Data Streaming Patterns for Distributed Data Exploitation

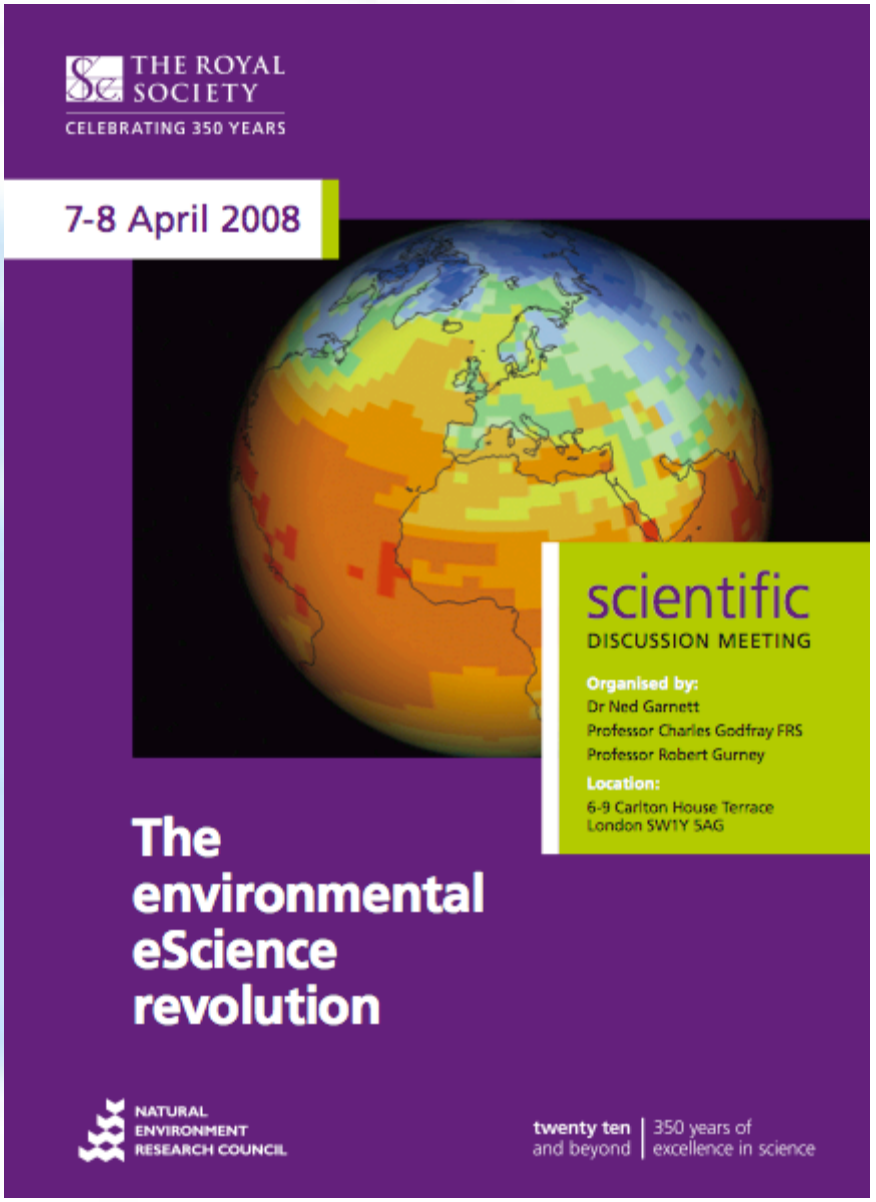
Data-Aware Distributed Computing Workshop

Boston, MA, USA

Malcolm Atkinson
Director e-Science Institute
UK e-Science Envoy

www.esi.ac.uk
24th June 2008

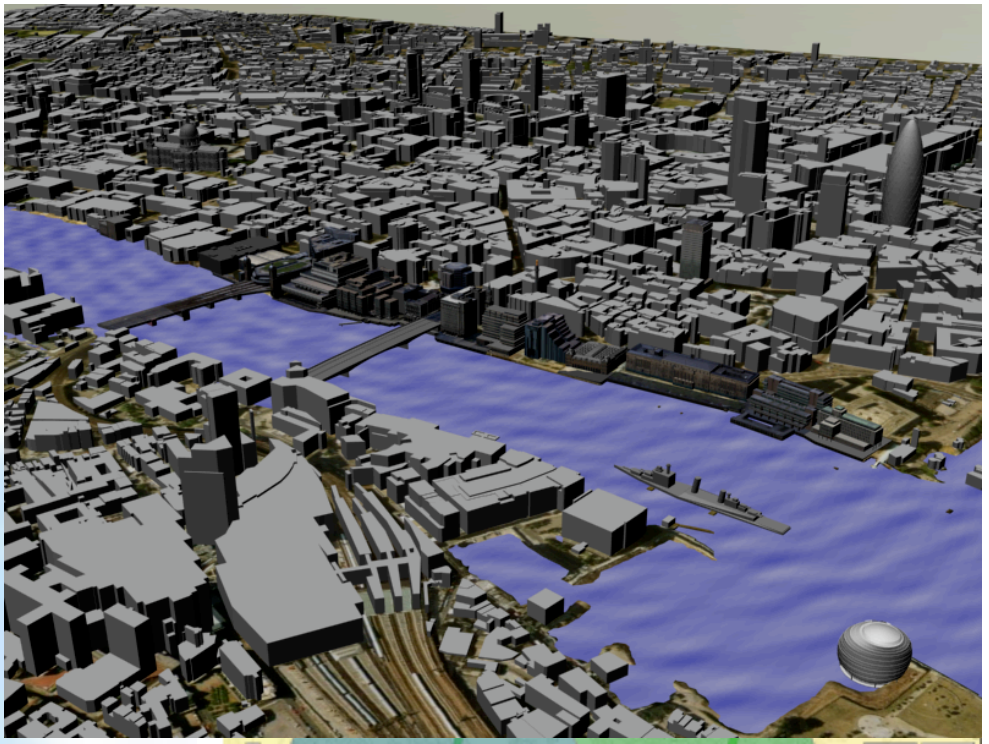
Data-Driven Science & Policy

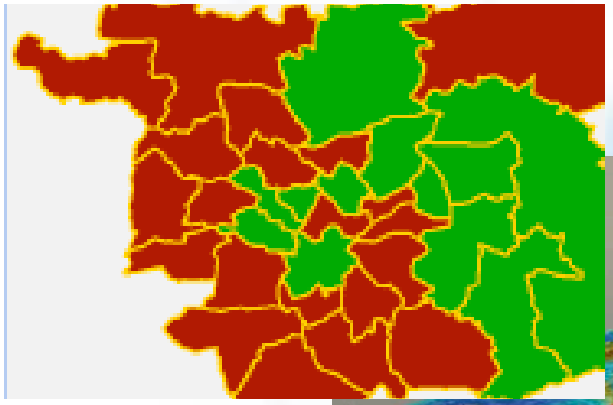


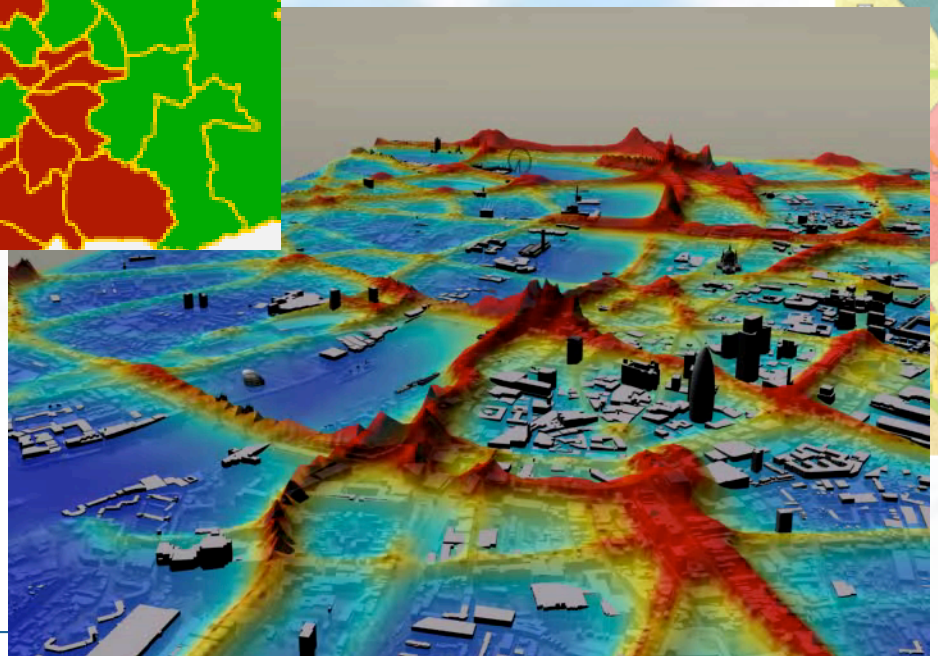
The 21st Century

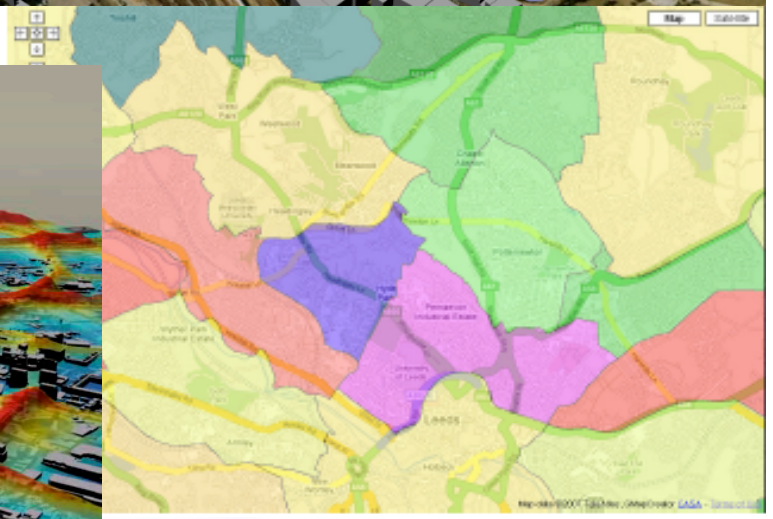
This is the century of information

Prime Minister Gordon Brown, University of Westminster, 25 October 2007





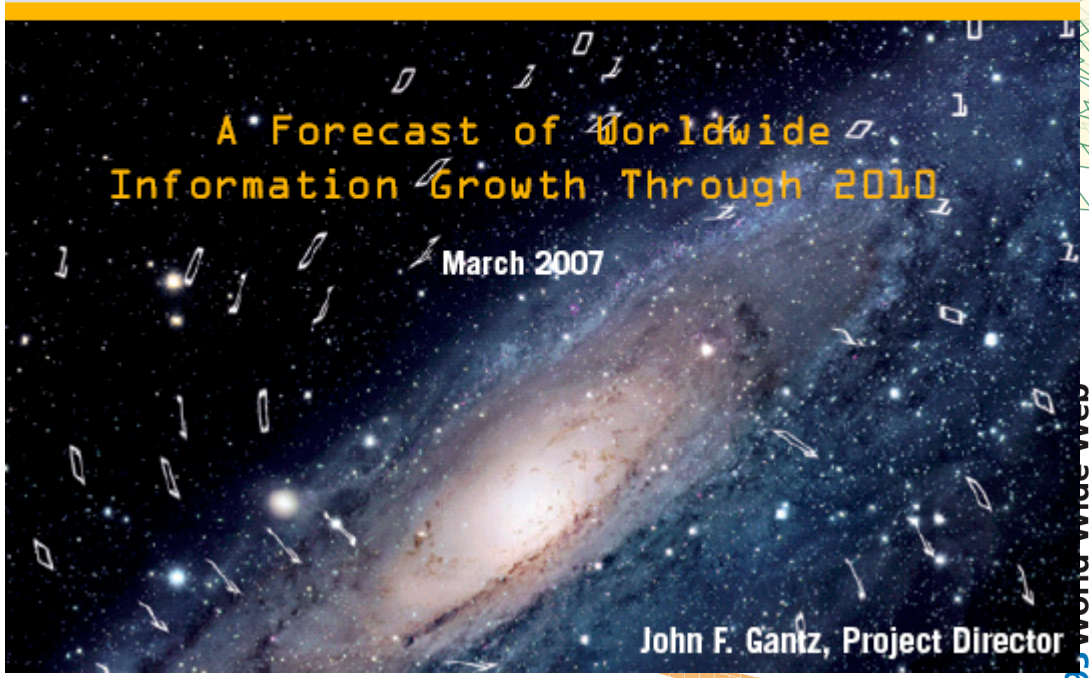




Thanks to Mark Birkin (MoSeS project) and Michael Batty (GeoVue project) for images

The Information Explosion

The Expanding Digital Universe



988EB = 1ZB
(2010)

161EB
(2006 by IDC)

billion GB
03

billion GB
02

.5 billion GB
01

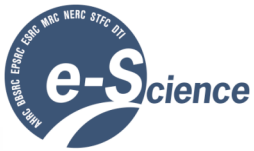
2 billion GB
00

1970
1980
1990
2000

Source: Horison Information Strategies,
cited from Storage New Game New Rules. p.34
(www.horison.com)

Slide: Satoshi Matsuoka

Data Streaming Hypothesis



There are a significant and growing number of data integration and exploitation applications that will benefit from decomposition into computational components coupled via data streams

Outline

- Hypothesis
- Context, History & Motivation
- Data-Aware Distributed Computational Patterns
- Multi-level Composition
- Streams: Content and Structure
- Notational and Functional Requirements
- Architectural Ideas
- Experience and Experiments
- Conclusions and Plans

Context: Traditional Distributed Data Strategies

1. Distributed applications communicating via a shared database
   - Including DB-supplier Distributed DBs
2. Large-scale file repositories
   - Including distributed file systems, replication, …
3. Compositions of the above
4. Data warehouses
5. Virtual data warehouses
6. Data steaming from instruments and Tx systems

These are the data resources that form the context for data-aware computing

Context: Autonomy, Diversity & Economy

1. *Independently* created and curated data resources
   - Individual, Organisational, National & International
   - Clusters of experts - ownership, skill & pride
   - Observational, Simulational, Transactional, Analytical & Synthetical
   - >50 ISO data standards for Environmental Data alone
2. *Growing number* of data resources
3. *Growing scale & complexity* of each data resource
4. *Independently evolving* data organisation at each resource
5. *Combinatorial explosion* of integration & exploitation opportunities

