

# High-Level Middleware for Data Management in Grids

Alvaro A A Fernandes  
Department of Computer Science  
University of Manchester

---

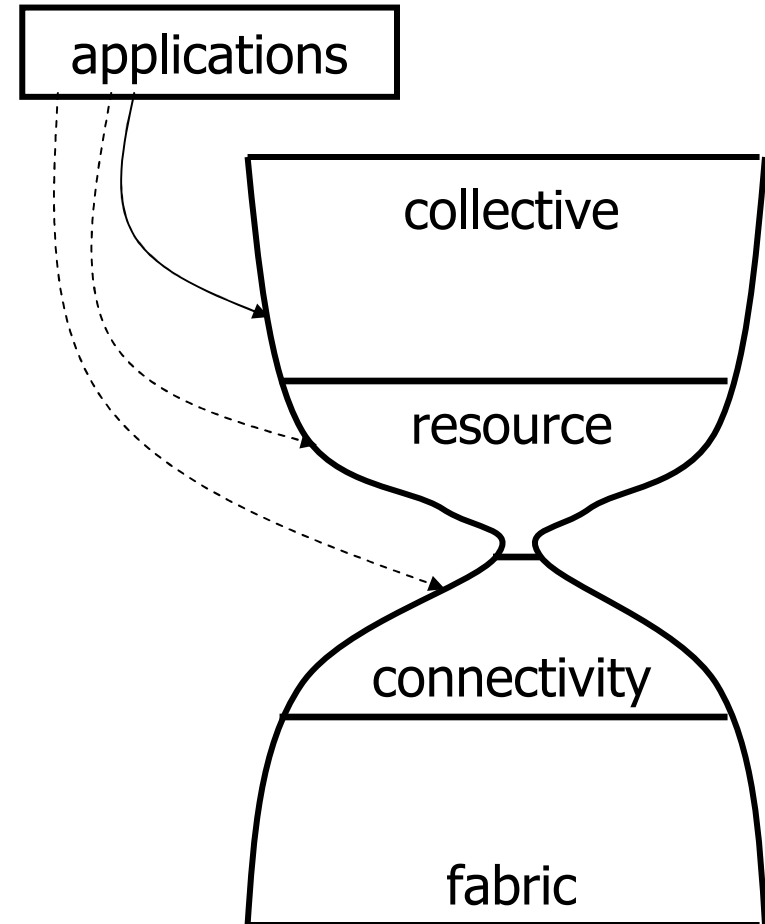
- What is a grid and what is grid middleware?
- What are the data management problems that arise in grids?
- What high-level grid middleware for data management is available and what can it do?

- Why grids?
- What are grids aimed at?
- What does a grid make possible?
- What are grids an infrastructure for?
- How are grids evolving?
- What is grid middleware?
- What is low and high level grid middleware?
- What is OGSA?

