

# GridSAM and OMII-AuthZ

Hugo Mills, OMII-UK Southampton

# GridSAM and OMII-AuthZ

- What OMII is and does
- GridSAM
  - What it does
  - High-level architecture
  - Components
  - Current issues
- OMII-AuthZ
  - What it does
  - How it sits in the OMII container
  - Current work

# Software Solutions for e-Research



OMII-UK provides free, Open Source software and support, enabling researchers to harness the power of e-Infrastructure.

- **Software Solutions** – easy-to-use and easy-to-install software to solve common e-Research problems

- **Development Toolkit** – inter-operable components that can be tailored to provide bespoke software functionality

- **Matching Requirements** – identifying the needs of the research community and commissioning software to fill the gaps

- **Comprehensive Support** – helping users from first contact through to implementation and beyond

# OMII DevKit components

- AHE – Application hosting environment
- BPEL – Workflow language & enactor
- GridSAM – Job submission engine
- GridSphere – Portlet container
- Grimoires – Semantic service registry
- OGM – Infrastructure management
- OGSA-DAI – Data Access Infrastructure
- Taverna – Workflow design & enactor
- WSRF-Lite – Base for AHE

# Forthcoming OMII components

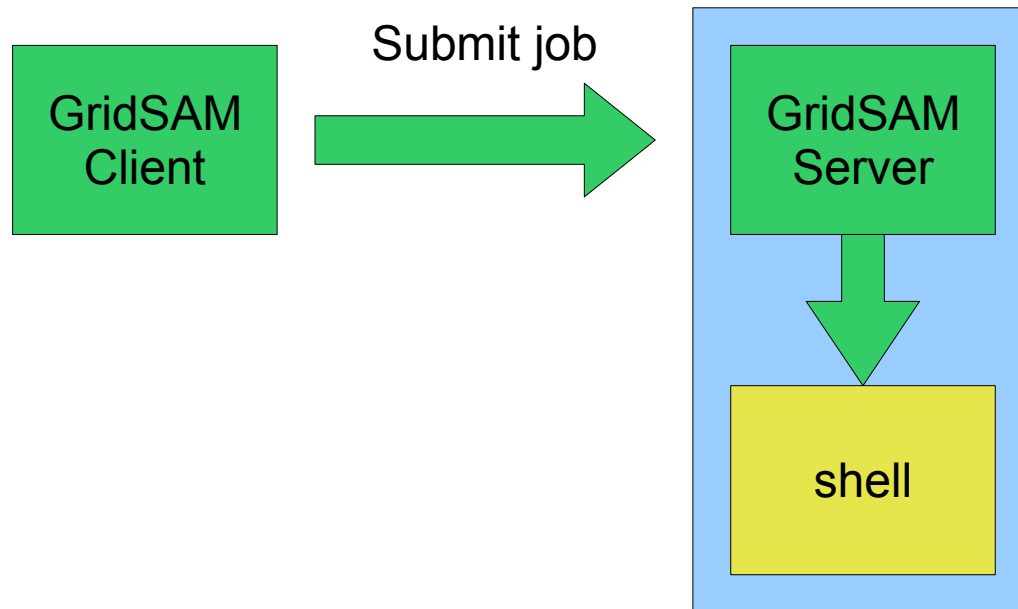
- OMII-AuthZ – Fine-grained authorisation
- SAGA – Grid API
- OGRSH – Desktop integration
- RAVE – Data visualisation
- VPMAN – VOMS/Permis (OMII-AuthZ)
- GridBS – Resource brokering
- SPAM-GP/SCAMP, WHIP, RAPID – Portlets

