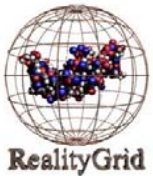
Web Services based Environments for Computational Science on Grids

Centre for Computational Science

University College London



open middleware infrastructure institute

Lightweight Middleware

Grid Computing - **distributed computing performed transparently across multiple administrative domains.**

A general computational grid should provide easy access to many different types of resources to enable users to pick and choose those required to achieve their intended scientific objectives.

WEDS

- We have developed WEDS--Web services hosting Environment for Distributed Simulation
  - Minimal dependencies on third-party software
  - Small learning-curve for new users – removing the need to learn new programming methods
  - Interoperable with other WSRF implementations
- Written in Perl, easy to install in user space, easy to modify
- Uses WSRF::Lite (interoperable with other WSRF implementations)
- "good enough" for rapid adoption, rather than waiting for a solution which will, supposedly, suit all needs.

About WEDS

- Developed to make life easier for application scientists
- Easy to deploy – sits inside a WSRF::Lite container, has no additional software requirements
- For use within an administrative domain
- Provides all the tools and glue required to:
  - expose an unaltered binary application as a service
  - create and interact with service instances

PolySteer Application

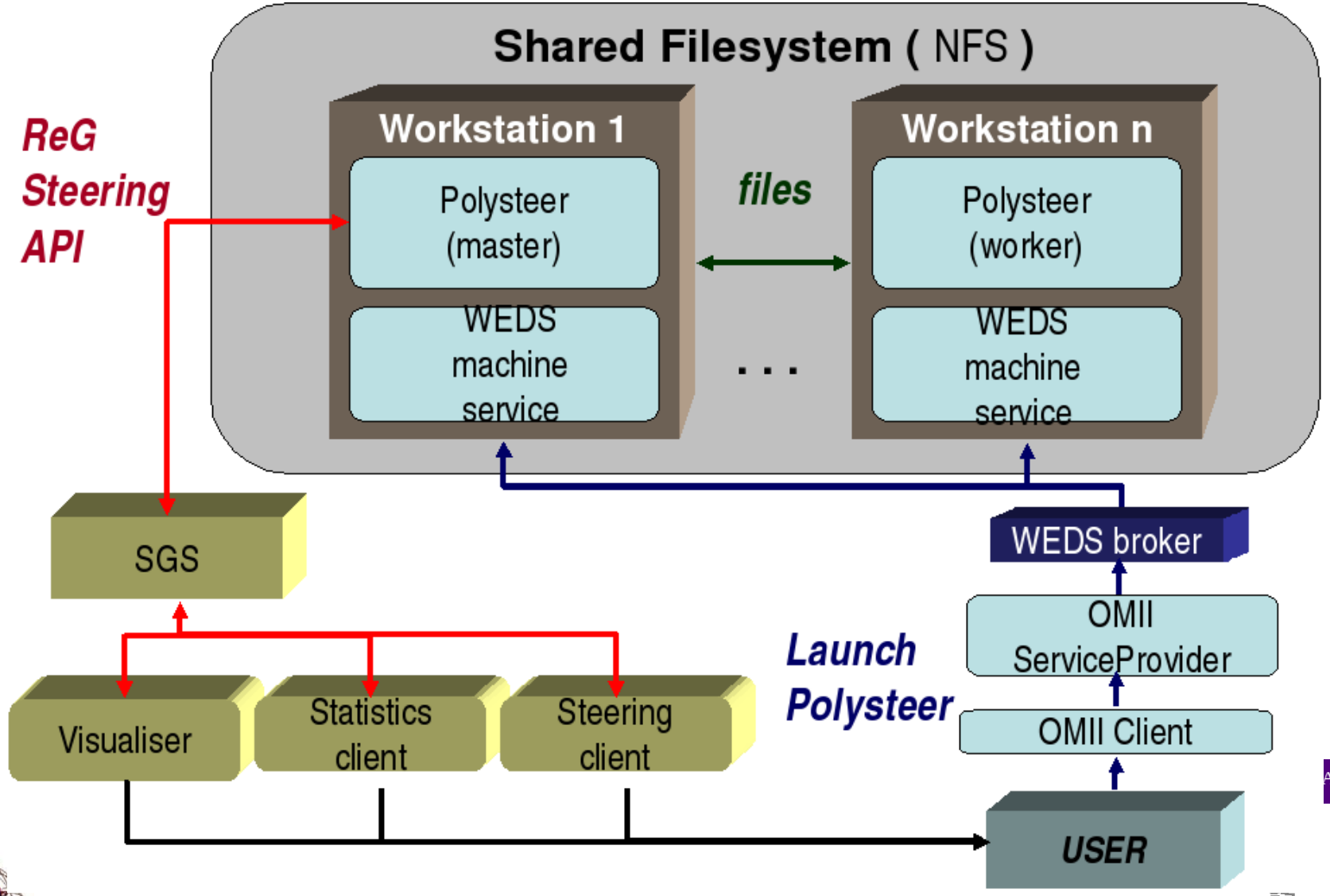
- Developed by Daniel Mason (Imperial; ReG partner)

