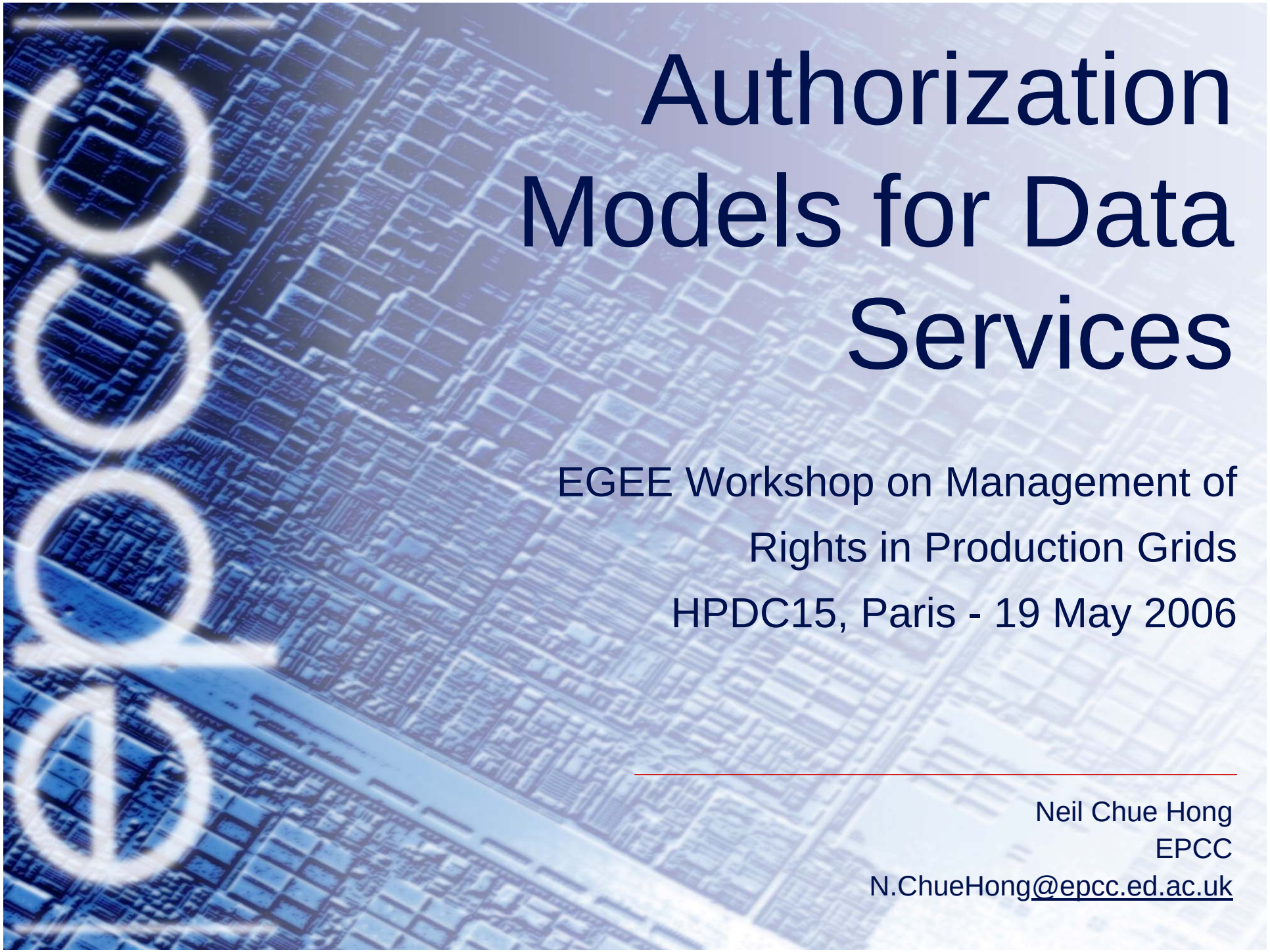


Authorization Models for Data Services

EGEE Workshop on Management of Rights in Production Grids

HPDC15, Paris - 19 May 2006

Neil Chue Hong
EPCC
N.ChueHong@epcc.ed.ac.uk

Overview

- Requirements for security on data services
- Current OGSA-DAI security model
- Authorization models in OGSA-DAI
- Future challenges for authorization in OGSA-DAI

- Thanks to:
  - Ally Hume, EPCC (OGSA-DAI)
  - Krzystof Kurowski, Michal Kulczewski, PSNC (inteliGrid)

Motivation

- Increased data sharing and data virtualisation
  - data security is still important
- Requirement for secure communication, authentication and authorization
- Different applications require different levels of security
  - some applications require no security at all
- Requirements for data services are typically more complex than for computation services
  - much finer grained security enforcement

Data location and context

- Data services (such as OGSA-DAI) provide virtualisation capabilities which allow data resources to be transparently accessed from within a VO
- Each part of the VO will want to retain full control over who can access their data resources
  - I will share my private data with you but not him
- How do we define cross-organisational security policies as data is shared and virtualised?

New Difficulties, New Challenges

- Data integration on the Grid brings new challenges
  - distributed queries when particular roles are denied access at the column/row level
  - geographical differences, e.g. in legal policy
  - delegation of sub-queries on behalf of other services
- Role based access may not be enough
  - Doctors can access records for patients IF they are treating them AND they are in a particular hospital

Medical data - anonymisation

- Epidemiologic data cannot be completely anonymous as long as it contains any useful information (e.g. place of residence, profession)
- Doctors must get permission of patients to release particular cases
  - Anonymise database and allow patient to deanonymise
  - Strip information – how to get back?
  - Legal protection only?
- Data protection is only as strong as the political environment allows

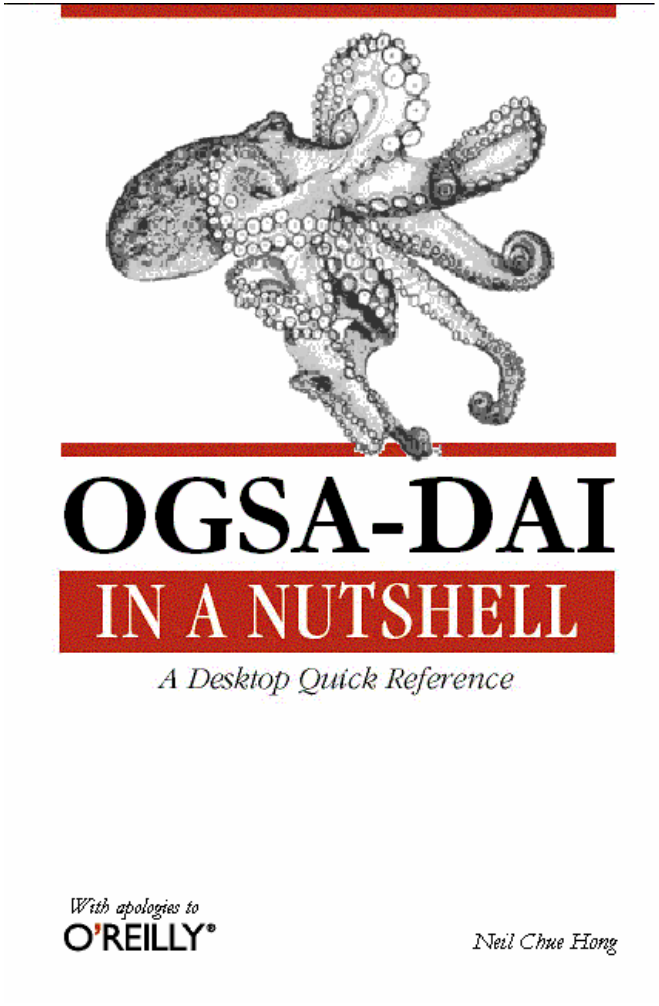
Levels of Authorization

- Currently, most data services authorize at the service level
  - allowed / denied access to data service, similar to computational resource
- Need a much subtler set of authorization points:
  - service (can / cannot utilise service)
  - resource (can / cannot send requests to resource)
  - activity (e.g. can get schema, but not run query)
  - sub-activity (e.g. can run some queries, but not others)
  - macro-level (e.g. block certain patterns of queries)

A Critical Mass of Data?