# GridSAM

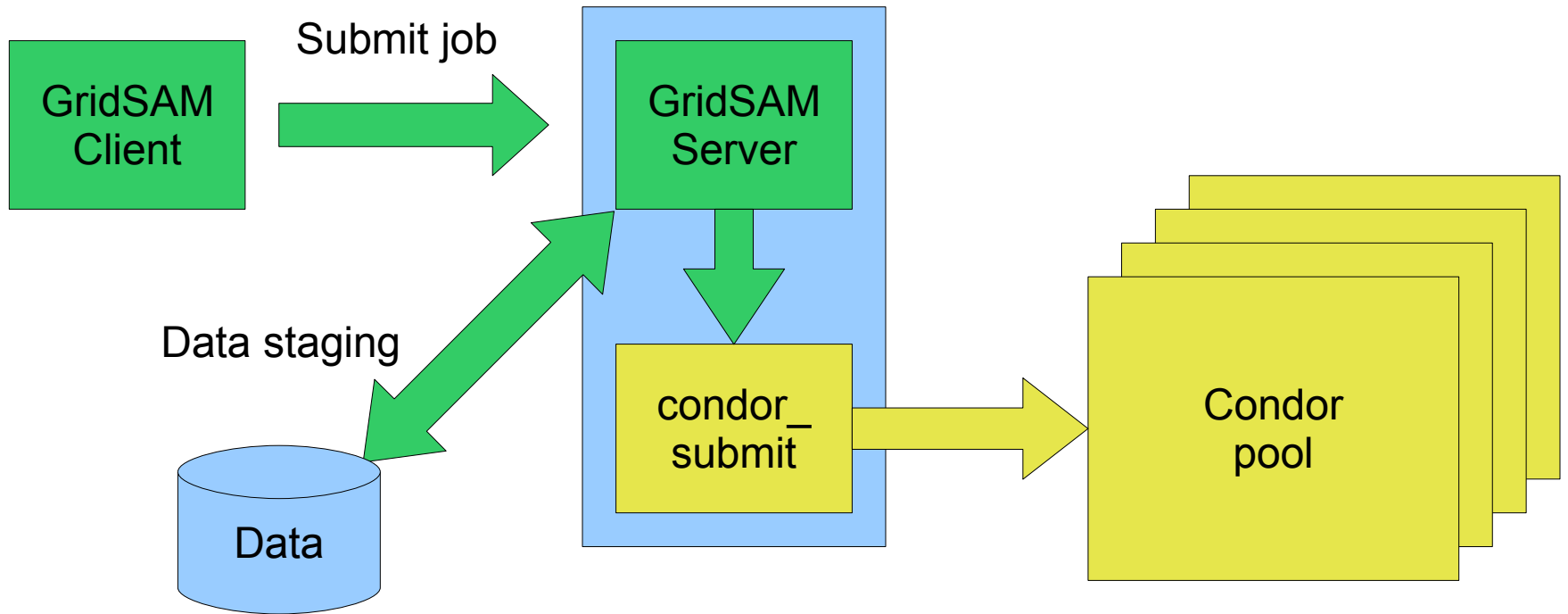
# GridSAM

- Job submission system
- SOAP service – Java servlet
- Uses OGF standard JSDL
  - File staging – many protocols
  - Extension for MyProxy
- Many back-ends:
  - Condor, GRAM, local, ssh, DRMAA, (PBS)
- Has an OGSA-BES interface

