Using Application Hosting Environment for Multiscale Computing

Stefan Zasada

University College London

The Application Hosting Environment

AHE is a tool for building Science Gateways – currently used at UCL to provide a gateway for our users

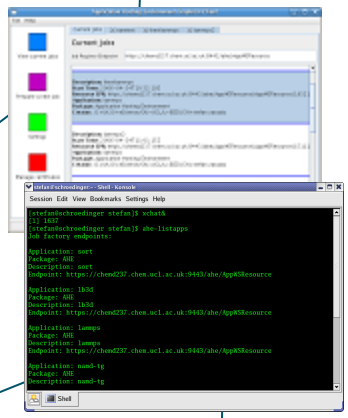
- Based on the idea of applications as stateful WSRF web services
- Lightweight hosting environment for running unmodified applications on grid and local resources
- Community model: expert user installs and configures an application and uses the AHE to share it with others
- Simple clients with very limited dependencies
- No intrusion onto target grid resources

Bridging the gap

Local resources

GridSAM

Globus



NGS

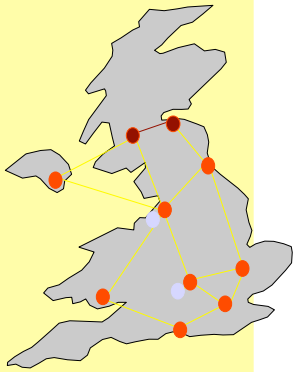
HPCx

Leeds

Manchester

Oxford

RAL



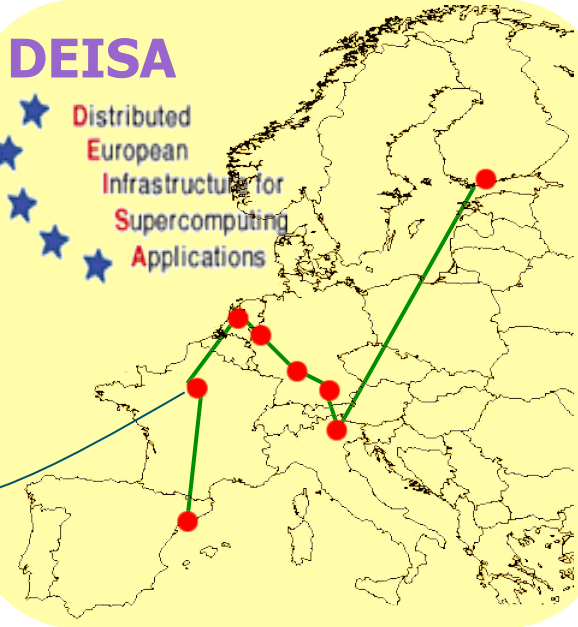
TeraGrid

Globus

UNICORE

DEISA

Distributed European Infrastructure for Supercomputing Applications



Motivation for the AHE

Problems with current middleware solutions:

- Difficult for an end user to configure and/or install
- Dependent on lots of supporting software also being installed
- Require modified versions of common libraries
- Require non-standard ports to be opened on firewall

We have access to many international grids running different middleware stacks, and need to be able to seamlessly interoperate between them

AHE Design Constraints

- Client does not have require any other middleware installed locally
- Client is NAT'd and firewalled
- Client does not have to be a single machine
- Client needs to be able to upload and download files but doesn’t have local installation of GridFTP
- Client doesn’t maintain information on how to run the application
- Client doesn’t care about changes to the backend resources