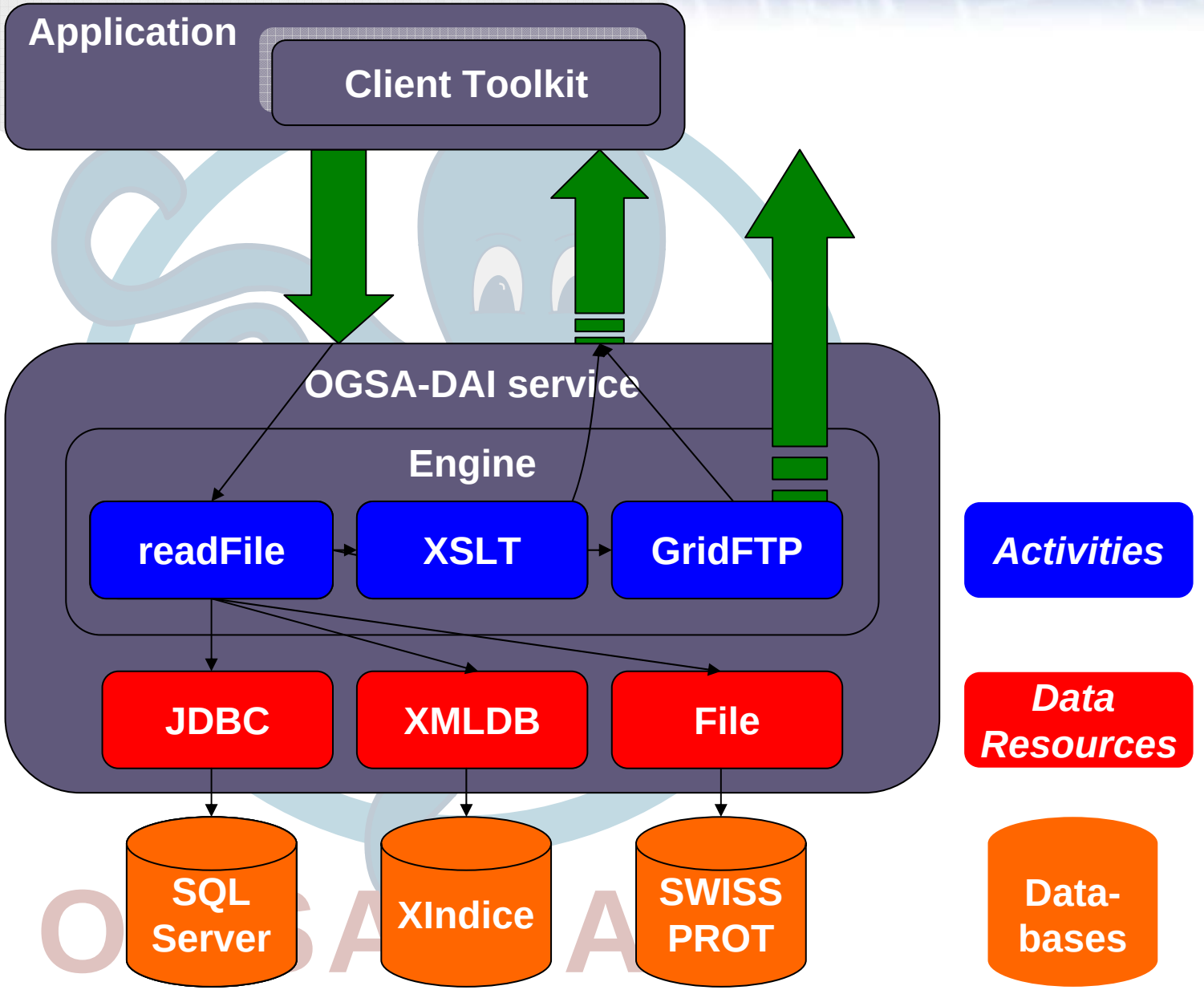
- Sometimes you do not want intermediaries to know what queries you ran
  - Know enough queries and you'll know what your competitor is searching for
- Sometimes you can extract information based on what isn't returned in the results
  - Query telephone directory with names from electoral register
    - Build up list of ex-directory names
- If you gather enough data together, does anonymisation and security become impossible?