- New Monte Carlo polymer simulation code
  - *Create as many chain conformations as possible*
- Task farming of configuration generation

- Equilibration is difficult from arbitrary start point
  - *Need to watch chains relax*
- Attach visualisation client

- Monte Carlo moves are complex
  - Modify parameters on-the-fly to optimise efficiency
- Attach steering client

Polysteer Application hosted in WEDS



About WEDS

- Lightweight middleware such as WEDS greatly facilitates deployment of applications on grids

- We're now working with several "computational user communities" from physics through to biology

- WEDS is in the public domain: Download from **www.realitygrid.org/WEDS**

AHE - Application Hosting Environment

- Based on the ideas of WEDS – applications as Web Services

- Lightweight hosting environment for running unmodified applications – NAMD, LB3D, LAMMPS on grid resources – NGS, TeraGrid

- Makes provision of a service oriented architecture a pleasant experience for all

AHE - Application Hosting Environment

- Purpose
- Design Constraints
- Functionality
- Architecture and Components
- AHE Client Interaction – Stefan Zasada

AHE - Application Hosting Environment

- AHE is based on the concept of deployed applications on the grid
- Starting applications is not the same as executing jobs
- Applications could be composed of multiple sub jobs eg a coupled models or an application that uses remote visualization

Application Hosting Environment - Applications not jobs…

- Application Instance/Simulation is central entity; represented by a stateful WS-Resource. State properties include -
  - simulation owner
  - target grid resource
  - job ID
  - simulation input files and urls
  - simulation output files and urls
  - job status
- Simple clients – limited dependencies.

AHE - Design Constraints

- Client does not have globus
- Client is firewalled
- Client does not have to be a single machine
- Client needs to be able to upload input files, doesn't support GridFTP
- Client needs to download output files (…and upload them to another resource)

AHE - Design Constraints

- Client doesn't know how to run the application on the grid resource -
  - Where is the executable on machine X?
  - What environment variables need to be set on machine Y to run the application?
  - How many CPUs can I use on machine Z?
- Client doesn't know about changes to the backend resources – e.g. if it moves from GT2 to GT4

AHE Functionality

- Query Application Registry to find list of applications AHE supports – NAMD, LB3D, LAMMPS
- Query for grid resources with NAMD installed, other constraints like CPUs, memory, NGS or Teragrid.
- Specify parameters and Start the Simulation
- Stage input files from
  - Localhost, webdav, gridftp server

AHE Functionality

- Stage output files to
  - Local host, webdav or gridftp server
- Simulation Registry to store list of simulation references.
- Query simulation state properties using the simulation reference
- monitor or terminate using the simulation reference