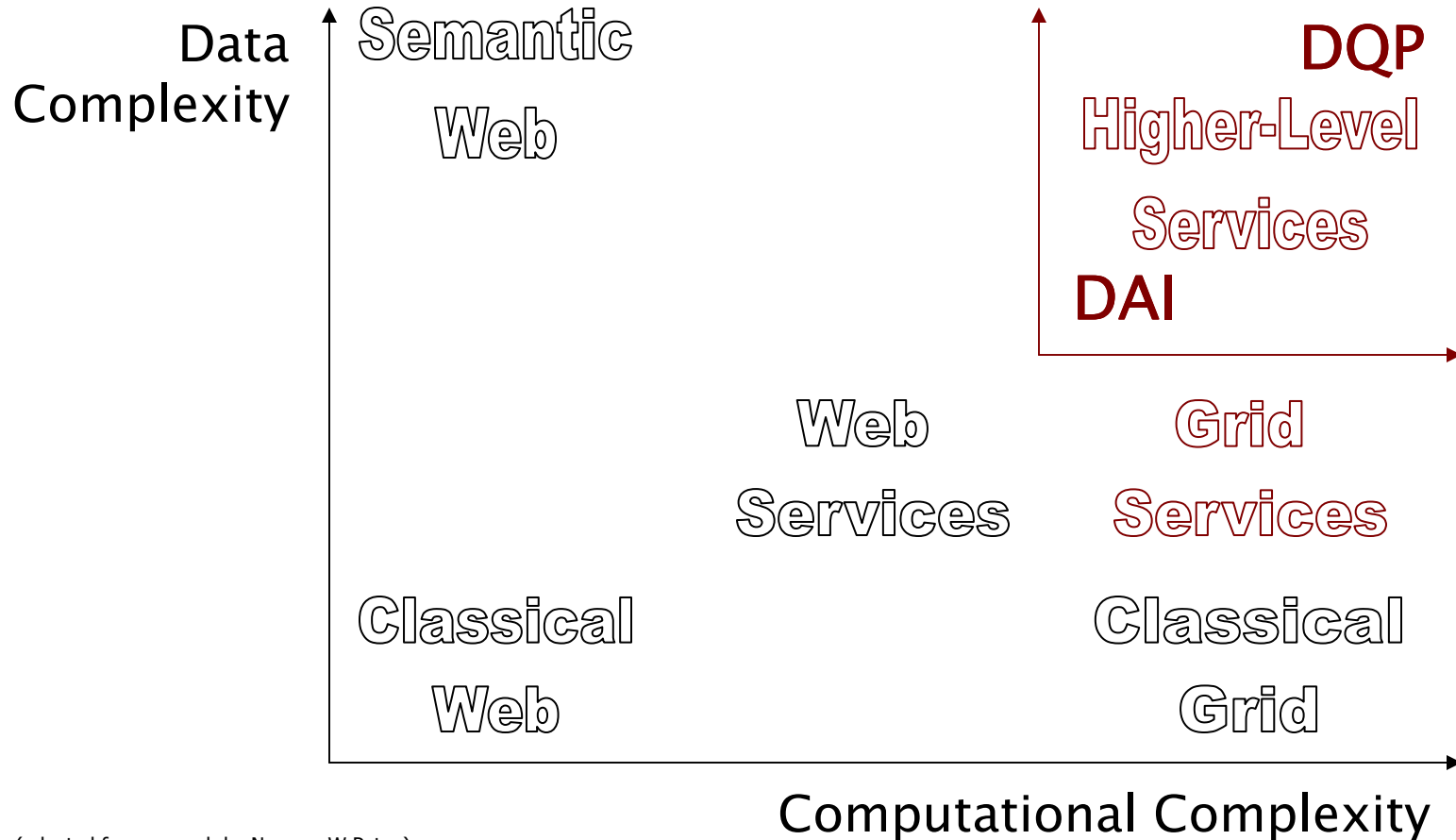
- advanced, complex patterns of business and scientific activity require
  - the dynamic, on-demand interaction of
    - people with
    - data and computing resources that are
  - geographically and organizationally dispersed
- a grid is aimed at
  - facilitating such interactions so as to
  - make them routine
  - (just as interaction with, e.g., power grids or telephone networks is)
- to support advanced, complex patterns of business and scientific activity

- a grid makes possible:
  - resource sharing and
  - coordinated problem solving in
  - dynamic, multi-institutional virtual organizations
- in spite of there being:
  - no central location,
  - no central control,
  - no pre-existing trust relationships,
  - minimal predetermination
- grids are an infrastructure for:
  - on-demand,
  - ubiquitous
  - access to data and computing services
- where:
  - new capabilities
  - can be constructed dynamically and transparently
  - from distributed services

- protocols
- organized in layers
- using an hourglass model
- the **neck**, i.e.,
  - resource protocols and
  - connectivity protocols
- facilitates the co-existence of
- a variety of
  - high-level (collective) protocols and
  - low-level (fabric) protocols
- delivering transparent discipline to grid applications