Complexity & rate of change limits conventional *maintained* integrations

Motivation for Streaming

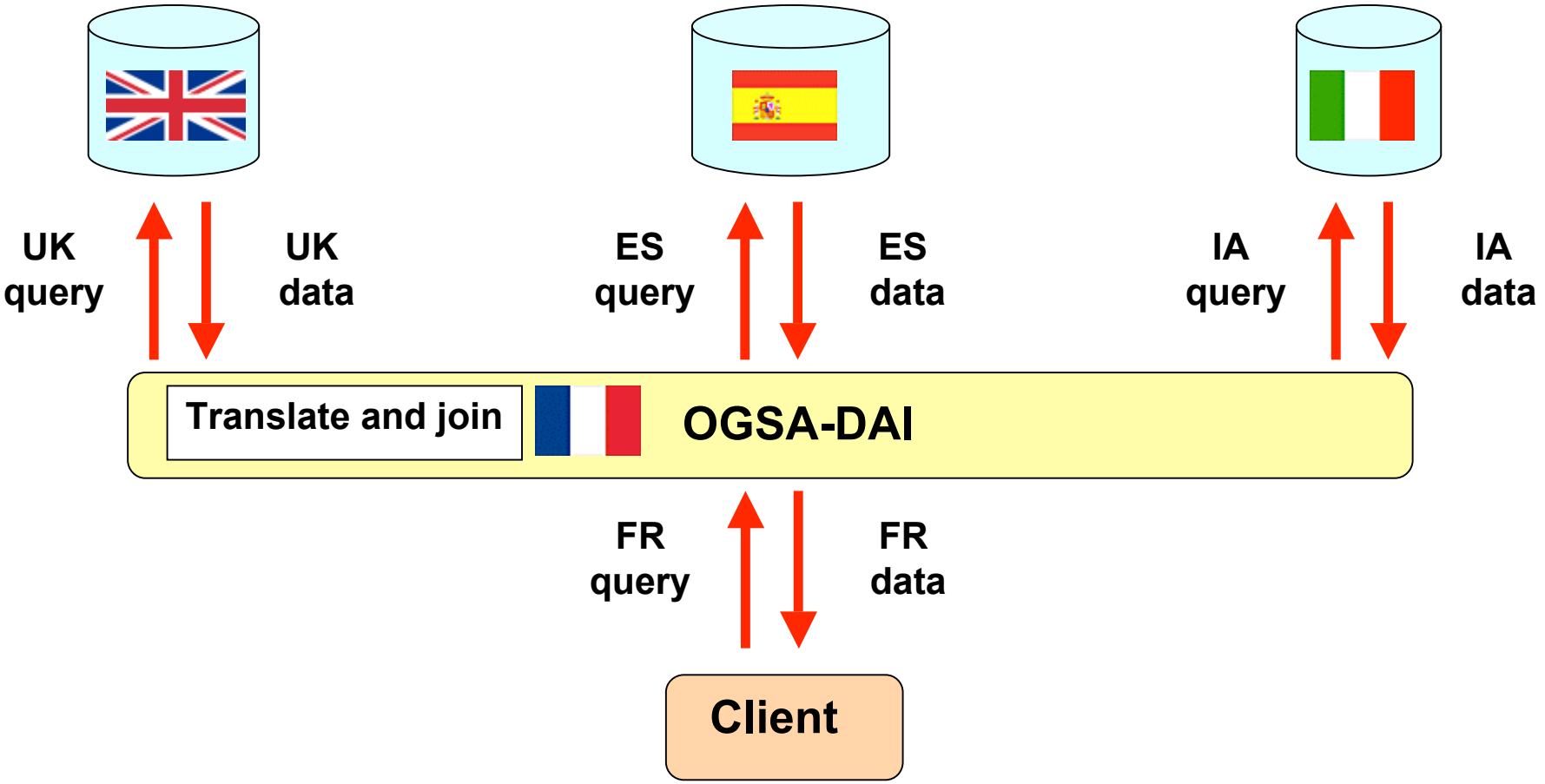
- Data movement is expensive
  - Bandwidth on and off chip may be scarcest resource
- Streaming can avoid data movement
  - Eliminating transfers to and from temporary stores
  - Pushing selectivity and derivation towards data sources
  - Earlier computation termination decisions
- Streaming can reduce computation
  - Pipelines of transformations overlap computation time
  - When co-located can pass on data via caches
- Streaming is scalable
  - Avoids locally assembling complete data sets
  - Sometimes this cannot be avoided
- Some data sources and consumers inherently streamed
- Permits light-weight composition and requires optimisation

Can we exploit streaming to deal *more* effectively with autonomous evolution?

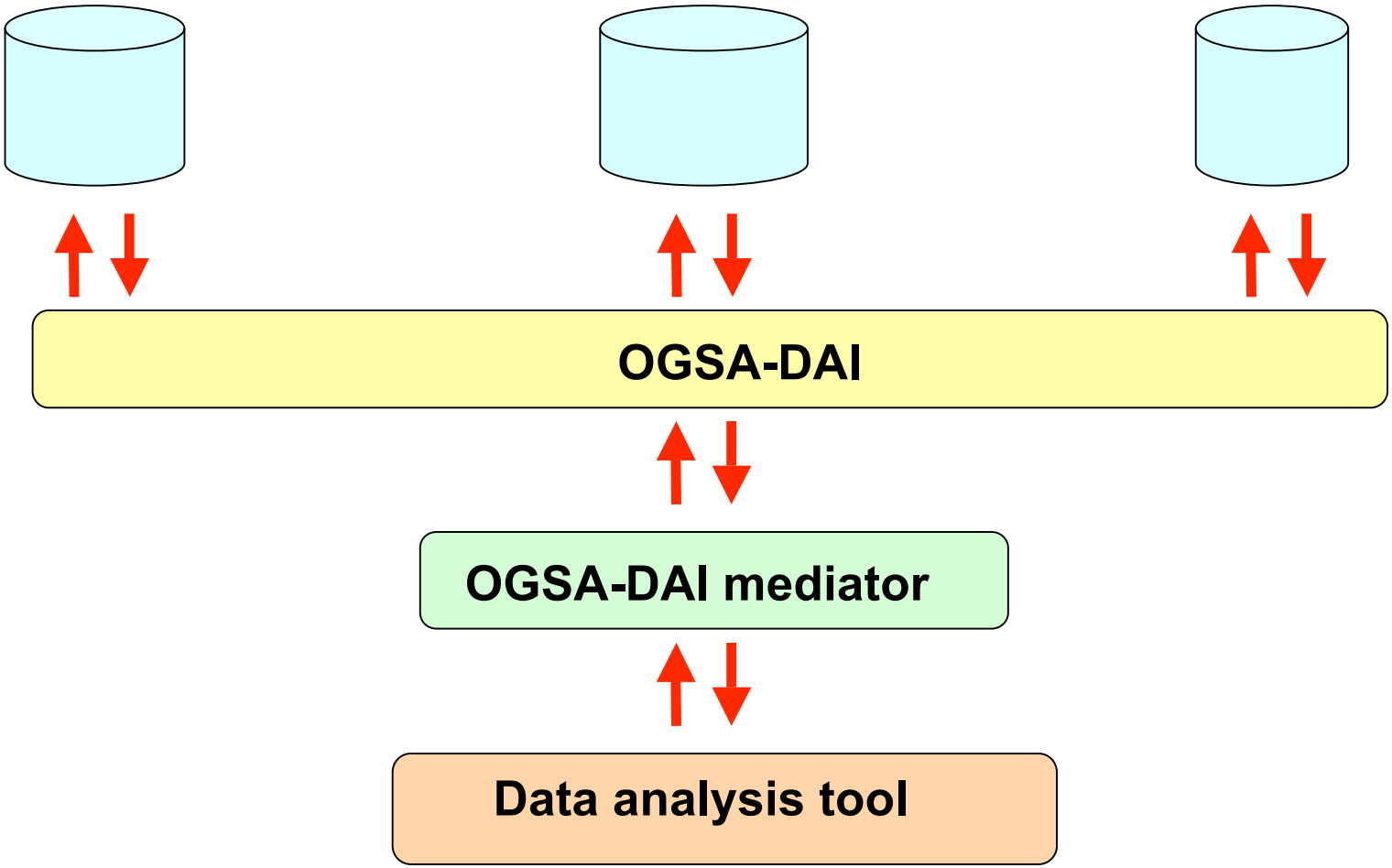
OGSA-DAI project

- Started February 2002
- Current partners
  - EPCC, The University of Edinburgh
  - National e-Science Centre
- OMII-UK
  - OMII, The University of Southampton
  - myGrid, The University of Manchester
  - OGSA-DAI, The University of Edinburgh
  - Funded by UK EPRSC – Engineering and Physical Sciences Research Council
- OMII-UK vision
  - Provide and support free, open source software to enable a sustained future for the UK e-Research community

Sharing distributed heterogeneous resources with OGSA-DAI



Grid-enabling existing data-related products



Security

- Authentication and authorization
- Access to
  - Service
  - Operation
  - Resource
  - Activity
  - Database
  - Table, column, file, XML document, XML element,…
- Read-only or read-write
- Role-based access

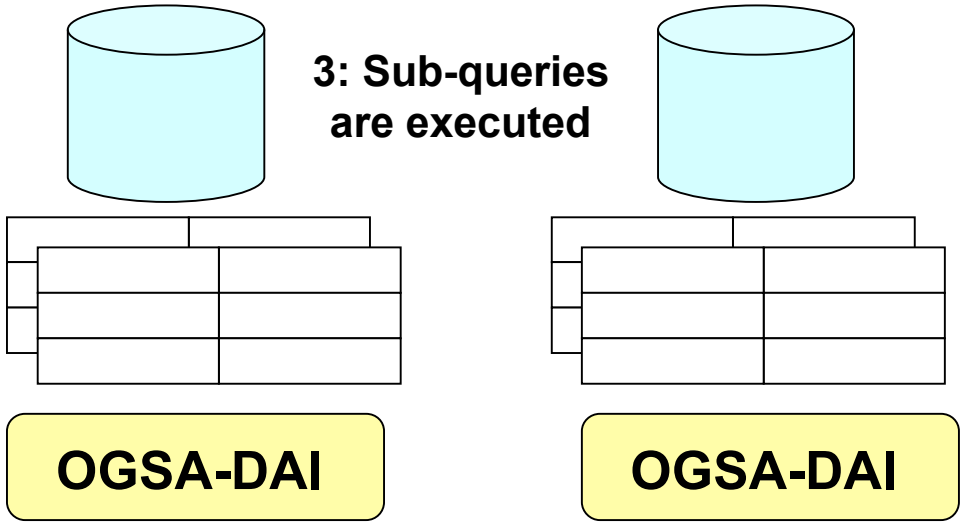
Extending OGSA-DAI

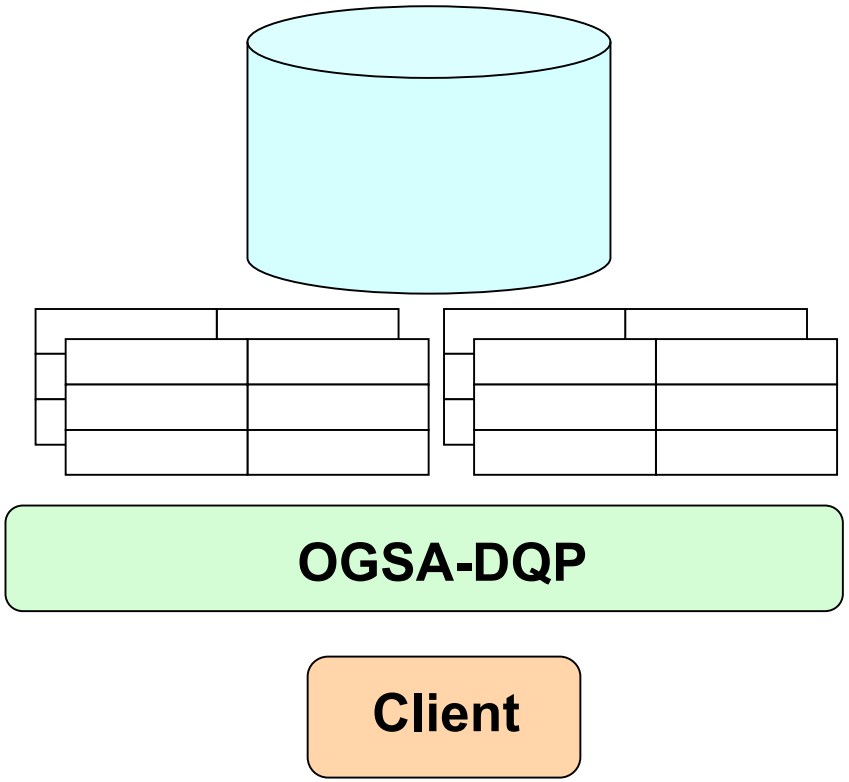
- OGSA-DAI is a framework
- Typically customised for application-specific requirements
- Extensibility points
  - Functionality – activities
  - Data – data resource plug-ins
  - Security
    - Database logins – login providers
    - Access to resources – resource authorizers
    - Access to activities – activity authorizers
    - Presentation layer security protocols – security contexts
  - Presentation layers

OGSA-DQP

- OGSA-DQP utilises OGSA-DAI
  - Developed by Universities of Manchester and Newcastle
  - Rewritten for OGSA-DAI 3.0 by EPCC as part of the NextGrid project
- Distributed query processing
  - Multiple tables on multiple databases are exposed to clients as multiple tables in one “virtual database”
  - Databases can be exposed
    - EITHER within one OGSA-DAI server
    - OR via multiple remote OGSA-DAI servers
- Clients are unaware of the multiple databases

