



Enabling Grids for E-scienceE

*JRA1 Workshop Padova, 2004 November 15-17*

# Information System

Ricardo Brito da Rocha

*Relationship between WM and DM services*

*on behalf of the JRA1 DM Cluster:*

*Paolo Badino, Ricardo Brito da Rocha, Ákos Frohner, Peter Kunszt, József Kovács, Gavin McCance, Krzysztof Nienartowicz, Daniel Rodrigues*

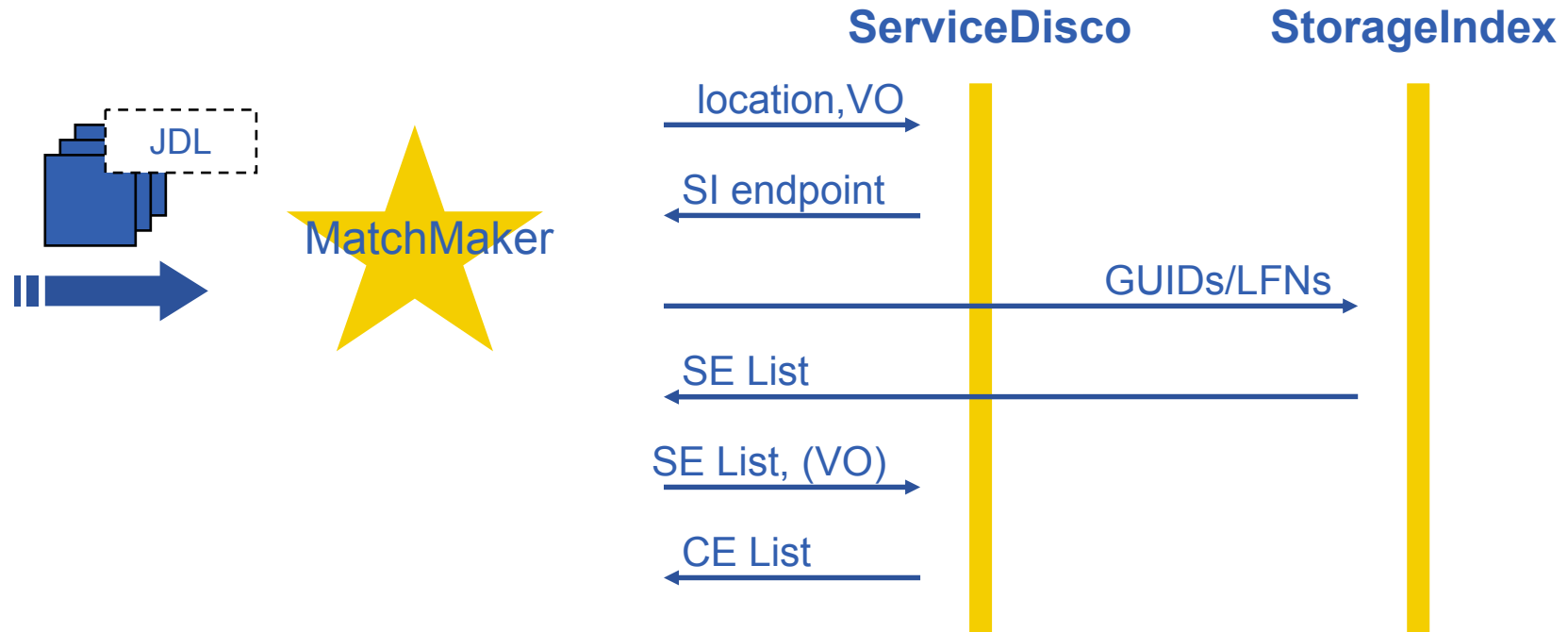
[www.eu-egee.org](http://www.eu-egee.org)