Application Hosting Environment

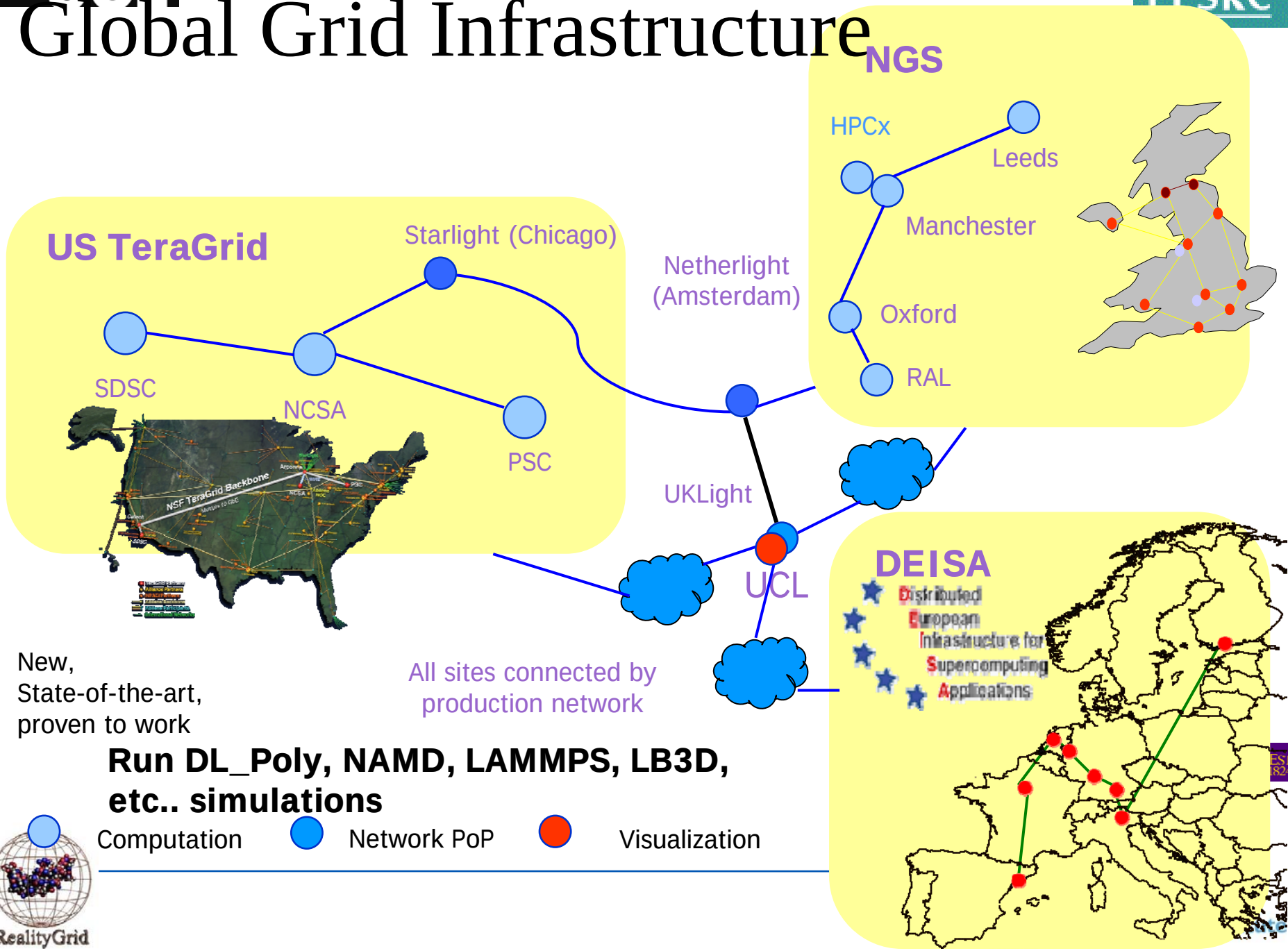
- Use Case
  - launch simulations on multiple grid resources(> 2-3 sites)
  - single interface to monitor and manipulate all simulations launched on the various grid resource..
  - some of which don't allow SSH access
  - use different machines to log in – simulation state is not saved on the client side – demo conditions(?)
  - enable collaboration – share provenance info, simulation results

AHE - Design considerations

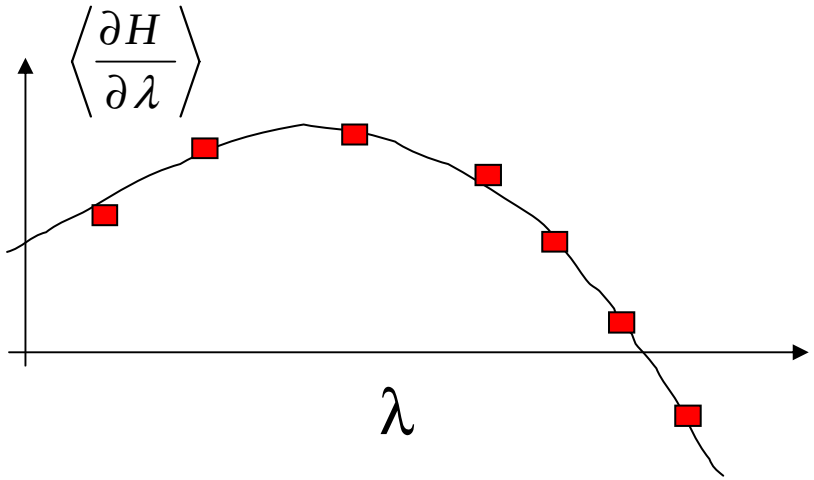
- Group of users run a particular super-computing class application on the grid resource. The installation and optimized run parameters may be the same for the whole group.