OGSA-DAI In One Slide

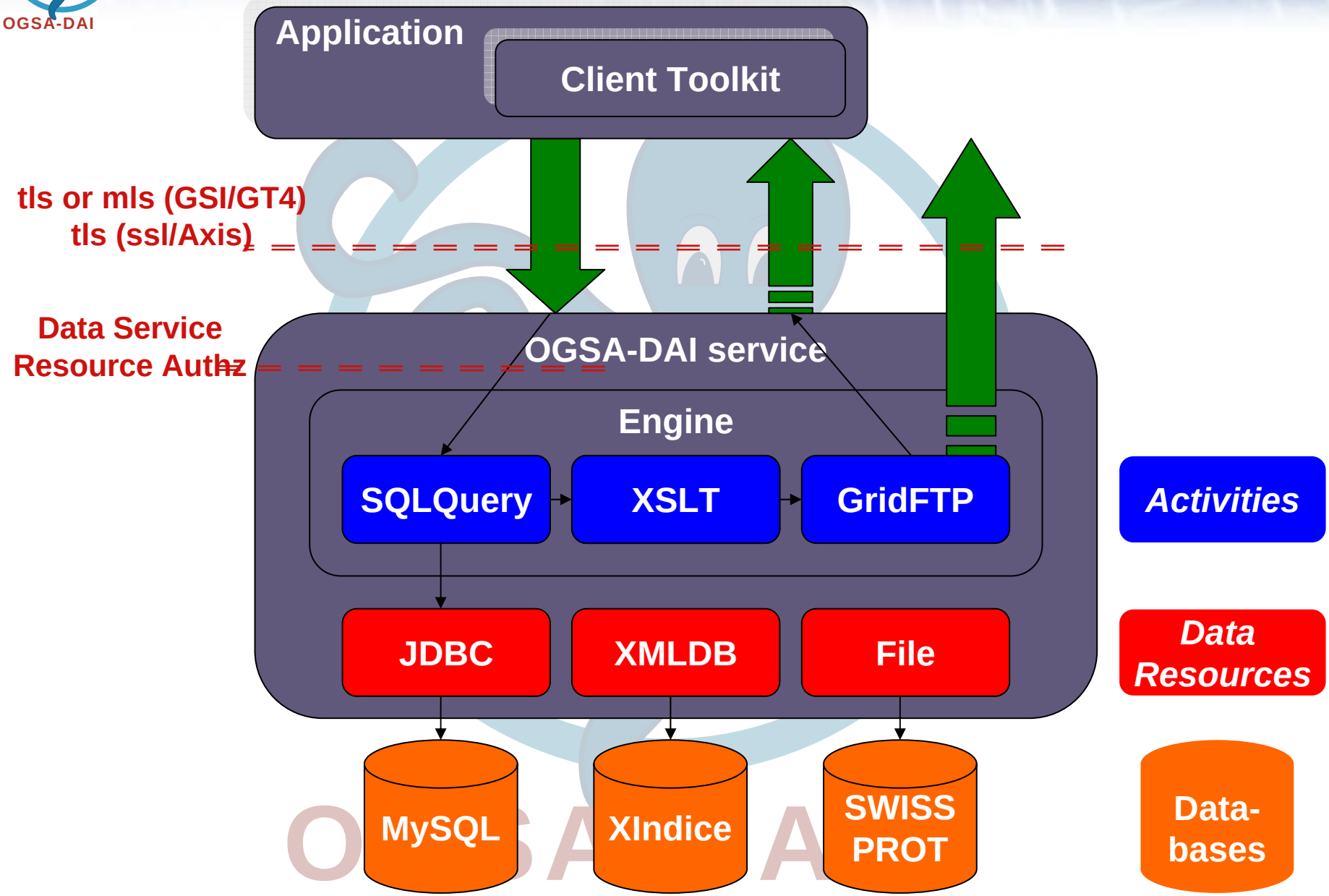


- An *extensible framework* for data access and integration.
- Expose heterogeneous data resources to a grid through web services.
- Interact with data resources:
  - Queries and updates.
  - Data transformation / compression
  - Data delivery.
- Customise for your project using
  - Additional Activities
  - Client Toolkit APIs
  - Data Resource handlers
- A base for higher-level services
  - federation, mining, visualisation,…

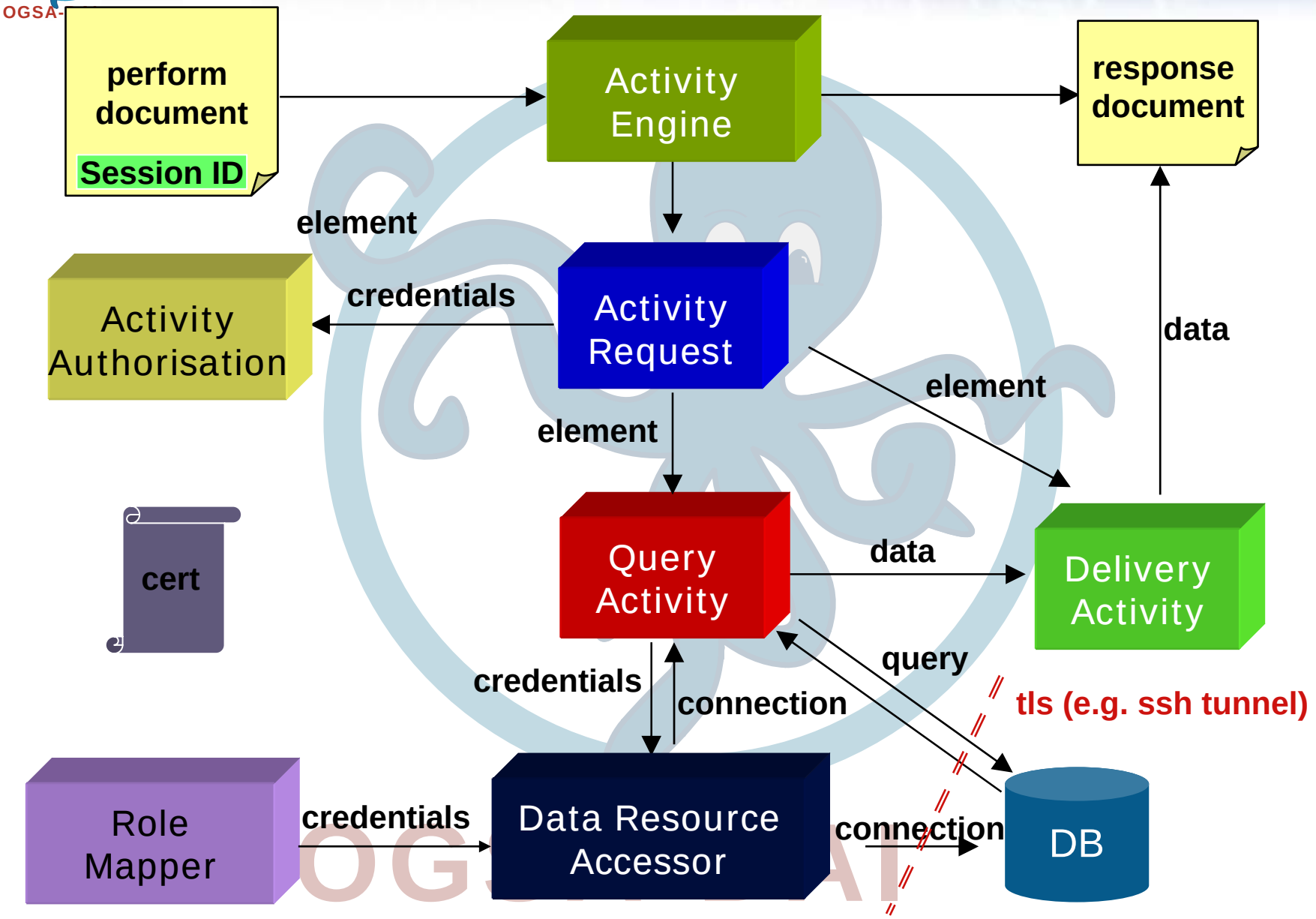
The OGSA-DAI Framework



The OGSA-DAI Framework



Data Service Internals



OGSA-DAI current model

- OGSA-DAI currently supports the Grid Security Infrastructure (GSI), through the Globus Toolkit, for mutual authentication and authorization
  - This supports both message and transport level security (TLS/SSL)
  - Used to secure communication between clients and services, and to identify grid users through DNs
- For OMII, utilises WSS4J plugin to access DNs
- Static and restrictive
- Want a dynamic model
  - change policy through lifetime of a service, resource, query…

Requirements from projects

- "*Is it possible to add/remove users of a data resource whilst a service is running*"
  - Larry Tan, University of Stirling, GEODE project
- "*Is it possible to secure the administration of the access control to services*"
  - Samatha Kotta, TU-Dresden, D-Grid
- "*Is it possible to make the delegation API implementation agnostic?*"
  - Ally Hume

Authorization in OGSA-DAI

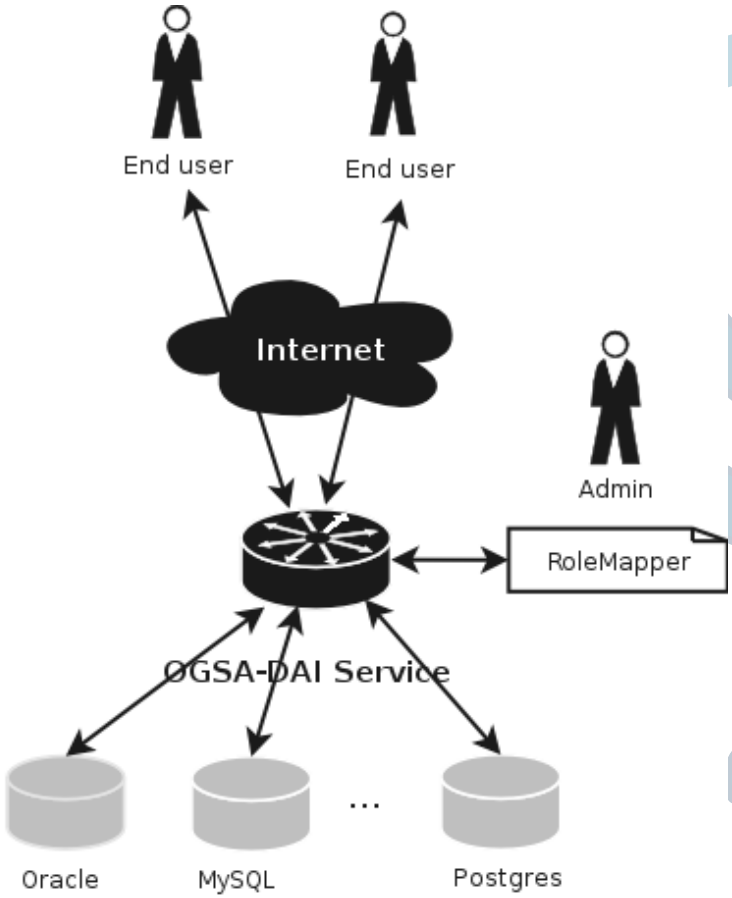
- Resource authorization determines whether a client can access a data service resource. If the authorization fails then the client cannot access the data service resource at all - they cannot access any activities or any properties or do anything with the resource.
- Activity authorization is applied on requests by a resource. It determines the activities that a specific client can execute on the data service resource. Activity authorization would allow a data service resource to be configured to allow some clients to execute query activities only whereas other clients to execute both query and update activities.
- Role mapping controls access to a database wrapped by a data service resource. If a client is denied access to the database this does not prevent the client from using the data service resource for non-database-specific activities