OGSA-DQP – what it does and the client's view



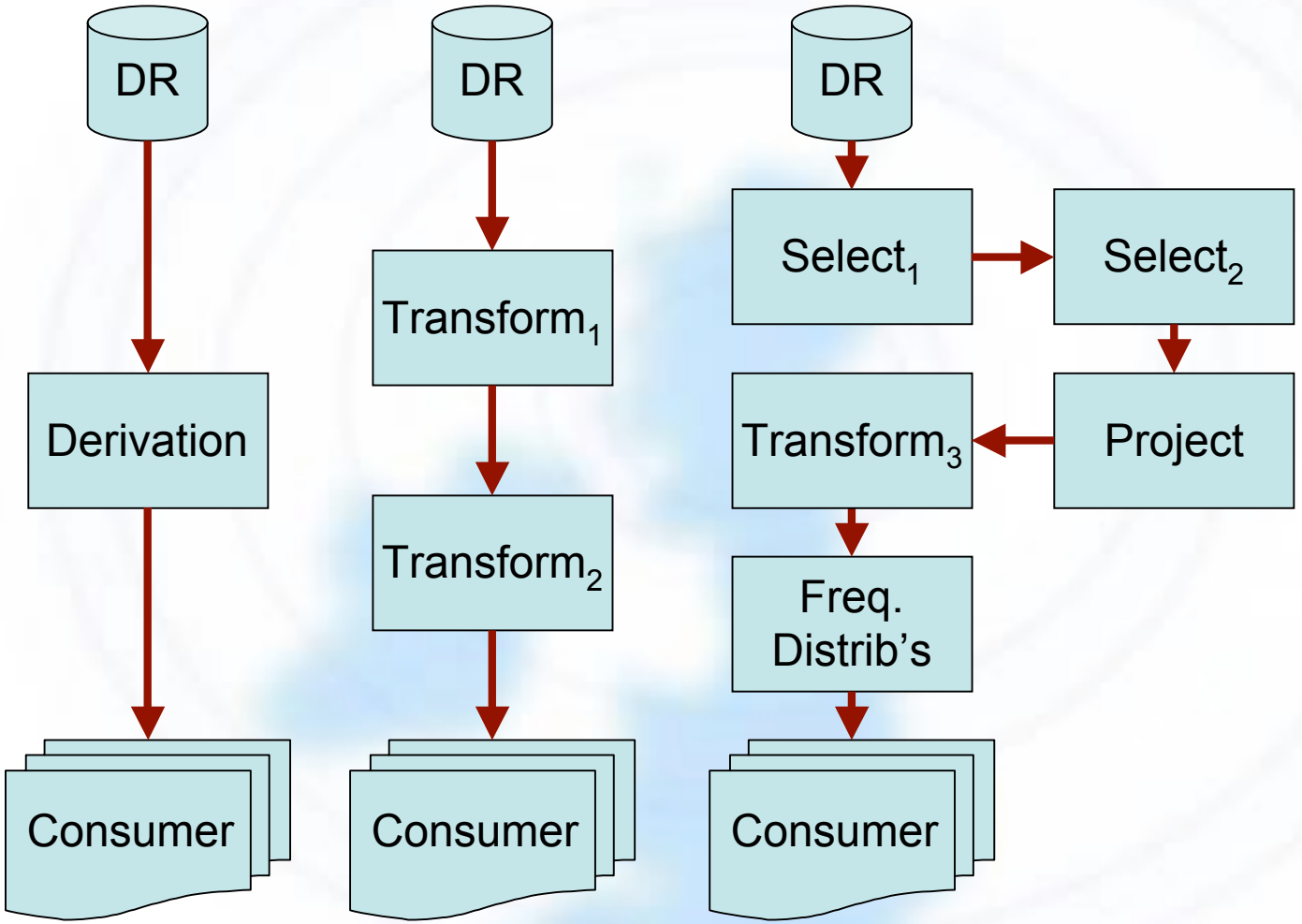


The client's view

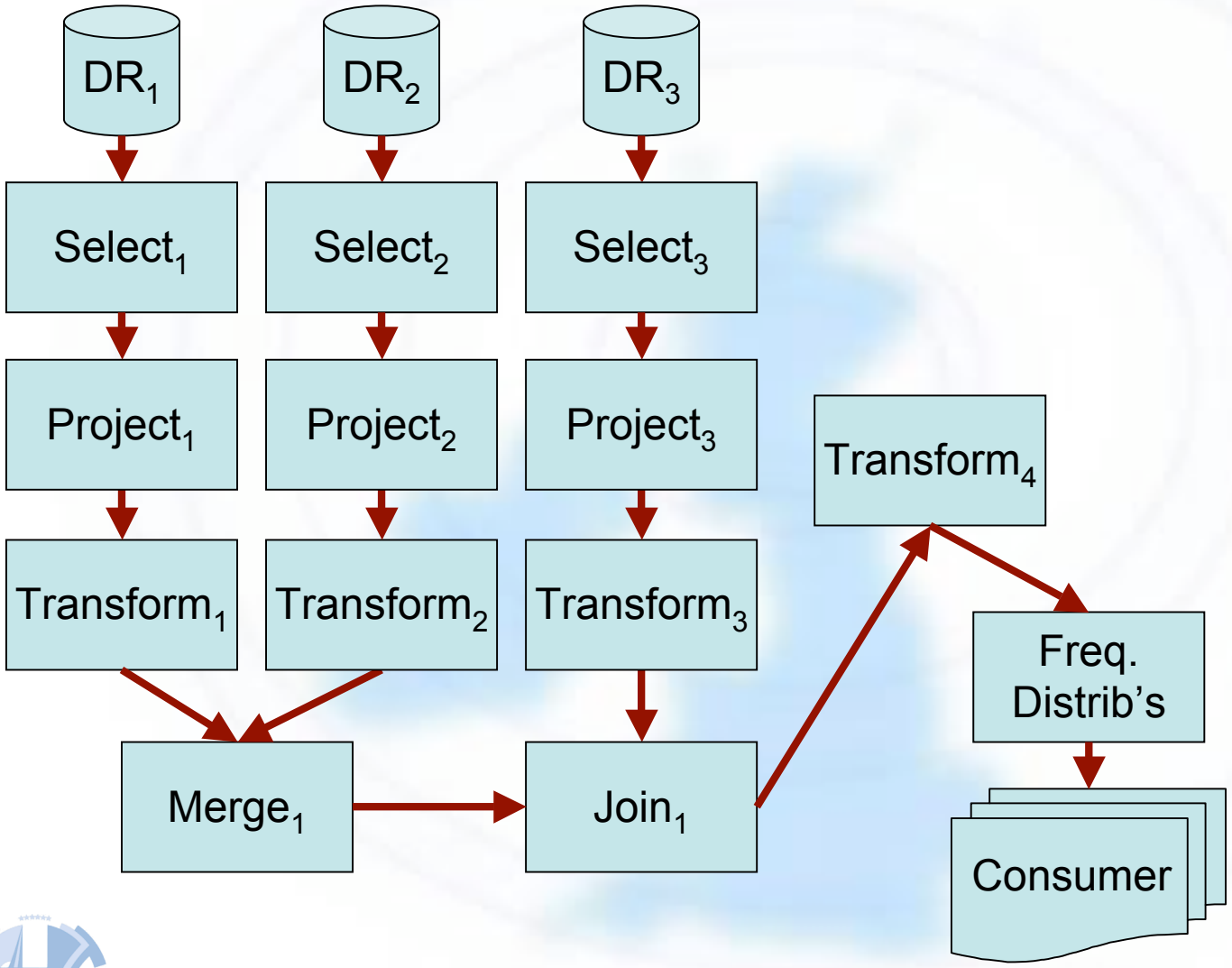
Outline

- Hypothesis
- Context, History & Motivation
- **Data-Aware Distributed Computational Patterns**
- **Multi-level Composition**
- **Streams: Content and Structure**
- **Notational and Functional Requirements**
- **Architectural Ideas**
- **Experience and Experiments**
- **Conclusions and Plans**

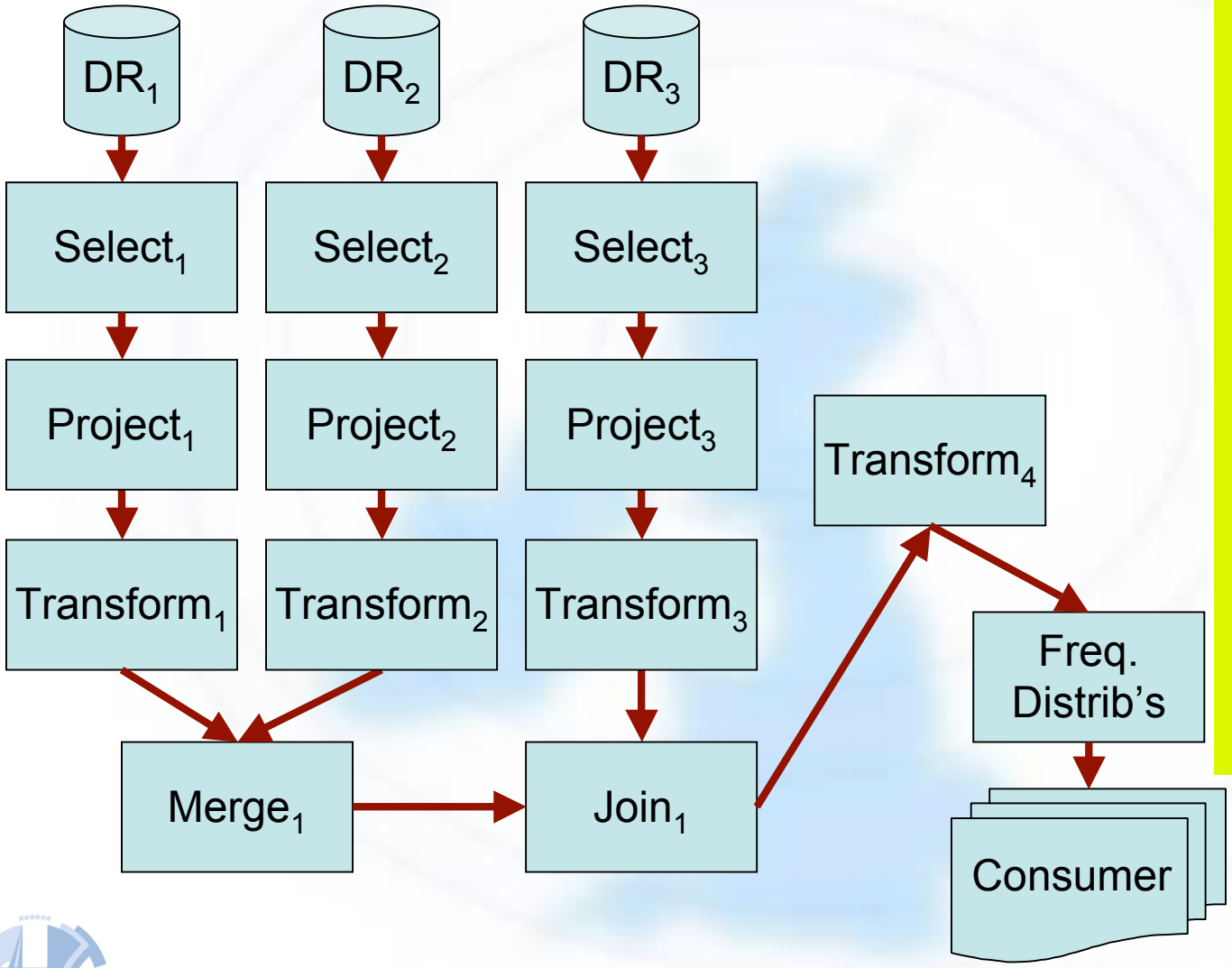
Simple Access Pattern



Accessing Multiple Resources

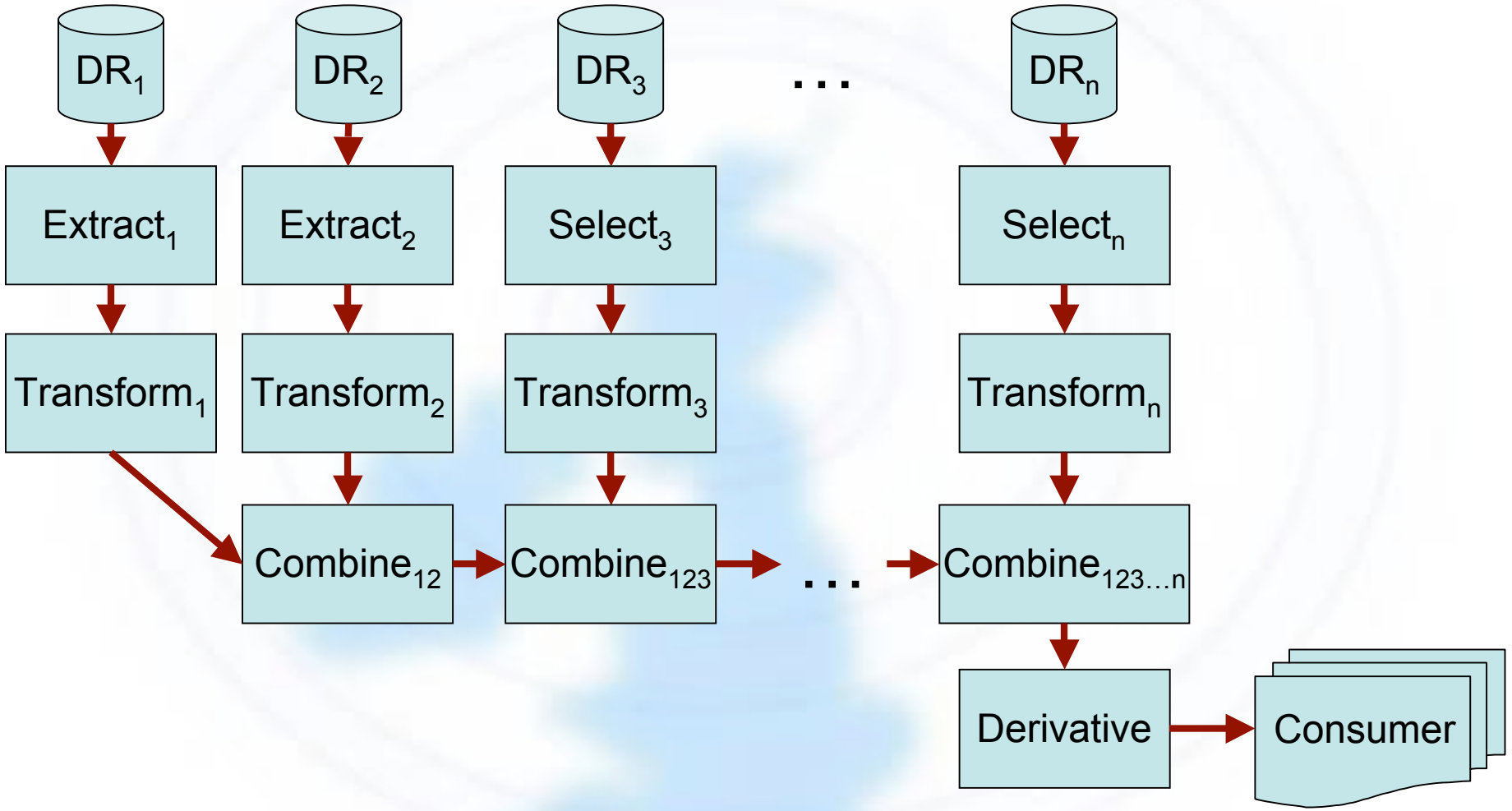


Accessing Multiple Resources

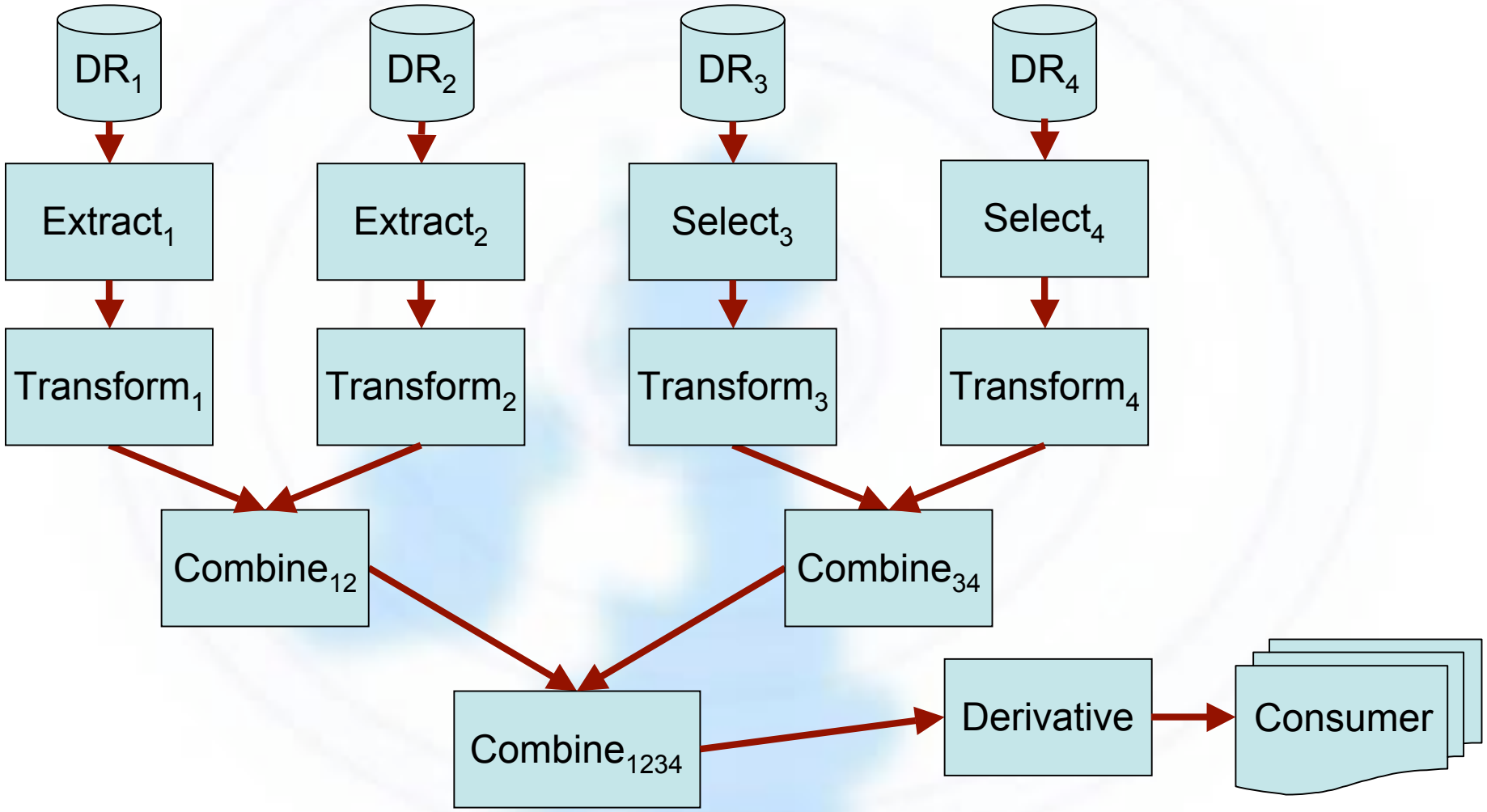


- Can partition across sites in many ways
- Can push select / project towards / onto data source
- Can co-locate proxies if source will not host
- Can combine or split computational activities
- Computational activities may be parameterised with multiple instances in use