Virtualizing Applications

- 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

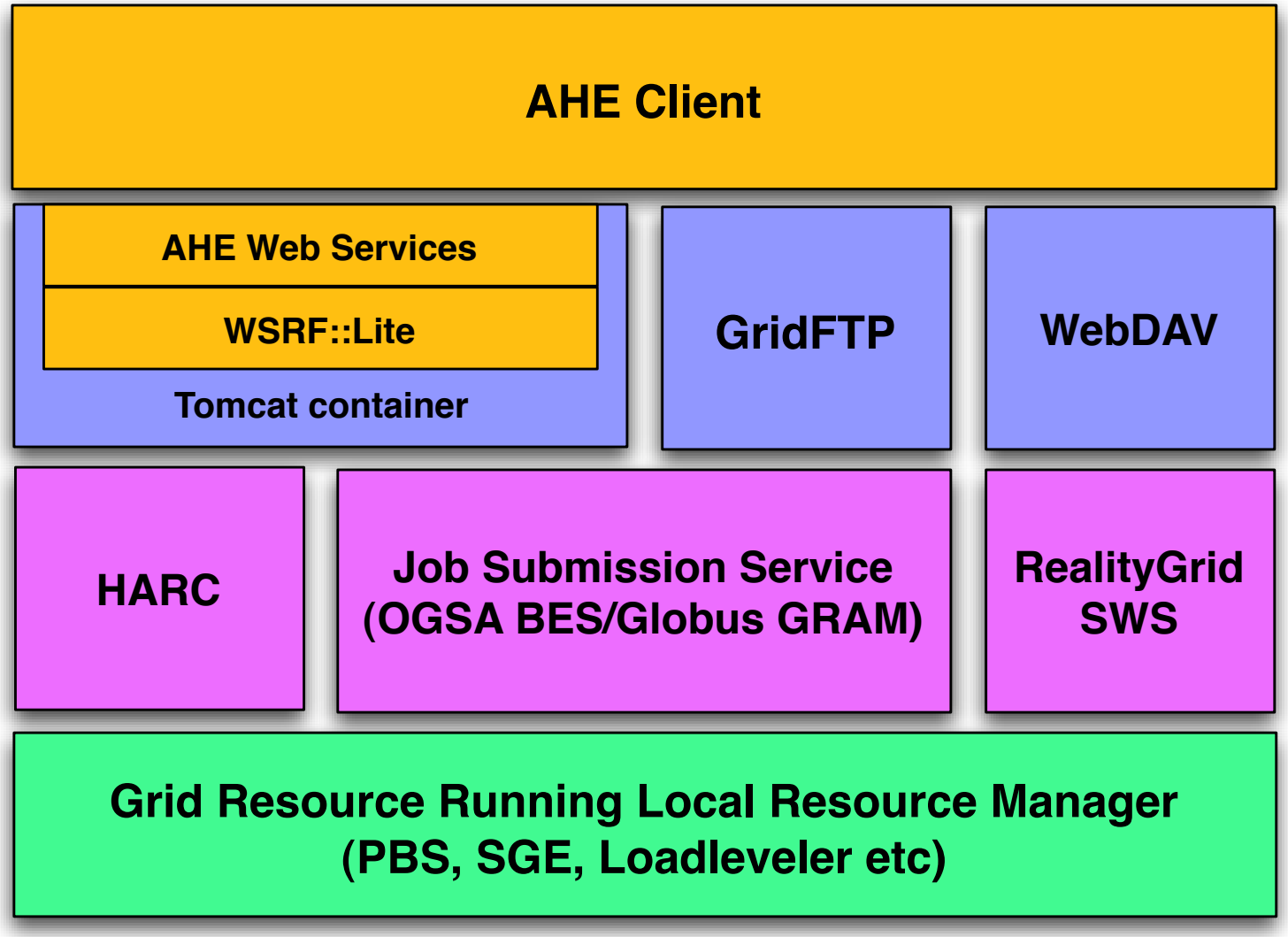
AHE 2.0 Provides a Single Platform to:

- Launch applications on Unicore and Globus 4 grids by acting as an OGSA-BES and GT4 client
- Create advanced reservations using HARC and launch applications in to the reservation
- Steer applications using the RealityGrid steering API or GENIUS project steering system
- Launch cross-site MPIg applications