Levels of Authz in OGSA-DAI

- Client submits request to data service resource exposed by a data service.
  - Data service receives request.
  - Resource authorization - data service authorizes whether the client can access the data service resource.
- IF not authorized
  - THEN return authorization problem to client
- ELSE continue...
  - Data service passes request to data service resource.
  - Data service resource parses request.
  - Activity authorization - data service resource authorizes whether the client can execute each activity in the request.
  - IF not authorized to execute all activities in request
    - THEN return authorization problem to client
  - ELSE continue...
    - Data service resource executes request.
    - Data service resource executes individual activity in the request.
    - IF activity connects to database
      - THEN data service resource authorizes access to the database (rolemap).
      - IF not authorized to access the database
        - THEN return authorization problem to client
      - ELSE continue…
        - Activity interacts with database.

Want the APIs to look the same across GT4/OMII implementations

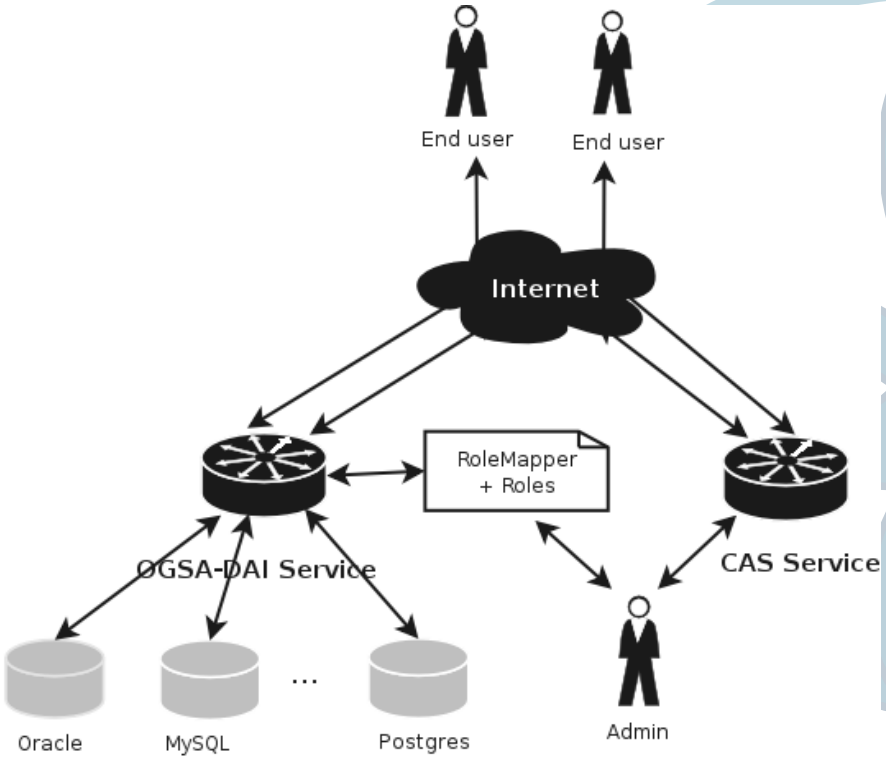
Basic OGSA-DAI Authz Model



- **Advantages**
  - Closed system
  - Local administration
- **Disadvantages**
  - Static model
  - No dynamic VO support
  - Only internal authorization possible

Slide provided by Krzysztof Kurowski, PSNC
inteliGrid project: www.inteligrid.com

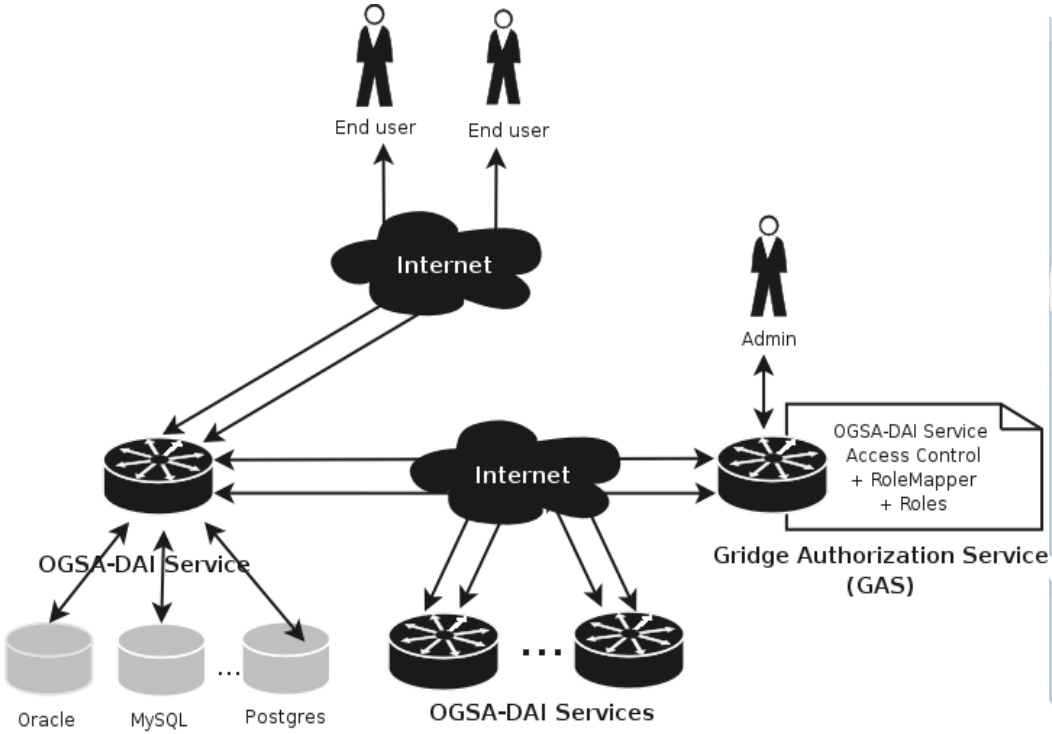
PULL (CAS) Authz Model



- **Advantages**
  - VO support
  - Fast model
- **Disadvantages**
  - Static model (as long as proxy is valid)
  - Consistent polices required in two places: CAS and Rolemapper
  - Specific user security policy for OGSA-DAI can be seen by various system components
- *Anil Pereira et al. "Role-based Access Control for Grid Database Services", Wright State University*