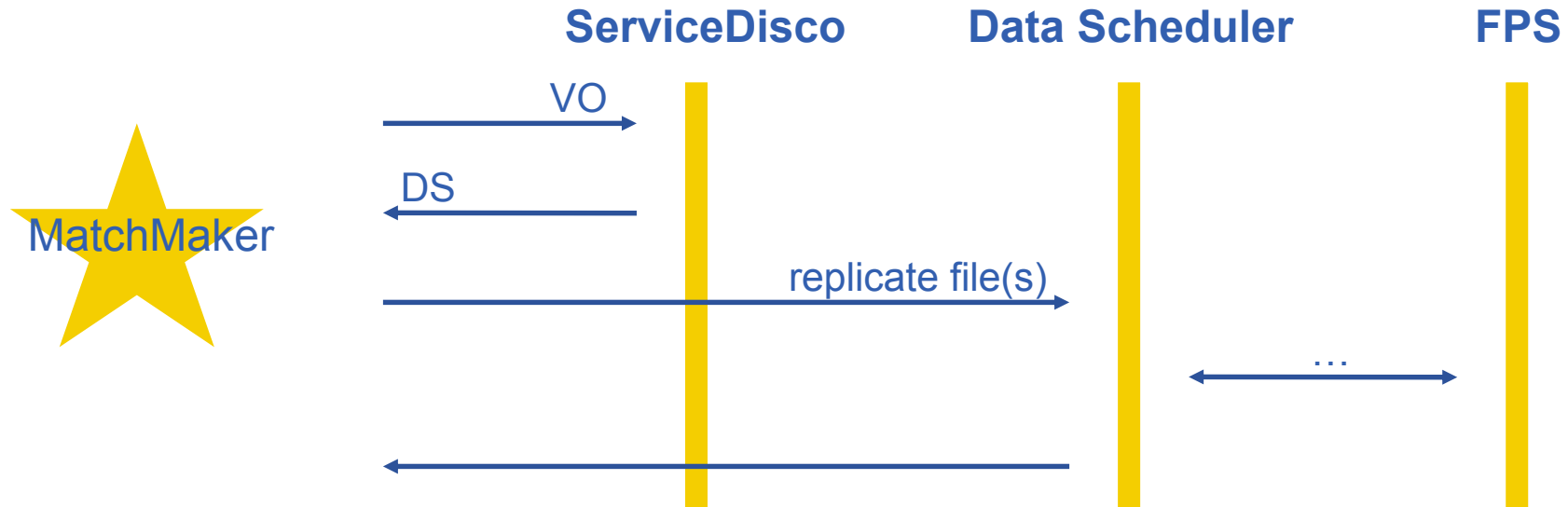
Information Society



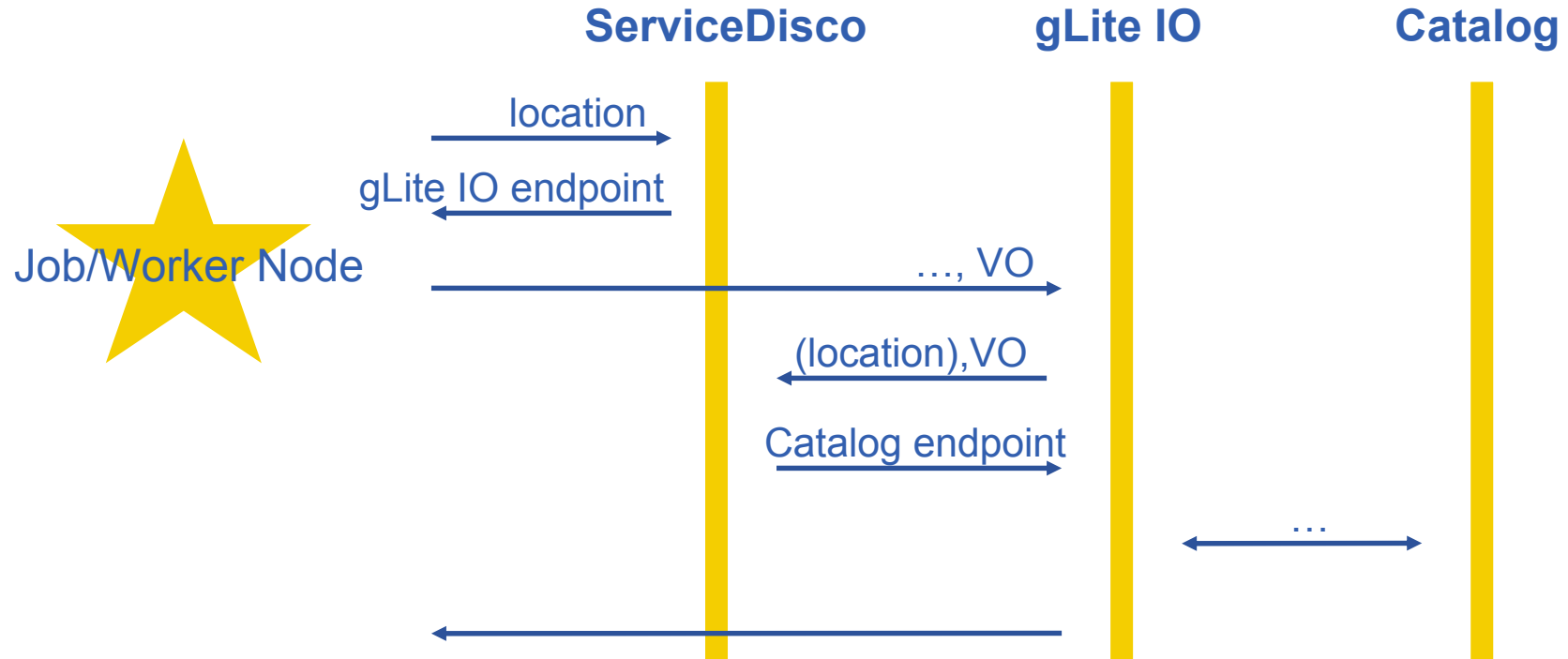
- **WMS/DM Interaction Scenarios**
- **Service Discovery Requirements/Proposal**
- **Examples**