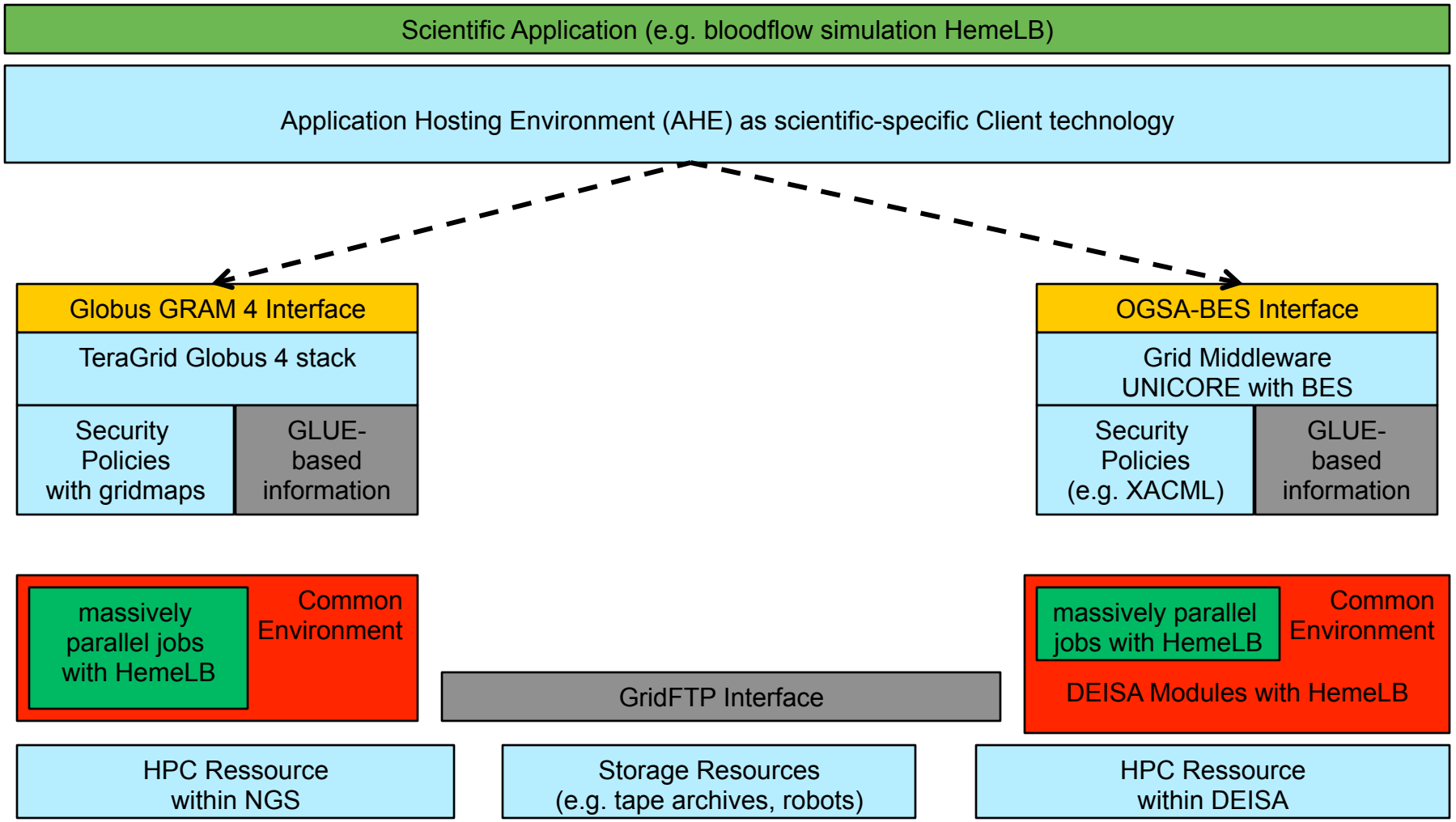
AHE 3 plans to support – due Q1 2012:

- SPRUCE urgent job submission
- Lightweight certificate sharing mechanisms

Service Architecture of the AHE



GINSC2008-001 – Architecture



Stefan Zasada, Steven Manos, Morris Riedel, Johannes Reetz, Michael Rambadt et al.,
Preparation for the Virtual Physiological Human (VPH) project that requires interoperability of numerous Grids

Client Implementation

- GUI & command line clients implemented in Java
- Client allows user to:
  - Discover appropriate resources
  - Launch application
  - Monitor running jobs
  - Query registry of running jobs
  - Stage files to and from resource
  - Terminate jobs
- GUI client implements application launching as a wizard

AHE Deployment

AHE Server

- Released as a VirtualBox VM image – download image file and import to VirtualBox
- All required services, containers etc pre-configured
- User just needs to configure services for their target resources/applications