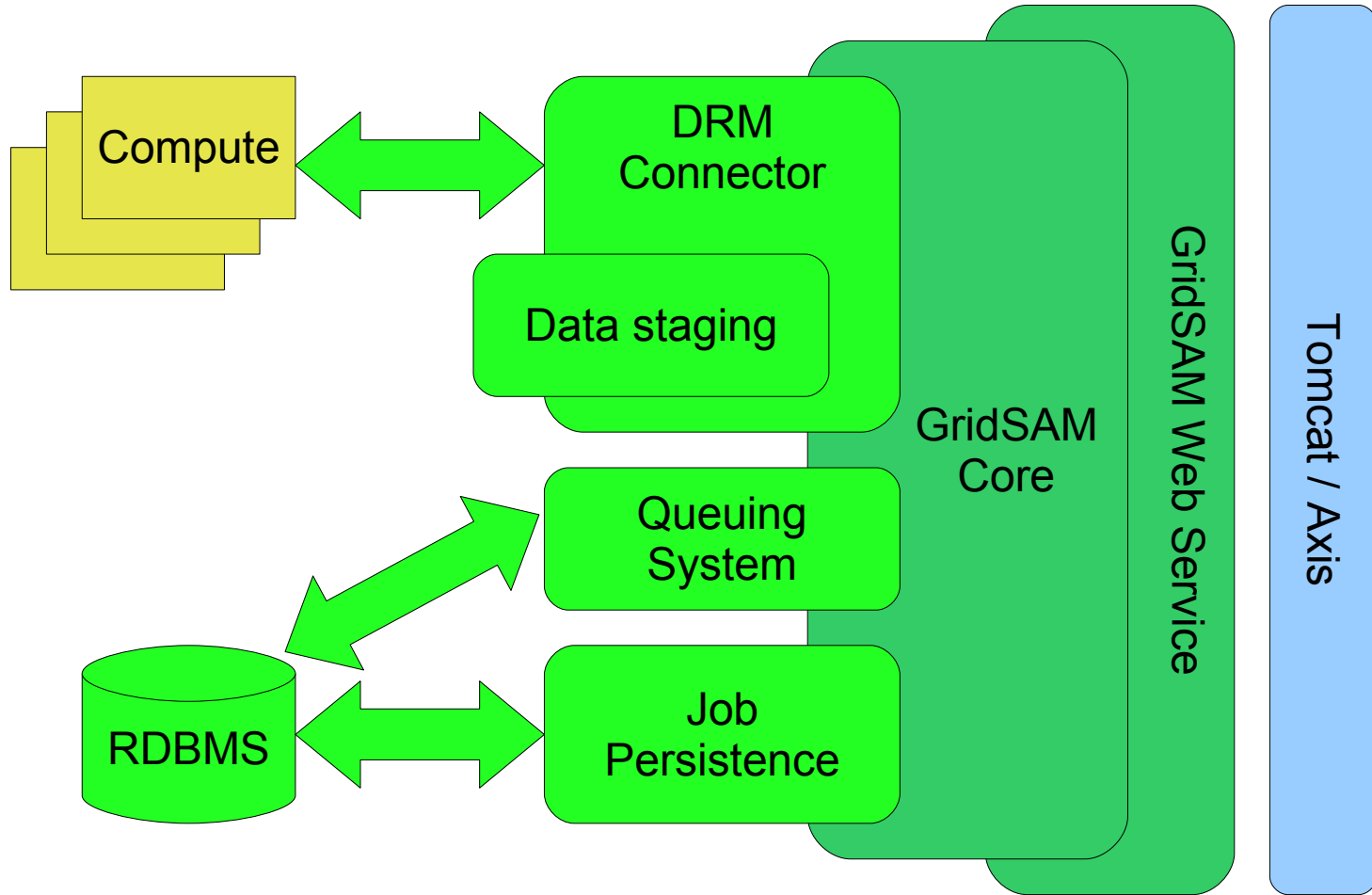
# Local “fork” configuration



# Condor configuration – with data staging



# Architecture overview



# DRM Connectors

- Submit job to back-end system (DRM)
  - DRM = Distributed Resource Manager
- Keep track of state transitions
  - DRMConnector drives internal state machine
  - States are implemented as **Stages**
  - Some Stages are common across several DRMs

# State machine sequence

## – Condor example

- gridsam.ValidatingStage
- gridsam.JobIdentificationStage
- condor.FileSystemInitialisationStage
- condor.StageInStage
- condor.CondorClassadStage
- condor.ShellBasedCondorSubmitStage
- condor.StageOutStage
- gridsam.DoneStage
- gridsam.FileSystemCleanupStage

# Implementing a new DRMConnector

- Required:
  - DRMConnectorManager – connector factory
  - DRMConnector – specifies sequence of stages
  - SubmitStage – how to submit a job
  - Hivemind config file – maps stage names to classes
- Optional:
  - StageInStage
  - StageOutStage

# Data staging

- GridSAM server handles staging of files
- Default data staging copies via GridSAM server
  - Used in most DRMConnectors
- Possible to copy direct from data store to DRM host
  - Needs a modified connector, and support in DRM for data staging