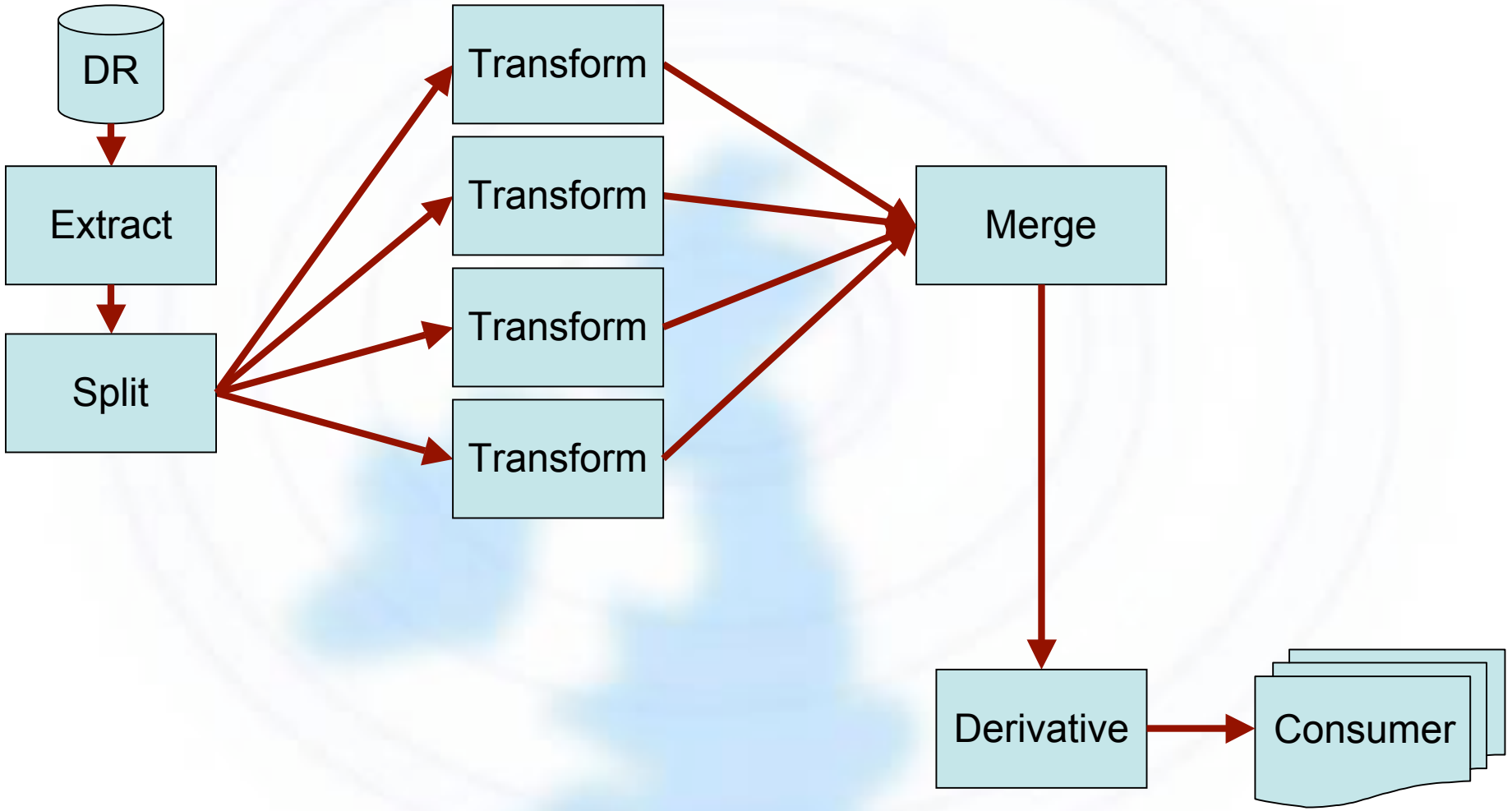
Multiple site collection



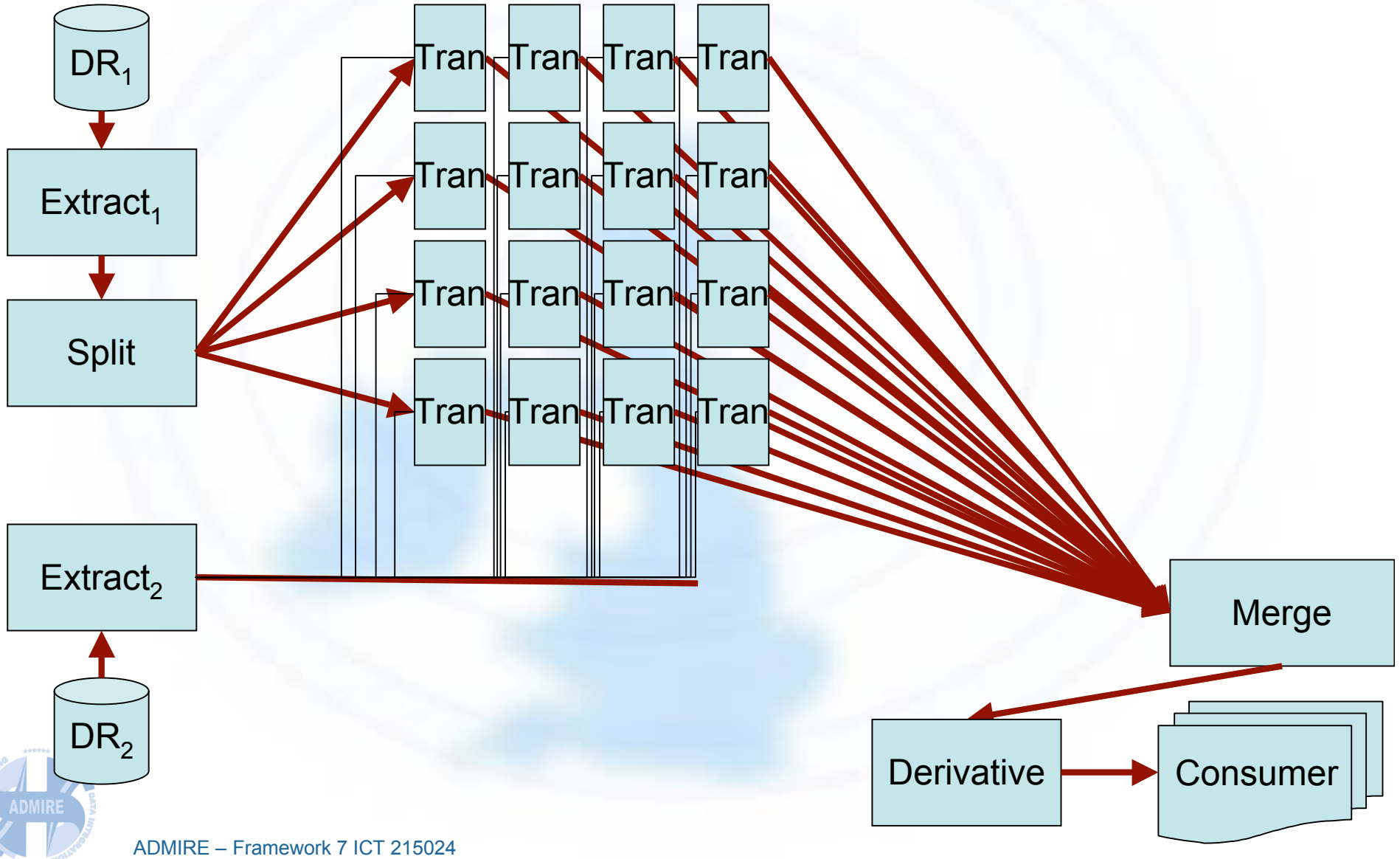
Multiple site tree pattern



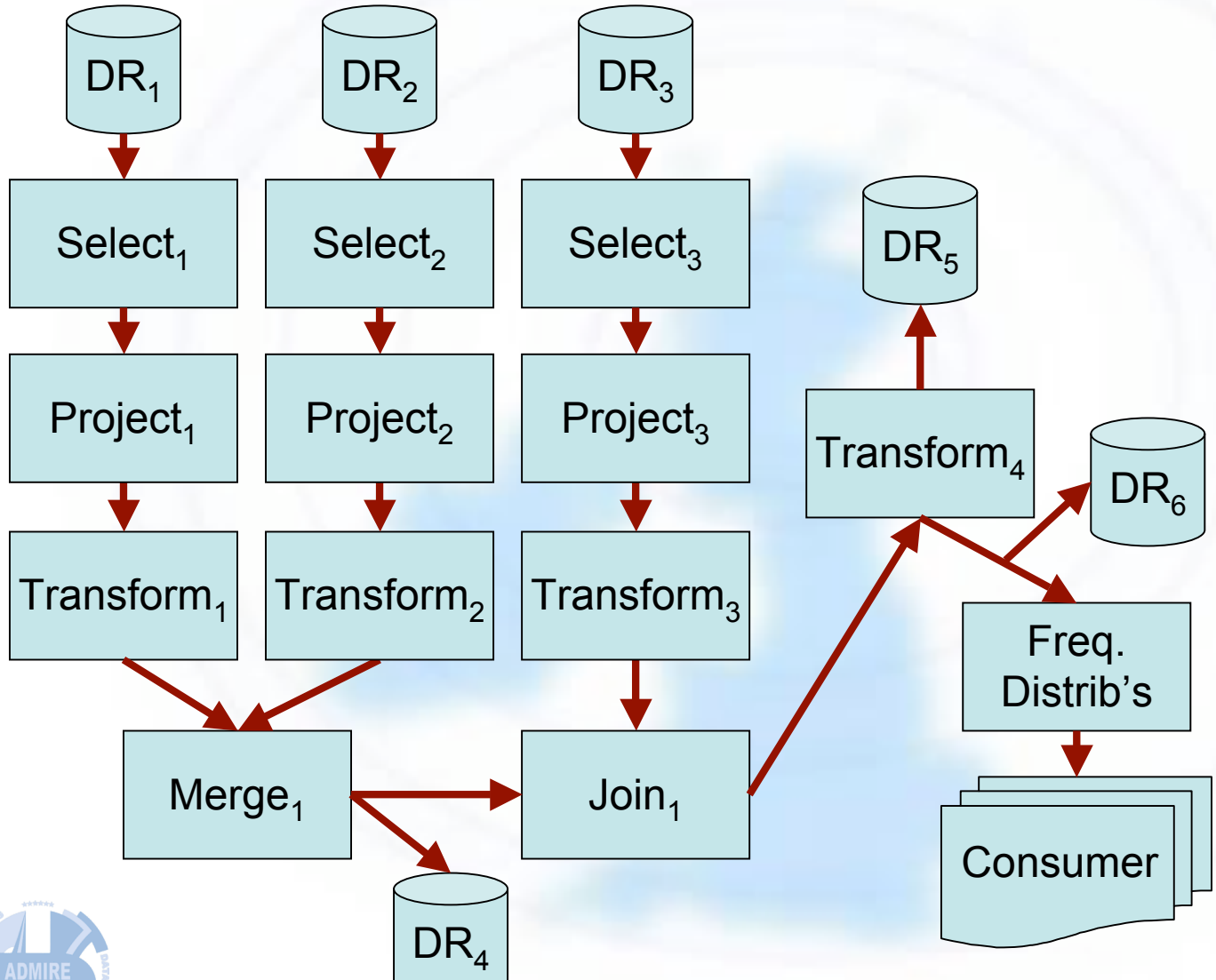
Simple parallel split pattern



All-against-all pattern



Delivering to Multiple Resources



- Further analysis
- Save expensive repeat requests
- Progress monitor
- Diagnostics
- Provenance

Outline

- Hypothesis
- Context, History & Motivation
- Data-Aware Distributed Computational Patterns
- **Multi-level Composition**
- **Streams: Content and Structure**
- **Notational and Functional Requirements**
- **Architectural Ideas**
- **Experience and Experiments**
- **Conclusions and Plans**

Composing via Streams

- Across long-haul networks
  - Higher overhead, more latency & slower
  - Combine large units of data-intensive computation
  - Use coarse stream granularity
- Local high-speed networks
  - Local area or on a cluster
  - Range of scales of data-intensive computation
  - Medium to coarse stream granularity
- On-chip - in VM or between VMs
  - Efficient fine-grained composition of computational units
  - Has to still handle high-volume streams
  - Fine-grained stream granularity can yield performance
- All logically equivalent
  - With identical API for the computational units

Fine-grained composition an alternative to endless data-adapter coding

Stream Content

- A Sequence of Items
  - Have to handle large items > stream granularity
    - Stream binary large objects (images, files, matrices, …)
    - Or pass by reference (while reference still valid)
  - Have to handle many small items
  - Require (internally at least) efficient binary forms
- Open range of items to accommodate
  - Extensibility & legacy systems
- Start & End Sequence markers
  - With metadata
  - To handle interpretation
  - To support auto-iteration
  - To support shut-down & clean up

Standard Stream Content

- Start Sequence Marker [
- Tuple Metadata < $name_1$ $type_1$; … >
- Tuples < 5, 13.135, “abc”, … >
  < 9, 18.279, “def”, … >
- End Sequence Markers]

- Sequences may be arbitrarily deeply nested
- Internally represented by data structures
- Mapped for external transmission

Types of Values in Streams

- Recursive composition of
- Base types (extensible)
  - Programming types: int, real, bignumber, …
  - BLOB types: image, ASCII doc, UNICODE doc, XML doc, SQL RowSet, …
- Matrix constructors
- Sequence of Tuples
- Ref to T (in a store S)

Need to know more

- For optimisation
  - Sequence represents
    - Bag or Set
  - Order significant
    - Implicit
    - Based on a sort of tuples with comparison test C
- Issue
  - How much do we need to know
  - How much can we avoid "knowing"
    - So compositions less sensitive to autonomous evolution
  - "we" equals the Data Access and Streaming Engines (DAISE) that do stream "plumbing" & "optimisation"

Outline

- Hypothesis
- Context, History & Motivation
- Data-Aware Distributed Computational Patterns
- Multi-level Composition
- Streams: Content and Structure
- **Notational and Functional Requirements**
- **Architectural Ideas**
- **Experience and Experiments**
- **Conclusions and Plans**

Introducing the Action

- **Each action is specification of some data task**
  - **Zero or more input streams**
    - Some may be thought of as parameters
    - But they all treated the same
    - They may receive their inputs from any composition of Actions
  - **Zero or more output streams**
    - Primary data for other Actions
    - Diagnostic, Progress and Metadata (provenance)
- **A host DAISE will run instances of actions (tasks)**
  - **Setting up and providing streams**
  - **Monitoring progress & providing diagnostics**
- **Some Streams are connected to tasks**
  - **Hosted by different DAISE**