# how are grids evolving?

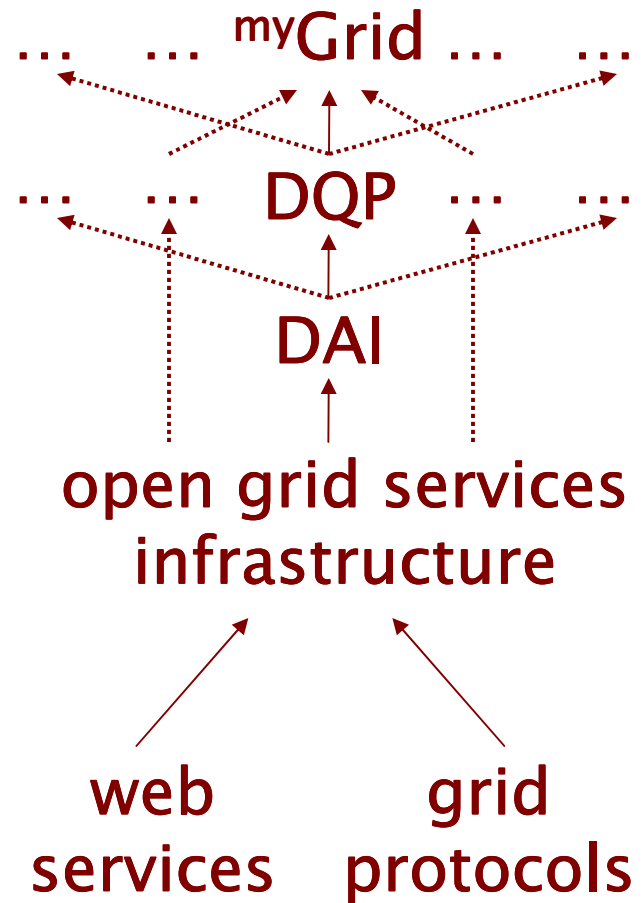


(adapted from a graph by Norman W Paton)

- software components that implement
  - (aspiring) standards,
  - protocols, and
  - bundled interaction patterns
- functionality that is convenient for domain-specific grid applications
- it lets domain-specific grid applications worry less about:
  - scale:
    - too many
    - too much
  - scatter:
    - too remote
    - too costly to access
  - scare:
    - too complex to understand
    - too daunting to use

# what is low and high level in grid middleware?

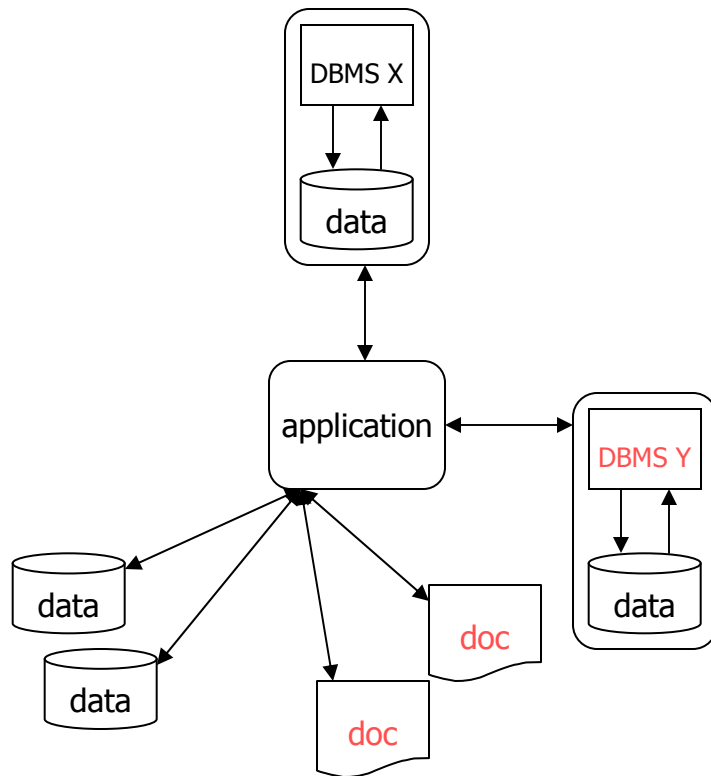
- classical grids ran as low-level middleware for
  - discovery
  - certification
  - allocation
  - sharing
  - transport, etc., of
  - primary resources
- latest grids use high-level middleware
  - service-based
  - value-added resources