# Data Staging

- Default code uses Apache commons-vfs
- Supports
  - http, https, ftp, sftp, scp, file, webdav, gsiftp
- Issues
  - webdav not supported upstream
  - gsiftp uses relative URLs
  - sftp (and others?) unreliable

# DRM Connector issues

- Data staging goes through GridSAM
  - Two network transfers, not one
  - Limited by storage on GridSAM server
- Limited capability with proxy certs
  - Service provider sees GridSAM as user
- Restart of partially-failed jobs difficult
- Re-staging of data is difficult

# Queueing and event handling

- Decides how and when to start jobs
- Implemented using Quartz
- Default system starts jobs immediately, as fast as possible
- Pluggable interface
  - Could implement some other queueing discipline
- Quartz also used in DRMConnectors

# Persistence

- For stability of job state data
- Allows resume after server restart
- Implemented with *Hibernate* and *HSQL*
- Job state objects stored

# Persistence issues

- Jobs are **too** persistent
- Completed jobs kept in database
  - Logging, audit trails
- Hibernate loads *all* persistent objects
  - All job status
  - All job logs
- System slows down
- Memory footprint grows

# Persistence issues

- What policy to use to reap jobs?
  - Keep for  $n$  days
  - Keep  $m$  jobs
- How to archive job details outside Hibernate?
  - What format(s) would be useful?

# OMII-AuthZ



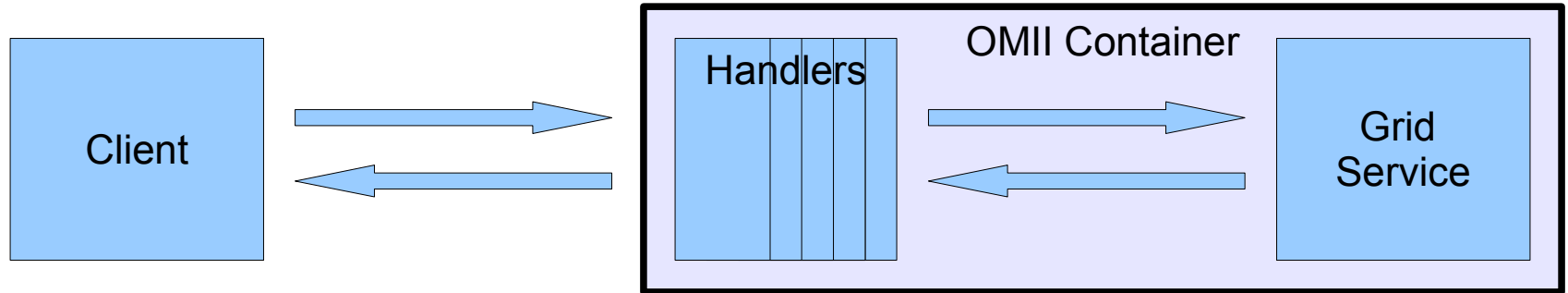
web: [www.omii.ac.uk](http://www.omii.ac.uk)

email: [info@omii.ac.uk](mailto:info@omii.ac.uk)