Libraries of Actions

- Provided as "Standard"
  - RDB & XDB query, update, bulk load, statements
  - Conversion between standard forms
  - Data generation
  - Third party fetch and delivery
  - Simple linear operations: select, project, …
  - Combinators: merges, joins, "set" operations
  - Sorting and grouping
  - Statistical derivatives
  - Distributed and resilient query processing
  - Generic Select-Project
  - Generic combinators

Extensible Libraries of Actions

- Built and provided by user communities
  - Data mining
  - Text mining
  - Geo-Information Systems data extraction
  - RDF & ontologies
- Built by application developers for specific tasks
  - Medical data access security and privacy
- Standard Interfaces
  - JDBC, ODBC, WS-DAI, WS-DAIR, WS-DAIX, WS-DAIRDF, …

Challenge: co-design of architecture, DAISE, Developer's kit & Users' aids

Streaming composition notation

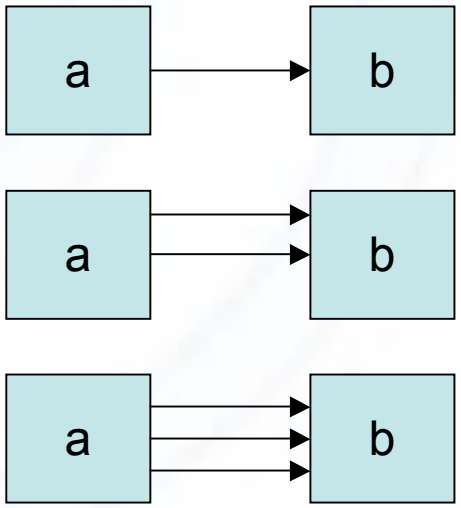
$q^{RDB} \rightarrow t \rightarrow d$

$u^{RDB}; q^{RDB}$

$r_1 \rightarrow d_1 \mid r_2 \rightarrow d_2$

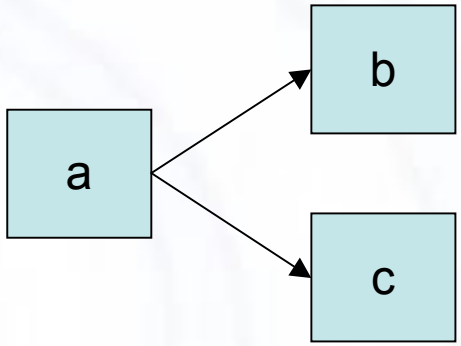
- Contrast with current OGSA-DAI requests
  - XML texts describing each task
- Lacks much required detail
- Useful foundation for discussion / language
  - How much can we do by describing activities
  - Once when they are defined
  - And inferring default behaviour?

$a \rightarrow b$

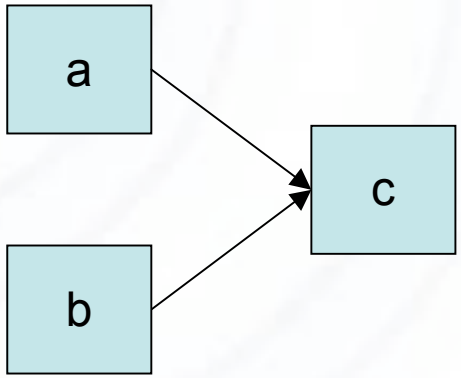


Streaming composition notation

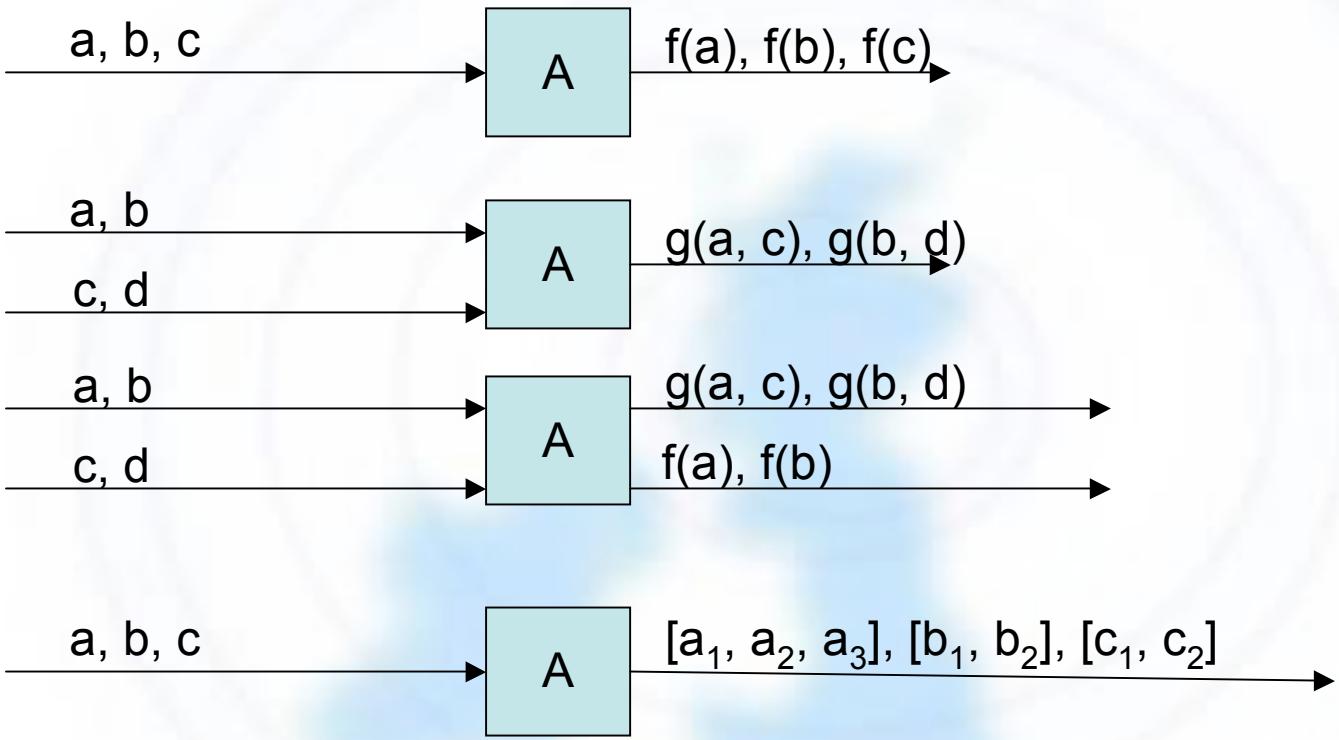
$$a \rightarrow [\, a_{o1} = b_{i1},\ a_{o2} = c_{i1}\,]\ (b \mid c)$$



$$(a \mid b) \rightarrow [\, a_{o1} = c_{i1},\ b_{o1} = c_{i2}\,]\ c$$



Activities Transform Streams



Describing activities

- **How do their outputs relate to their outputs**
  - **Sequence Structure**
    - Order preserved?
    - Set property preserved?
    - Cardinality preserved | increased | decreased?
    - Degree preserved | increased | decreased?
    - Sequence depth unchanged | decremented | incremented
    - Too complex - annotation of output required
  - **Tuple Structure**
    - Elements changed | unchanged
    - Deleted elements
    - New elements
  - **Type System**
    - Specific to validate & infer correctors
    - Non-specific to avoid unnecessary dependencies
  - **Naming System & Metadata**
- **And ontological descriptions …**
  - **Input 1 must be Environmental data complying with ISO…**
  - **Output 1 will then comply with ISO… and OGC… version …**

Functional requirements

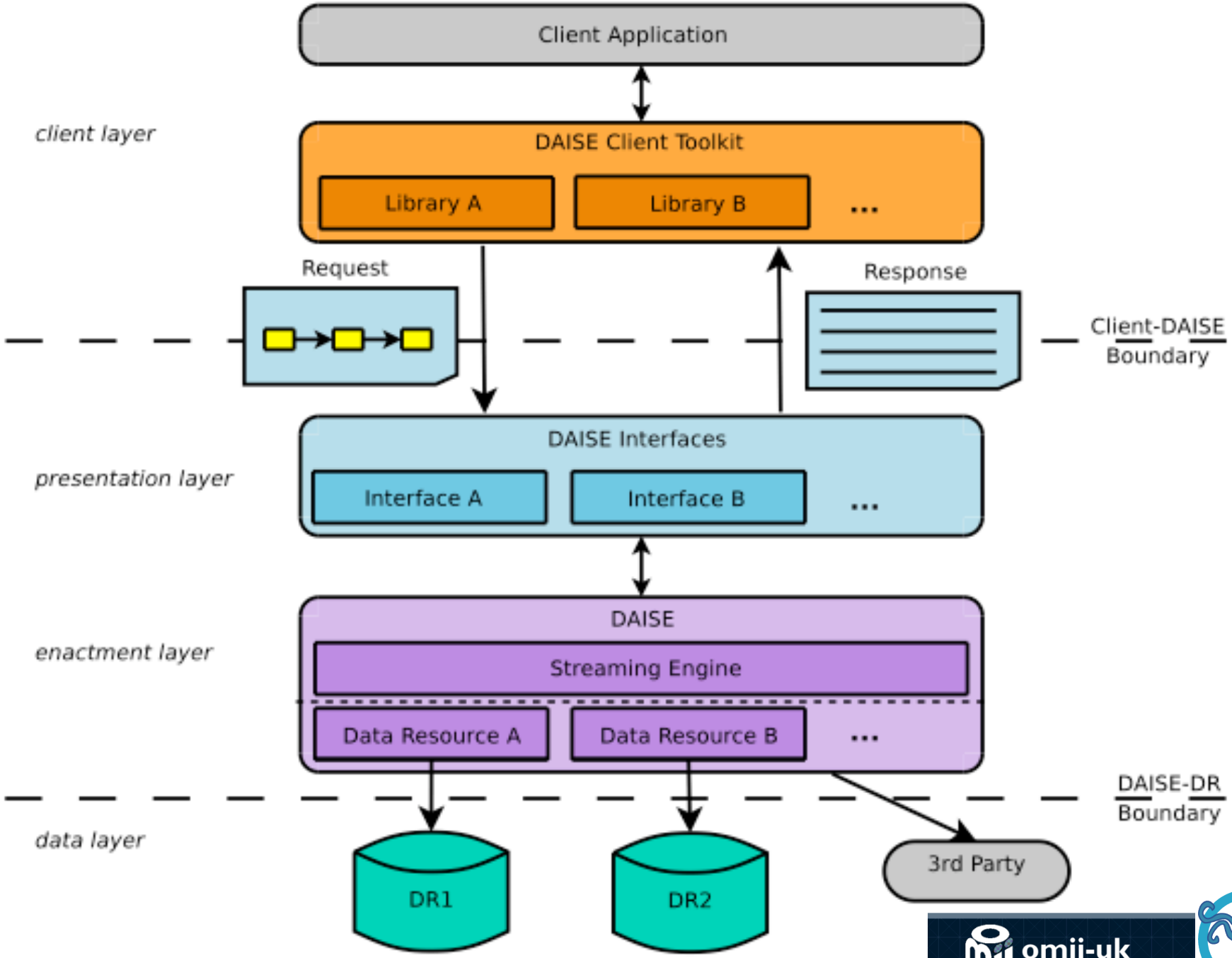
- Deployed activities
  - Hosted in DAISE
  - Instances initialised as tasks
  - Each task monitored
- Pipes for streams
  - Buffering
    - Local & long-haul
  - Wait for input
  - Detect stream exhausted
  - Detect recipient not consuming

Functional Requirements

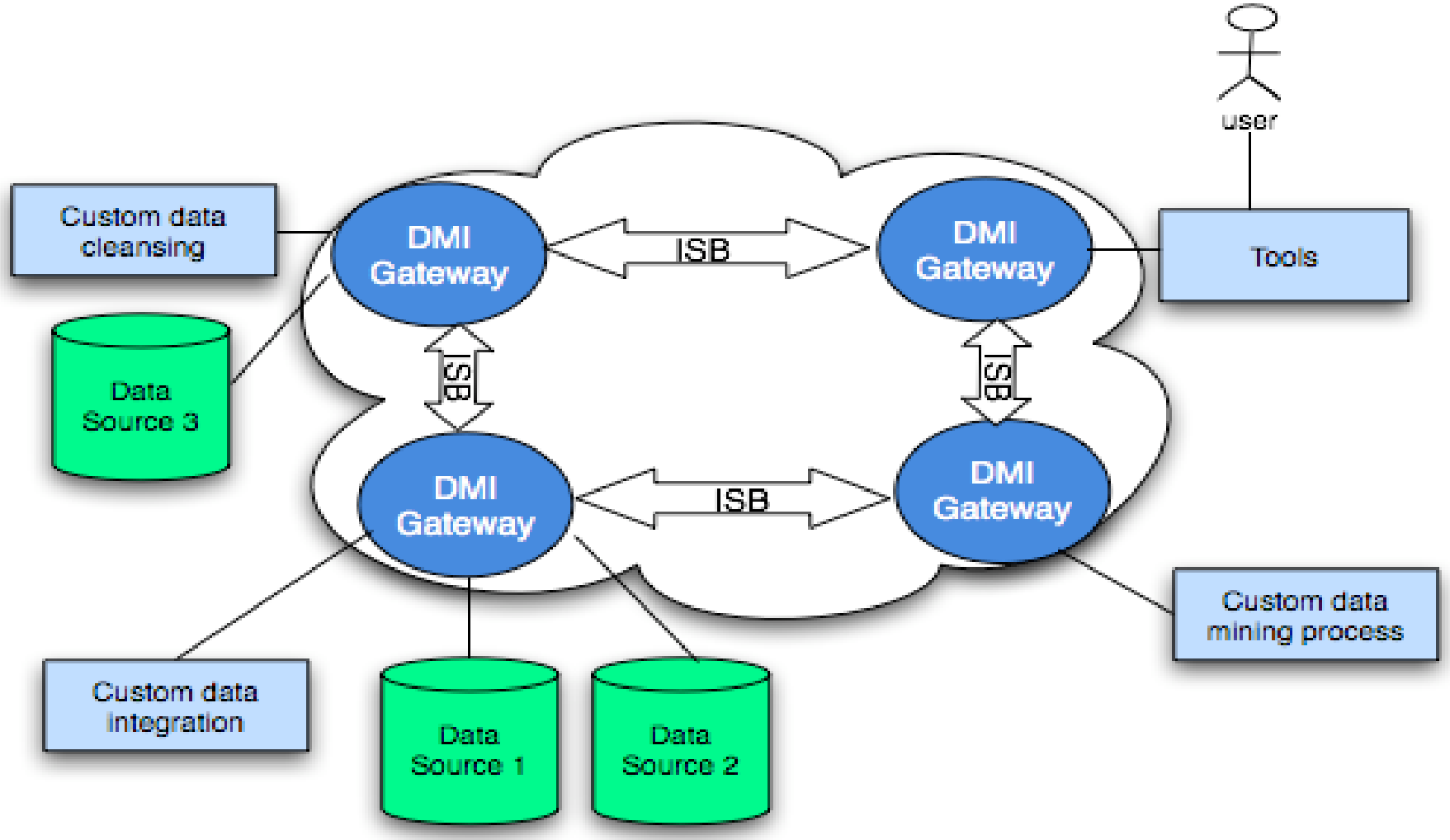
- **Shutdown**
  - **When receiver indicates “no more”**
  - **Propagate shut down along pipes**
    - Until encounter alternative receiver or all shut down
  - **Asynchronous third party delivery not affected**
  - **When supplier indicates “no more”**
    - Flush other inputs if logically appropriate
    - Flush outputs
    - Propagate the “no more” signal down outputs & stop
  - **When control signal to DAISE says “stop”**
    - Stop all tasks associated with this request enactment
    - But reflexively spawned enactments continue
  - **When an error is detected (signalled by a Task)**
    - Stop all (or just part of?) the enactment