Slide provided by Krzystof Kurowski, PSNC
inteliGrid project: www.inteligrid.com

PUSH (GAS) Authz Model



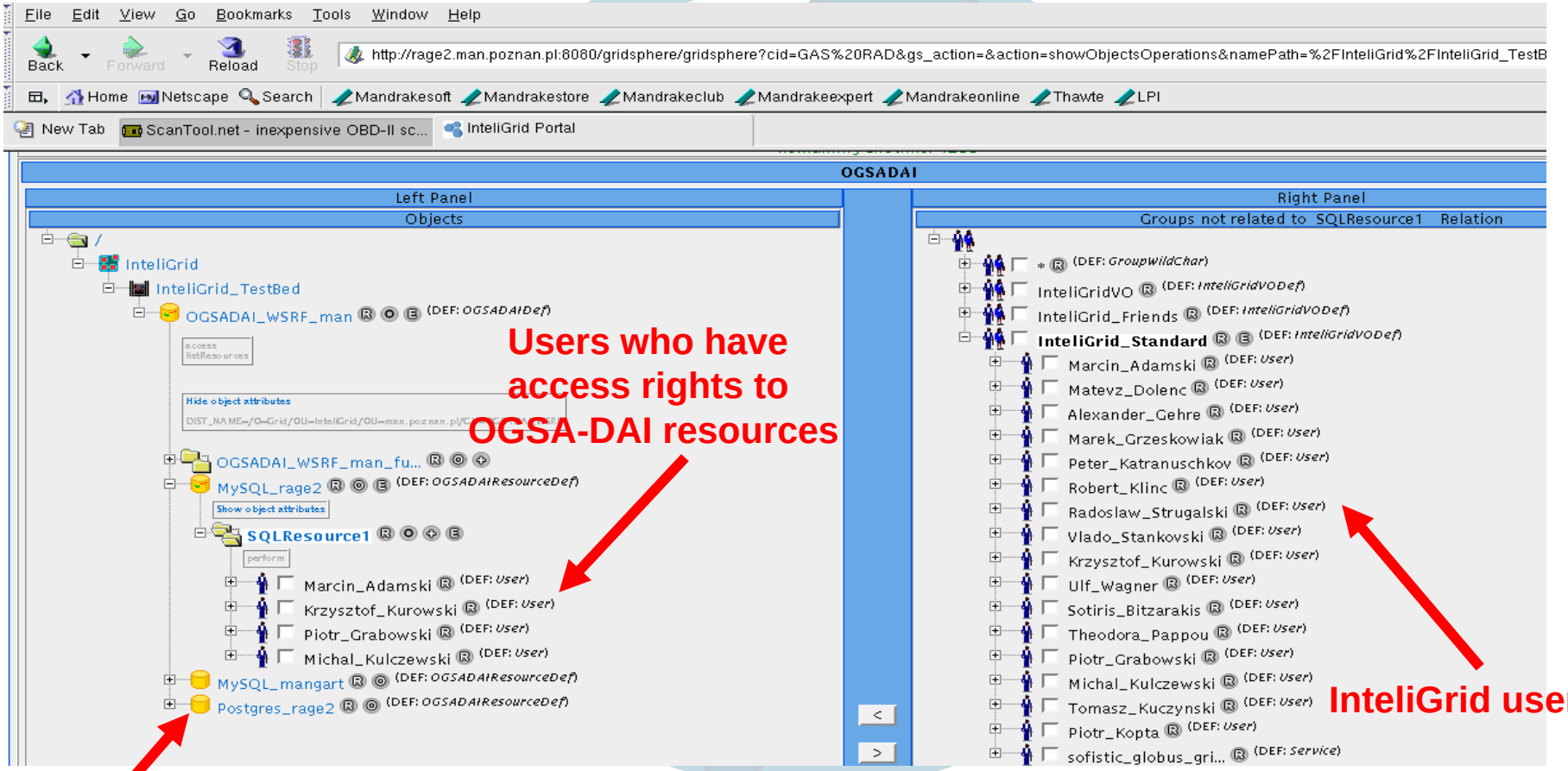
- **Advantages**
  - VO support
  - Dynamic model
  - Full security control in one place GAS
  - Real RBAC model (admin can change roles dynamically during execution)
  - No modification to source code of OGSA-DAI
- **Disadvantages**
  - Slow model (many iterations required)
  - DOS attacks possible

- Implemented in inteliGrid
- Similar PBAC model in SIMDAT

Slide provided by Krzystof Kurowski, PSNC
inteliGrid project: www.inteligrid.com

inteliGrid dynamic policy control

www.inteligrid.com



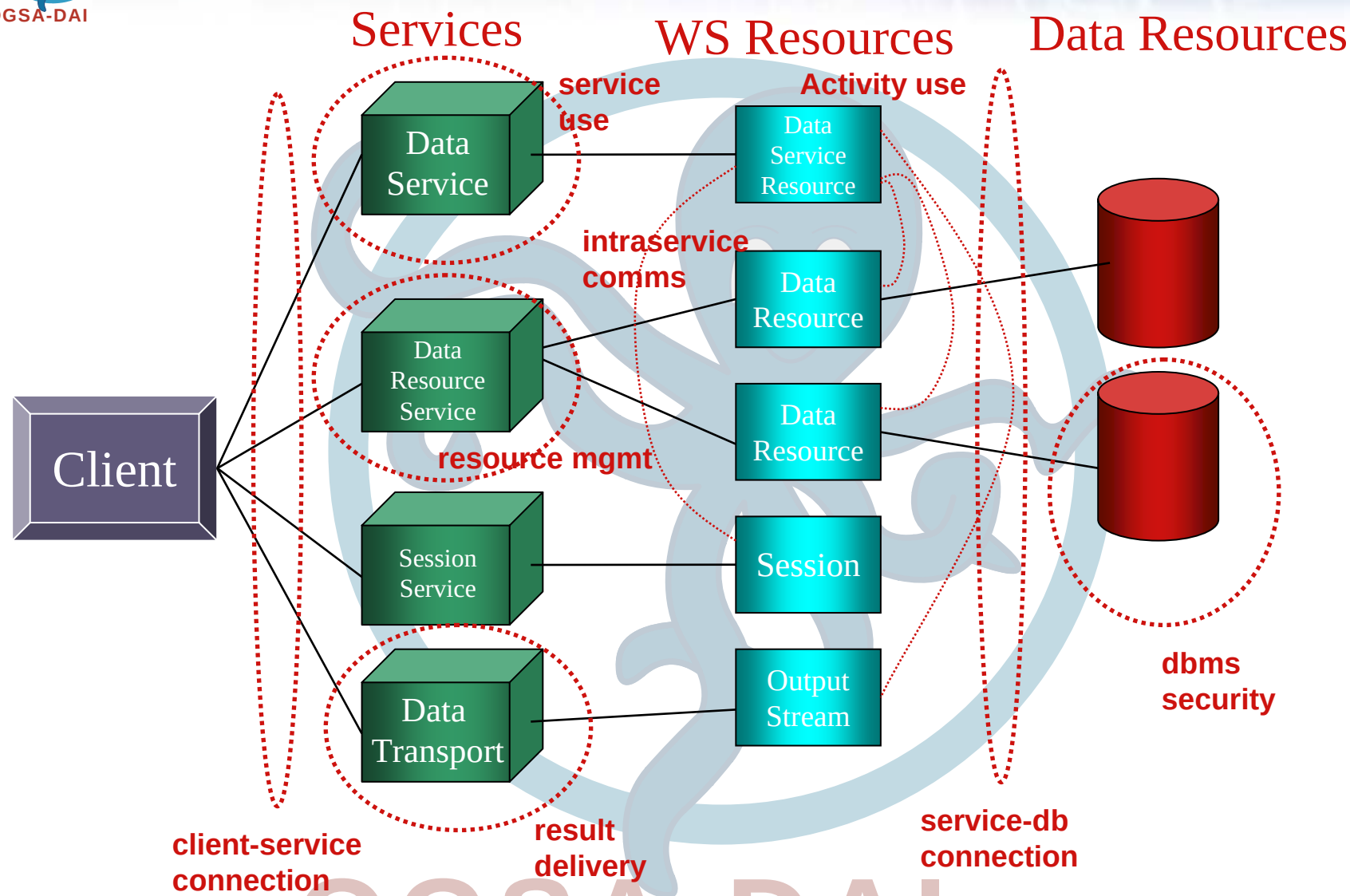
Slide provided by Krzystof Kurowski, PSNC
inteliGrid project: www.inteligrid.com