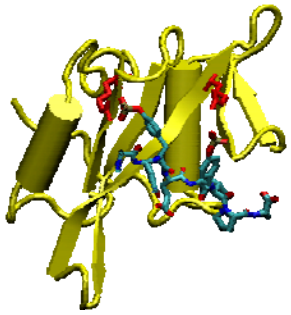
- With the AHE all application specific information for running simulations on the grid resource is maintained on a central service.

Global Grid Infrastructure



Thermodynamic Integration on Computational Grids





Starting conformation

λ=0.1

λ=0.2

λ=0.3

λ=0.9

λ=1.0

Simulation time

Run each independent job on the Grid

Combine and calculate integral

RealityGrid

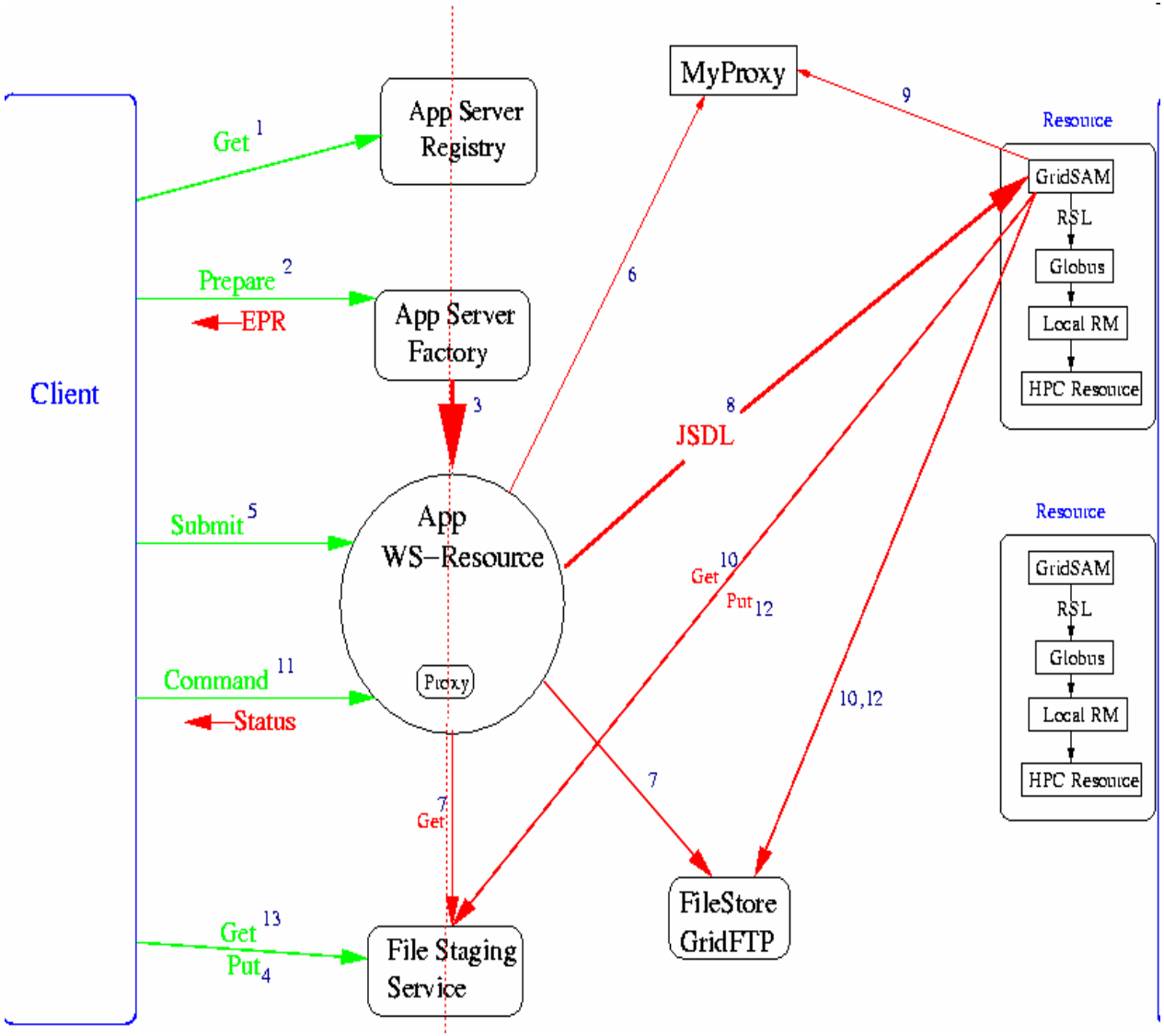
open middleware infrastructure institute

