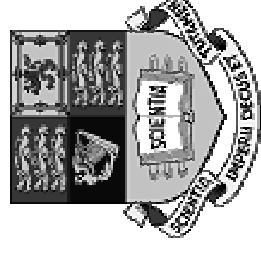


# Building Computational Communities From Federated Resources

Nathalie Furmento, Steven Newhouse,

John Darlington

Grid Middleware Group  
London - Science Centre  
Imperial College, London



Euro-Par2001, Manchester, 28<sup>th</sup>-31<sup>st</sup> August 2001

# Outline

- Computational Grids and e-Science
- Using Java & Jini as a Grid Middleware
- Exploiting GridMeta - Data
- Current Status
- Summary & Further Work

# Computational Grids & e - Science

- **Connect networked resources (the Grid)**
  - Linked instruments/storage/computation
  - Coupled computing (meta - computing - high performance)
- **Optimise use of resources**
  - Wider access
  - Better utilisation of resources (throughput)
- **High-level use (e - Science)**
  - Ease of use for domain specialists
  - Support scientific collaboration and knowledge management (connect & build communities)

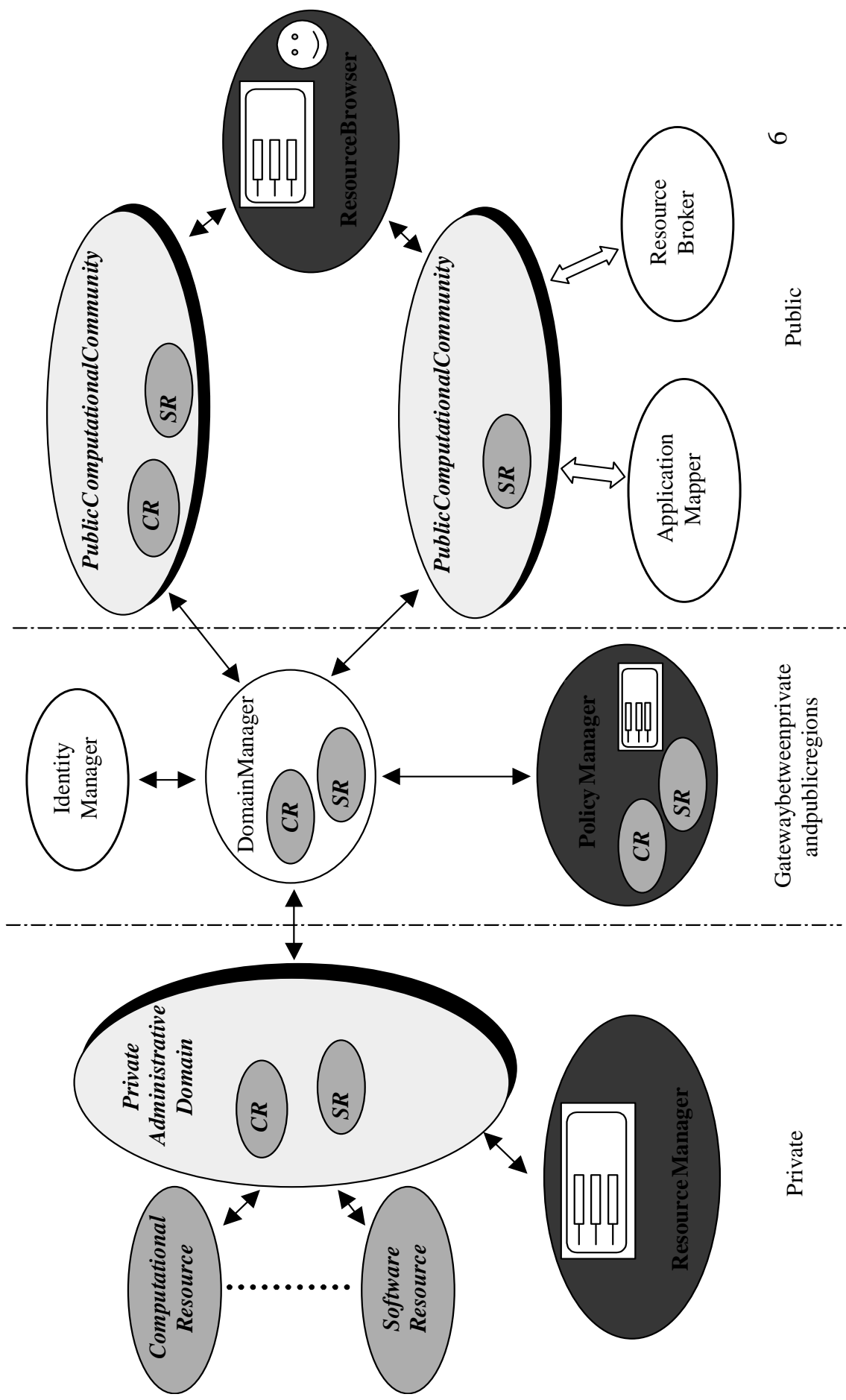
# Building Computational Communities

- By federating resources from real organisations (to form *virtual organisations* )
- Share, combine and use computational resources (software and hardware) in an open, transparent and optimal manner
- Need information about:
  - The resources
  - The users
  - The applications (See presentation 4.00, Thursday, Room D)