- the Open Grid Service Architecture mirrors in a grid context the move towards a service-based view initiated by the web
- a service hides from grid user applications inessential details about a complex tools it needs
- there are also standards for describing, exposing and interacting with services
- grid services and web services are not yet fully compatible
- current status:
  - Globus Toolkit 3.2, released in March 2004
  - implements one approach to OGSA
  - is used by OGSA-DAI and OGSA-DQP
- current drive:
  - better integration with web services in GT 4

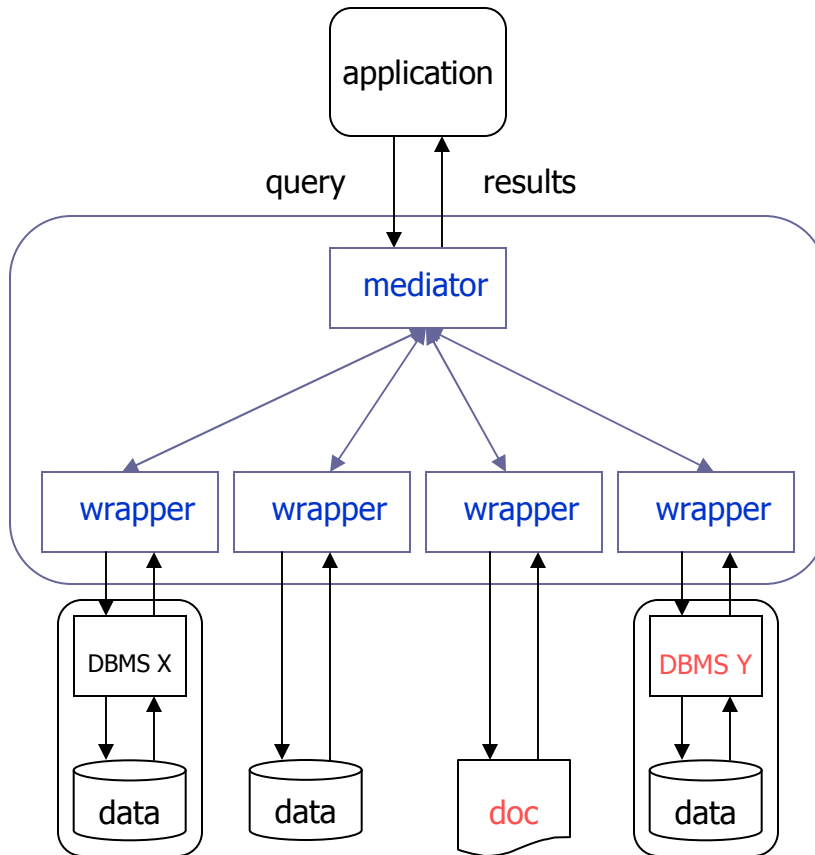
- How do scale, scatter and scare manifest themselves in data management?
- How can it be ameliorated?
- What are OGSA-DAI and OGSA-DQP?
- What are OGSA-DAI and OGSA-DQP for?