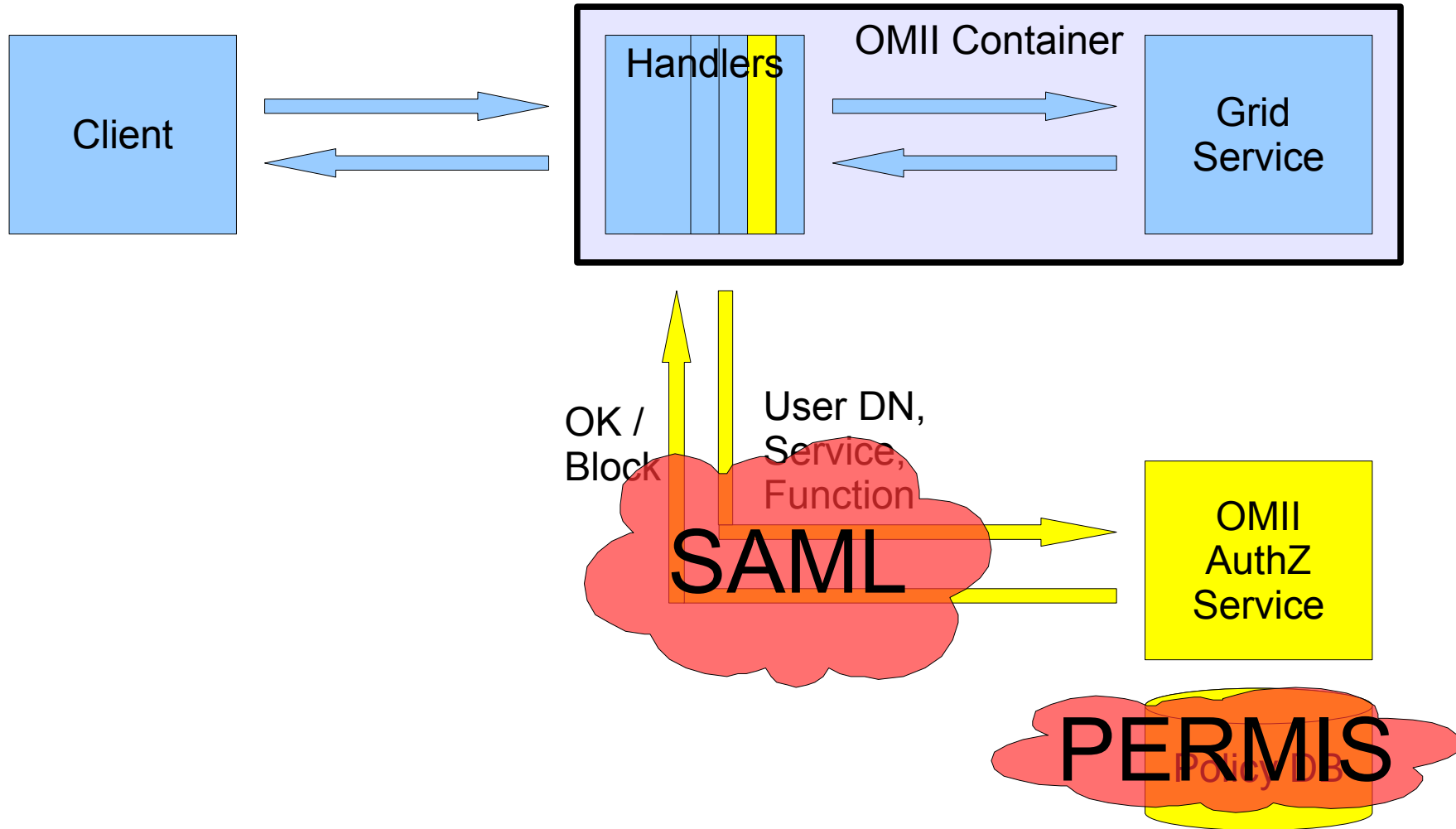
# OMII-AuthZ

- Authorisation service
- “Can this person use this function?”
- Fine-grained control
  - Individual functions within a service
- Integrated into software stack
  - Applicable to **all** services
- Uses SAML for exchanging security info
- Uses PERMIS for handling policy