# Federating Distributed Resources

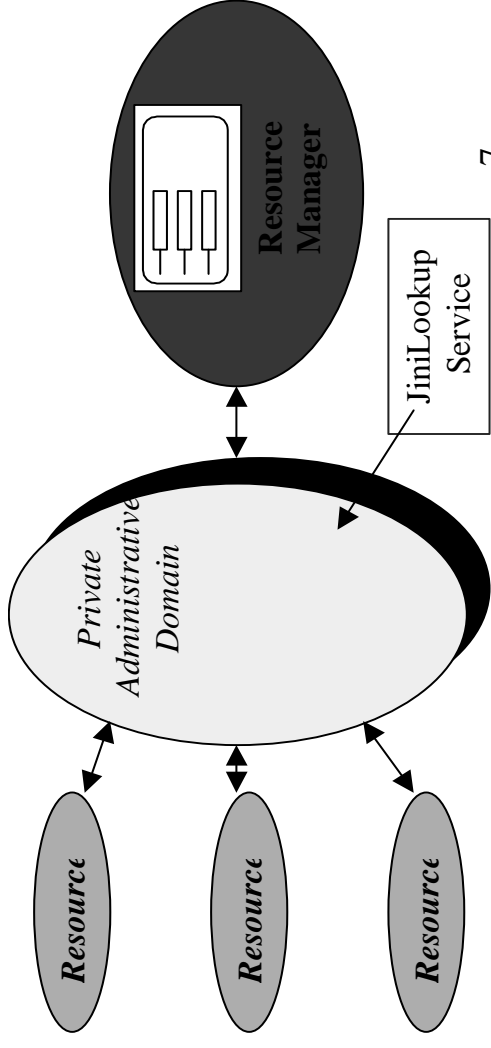
- Real Organisations
  - Need to express ownership and retain control
  - Grid resources are 'not' free – someone must pay!
  - Manage the resources as a single unit
- Virtual Organisations
  - Hide complexity from the user
  - Optimise resource utilisation for all jobs and users
  - Easy to contribute (and withdraw) resources

# Java/Jini Grid Architecture



# Managing Internal Resources

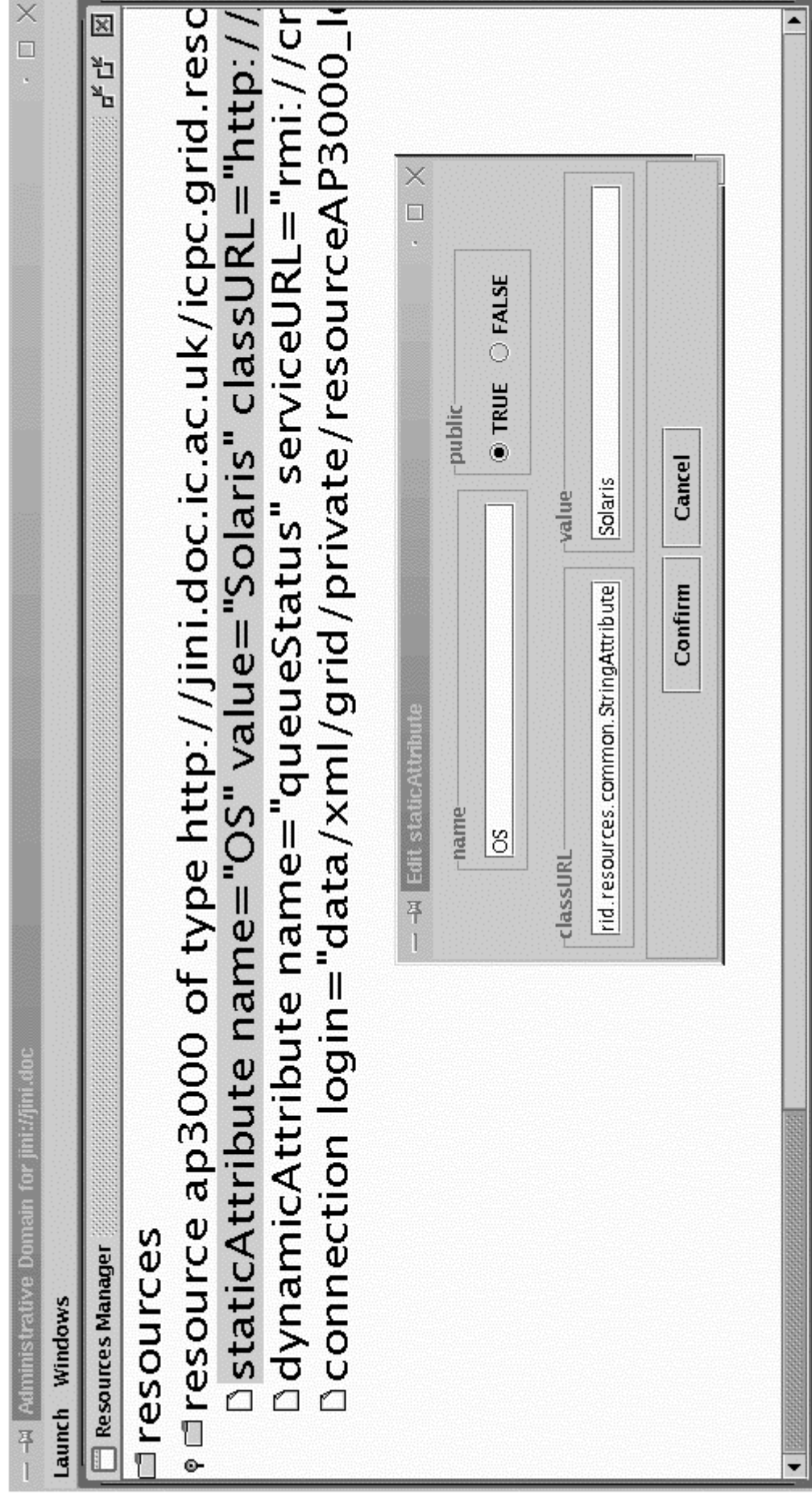
- The resources advertise their capabilities to a private domain
- Three types of resources:
  - Computational resources
  - Storage resources
  - Software resources
- Annotate using XML



# A Computational Resource: the AP3000

```
<resource name="ap3000" publish="//jini.doc/"
type="...ComputationalResource">
  <staticAttribute name="OS" value="Solaris"
    public="TRUE" classURL="...stringAttribute"/>
  <dynamicAttribute name="queueStatus"
    serviceURL="rmi:...getAP3000QSStatus"
    classURL="...AP3000" public="TRUE"
    interval="5" property="...queueStatus.prp"/>
  <connection login="...AP3000_login.xml.enc"/>
</resource>
```