AHE Components

- AHE Client
  - Java GUI
  - WSRF::Lite(perl) command-line

- AHE Server-Side
  - WSRF::Lite
  - WWFS/Webdav server
  - GridSAM => Globus grid
    => SGE resource (UCL CCC)
    => Condor grid
  - MyProxy
  - GridFTP servers – ral, man NGS nodes

Architecture of the AHE



http://**kato**.mvc.mcc.ac.uk/rss-**wiki**/ApplicationHostingEnvironment

WSRF::Lite

- WSRF::Lite – An Implementation of the Web Services Resource Framework

  http://www.sve.man.ac.uk/Research/AtoZ/ILCT

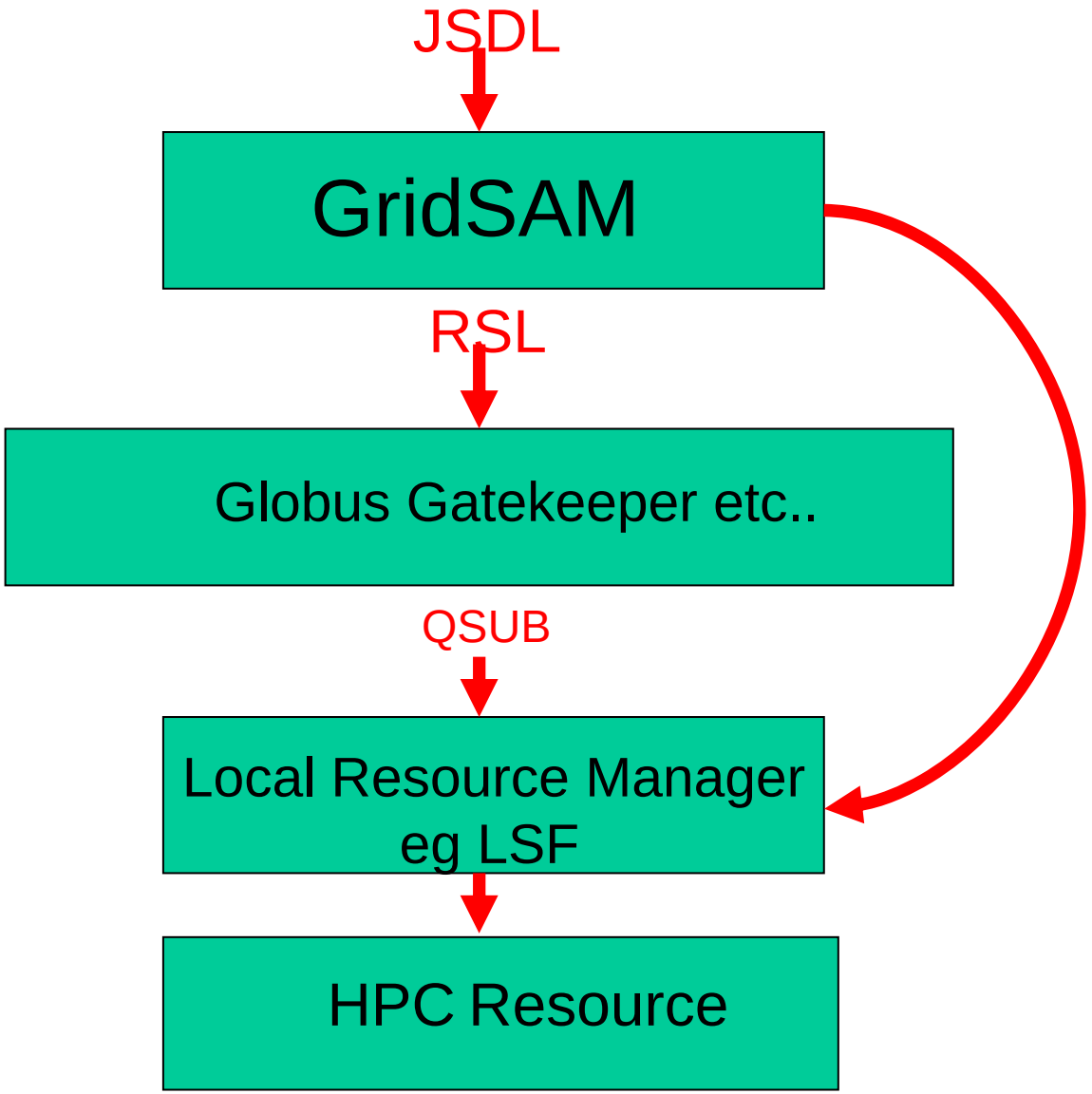
- We use WSRF::Lite **http://www.sve.man.ac.uk/Research/AtoZ/ILCT** as the AHE middleware.