Technical details of GAS approach

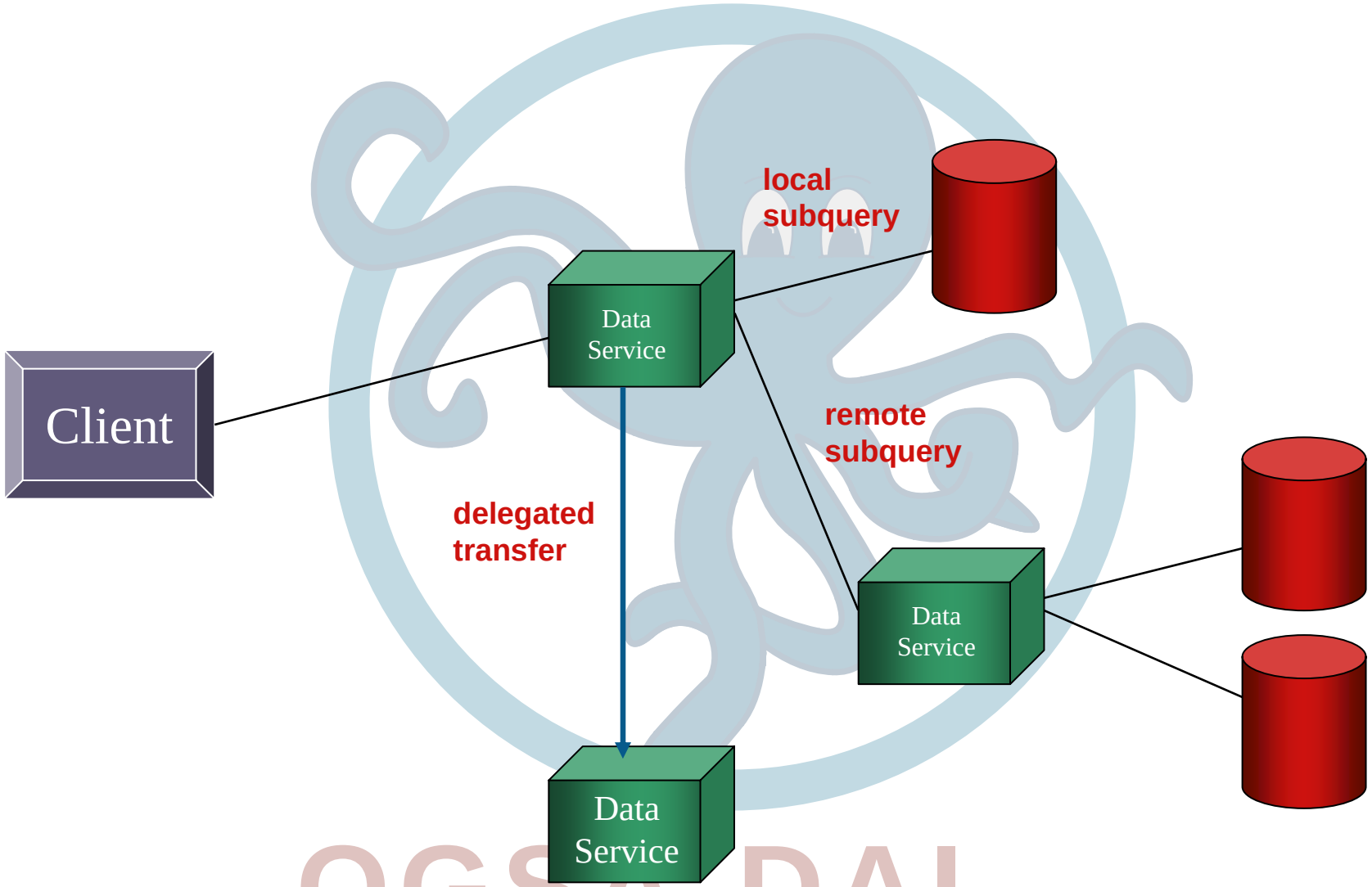
- The OGSA-DAI service is secured by standard Globus security mechanism called security descriptor. Upon OGSA-DAI service startup the security descriptor location is being read from OGSA_DAI ***server-config.wsdd*** file.
- In the ***security-descriptor.xml*** we tell Globus to authenticate users via transport security, and to authorize them via our PDP (Policy Decision Point) called GAS PDP. In security-descriptor.xml we only point, what class will be responsible for authorizing users - it must implement mehtods init (for getting initial configuration, e.g. GAS server URL) and ***isAuthorized*** (for returning authorization decision). Please note that any configuration variables GAS PDP should obtain (e.g. GAS server URL) must be placed
in OGSA-DAI server-config.wsdd.
- When OGSA-DAI receives a request, isAuthorized method of GAS PDP is called to authorize the user. GAS PDP first asks GAS server whether user is entitled to access OGSA-DAI service. Upon successful response GAS PDP ask GAS once again whether user can perform requested action. This action may be of perform document (select, insert, update etc), listResources (for listing available data service resources) or property (e.g. databaseSchema).
- Currently we do not recognize the meaning of perform documents (whether it is **select** or **update** or **insert** etc). But it is possible…
- Uppon successful response (user is authorized to access OGSA-DAI service and entitled to perform some action), RoleMapper work is about to begin (if user requested some operation on database). In most cases RoleMapper file contains mapping between user's DN and database credentials. But when role-based authorization must be done, RoleMapper file contains mapping between role and database credentials.
- The role is obtained from the GAS server - RoleMapper asks GAS for the role of the user (giving GAS server user's DN). After the mapping OGSA-DAI data service connects to data service resource database and sends back to the user the response.