# TheResourceManagerGUI

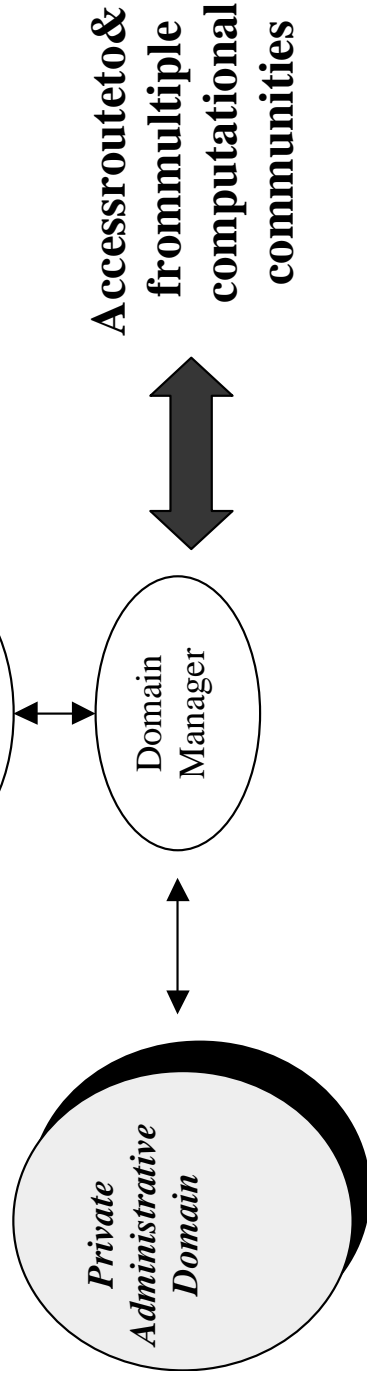


# The Domain Manager

Sole route between the private and the public areas of the infrastructure

Private → Public  
Imposes local usage requirements and access and publicly advertises its resources

Public → Private  
Authenticates requests to use the local resources (through the Identity Manager)



# Trusting Organisations & Users

- Recognise three entities: individuals, groups and organisations (or domains)
- Entities verified through public key architecture (Identity Manager)
- Locally managed access control list determine which entities have access to local resources
- Non-local users can be mapped to individual, single, or guest accounts (resource dependant policy)

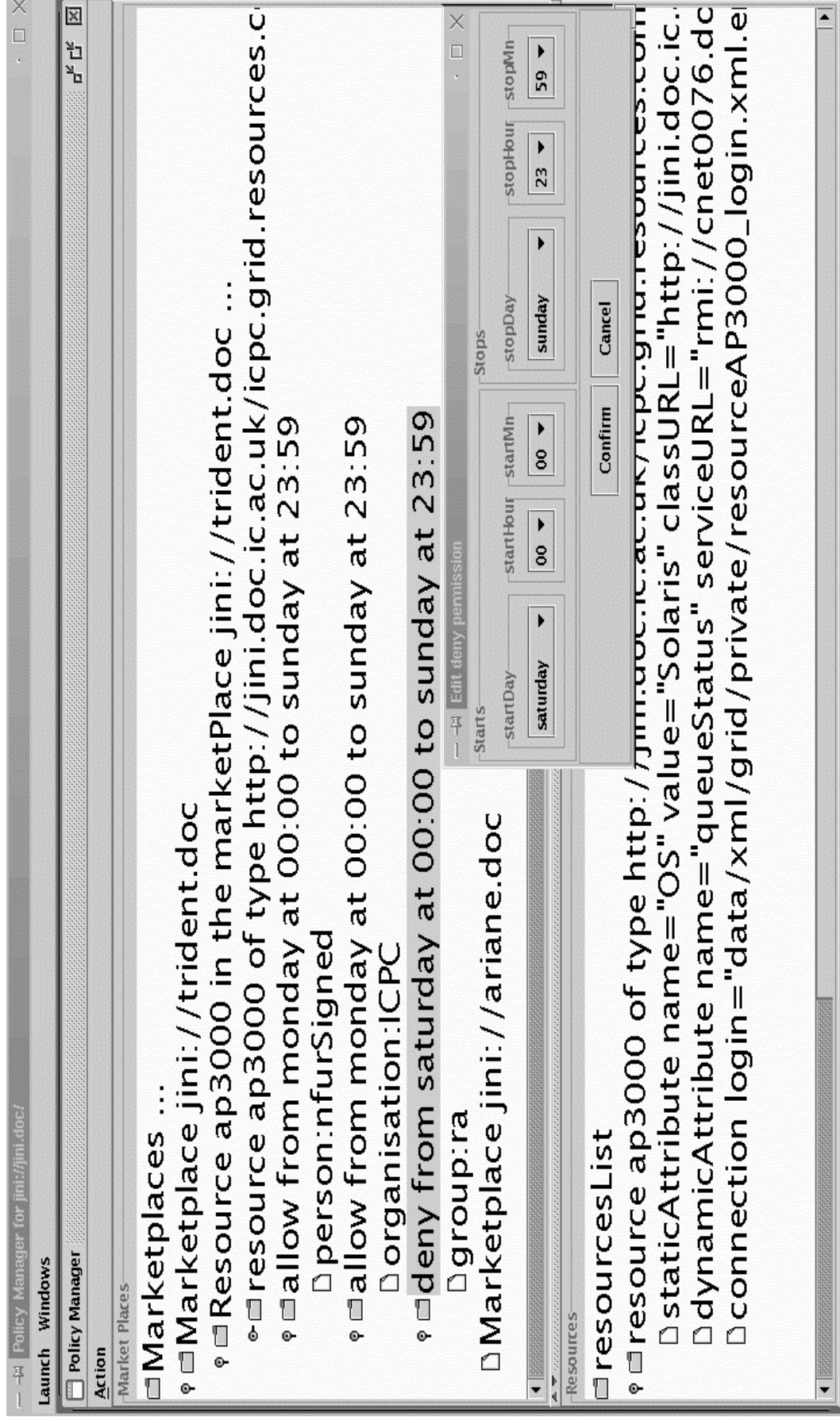
# AccessControllist

```
<domainManager manage="jini://jini.doc/" name="icpc">
  <promote domain="jini://trident.doc">
    <deny startDay="saturday"> <entity type="group"
      name="ra"/> </deny>
    <allow> <entity type="organisation" name="ICPC"/>
  </allow>
  <resource name="ap3000">
    <allow> <entity type="person" name="nfuncsigned"/>
  </allow>
</resource>
<resource name="atlas"/>
</promote>
<!--DomainManager promotes resources in a second JINI -->
<promote domain="jini://ariane.doc"> .. </promote>
</domainManager>
```