# how do scale, scatter and scare manifest themselves in data management?



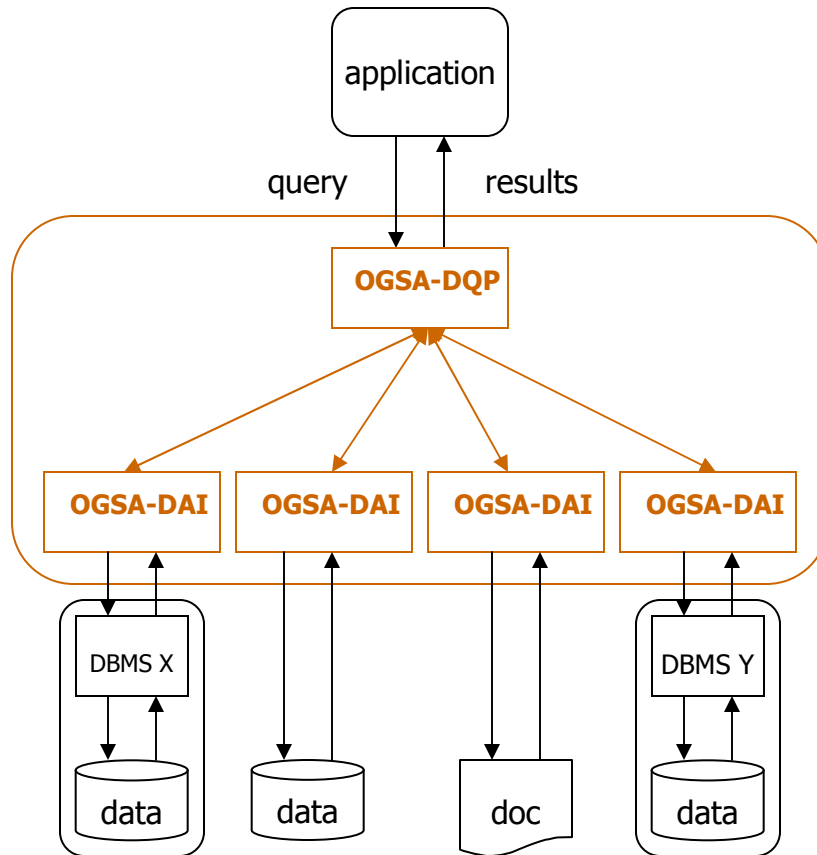
- distributed data resources lead to the risk of non-dependable applications
- multitude of data resources lead to the risk of non-scalable applications
- heterogeneous data and data environments lead to the risk of over-complicated applications
- the grid is bound to exacerbate these issues in comparison with the Web

# how can it be ameliorated?



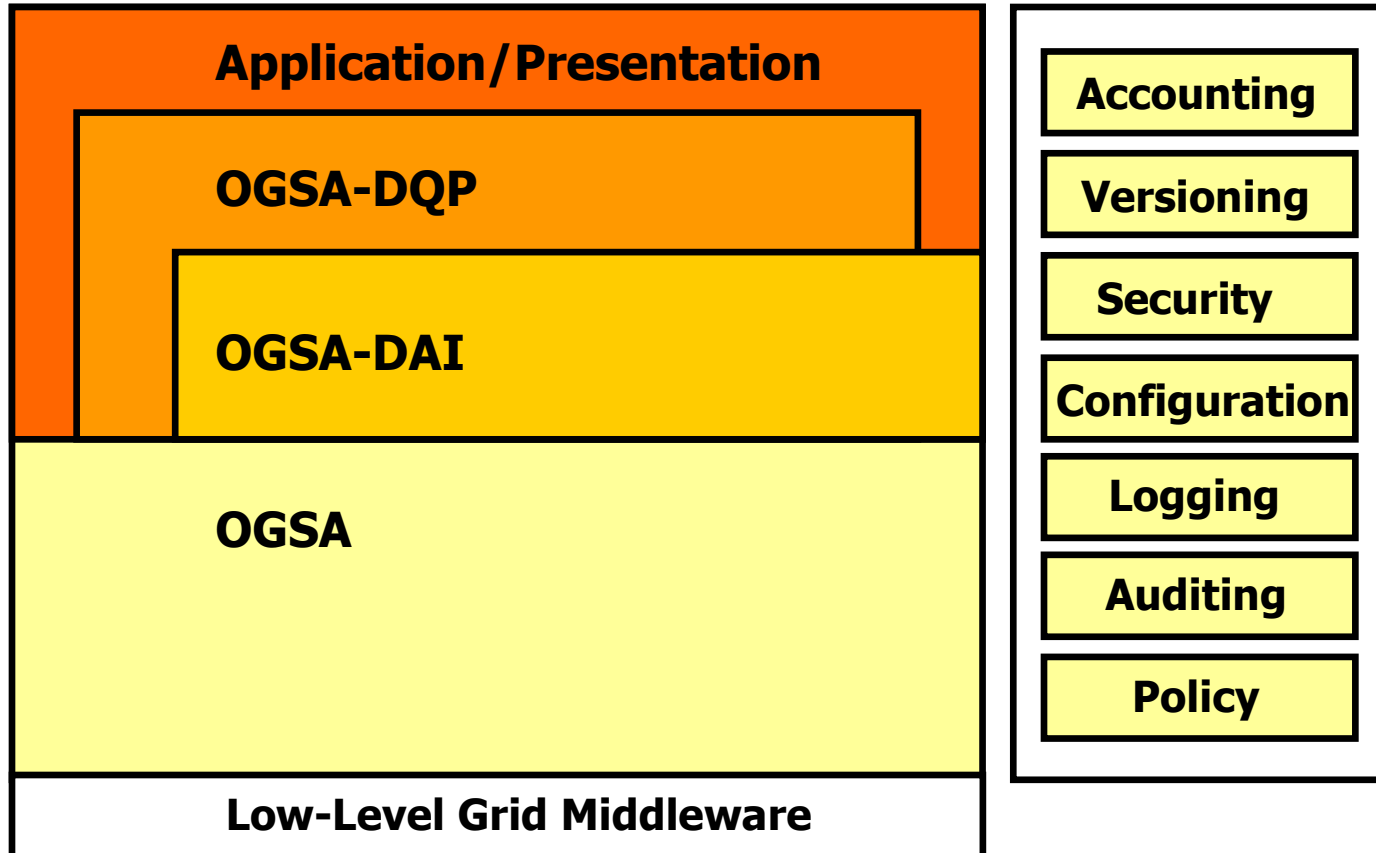
- one approach is to
  - use wrapper middleware to project a uniform view and reduce scare
  - use mediator middleware to reduce impact of scatter and cope with scale
- the effect is to make dependable, scalable, simpler data management more likely
- applications stand to reap the benefits

