034115

PHOSPHORUS

Lambda User Controlled Infrastructure for European Research

Integrated Project

Strategic objective:
Research Networking Testbeds

Deliverable reference number D.3.2

Report on middleware extensions and implementation

Due date of deliverable: 2007-09-30
Actual submission date: 30/09/07
Document code: Phosphorus-WP3-D.3.2

Start date of project:
October 1, 2006

Duration:
30 Months

Organisation name of lead contractor for this deliverable:
FHG

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission	
RE	Restricted to a group specified by the consortium (including the Commission	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Report on middleware extensions and implementation

Abstract

This report describes the status of the changes and implementation of the first version of the middleware as defined for the milestone M3.2.

Project:	Phosphorus
Deliverable Number:	D.3.2
Date of Issue:	30/09/07
EC Contract No.:	034115
Document Code:	Phosphorus-WP3-D.3.2

Table of Contents

Table of Figures

0 Executive Summary

This report describes the status of the changes and implementation of the first version of the PHOSPHORUS middleware.

In the first phase of the project these activities comprise

- the definition of interfaces to the PHOSPHORUS Network Resource Provisioning System,
- the definition of interfaces to the PHOSPHORUS G^2MPLS development,
- the resulting changes of the MetaScheduling Service for the integration with the Network Resource Provisioning System,
- changes for an improved integration of the MetaScheduling Service in the UNICORE middleware,
- modifications of UNICORE towards a fully web-service compliant version (UNICORE 6),
- deployment of UNICORE 6 to the sites participating in the WP3 application experiments,
- changes necessary for the integration of MetaScheduling Service with UNICORE 6,
- and finally the design of a framework for automatic selection of resources based on semantic annotation of applications and first steps towards an implementation of this framework.

1 Middleware design changes and extensions

1.1 UNICORE

1.1.1 Planned design changes and extensions

During the specification phase of the middleware and application design changes and extensions it was decided to use the new version 6 of the UNICORE Grid middleware [UNICORE] in this project. There have been several reasons for that decision:

- In contrast to previous versions, UNICORE 6 is supporting a broad spectrum of OGF [OGF] and OASIS [OASIS] standards, in particular Web Services Resource Framework 1.2 (WSRF) [WS-RF], and other WS-* standards. This will simplify future interoperability with other middleware stacks.
- Improved performance and scalability of the core components
- Improved flexibility of the client framework (UNICORE GPE clients) to support Phosphorus application use-cases

There are only a few Phosphorus specific modifications required for UNICORE 6. They are all related to the integration of the MetaScheduling Service (MSS):

- The UNICORE Target System Interface (TSI) has to be extended to support the in-advance reservation of computing resources and related functions like query of availability of the resource, cancelling of reservations, etc.
- It has to be ensured that the other UNICORE server components such as the Gateway and the XNJS (eXtended Network Job Supervisor) support the protocol extensions for the communication between the MSS and the TSI.
- The UNICORE client has to be extended to access the MSS services, namely making complex orchestrated in-advance or immediate resource reservations.