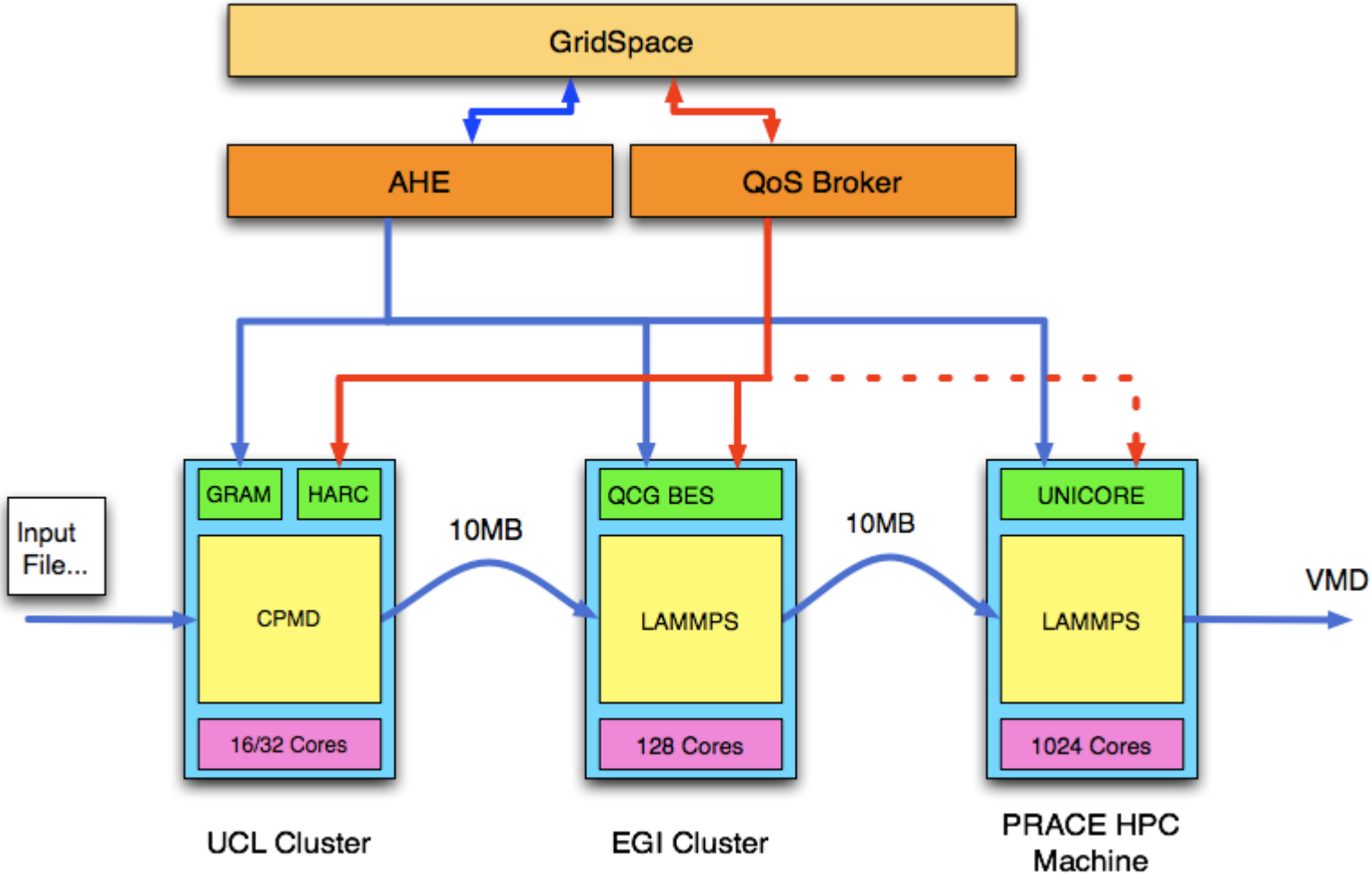
AHE Client

- User's machine must have Java installed
- User downloads and untars client package
- Imports X.509 certificate into Java keystore using provided script
- Configures client with endpoints of AHE services supplied by expert user