Slide provided by Krzystof Kurowski, PSNC
inteliGrid project: www.inteligrid.com

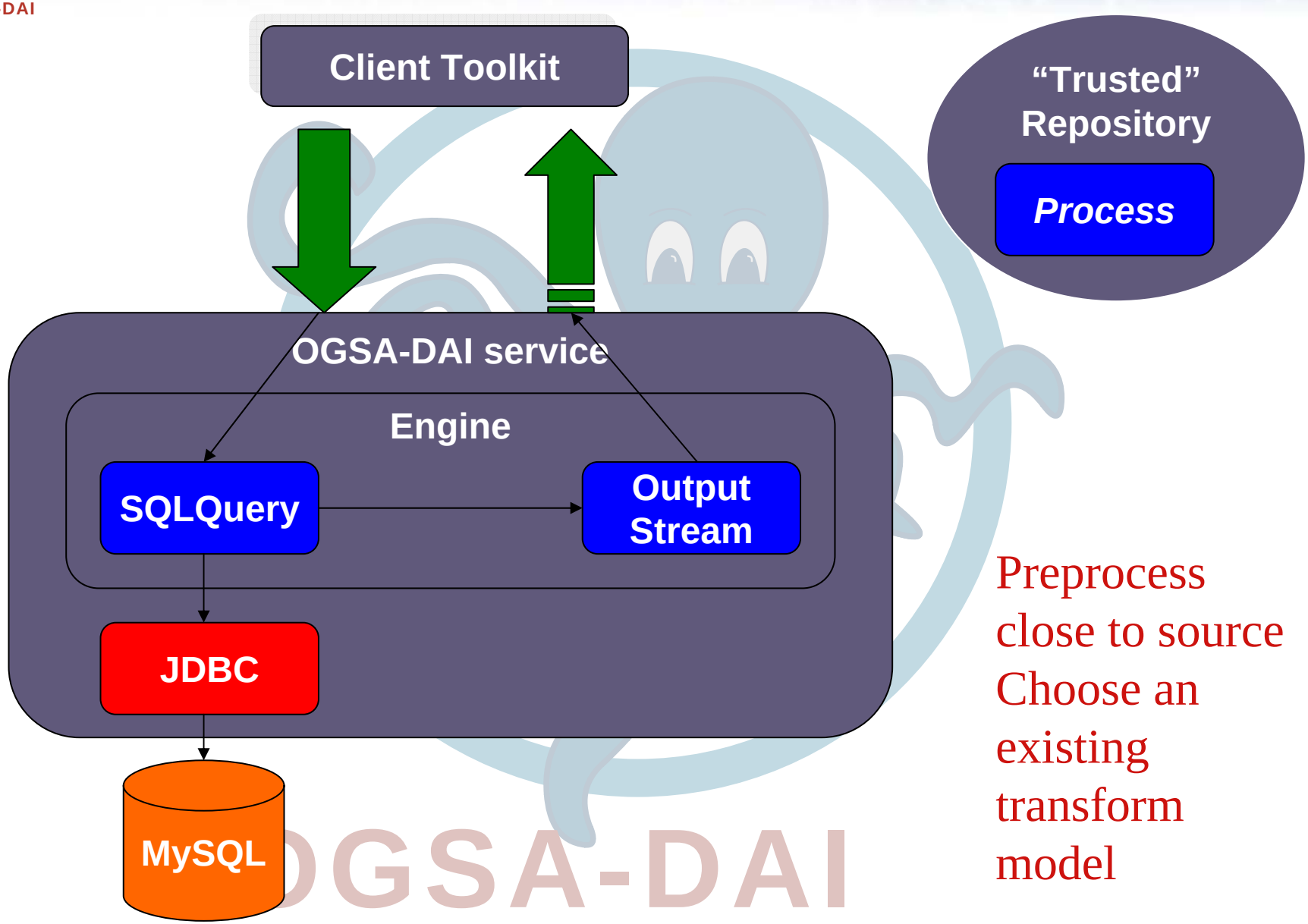
Future architecture of OGSA-DAI



Delegation in sub-querying



Repository of Activities



Preprocess close to source
Choose an existing transform model

Continuing Issues

- How to protect data whilst being processed by a service
- How to protect against macro-level security attacks

Conclusions

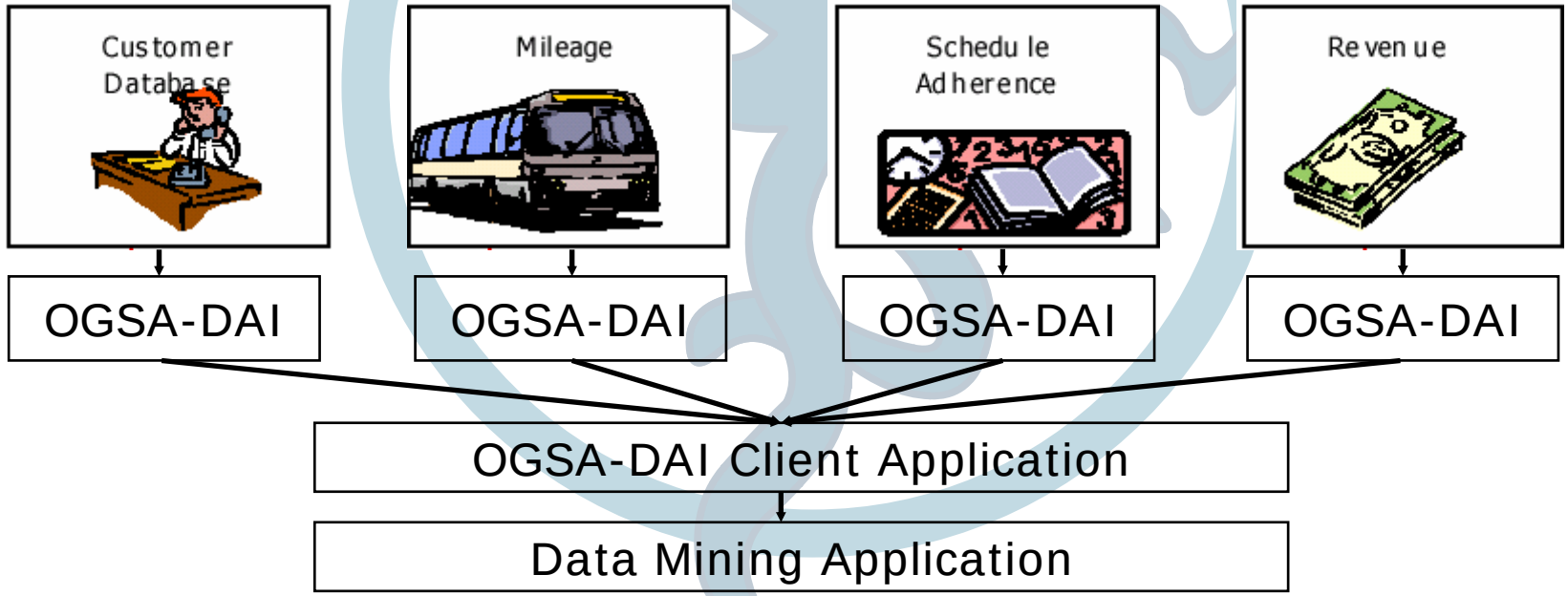
- Data Service security is a major challenge
  - perhaps legal protection is the best policy
  - Service level authorization is not sufficient
- OGSA-DAI attempts to implement a multi-level authorization model
  - a developer at the data manipulation level should not have to worry about the implementation of e.g. delegation, just the delegation model
- Other projects have extended OGSA-DAI's static authz model to provide new capabilities
  - inteliGrid, SIMDAT, Wright State University
- As we move towards new composition models for data services, we require a better trust model