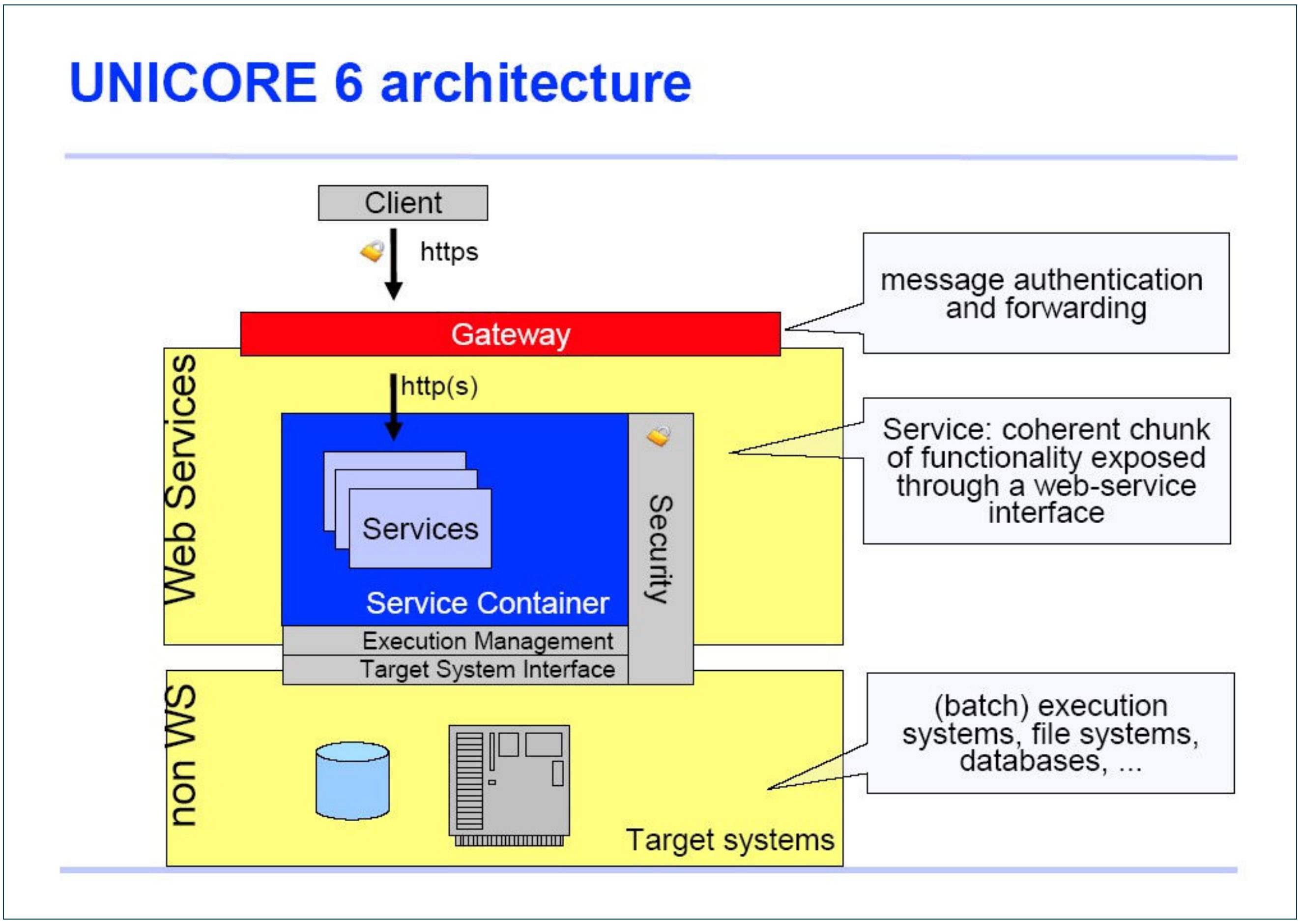

Figure 1.1: UNICORE 6 architecture

The modifications related to the MMS integration of the UNICORE server components is described in more detail in section 1.2. The modifications related to the UNICORE client are implemented in the context of the application use-cases, in particular KoDaVis and WISDOM. Therefore, they are described in the respective Deliverables.

1.1.2 Status of implementation and deployment

UNICORE 6.0 has been released at the end of August 2007. The protocol enhancements for MSS support are implemented. They are available on SourceForge [U@SOURCEFORGE] and will be included in the forthcoming release 6.1 of UNICORE.

A UNICORE 6.0 server has been installed at the following four sites in three different local test-beds that are participating in the WISDOM and KoDaVis application experiments:

- Forschungszentrum Juelich / VIOLA,

- Fraunhofer SCAI / VIOLA,
- PSNC,
- University of Essex.

Currently, a shared registry is being set up, that will make it easier for applications to access these sites as a single Grid during the tests planned for month 13 and 14.