- Is it up to the Matchmaker or the Service Discovery service to filter the CE list according to VO?
- The information on the services associated with the SE (gLite IO, FPS, ...) is queried later by the job in the WN



- **File transfer requests to DS and FPS are asynchronous**
- **Data Scheduler is per VO**
- **File Placement Service is per site and per VO**



- Both reading and writing (placing) new files
- gLite IO deployed one per SE
- Catalog can be local or global
- Similar situation if interacting with the FPS

- We Propose a unique service naming scheme
  - name.type.VO.grid.domain**
  - *type and domain are mandatory*
  - *ex: atlas.wms.glite.cern.ch, alice.ce.alien.infn.pd.it, ...*
  - *Advantages:*
    - *DNS friendly (ie. DNS itself can be used to implement SD)*
    - *DNS itself can be used to ensure uniqueness*
    - *Bootstrapping problem might be easier to solve*

- **Data Structures**

Service
String name
String type
String endpoint
String version

ServiceDetail
... (same as Service)
String site
String administrator
String[] VO
String WSDL
Service[] associatedServices
ServiceData[] data

ServiceData
String key
String value
boolean mandatory

- name, type, endpoint, site are mandatory

- **Inquiry Operations**
  - Service getService(String name);
  - ServiceDetail getServiceDetails(String name);
  - String getServiceParameter(String name, String key);
  - Additional helper methods:
    - String getServiceWSDL(String name);
    - String getServiceSite(String name);
    - ServiceData[] getServiceData(String name);
- **Query Operations**
  - Service[] listServices(String type, String site, String[] VO);
  - Service[] listServicesByData(String type, String site, String[] VO, String[] keys, String[] values);
  - Service[] listAssociatedServices(String name, String association, String type, String site, String[] VO);

- **Matchmaking**

- Get the Storage Index endpoint

- Request: `listServices("SI", "cern.ch", {"atlas"});`
- Response: `Service {"mysi.si.atlas.cern.ch", "SI", "http://cern.ch/endpoint/to/mysi", "1.0"};`

- Get the Computing Elements list

- Request: `listAssociatedServices("myse.se.atlas.cern.ch", "", "CE", "cern.ch", {"atlas"});`
- Response: `Service {"ce1.ce.atlas.cern.ch", "CE", "http://cern.ch/endpoint/to/ce1", "1.4"};`

- **Job Preparation**

- Get the Data Scheduler endpoint

- Request: `listServices("DS", "cern.ch", {"atlas"});`
- Response: `Service {"ds.atlas.cern.ch", "DS", "http://cern.ch/endpoint/to/ds", "1.2"};`



- **Access to Files**
  - Get the gLite IO endpoint
    - Request: `listServices("IO", "cern.ch", {"atlas"});`
    - Response: `Service {"io.atlas.cern.ch", "IO", "http://cern.ch/endpoint/to/gliteio", "1.0"};`