Further information

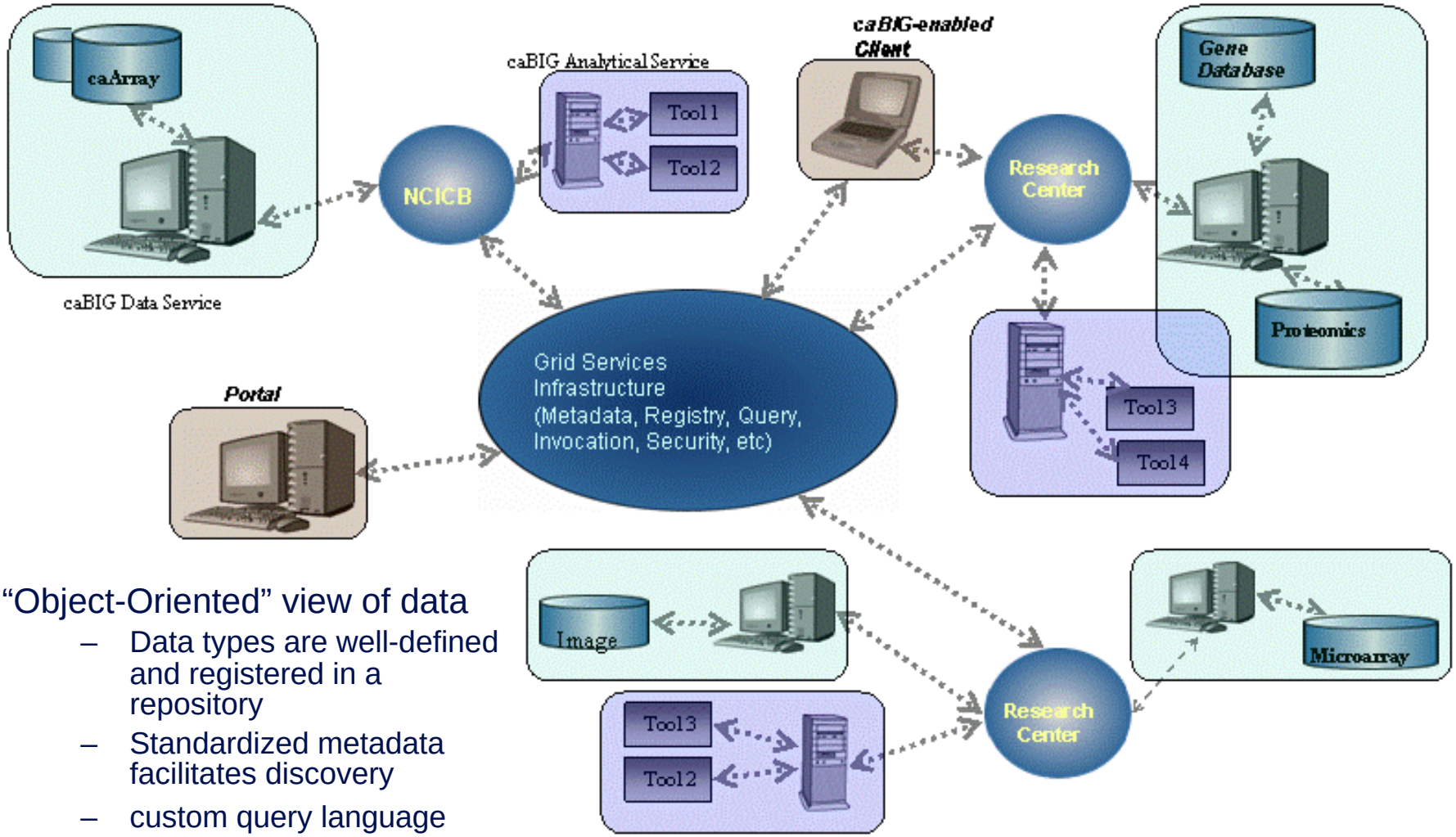
- The OGSA-DAI Project Site:
  - http://www.ogsadai.org.uk
- OGSA-DAI Users Mailing list
  - users@ogsadai.org.uk
  - General discussion on grid DAI matters
- Formal support for OGSA-DAI releases
  - http://bugs.ogsadai.org.uk/
- OGSA-DAI training courses
  - Announced on the OGSA-DAI project site
- The DAIS-WG site:
  - http://forge.gridforum.org/projects/dais-wg/

FirstDIG

- Data mining with the First Transport Group, UK
  - Example: “When buses are more than 10 minutes late there is an 82% chance that revenue drops by at least 10%”
  - *"The results of this exercise will revolutionise the way we do things in the bus industry.“,* Darren Unwin, Divisional Manager, First South Yorkshire.



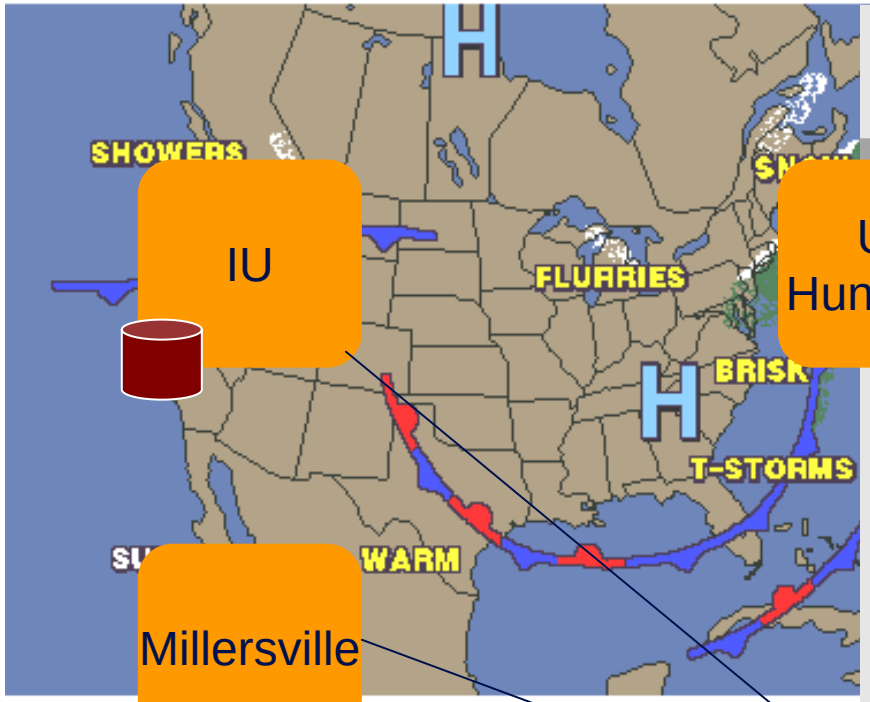
caBIG

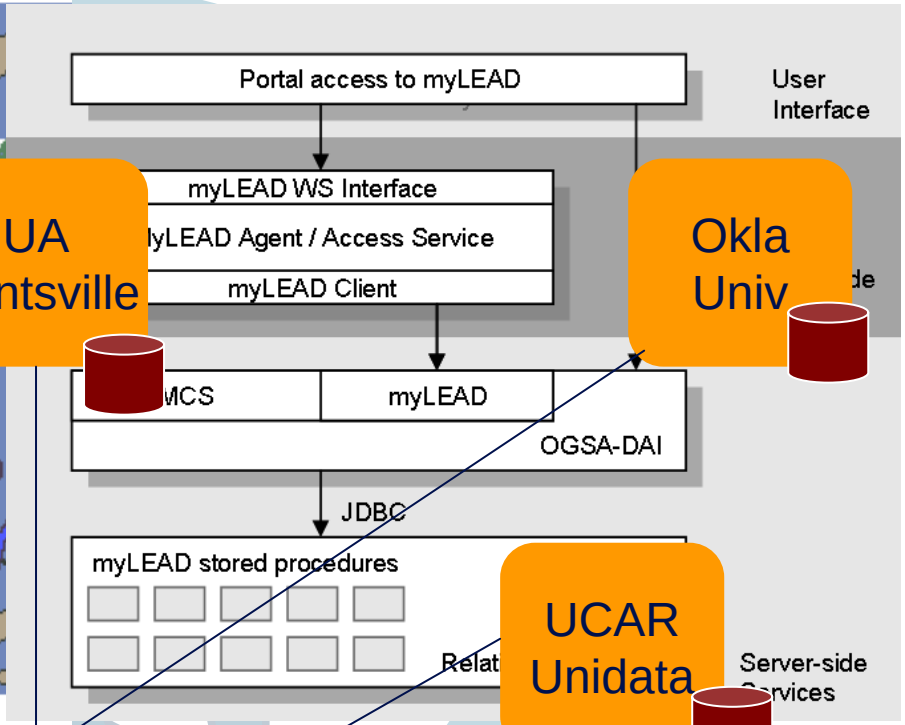


"Object-Oriented" view of data

- Data types are well-defined and registered in a repository
- Standardized metadata facilitates discovery
- custom query language implemented as an activity

LEAD





IU

UA Huntsville

Okla Univ

Millersville

UCAR Unidata

NCSA Illinois

Master catalog

Each satellite replicates its contents to the master catalog