1.1.3 Planned test-bed experiments

UNICORE 6 will be used as a Grid middleware platform in the test-bed experiments of the WISDOM and KoDaVis applications. There are no explicit dedicated tests of the UNICORE middleware planned. However, it should be noted that the Phosphorus test-bed will be the first major multi-site installation of UNICORE 6. Therefore it is expected that a lot of practical experience will be gained, in particular with respect to the following issues:

- Operation in multi-homed installations (with network interfaces towards the internet and towards the test-bed);
- Stability, performance and scalability, in particular when executing the demanding WISDOM use-case;
- Easy installation, configuration and management of the software.

1.2 MetaScheduling Service (MSS)

1.2.1 Planned design changes and extensions

In the first phase of the project the focus of the changes and extensions of the MSS [MSS] was on the following issues:

- the definition of interfaces to the PHOSPHORUS Network Resource Provisioning System,
- the definition of interfaces to the PHOSPHORUS G^2MPLS development,
- the resulting changes of the MetaScheduling Service for integration with the Network Resource Provisioning System,
- changes for improved integration of the MetaScheduling Service in the UNICORE middleware,
- changes necessary for the integration of the MetaScheduling Service with the new web-service based version of UNICORE 6.

The new network service plane interface to the PHOSPHORUS Network Resource Provisioning System (NRPS) developed in WP1 has been defined in collaboration with WP1. A first stable version of this interface has been implemented for the PHOSPHORUS demonstrations at the Supercomputing Conference 2007 in Reno (November 2007). The implementation of the corresponding interface on the side of the MetaScheduling Service on the basis of the first stable version is planned for the next months. This interface will supersede the previous interface to the ARGON [ARGON] NRPS the MSS used in the VIOLA testbed [VIOLA] before.

In parallel, the definition of interface to the PHOSPHORUS G^2MPLS development was done jointly with WP2. This interface did not exist before and will be implemented for the MSS once a stable version of the G^2MPLS is available.