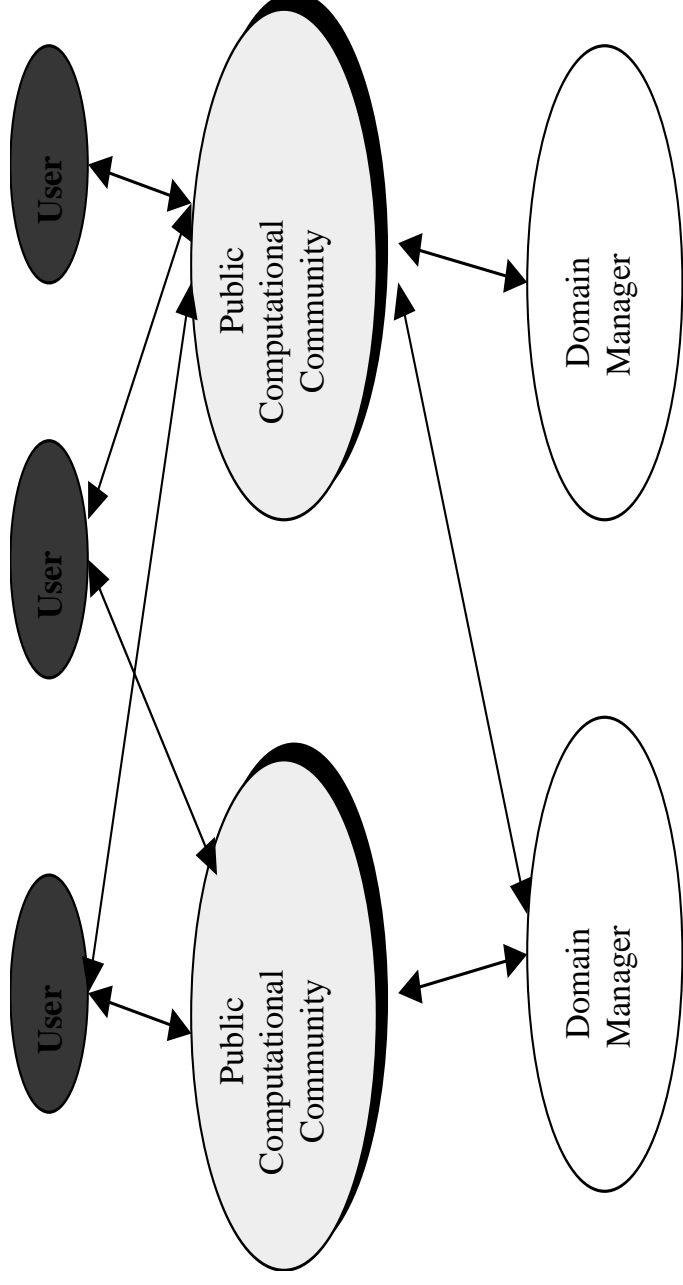
# Individuals, Groups & Organisations

```
<identificationOrganisation
  publish="jini://jini.doc" name="ICPC"
  keystore=". /data/security/keystore/keystore.ICPC">
  <group name="ala">
    <user name="aem3signed" />
    <user name="asm100signed" />
    <user name="nforsigned" />
  </group>
  <group name="ra">
    <user name="asm100signed" />
    <user name="nforsigned" />
  </group>
  <knownOrganisation name="inria" />
</identificationOrganisation>
```