# Architecture – Normal usage

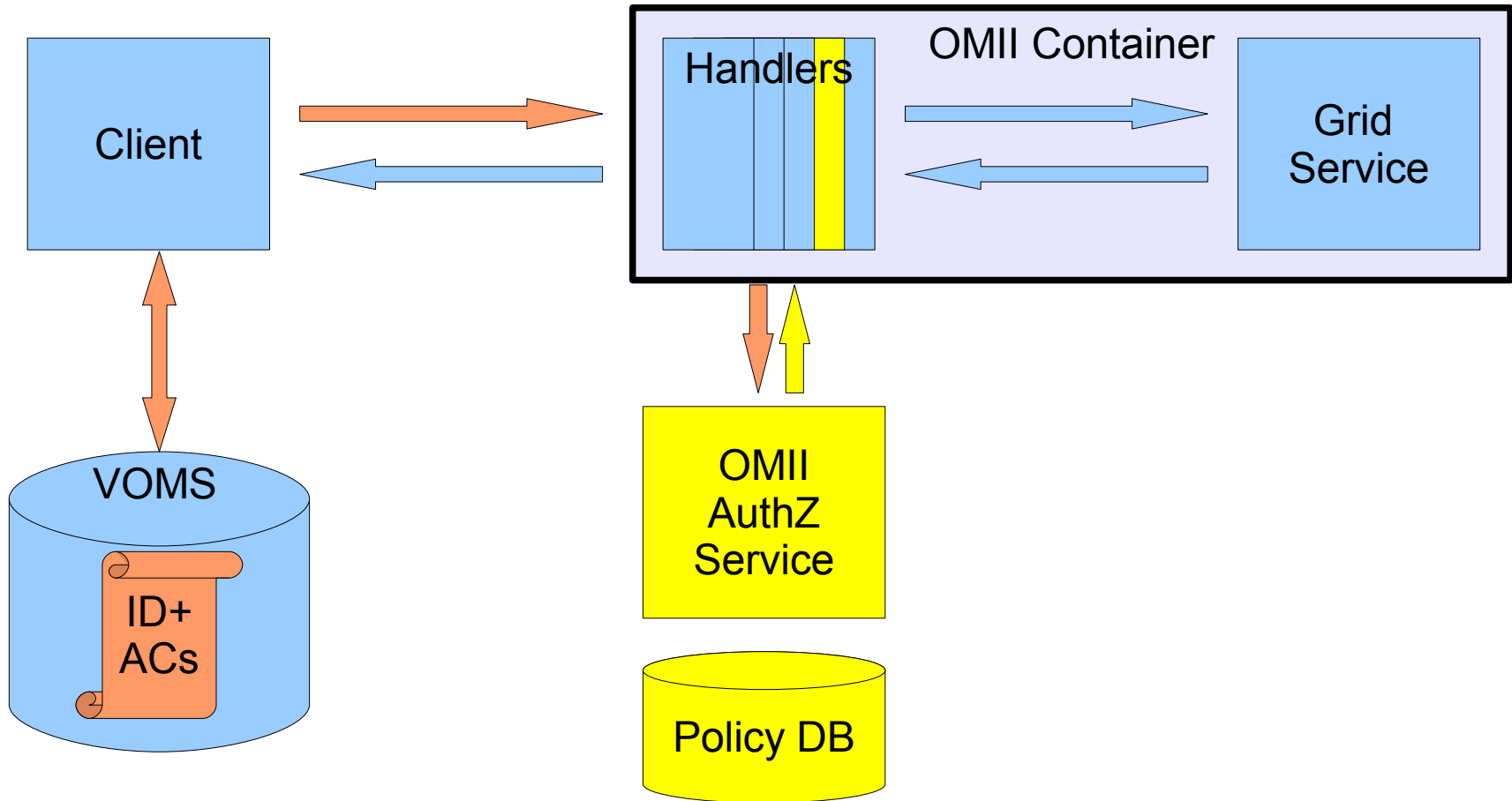


# Architecture – With AuthZ

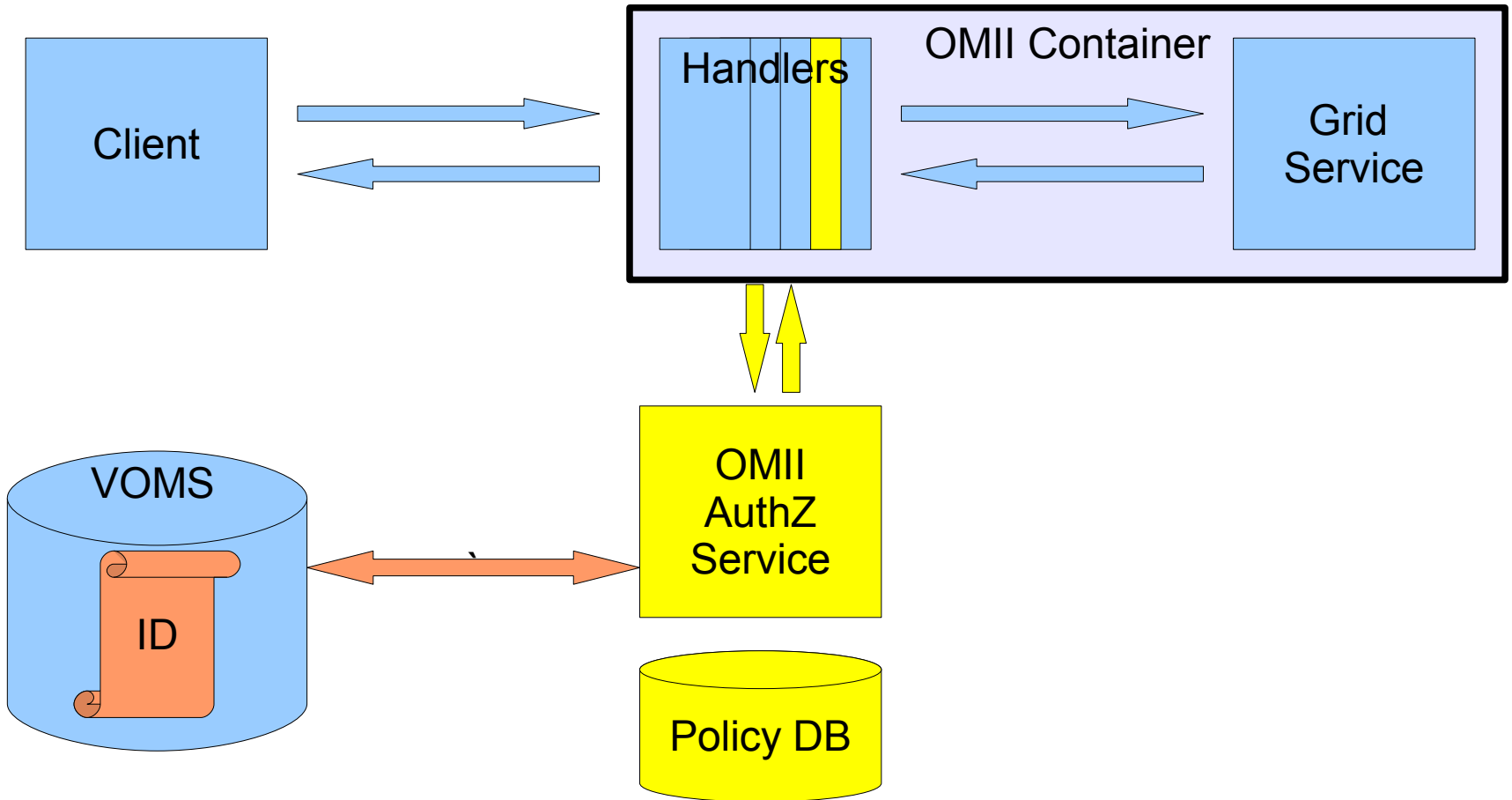


- VOMS – VO Management System
  - Holds security tokens for a virtual organisation
  - X.509 Attribute Certificates
- PERMIS – Security policy
  - Holds policy
  - Can *person* do *action* according to policy?
  - Uses X.509 Attribute Certificates to do checks
- VPMan – Integrate VOMS and PERMIS

# VOMS/PERMIS Push model



# VOMS/PERMIS Pull model



# More information

- Web: <http://omii.ac.uk/>
- Mail:
  - Hugo Mills <[hugo@omii.ac.uk](mailto:hugo@omii.ac.uk)>
  - Tech Support <[support@omii.ac.uk](mailto:support@omii.ac.uk)>
  - General info <[info@omii.ac.uk](mailto:info@omii.ac.uk)>
- GridSAM: <http://gridsam.sf.net/>