Interfacing ARC and UNICORE

Csaba Anderlik,
Bergen Center for Computational Science,
University of Bergen
Motivation:

- some users prefer to use GUIs
- use the advanced features of the UNICORE client to access ARC managed resources
- platform independent access to ARC resources
- easy portal type interface through plugins
Uniform Interface to Computing Resources (UNICORE)

History:
• development started in 1997, German projects: UNICORE/UNICORE plus
• continued in European projects: EUROGRID, GRIP, OpenMoleGRID, UniGrids, NextGrid.
• opensourse software, available from: http://unicore.sourceforge.net

Highlights:
• support for batch applications
• heterogeneous meta-computing
• transparent data transfer/staging
• command line and graphical interface for job creation and management
• workflow support
Challenges (differences):

**ARC**
- GSI based security model: proxy certificates included with a job -> dynamic delegation
- dynamic resource description
- resource discovery through Globus MDS
- data transfer through modified Globus GridFTP

**UNICORE**
- X509v3 certificates, job objects are signed with the user’s private key -> static delegation
- static resource description
- resource discovery using HTTP and UPL
- data transfer using JAVA Zip streams (alternatively, Globus GridFTP)
Solution:

- proxy plugin for the UNICORE client
- currently on the server side the interfacing is on the TSI level
- UNICORE resource description translated to XRSL and submitted through the ARC standalone client
- properly it should be on the NJS level using jarclib
- resource description from ARC is not propagated back to the client