Outline

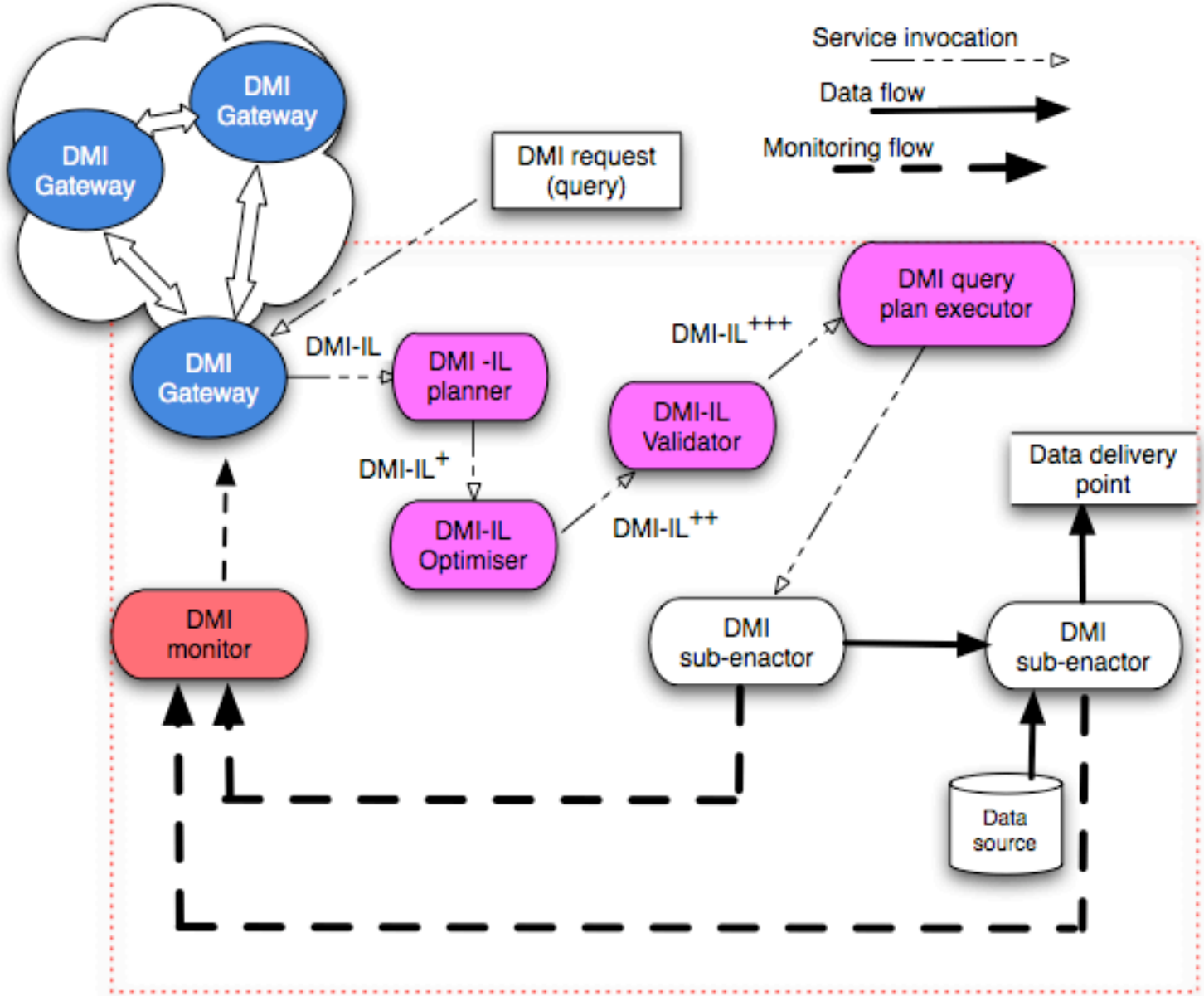
- Hypothesis
- Context, History & Motivation
- Data-Aware Distributed Computational Patterns
- Multi-level Composition
- Streams: Content and Structure
- Notational and Functional Requirements
- **Architectural Ideas**
- **Experience and Experiments**
- **Conclusions and Plans**


Client Application
client layer
DAISE Client Toolkit
Library A
Library B
...
Request
Response
Client-DAISE
Boundary
DAISE Interfaces
presentation layer
Interface A
Interface B
...
DAISE
enactment layer
Streaming Engine
Data Resource A
Data Resource B
...
DAISE-DR
Boundary
data layer
DR1
DR2
3rd Party

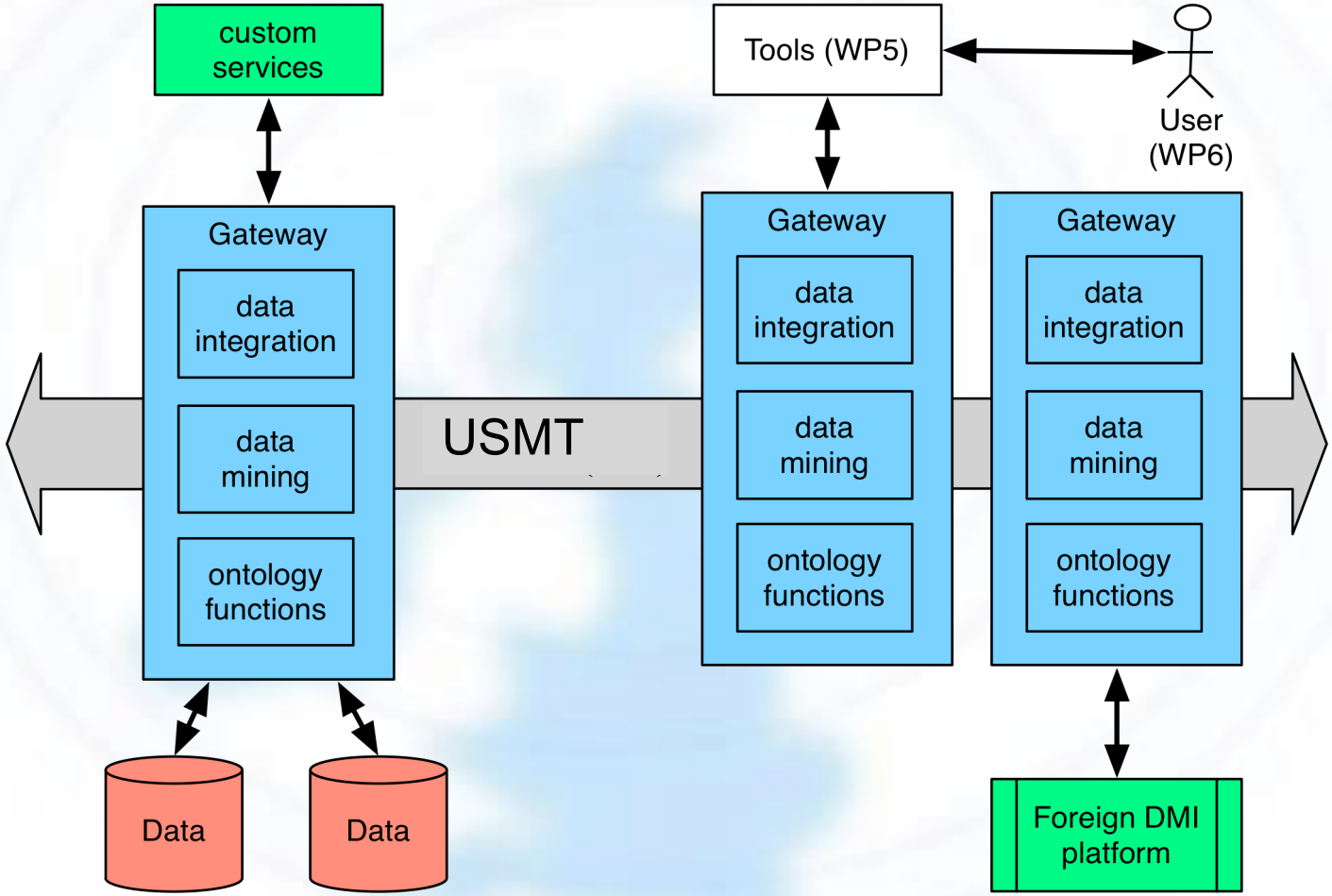
ADMIRE System Architecture



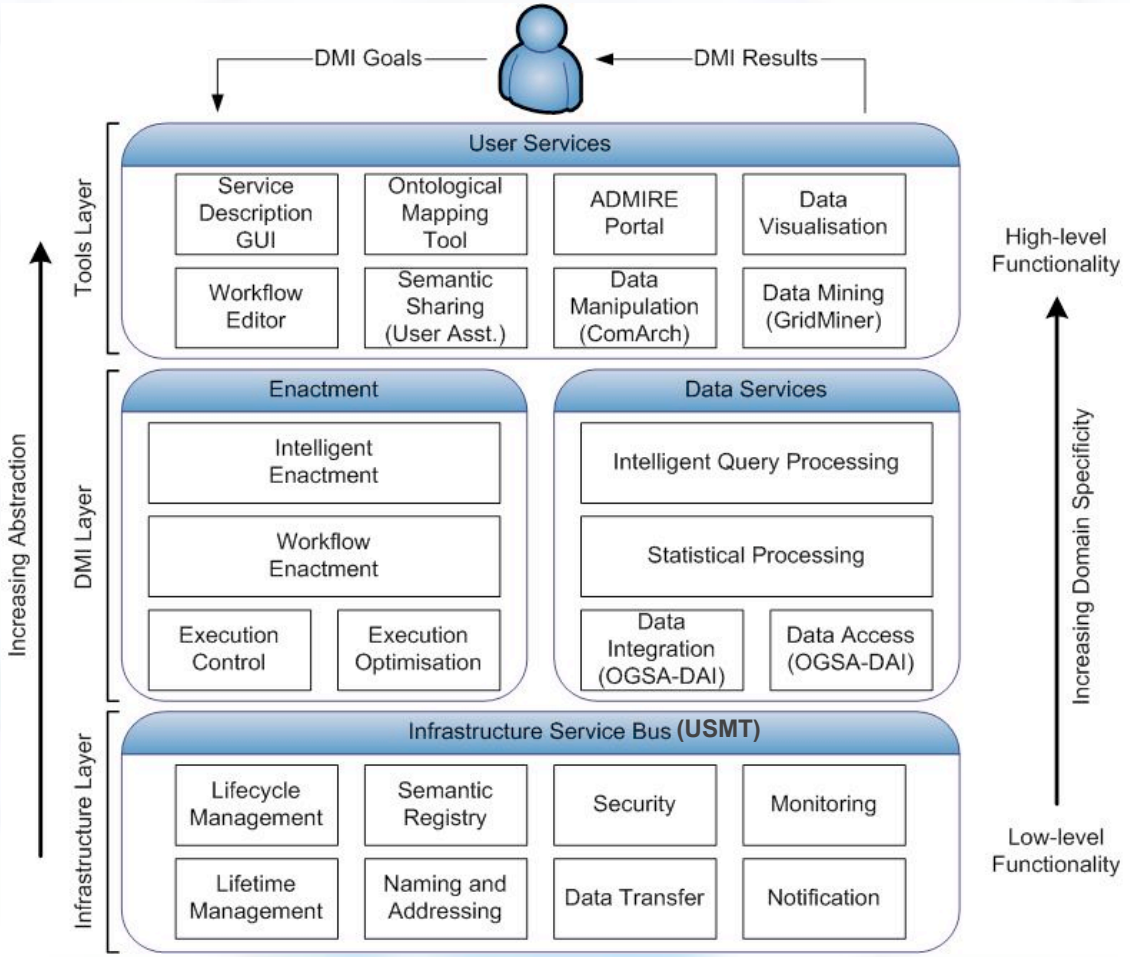
Envisaged ADMIRE architecture



ADMIRE Gateways



ADMIRE Architecture Components



Outline

- Hypothesis
- Context, History & Motivation
- Data-Aware Distributed Computational Patterns
- Multi-level Composition
- Streams: Content and Structure
- Notational and Functional Requirements
- Architectural Ideas
- **Experience and Experiments**
- **Conclusions and Plans**

FirstDIG – data with different schema distributed across multiple databases within an organisation

- FirstDIG – First Data Investigation on the Grid
  - http://www.epcc.ed.ac.uk/firstdig/
  - 2003-2004
- FirstGroup plc – UK's largest transport operator
- Distributed databases
  - Customer contact
  - Vehicle mileage
  - Ticket revenue
  - Schedule adherence
- Database types
  - Relational, ODBC-enabled, COBOL files,…

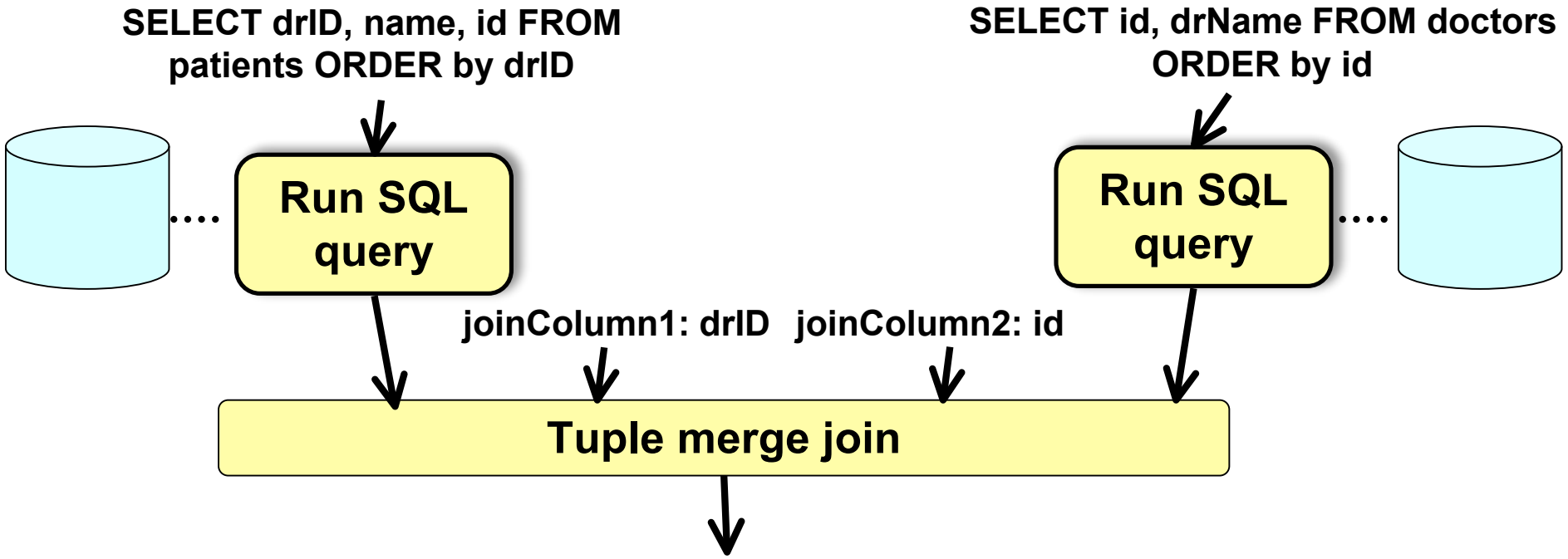
BEinGRID – data with same schema distributed across multiple databases within an organisation

- Business Experiments in Grid
  - http://www.beingrid.eu
  - Demonstrate applicability of Grid technology to business
- BE12 – Sales Management System
  - ENEA, CINECA, Tecnocassa, Pizza New
- OGSA-DAI
  - Standardised and uniform layer to distributed heterogeneous databases
- In conjunction with sales management system