# what are OGSA-DAI and OGSA-DQP?



- OGSA-DAI and OGSA-DQP are high-level middleware for data management in grids
- OGSA-DQP plays a mediator role over OGSA-DAI-wrapped resources

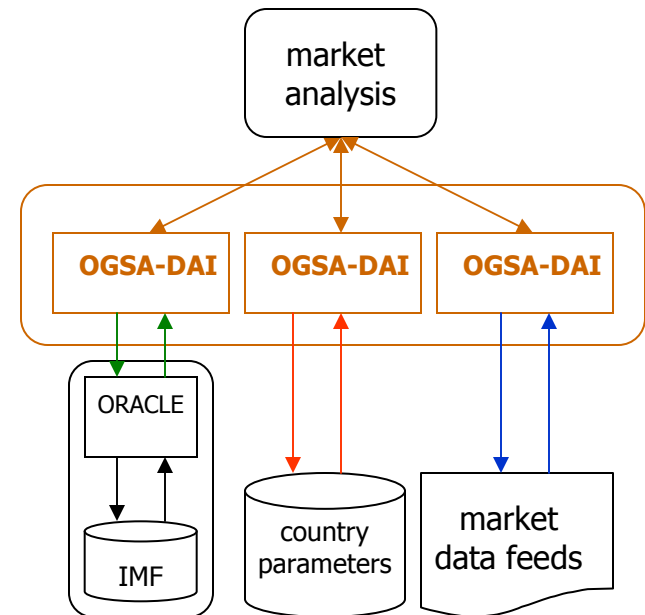
# how do OGSA-DAI and OGSA-DQP relate to other grid software?



- deploy it through integration with other Grid services
- provide standard interfaces
- the guiding principle is **virtualization**
- for all practical purposes, it is a global database as far as applications are concerned
- OGSA-DAI offers a bundle of interaction patterns that
- uniformly applies across heterogeneous data and heterogeneous data environments
- OGSA-DQP uses those to offer higher added-value interaction patterns