Constructing workflows with the AHE

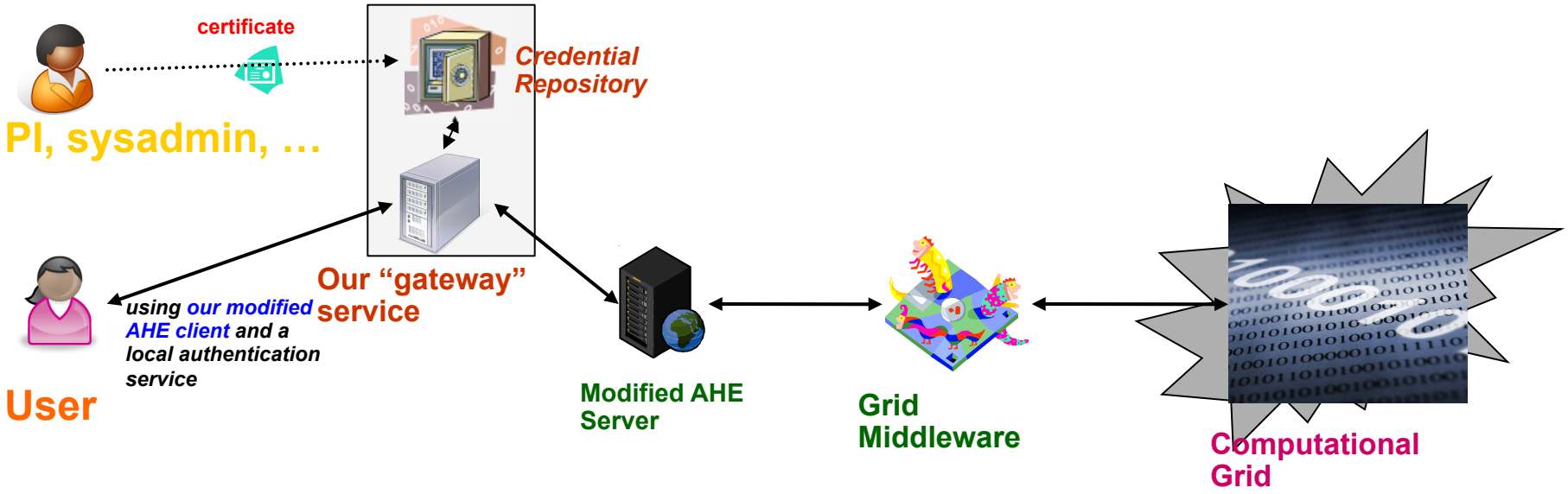
- By calling command line clients from Perl script complex workflows can be achieved
- Easily create chained or ensemble simulations
- E.g.:
  - ahe-prepare → prepare a new simulation for the first step
  - ahe-start → start the step
  - ahe-monitor → poll until step complete
  - ahe-getoutput → download output files
  - repeat for next step

AHE in MAPPER Scenarios



Authentication via local credentials

Audited Credential Delegation



- A designated individual puts a *single* certificate into a credential repository controlled by our new "gateway" service.
- User uses a *local authentication service* to authenticate to our gateway service.
- Our gateway service provides a session key (*not shown*) to *our modified AHE client* and *our modified AHE Server* to enable the *AHE client* to authenticate to the *AHE Server*.
- Our gateway service obtains a *proxy* certificate from its credential repository as necessary and gives it to *our modified AHE Server* to interact with the grid.
- User now has no certificate interaction.
- Private key of the certificate is never exposed to the user.

Summary

- AHE provides a single user interface to both grid and local resources
- AHE middleware stack allows federated access to resources from multiple independent grids, running different middlewares – Globus, Unicore etc.
- AHE 2.0 provides a single interface to several other tools, including HARC and RealityGrid steering
- The client is easily installed by any end user, requiring no intervention by system/network administrators
- By calling the command line clients from scripts, complex scientific workflows can be implemented
- User federation of resources allows us to make efficient use of our allocations, and investigate problems that would take too long on a single grid

Further Information

stefan.zasada@ucl.ac.uk

- **RealityGrid web site:**

  http://www.realitygrid.org/AHE

- **Wiki:**

  http://wiki.realitygrid.org/wiki/AHE