VOTES – data with different schema distributed across multiple databases within a group of strategic partners

- Virtual Organisations for Trials and Epidemiological Studies (VOTES)
  - http://labserv.nesc.gla.ac.uk/projects/votes/index.html
  - UK Medical Research Council project
- Provision of data access and integration functionality for the clinical domain
- Distributed databases
  - Patient information
  - Clinical trials results
- Database types
  - Microsoft SQL Server, Access,…

VOTES – cross-database join activity



- This is equivalent to running:

```
SELECT id, name, drID, drName FROM patients, doctors
        WHERE patients.drID = doctors.id;
```

- patients and doctors are in two different databases

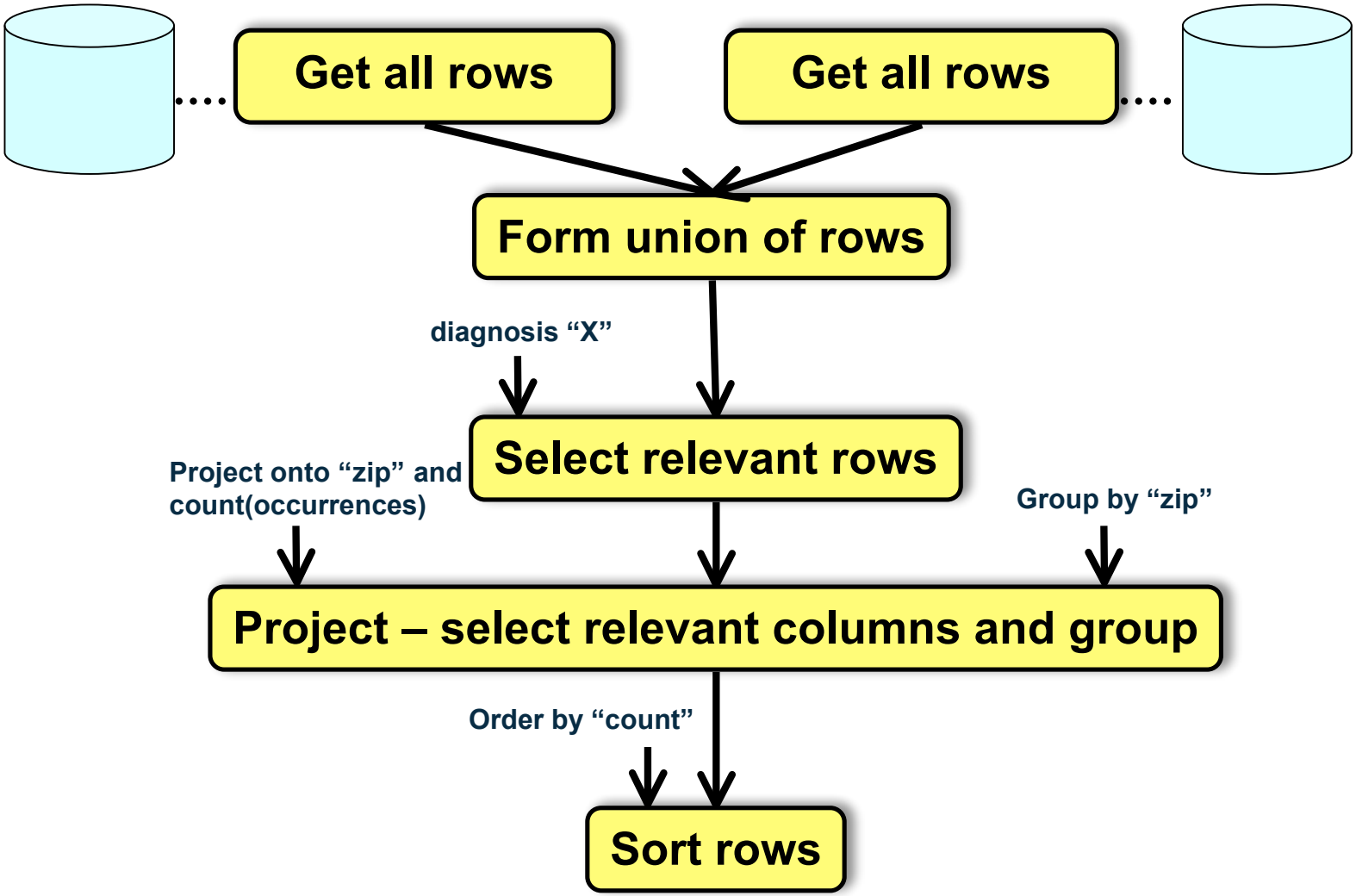
SIMDAT – data with different schema distributed across multiple databases within a group of strategic partners

- SIMDAT
  - http://www.simdat.eu
  - Grids for industrial product development
  - Automotive, aerospace, pharmaceutical, meteorological
    - e.g. Audi, BAE Systems, EADS, Meteo France, UK Met Office
- GRIA OGSA-DAI ontology and data services
  - Transparent access to files and databases
  - Industrial-strength fine-grained access control
  - Common interface to local data and remote data resources
- Semantic Bus
  - Automotive semantic mediation for CAE/CAD
  - Access local database with same schema
  - Go through semantic mediation steps to access remote data resources

Public Health Grid – data with different schema distributed across multiple databases within a group of strategic partners

- US Public Health Grid
  - US Centers for Disease Control
  - University of Pittsburgh
  - Tarrant Country Public Health Department
  - Dallas County Public Health Department
- Health query system
  - Look for incidences of some disease on the rise over an area
  - Historical and live data
- Health centres maintain their own databases
  - Distributed databases
  - Different products
  - Different schemas
    - e.g. PatientID, Id, PatientIdentifier, PatientNumber
  - Security and privacy is important

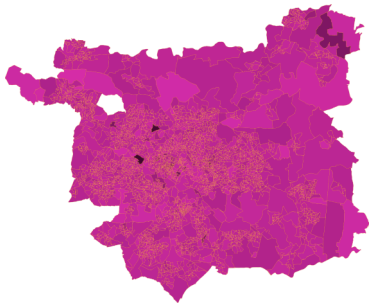
Public Health Grid – demonstration



SEE-GEO – working with private and public data

- SEcurE access to GEOspatial services
  - http://edina.ac.uk/projects/seesaw/seegeo/index.html
  - EDINA, MIMAS, NeSC, NCeSS
  - UK JISC project
- Geographical information systems
- Virtual integration of and access control to
  - Census data
  - Borders data
  - Data hosted by other organisations and exposed as services
- OGSA-DAI for federation of heterogeneous data resources

SEE-GEO – geo-linking service portal



1: GLSQuery submited via portal e.g. “Leeds population distribution by census output area”

GLS Portal

Maps

5: Portal gets image using URL

4: URL of image is returned to portal – avoids costly SOAP/HTTP transfer of image

MIMAS Census

UK BORDERS

OGSA-DAI

Get

Get

Join

Transform

Deliver

3: Image is placed on a map server

2: Workflow is populated with query parameters and run

Image Creation Service

GEOGrid – work with private data and manage access to distributed data

- Global Earth Organisation (GEO) Grid
  - http://www.geogrid.org/
  - National Institute of Advanced Industrial Science and Technology, Japan
- Geospatial data and services
  - Disaster mitigation
  - Environmental monitoring
  - Natural resource exploration
- Virtual integration and access control
  - Satellite imagery
  - Geological data
  - Ground-sensed data
  - Each with different ownership and access policies
- OGSA-DQP and OGSA-DAI
  - Federation of heterogeneous data resources
  - Additional access control in conjunction with VOMS

GEOGrid – manage access to distributed data

- Virtual organisations access data resources
  - VO membership determines data that can be accessed

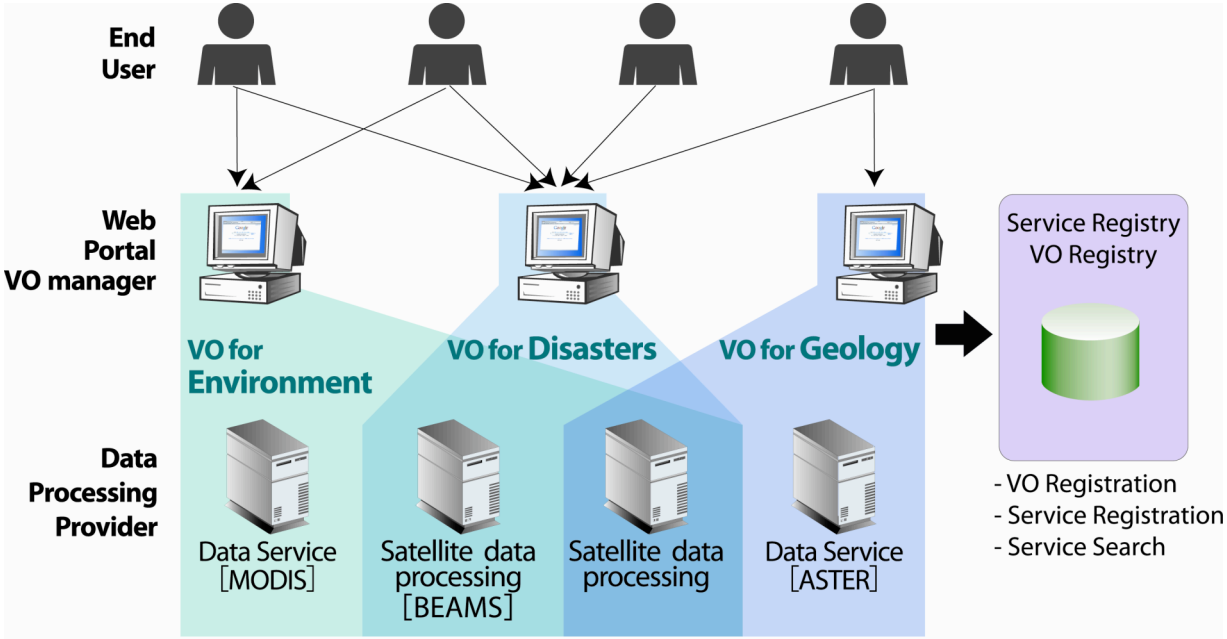


Image courtesy of Isao Kojima, GEOGrid

GEODE – work with private and public data

- Grid Enabled Occupational Data
  - http://www.geode.stir.ac.uk/
  - University of Stirling
  - UK ESRC e-Social Science small grants project
- Online standardised access, management and translation of occupational statistical data for social sciences
  - e.g. gender segregation for occupational codes used in UK 1991 census (SOC90)
- OGSA-DAI for uniform access to heterogeneous data resources
  - Relational and CSV
  - Driver management => lighter-weight data linking tool
- Generic data linking tool
  - Link data resources exposed by OGSA-DAI
  - Link to researchers' own, possibly private, data resources

OGSA-DAI pros

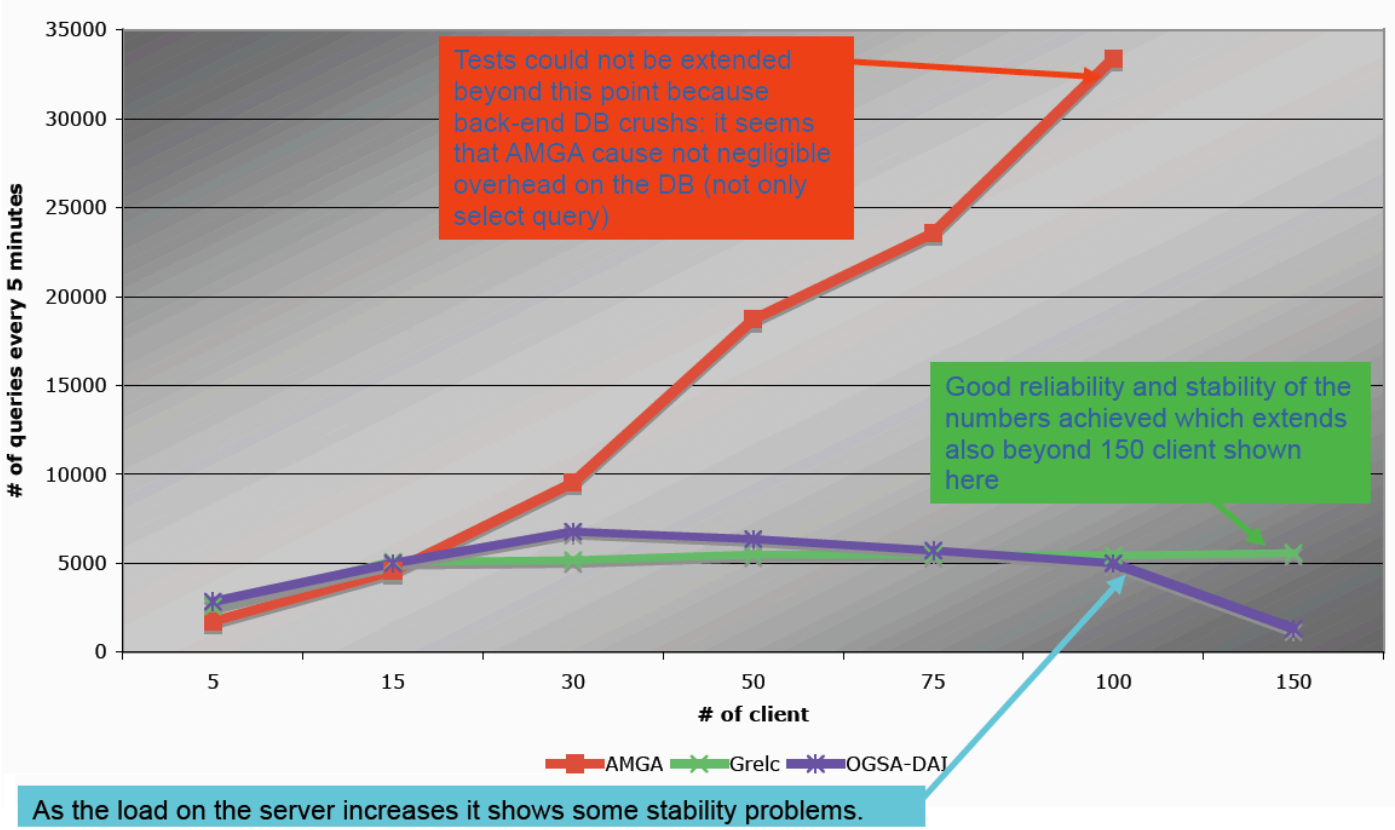
- 100% Java open source freeware
  - Runs under Windows, UNIX, Linux
- Compliant with
  - Globus Toolkit 4.0.x
  - Apache Axis/Tomcat
  - OMII 3.4.0
  - UNICORE – developed by OMII-Europe
  - VOMS – security components developed by OMII-Europe

JDBC and ODBC-compliant facades

- Analyse federations of data resources managed by OGSA-DAI
- Existing data analysis products are “Grid-enabled”
- INWA project – Innovation Node Western Australia
  - http://www2.epcc.ed.ac.uk/~inwa/
  - EPCC, Curtin Business School Australia, Chinese Academy of Sciences Beijing
  - Secure distributed data mining of property and mortgage data
  - Use OGSA-DAI with ODBC-compliant SPSS statistical analysis tool
- JDBC driver for OGSA-DAI
  - Currently under development by Mathias Brito and BEinGRID
- ODBC driver for OGSA-DAI
  - Also of interest to BEinGRID