# The Policy Manager GUI

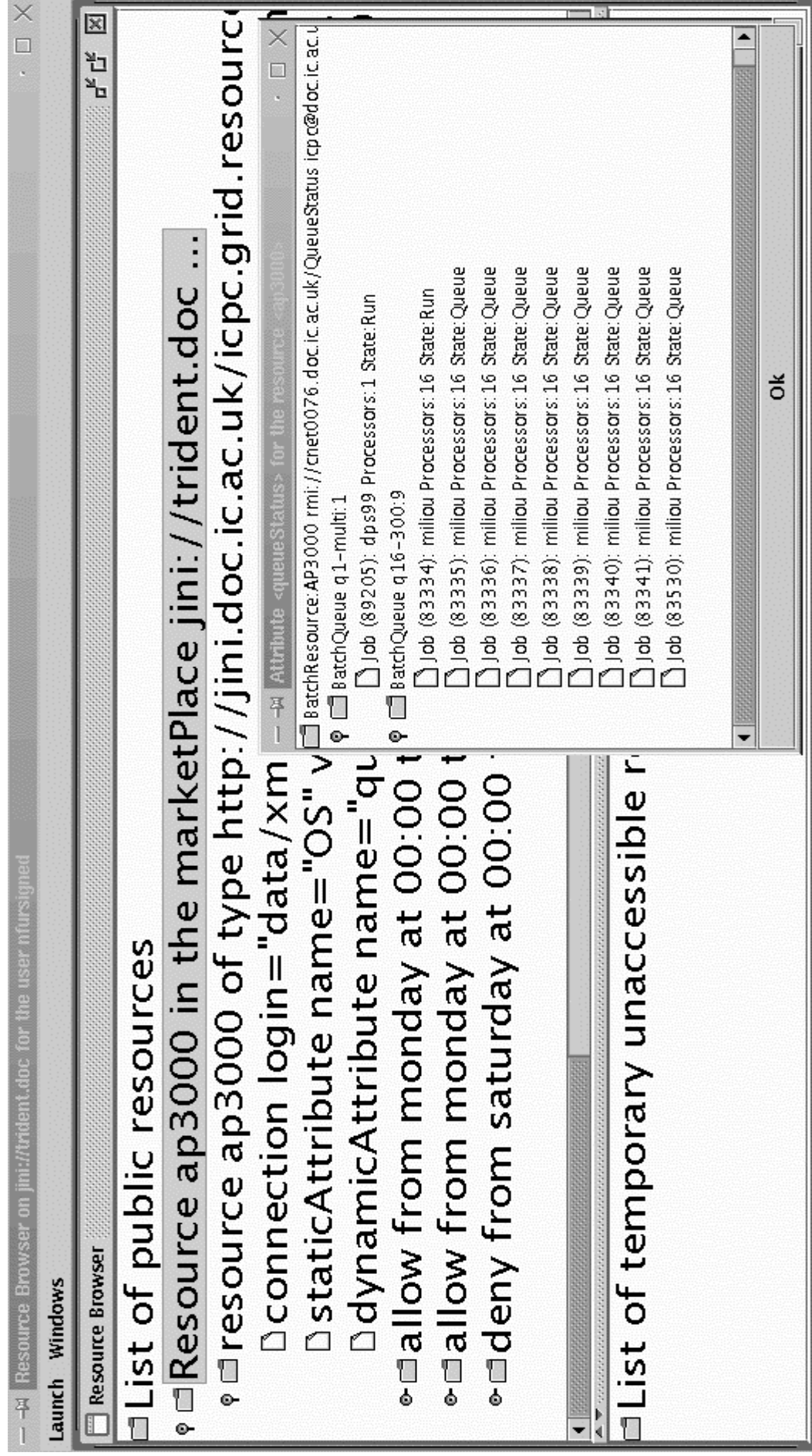


# Building Computational Communities



The Computational Community allows Users  
& Domain Manager to advertise  
their capabilities and requirements.

# TheResourceBrowserGUI

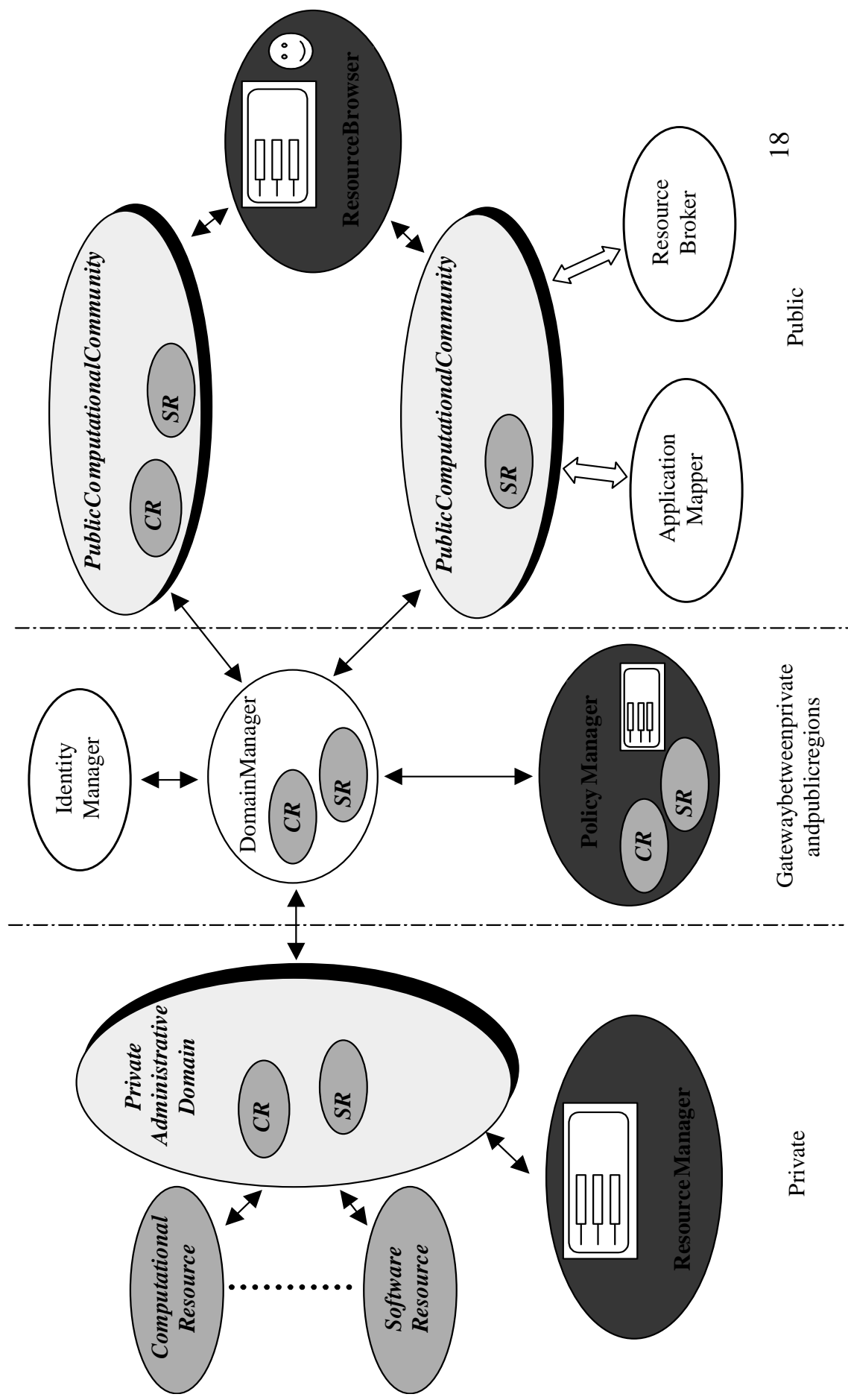


# Resources in a Computational Community

A resource might be:

- *Available* → shown with all its public attributes and access policies. It's possible to get the values of the dynamic attribute(s) and to connect to the resource
- *Temporarily unavailable* → only the access policies are shown
- *Always unavailable* → no information is shown

# Java/Jini Grid Architecture



Private

Gateway between private and public regions

Public

# Exploiting GridMeta -data

- The Computational Community contains information:
  - On resource capability, usage and access policy
  - On the user's job and their application
- Use this information to ensure:
  - The user's job is run as specified (e.g. deadline)
  - All resources are fully utilised (if possible)
  - The mapping of jobs to resources is 'optimal'

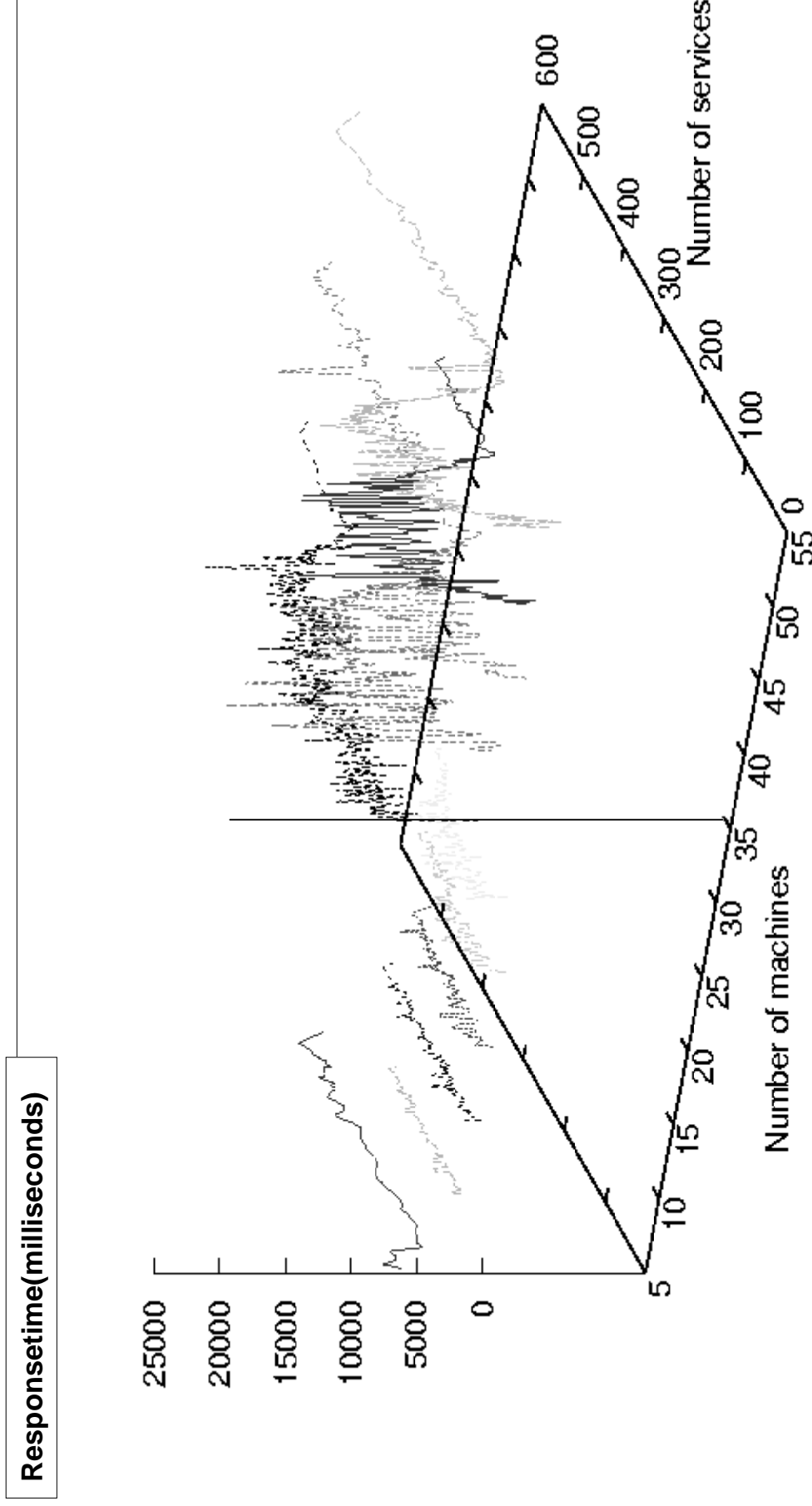
# Higher-level Grid Services

- **Application Mapper**
  - Use resource and application information to optimise resource selection for an application  
(e.g. are 16 PCs better than 8 Alpha processors?)
- **Resource Broker**
  - Use computational economic to balance priorities  
(e.g. 16 underused PCs may cost less for longer than 8 subscribed Alpha processors)
  - Factor queue length into resource selection  
(e.g. Will a slow loaded resource provide shorter job turnaround than a faster but heavily loaded resource?)

# Current Status

- Improving functionality & performance
- Merging Application Mapper and Resource Broker infrastructure into middleware
- Allow higher -level services to 'plugin' to the computational community
- Simulated loading using local Condor pool