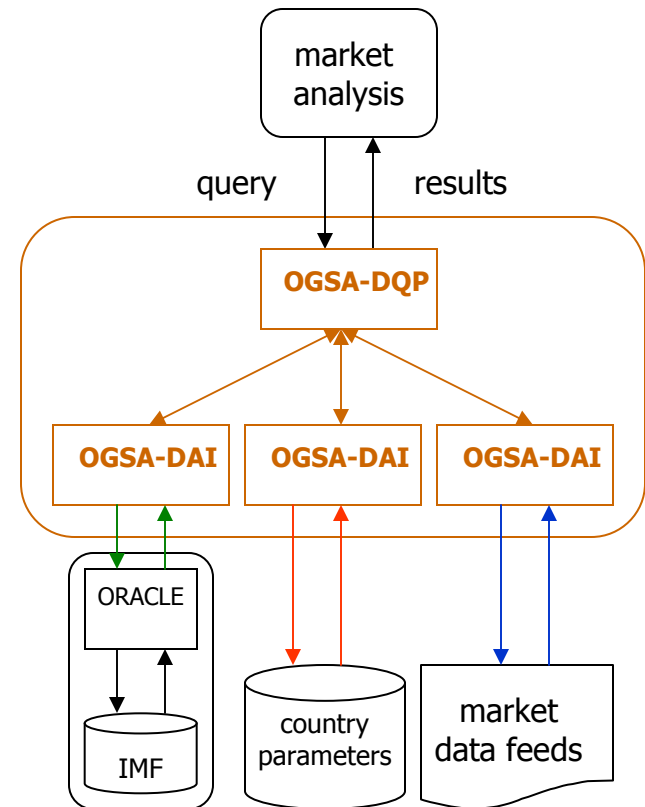
- OGSA-DAI is middleware to assist with access and integration of data from separate data sources via the grid
- the OGSA-DAI team has defined and developed generic Grid data services
- generic Grid data services are for
  - providing uniform means of access to data
  - held in relational database management systems, as well as
  - semi-structured data held in XML repositories
  - (flat files and unstructured data to come)
- there is also ELDAS, using J2EE and EJB

- OGSA-DAI can
  - perform queries, transformations and compression on the returned results
  - deliver those synchronously or asynchronously
  - perform updates and bulk loads
- it is a bundle of (query-transform-deliver) data access interaction patterns over
  - SQL/XPath relational/XML databases
  - flat files will be supported



- OGSA-DQP is middleware that uses OGSA-DAI to provide querying against global virtual databases
- the OGSA-DQP team defined and developed Grid distributed query services
- Grid distributed query services are for
  - efficiently querying distributed data resources
  - effectively providing a global view of many local resources
  - the ability to invoke services in queries opens the way for basing many data-centred applications on OGSA-DQP alone

- OGSA-DQP can issue a single ad-hoc query over remote OGSA-DAI wrapped data resources stored at multiple sites
- the locations of the data are transparent to the application from which the query is issued
- OGSA-DQP, not the application, is responsible for choosing the most efficient way of obtaining results



# can you give me a concrete example?

(from molecular biology)

```
select p.proteinId, Blast(p.sequence)
from protein p, proteinTerm t
where t.termId = 'GO:0005942' and
p.proteinId = t.proteinId
```