INFN's results

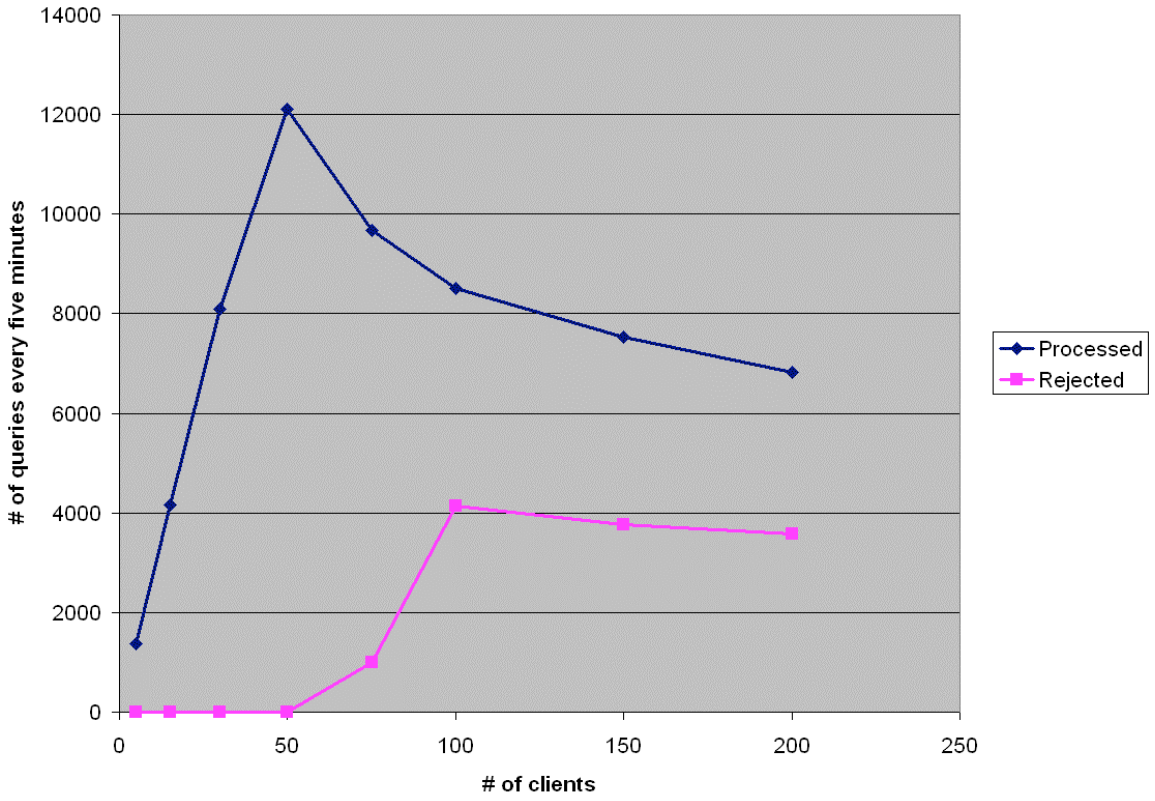
- OGSA-DAI did not perform well with 30+ clients



Graph courtesy of Giacinto Donvito, INFN

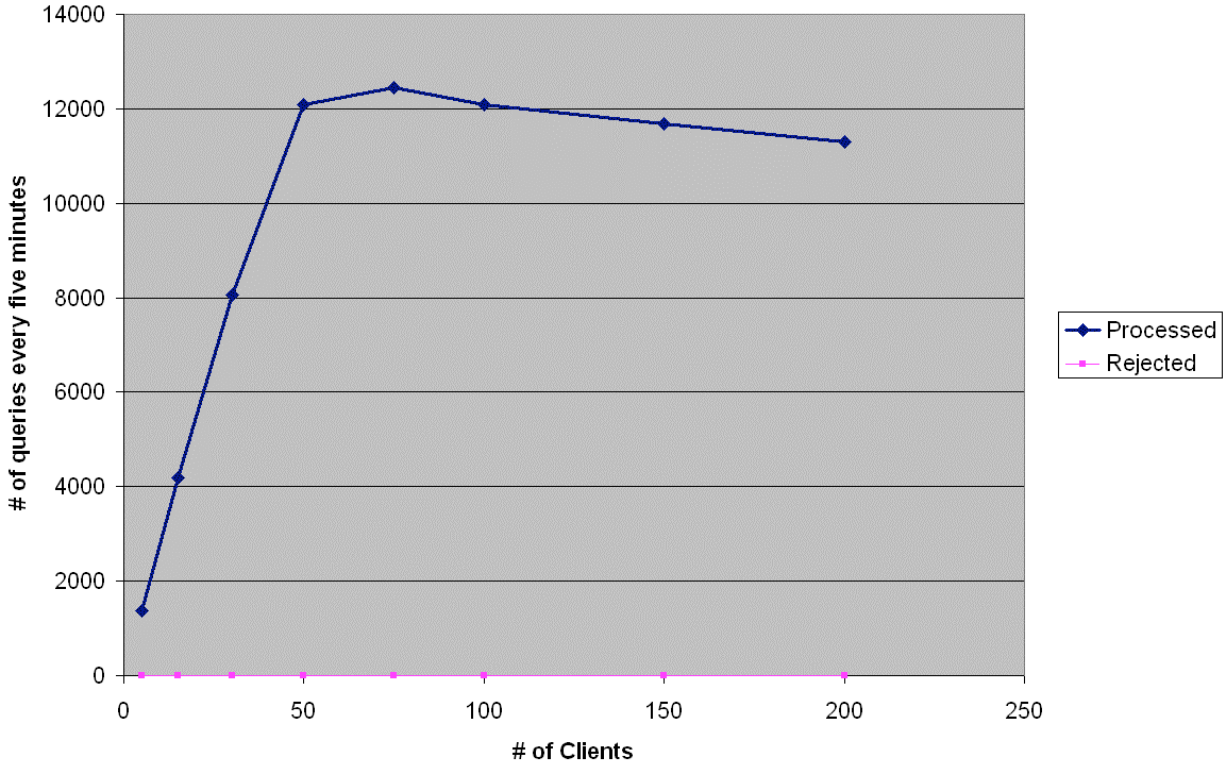
Reproducing INFN's results

- Performance drop starts when OGSA-DAI becomes too busy to process requests
- Even rejected requests consume resources



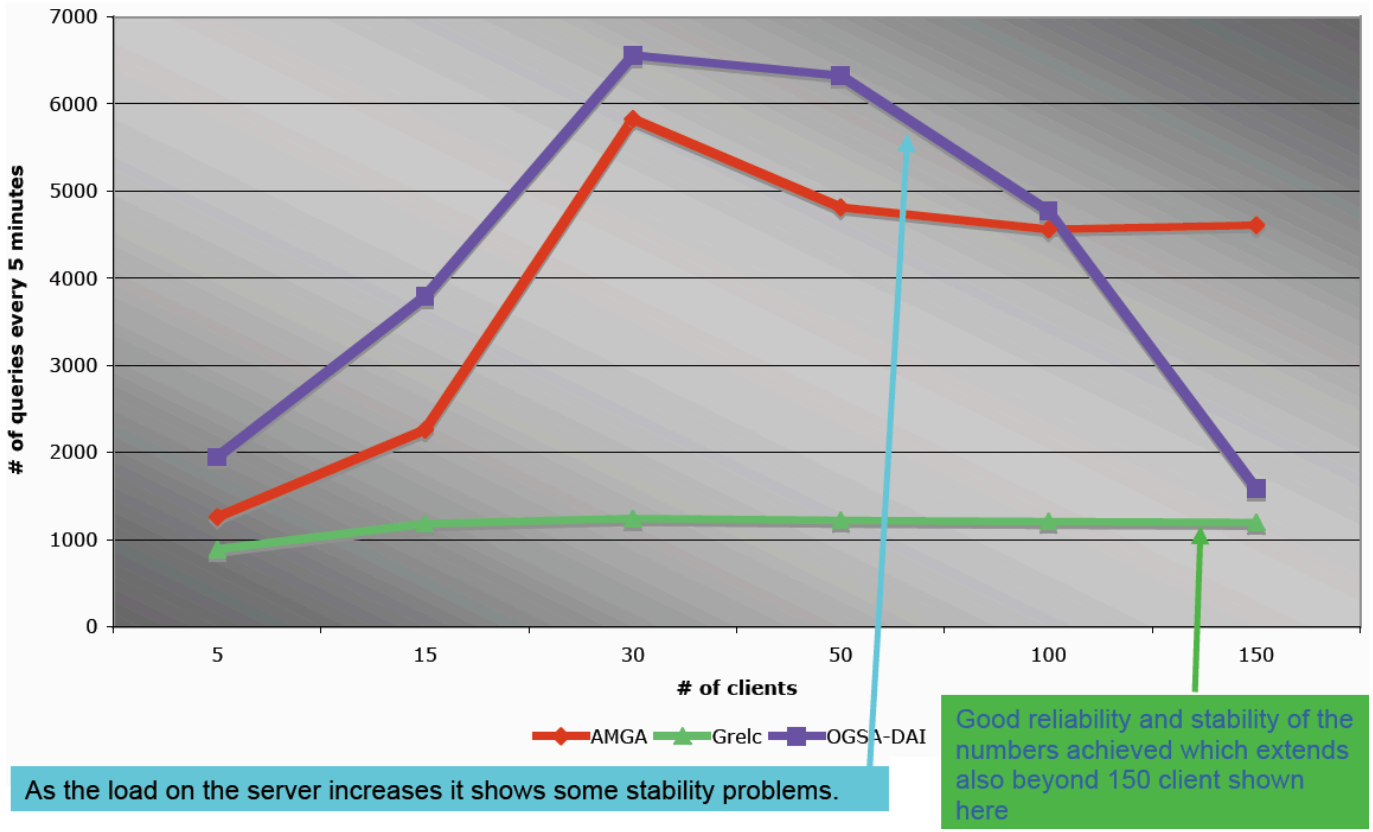
Experimenting with the queue size

- Keep the OGSA-DAI request pool at 10 but increase the queue to 200
- Make clients wait until their request has been processed (synchronous)
- Server resources are not wasted by rejecting requests, resulting in more stable performance



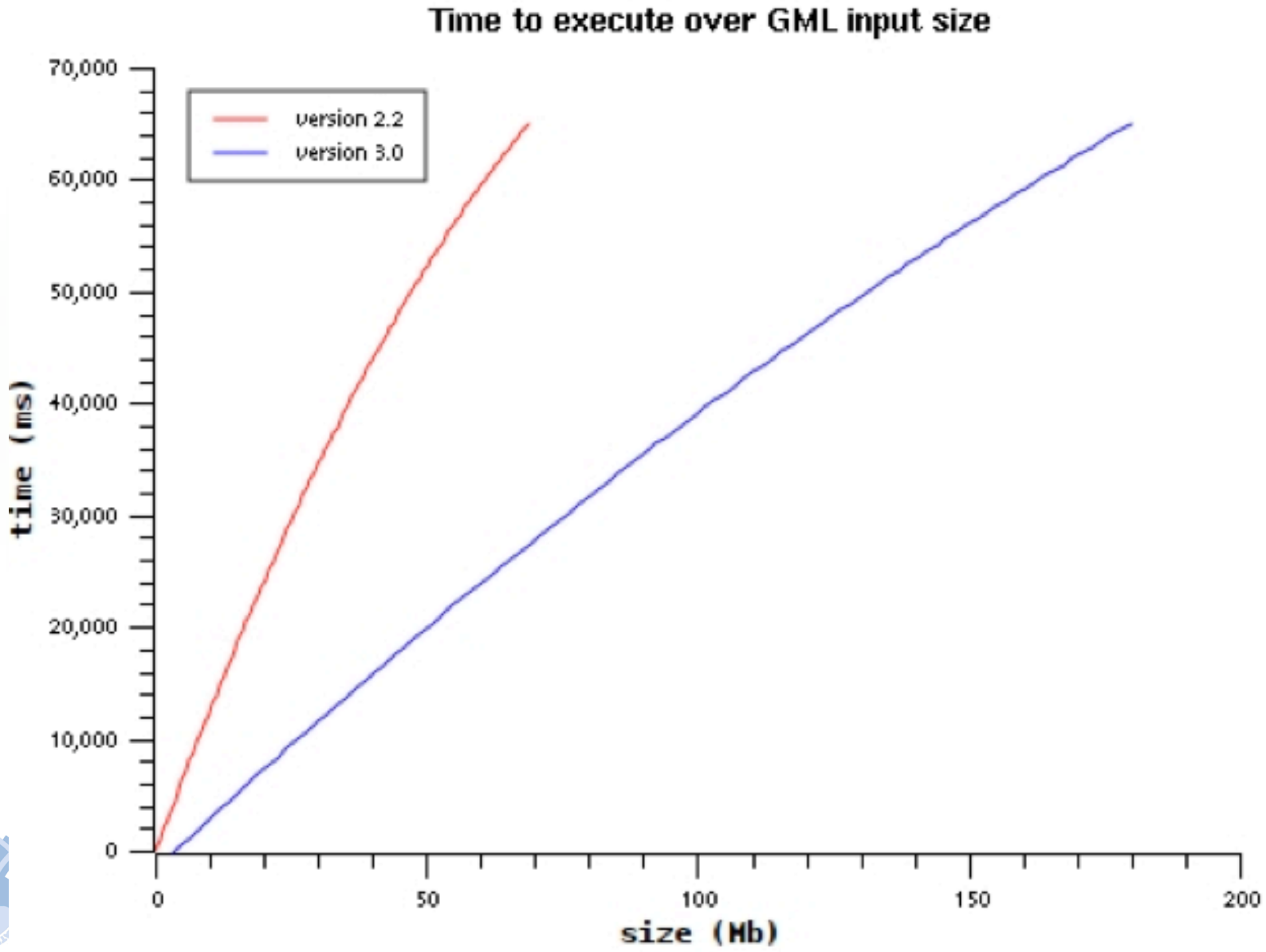
Investigate performance – INFN's evaluation

- Test done by INFN with default OGSA-DAI settings but for 1000+ row results
- OGSA-DAI shows good performance



Graph courtesy of Giacinto Donvito, INFN

Geospatial Performance



Outline

- Hypothesis
- Context, History & Motivation
- Data-Aware Distributed Computational Patterns
- Multi-level Composition
- Streams: Content and Structure
- Notational and Functional Requirements
- Architectural Ideas
- Experience and Experiments
- **Conclusions and Plans**

Stream-based DAI & DM

- **Challenges**
  - Scalable model from threads to wide area
  - Light-weight composability
  - Succinct high-level language
  - Inference providing most detail
  - Automated request transformation, deployment & enactment management
    - Optimisation
    - Recovery

OGSA-DAI evidence

- Extensibility demonstrated & exploited
- Deployability & Homogeneity Demonstrated
- Scalability
  - Demonstrated within “scope”
- Power
  - Extensive libraries of activities
  - Composible
  - Programmable

ADMIRE Vision

- Scale up and Integration
  - Coherent open & extensible environment for DMI
  - Incorporating legacy systems
  - Hierarchy of Composable DMI enactors
  - Autonomous operation
  - Dynamic load adaptation
- Expressive power
  - High-level DMI language
  - (Semantic) Description of Components
  - Inference reducing labour
- Conceptual Partition
  - End-User domain experts' View
  - DMI experts' View
  - Linked but Separated by Sophisticated Engineering

Data-Aware Distributed Computation

- Challenging & Complex
  - Diversity & Change
  - Dynamic requirements
  - Expert users need to steer & explore
- Growing in Scale, Importance & Complexity
- Stream-based Computation Composition
  - Will have a significant role

?

www.admire-project.eu

www.ogsadai.org.uk

www.nesc.ac.uk

omii-uk

www.omii.ac.uk

epcc

Picture composition by Luke Humphry based on prior art by Frans Hals