- Each instance of the running Application on the Grid is represented by a stateful Web Service Resource that conforms to the WSRF specification.

GridSAM

- Produced by Imperial College through the OMII managed programme, distributed with the OMII distribution
- Hosted in the OMII Container,jakarta-tomcat Container
- Key to GridSAM is JSDL – Job Submission Description Language. GridSAM provides a uniform abstraction of resource managers – AHE only has to understand JSDL.
- JSDL is a new standard coming out of GGF for replacing things like Globus RSL, etc.

Create an Abstraction of the different Grid middleware…



Adapted from Mark McKeown

MyProxy

- MyProxy provides a mechanism for giving services GSI Proxy certificates.
- It is an alternative to proxy delegation.
- MyProxy choice was natural given GridSAM's dependence.

File Staging Area

- AHE supports the case where the client has the required input files.
- The File Staging Area, FSA, allows the client to stage files to a place that the application can access them from.
- The client uses HTTP POST to send a file to the FSA and HTTP GET to download a file.
- Output files are placed in FSA by the AHE for the client to download.

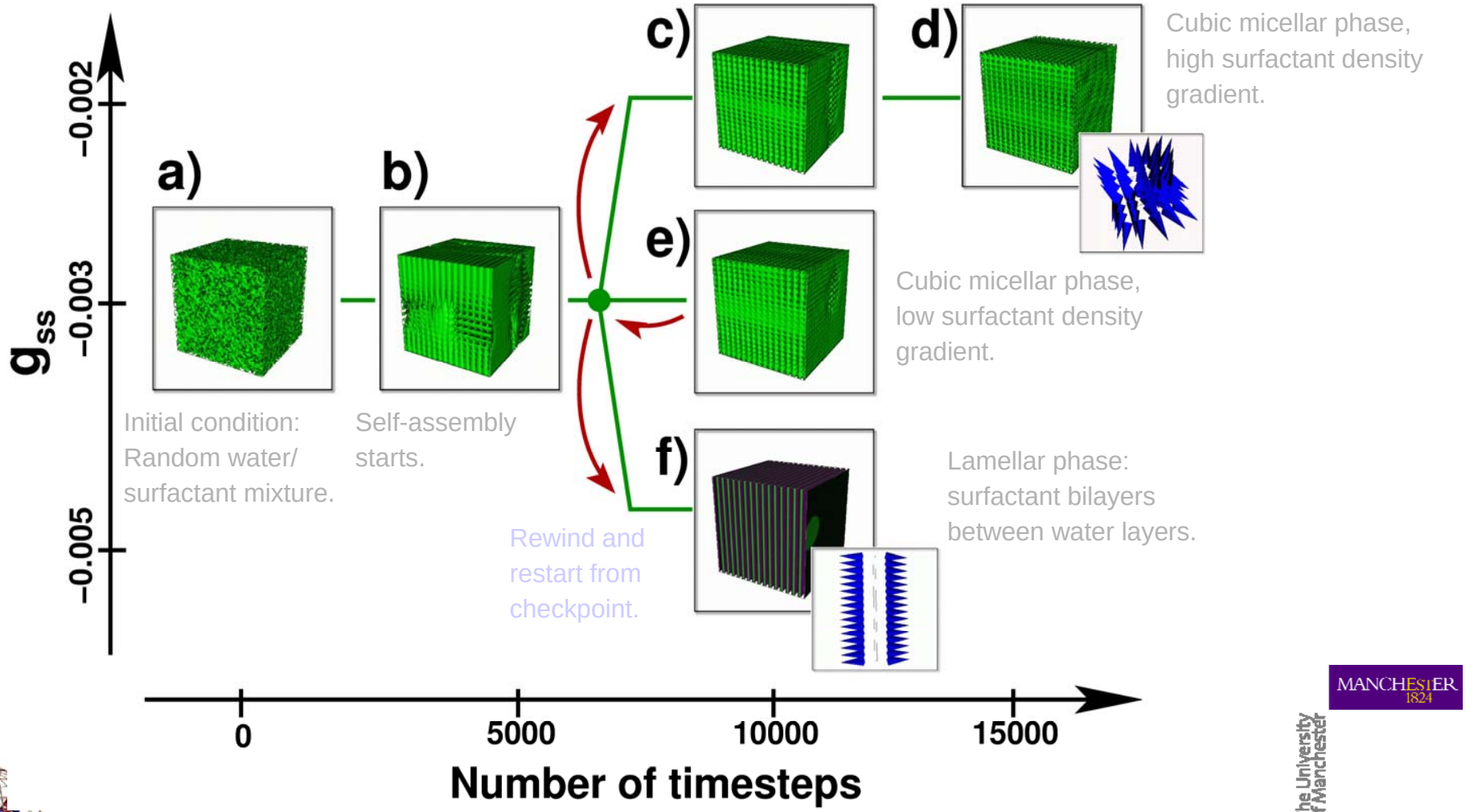
FileStore

- Not all files may be stored on the clients machine.
- The FileStore is anywhere that a files are stored that are required by the application eg on a GridFTP server.
- The client may not be able to access the FileStore directly - the client may not support the required protocol.

Future Plans

- Release version for vanilla applications
- Steering Web Service(SWS) , RealityGrid Steering Library based on WSRF::Lite
- Coupled models – host applications which are made up of other application components

Computational Steering – Parameter Space Exploration

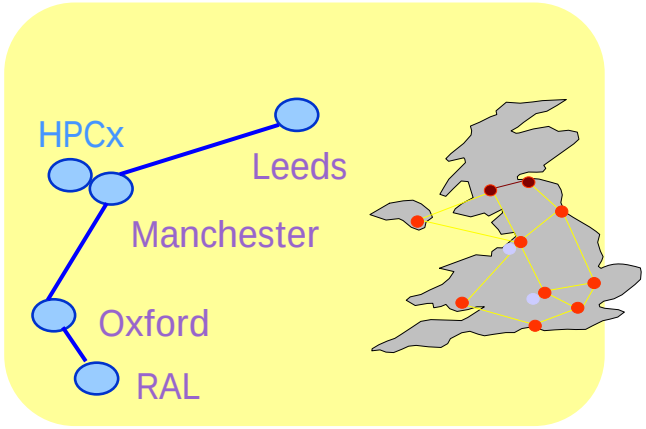


RealityGrid

open middleware infrastructure institute

Coupled Models

Computation