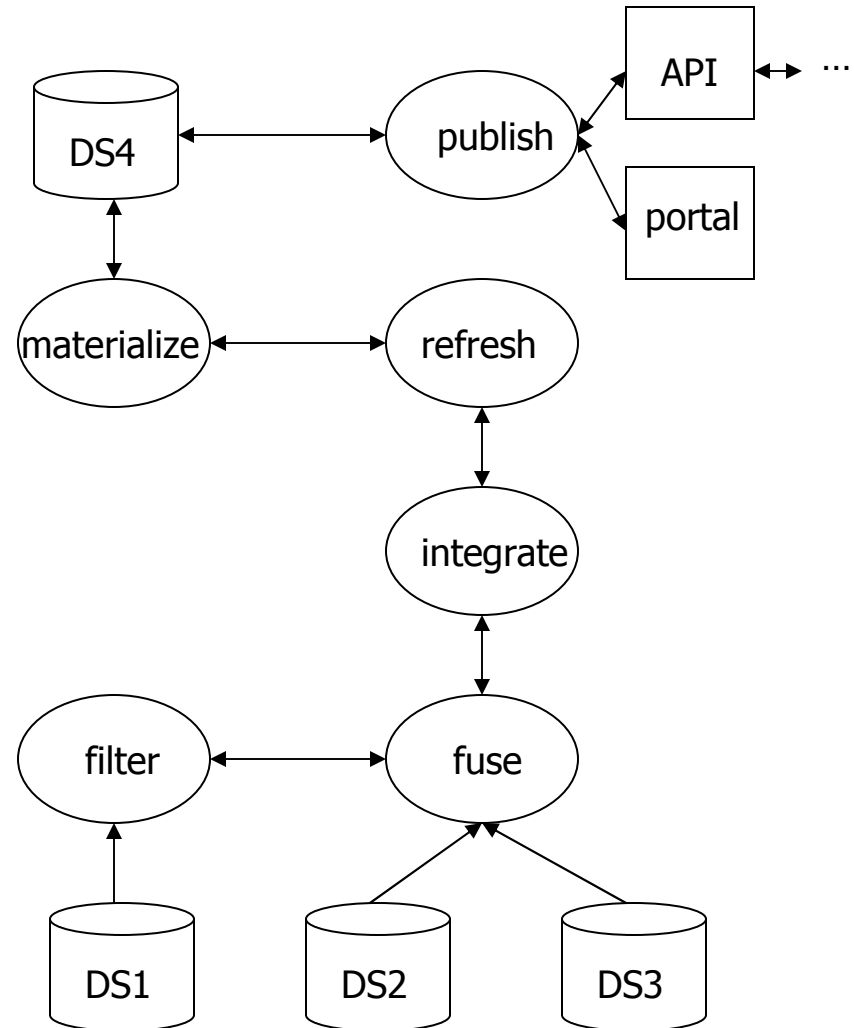
- take two distributed database queries:
  - **proteinTerm** to a GO Gene Ontology running as a mySQL DB
  - **protein** to a GIMS Genome Warehouse running as a Polar parallel ODMG-compliant DBMS
- and one external remote process exposed as service:
  - **Blast** (sequence alignment scoring)

- Which one? OGSA-DAI or OGSA-DQP?
- What can you do with them?
- What can you conclude?
- Where do you find out more?

## which one? OGSA-DAI or OGSA-DQP?

- with OGSA-DQP grid applications become
  - simpler
  - more reliable
  - more maintainablecompared to using OGSA-DAI alone
- i.e., higher-added value
- and also, more opportunities for further exploitation by grid applications
- OGSA-DQP queries are more flexible than OGSA-DAI requests
- they can be used to
  - define views on data
  - these views can be materialized, or not, on the client, and
  - exposed as services themselves
  - to participate in other applications or be exposed as portals

- queries have many uses:
  - filter
  - fuse
  - integrate
  - materialize
  - refresh
  - analyse
  - publish
- a query can be seen as a function
  - query: data → data
  - and so can be composed



## isn't this like a workflow?

- once you have queries, you can move on to compose them
- once you compose them, you have workflows
- if queries can (as does OGSA-DQP) retrieve from a web service, you have **declarative service orchestration**
- applications get for free:
  - declarativeness
    - what you want, not how to get it
  - optimization
    - the query processor pounces on resources, schedules on the fly
    - the effect is to choose a better execution plan than most people could
  - reliability
    - queries are tractable and terminate

## what can you conclude?

- high-level middleware for data management in grids is a concrete reality
- OGSA-DAI reduces barriers for making the most of distributed, autonomous, heterogeneous data resources
- it is already in use in the majority of e-Social Science demonstrator projects
- OGSA-DQP offers higher added-value still
- declarative service orchestration
- if handcrafting access paths to data sets is all you do, OGSA-DQP(++) may well be all you need
- this combination should yield significant benefits in terms of development productivity and run-time performance for large-scale data-intensive grid applications
- think of OGSA-DAI and OGSA-DQP as very versatile building blocks for use in your own domain-specific, data-intensive grid applications
- what does the future hold?
  - OGSA-DAI is likely to go both GT4 and WS-I
  - OGSA-DQP will track the former, but will require effort to track the latter

where do you find out more?

<http://www.ogsadai.org.uk/>



**OGSA – DAI**

Open Grid Services Architecture – Data Access and Integration



**EPSRC**



**epcc**



**ORACLE**