# Response to middleware underload



Server: 300MHz PII 160Mb

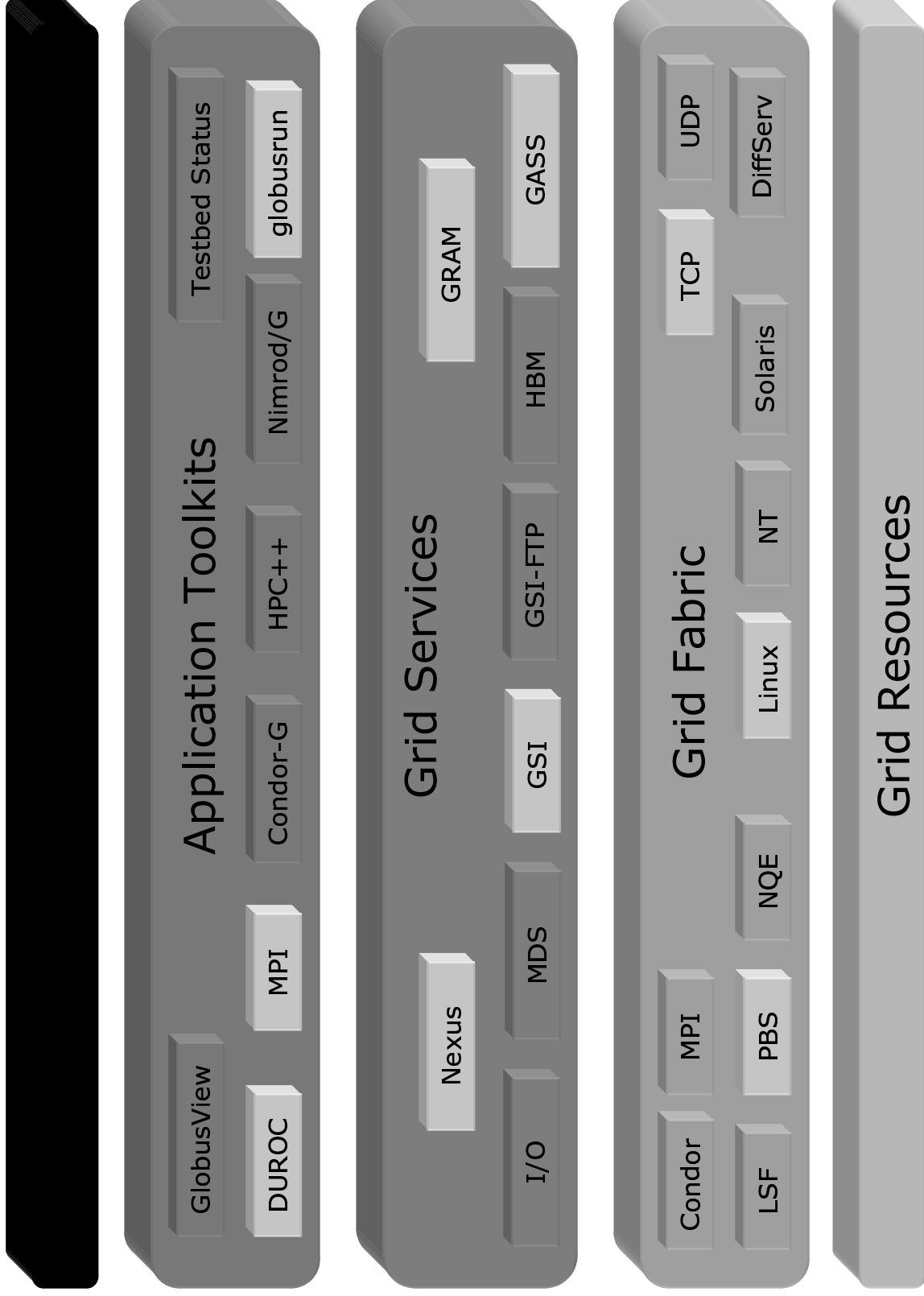
# Summary

- Use Java & Jini to federate Grid resources
- Hold meta-data from the user, their applications and the resource to guide decision making
- Use the meta-data to provide automatic higher-level services to hide complexity from the user
- Application Mapper and Resource Broker enables the Scientist to fully exploit the resources within their computational communities

