status, etc) provided by an implementation of the WS-ResourceProperties specification.
  - A WSRF version of the GridDataTransport portType supporting asynchronous data delivery between data services.
- Caveats/Issues:
  - This preview of OGSA-DAI WSRF does not support data service security.
  - Document schema and interfaces WILL change.
  - Will not be supported to same level as main release.
- Will be released as part of the Globus Toolkit 4 beta (3.9.x)

OGSA-DAI Project Webpage

- http://www.ogsadai.org.uk



[redacted]

Home
Background
Software Releases
DQP
Documentation
Support
Courses
News & Events
Links

**Open Grid Services Architecture
Data Access and Integration
OGSA-DAI**

The OGSA-DAI project is concerned with constructing middleware to assist with access and integration of data from separate data sources via the grid. It is engaged in identifying the requirements, designing solutions and delivering software that will meet this purpose. The project was conceived by the UK Database Task Force and is working closely with the Global Grid Forum DAIS-WG and the Globus team.

The first stages of the OGSA-DAI project ended at the end of July 2003. A follow on project, DAIT, will fund further research and development of the software for two years, starting in October 2003.

Latest news related to the OGSA-DAI/DAIT project can be found here.

You can get the software here. Release 3.0.2 is now available.

OGSA-DAI is a UK DTI funded e-Science Grid Core Project involving the following partners:

Background

News & Events

Software Releases

Documentation

On-line Tutorials

Support

Training Courses

Links

OGSA-DAI Users Group

- User Group Chair
  - Prof. Beth Plale, Indiana University
- A separate independent body to engage with users and feedback to developers in a formal way
- Held meetings in Edinburgh and Brussels in 2004
  - Presentations from projects using OGSA-DAI
  - Discussion of requirements and issues
  - Discussion of roadmap
- Meetings being arranged for 2005

- Contact Beth Plale (plale@cs.indiana.edu) for more details

FAQ, Support, Mailing List

- Frequently Asked Questions
  - http://www.ogsadai.org.uk/support/faq.php
  - Updated as common problems become clear
- Support for OGSA-DAI releases
  - http://www.ogsadai.org.uk/support
  - support@ogsadai.org.uk
  - Use to report problems
- Discussion list
  - users@ogsadai.org.uk
  - http://www.ogsadai.org.uk/support/list.php
  - General discussion of OGSA-DAI, data and the Grid

Conclusions

- Still early days
  - Standardisation process not stabilising quickly enough
  - Infrastructure still developing and prone to change
- OGSA-DAI acting as an enabler
  - Showing people what can be done
  - Evolving and improving with each release
- Usage patterns are similar
  - Call for people to work together to solve similar problems
  - Try to implement in core OGSA-DAI
- Some problems are not OGSA-DAI specific
  - Metadata, time zones, security, …
- Data discovery opens up a window of integration opportunity
  - Should we continue with registries ourselves?
- Please try it out!
  - It’s free and supported
  - Make suggestions, extend functionality, contribute to DAIS-WG