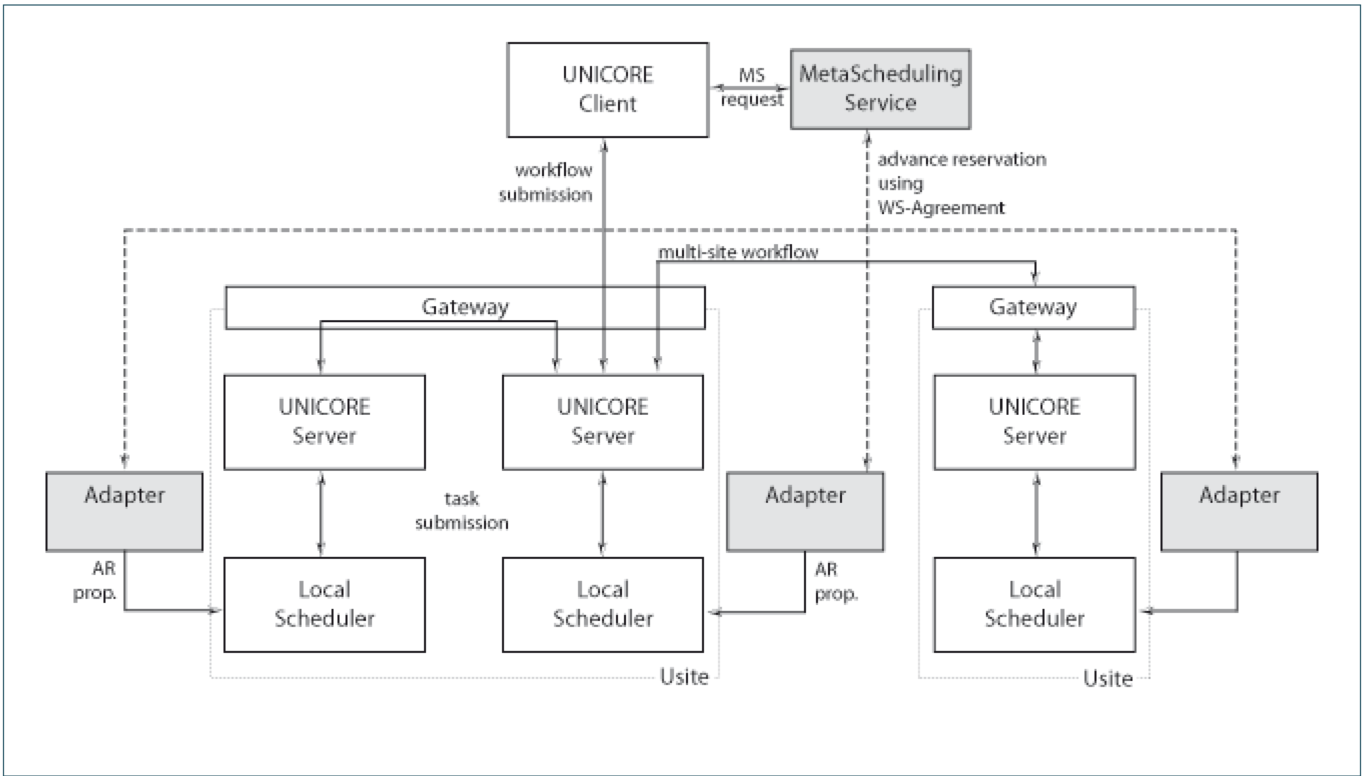

Figure 1.2: Integration of the MetaScheduling Service with UNICORE in the VIOLA testbed

The integration of the MetaScheduling Service with the UNICORE middleware in the VIOLA testbed was driven by the fact, that the focus in the VIOLA project was on experiments with the optical network, e.g. by using the selected applications to stress the network. This implied to time constraints for the set-up of the middleware leading to a number of design decisions like using external adapters as interface of the MSS to the local scheduling and queuing systems, benefiting from the fact that the VIOLA testbed was comprised of a small number of resources where the owners could easily establish a federation of trust and omit firewalls between the sites. The resulting architecture is depicted in Figure 1.2:

The PHOSPHORUS testbed is larger than the VIOLA testbed and spans across several countries thus, we had to follow another approach. Given the basic functionality of the MSS was already available from the VIOLA project we could concentrate on the enhancements of the MSS for using it in an environment with more general requirements regarding e.g. security. This led to a new approach for the integration of the MSS and the UNICORE middleware, which is depicted in Figure 1.3.

In contrast to the VIOLA implementation, the improved single adapter interface now utilises both the UNICORE Target System Interface (TSI) as adapter to the local scheduling and queuing systems and the secure single port access via the UNICORE gateway. The integration also includes some work to extent the TSI functionality to allow the MSS requesting information necessary for the negotiation of the service level agreements from the local scheduling and queuing systems. The changes of the architecture further on support connecting the MSS to a Network Resource Provisioning System like ARGON or to the Network Service Plane. Moreover, it will also allow integration with other Grid middleware systems like Globus Toolkit 4 [GT4] through a dedicated adapter that will be developed in the second phase of the project.