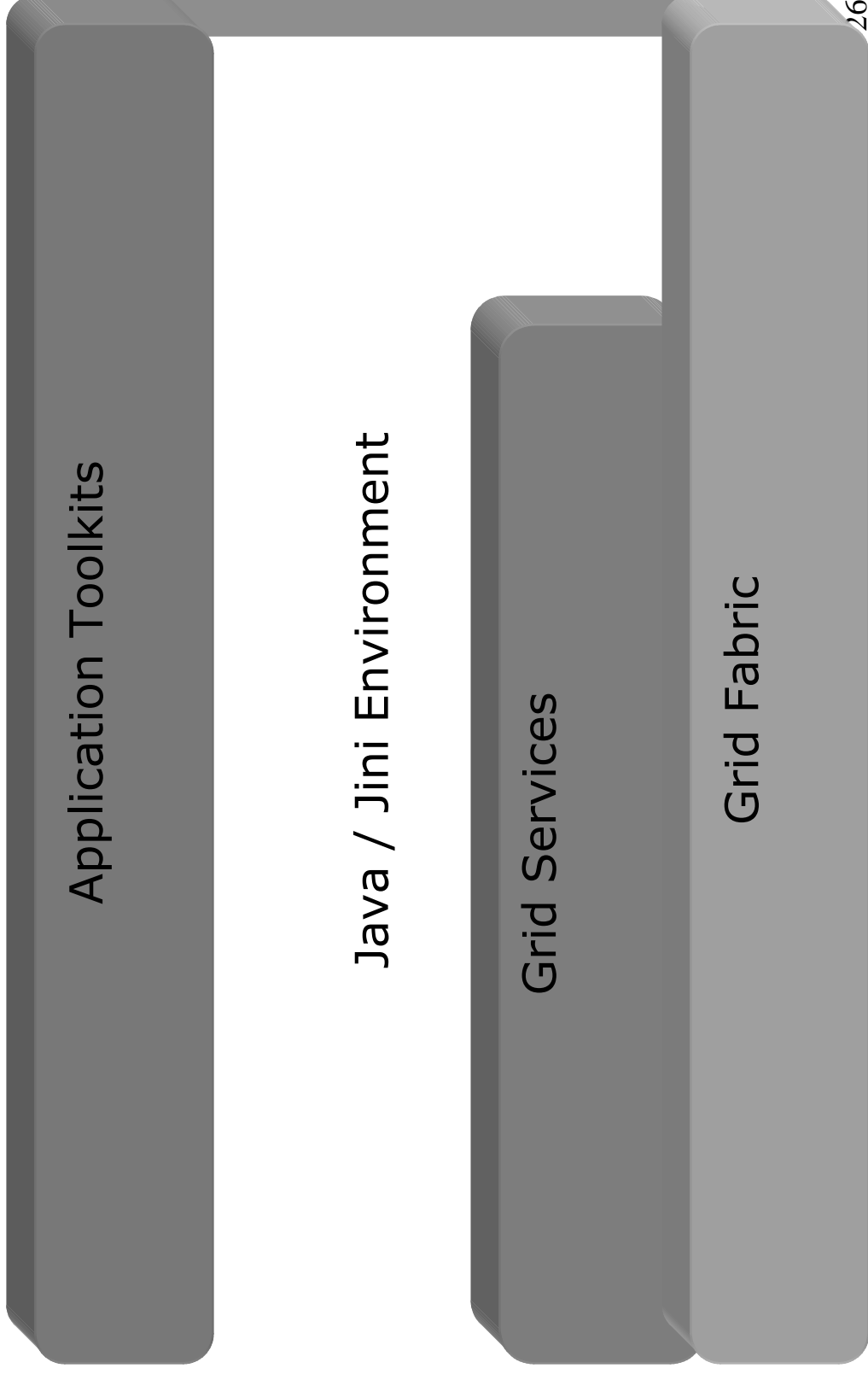
# Future Work

- Integration with other execution environments
- Developing code and user level documentation
- Integration with other Grid Infrastructures

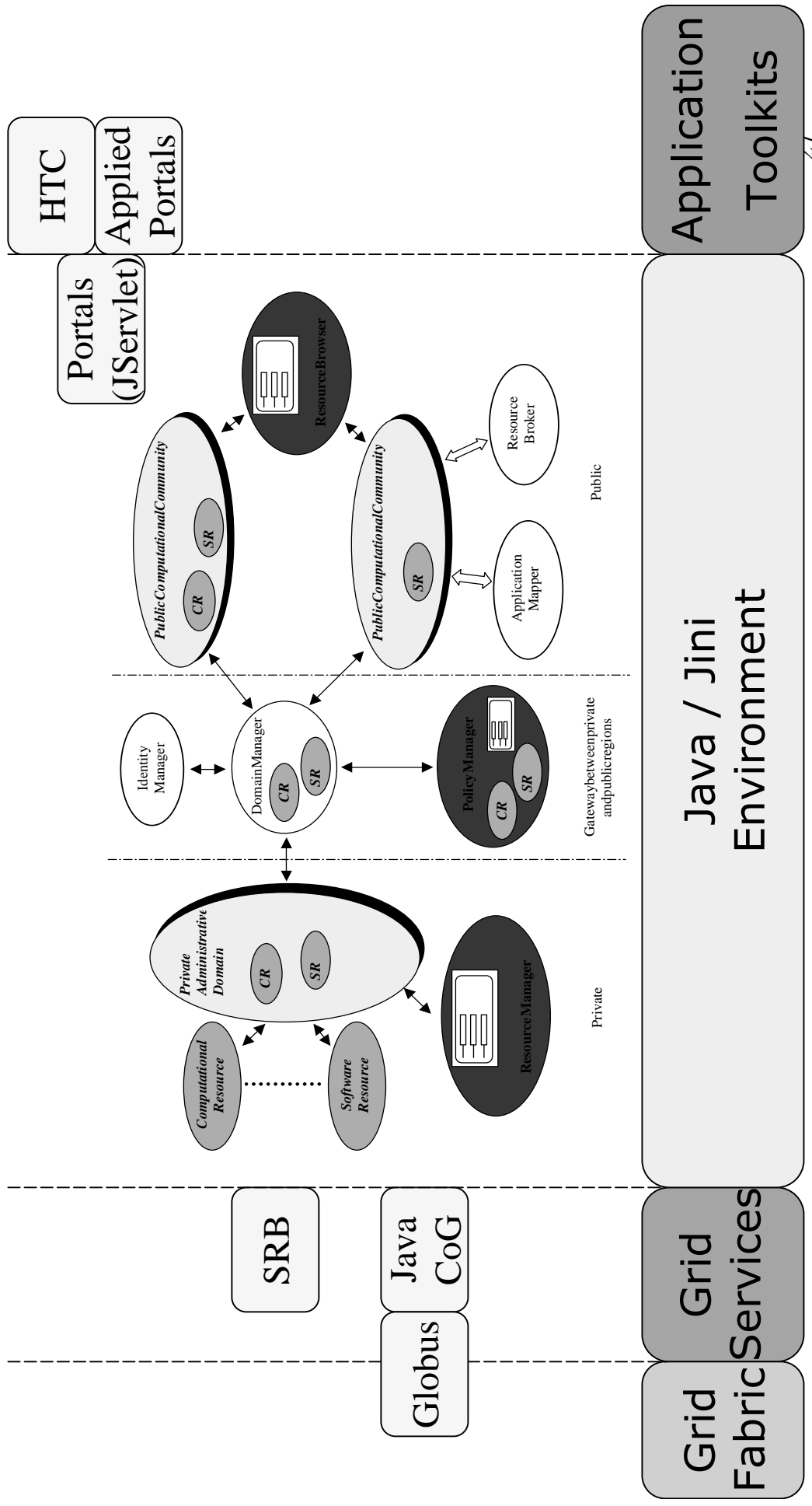
# Layered Grid Architecture



# Exploit Layered Architecture



# Java and Jini Within the Grid



# Acknowledgements

- Londone -ScienceCentre, GridMiddlewareGroup:
  - JohnDarlington
  - Steven Newhouse
  - AnthonyMayer
  - Nathalie Furmento
  - Stephen McGough
  - TonyField
- Furtherinformation:
  - <http://www-icpc.doc.ic.ac.uk/components/>
  - Contact: [icpc-sw@doc.ic.ac.uk](mailto:icpc-sw@doc.ic.ac.uk)
  - Funding:EPSRCGR/N13371