Data Flow

Vizualisation

Hybrid Molecular Dynamics

Particle based MD

Data Flow

Mesh based Hydrodynamics

RealityGrid

open middleware infrastructure institute

Links

- Thanks to Mark McKeown, University of Manchester for some of the AHE slides - Mark McKeown's talk at the All Hands Meeting 2005 - uploaded on the AHE Wiki
- AHE Wiki

  http://kato.mvc.mcc.ac.uk/rss-wiki/ApplicationHostingEnvironment
- WEDS download site

  http://www.realitygrid.org/weds
- GridSAM – Grid Job Submission and Monitoring Web Service

  http://www.lesc.ic.ac.uk/gridsam/
- WSRF::Lite – An Implementation of the Web Services Resource Framework

  http://www.sve.man.ac.uk/Research/AtoZ/ILCT

Acknowledgements

- Peter Coveney, Matt Harvey, Laurent Pedesseau, James Suter, Stefan Zasada, Phil Fowler, Kashif Sadiq, Mary-Ann Thyveetil, Giovanni Giupponni, Simon Clifford

- Mark McKeown, Stephen Pickles, Rob Haines, Andy Porter

- EPSRC

- OMII

RealityGrid

open middleware infrastructure institute

AHE GUI CLIENT Functionality

- Discover Appropriate Resources
- Launch Application
- Monitor Running Jobs
- Query Registry of Running Jobs
- Stage Files to and from Resource
- Terminate Jobs

Client Assumptions

- Client is NAT'd and firewalled
- Client doesn't use GridFTP
- Client doesn't require Globus to be installed
- Client doesn't know anything about how to run the particular job
- Client doesn't maintain any information about job state

GUI Client Implementation

- Implemented in Java using Apache Axis WS API
- Demonstrates interoperability between WSRF::Lite and Java
- Implement job launching as a wizard, with each frame of the wizard covering a separate step in the launching process

Job Building Wizard

Wizard guides user through stages of submitting a job:

- Specifying constraints for job, e.g. number of processors to use
- Sends prepare message to cause the WS-Resource to be created
- Allows user to choose appropriate resource to run job

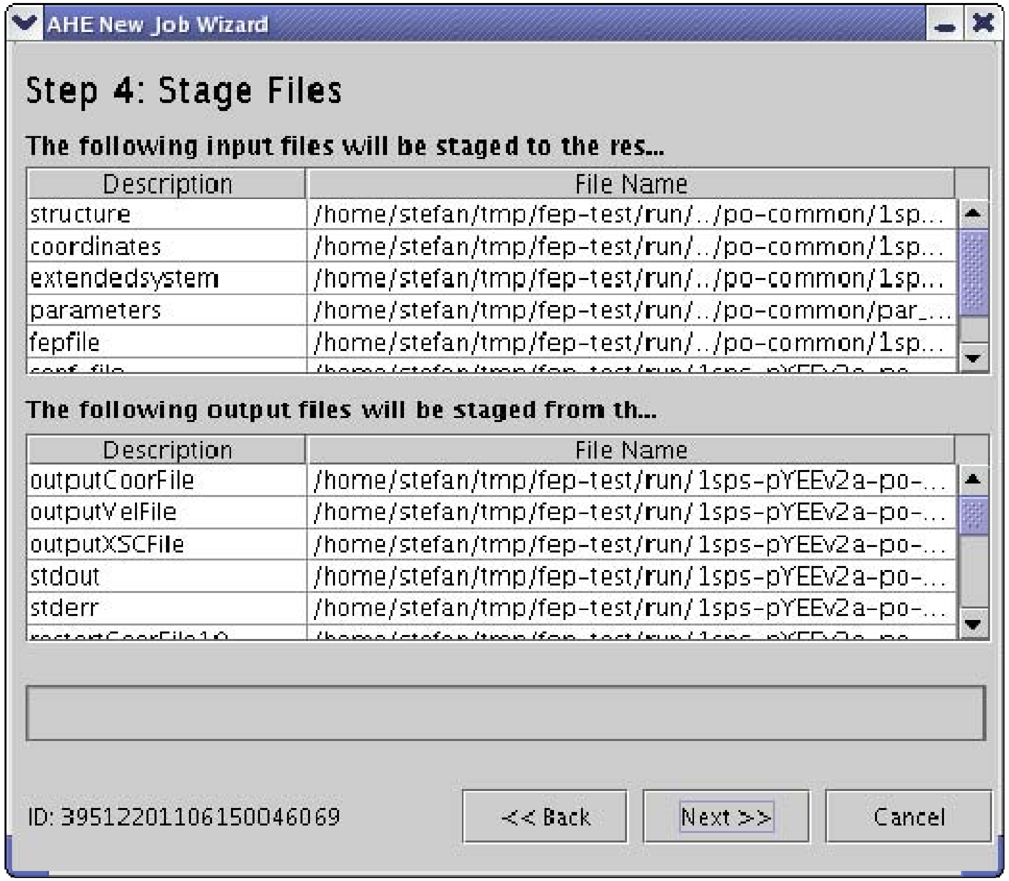
Job Building Wizard

- If plugin is available for particular type of job (e.g. NAMD) then client will attempt to process job configuration file to discover input and output files to be staged, otherwise the user will have to specify manually

- Client stages file to AHE file staging area (WWFS, Web-Dav etc.)

- User checks job details then submits it to AHE

AHE Job Building Wizard



Client Extensibility

- Plugins will be added to process application input files to automatically discover the input and output files that need to be staged
- If no plugin is available then a default case will allow users to specify input and output files manually
- Framework will allow experienced users to create new plugins for client for applications beyond the initial use cases

AHE Job Interaction

Once job has been prepared and submitted:

- Client will display job information, and poll resource at regular intervals to update status
- Client can terminate and destroy current jobs
- Once job is complete, client will stage input files back from WWFS if required
- Client allows user to query resource for list of current jobs
  - no state need be maintained in client

Java Client Development

- As AHE develops functionality, client will be extended (e.g. to support RealityGrid steering and resource co-allocation).
- Java codebase can be used to easily create other clients to interact with AHE (e.g. Java command line clients).
- GUI Client development will be greatly influenced by user feedback from early adopters

Links

- Thanks to Mark Mckeown, University of Manchester for some of the AHE slides - Mark McKeown's talk at the All Hands Meeting 2005 is uploaded on the AHE Wiki
- AHE Wiki

  http://kato.mvc.mcc.ac.uk/rss-wiki/ApplicationHostingEnvironment
- WEDS download site

  http://www.realitygrid.org/weds
- GridSAM – Grid Job Submission and Monitoring Web Service

  http://www.lesc.ic.ac.uk/gridsam/
- WSRF::Lite – An Implementation of the Web Services Resource Framework

  http://www.sve.man.ac.uk/Research/AtoZ/ILCT