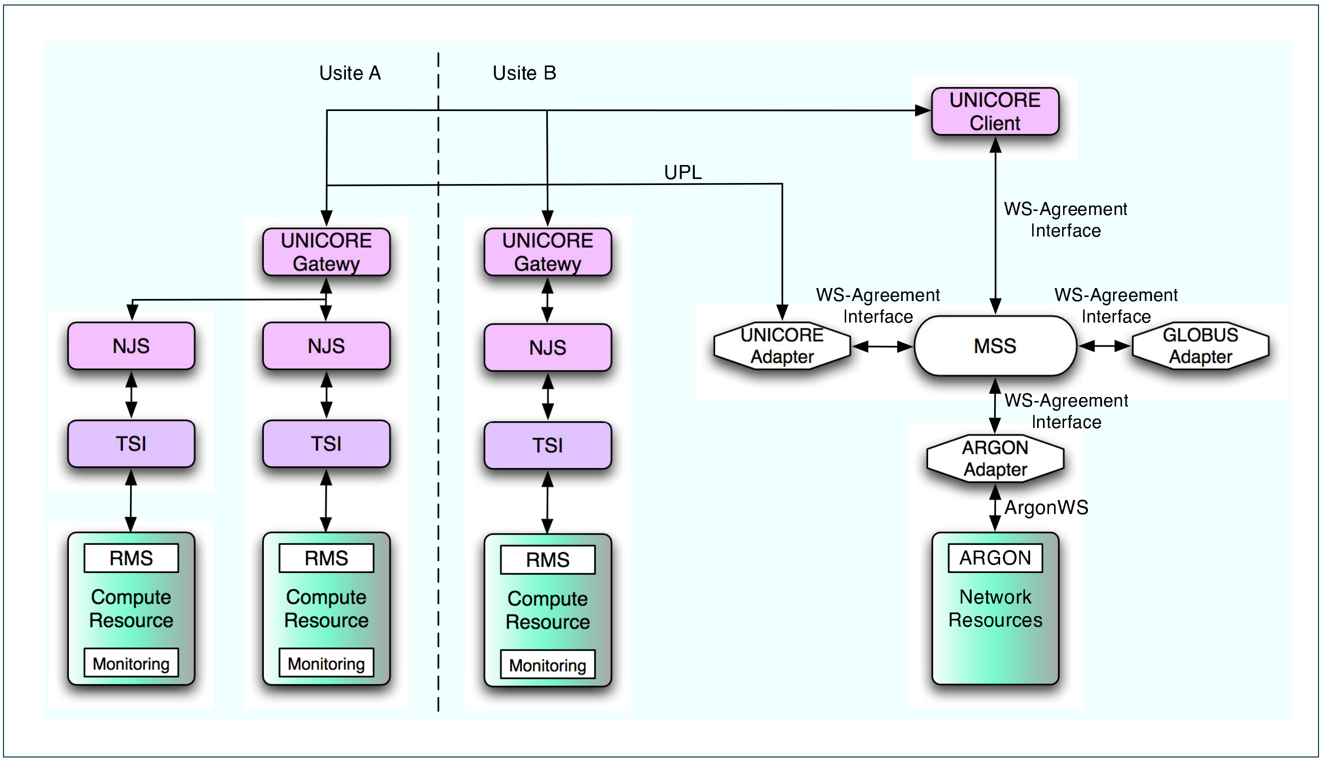

Figure 1.3: Improved single adapter interface to UNICORE in the PHOSPHORUS testbed

The new web-service based version of UNICORE delivered in August 2007 requires modifications of the MSS as some of the Unicore interfaces have been modified slightly. However, as we have been in close contact with the developers during the development of UNICORE 6 we could already foresee some of the necessary changes. Also, the fact that the MSS has been implemented as web-service from the very beginning made it easier to integrate with the new UNICORE version. Thus, the amount of work for the integration with UNICORE 6 is limited to changes in the protocol and smaller modifications of the TSI.

1.2.2 Status of implementation and deployment

The work on the interface to the network service plane for resource reservation started recently.

A first prototype of the new, improved interface to the UNICORE 5 middleware has been implemented. This interface supersedes the previous one that was used in the VIOLA testbed. Figure 1.2 depicts the interface used in the VIOLA testbed where the MetaScheduling Service framework included adapters to the local scheduling systems, like the EASY scheduler, or batch queuing systems, like Torque [TORQUE] while adapters with almost equivalent functionality are realised through the UNICORE Target System Interface (TSI). Another feature that is specific for the VIOLA approach is the direct connection to these adapters thus bypassing to some extent the UNICORE middleware and the built-in features to use a single port of the firewall to establish a secure connection between client and the UNICORE middleware.

In contrast, the improved single adapter interface now utilises both the UNICORE TSI as adapter to the local scheduling and queuing systems and the secure single port access via the UNICORE gateway. The integration includes some work to extent the TSI functionality to allow the MSS requesting information necessary for the negotiation of the service level agreements from the local scheduling and queuing systems. The new architecture is depicted in Figure 1.2. The modification also allows connecting to a Network Resource Provisioning System like ARGON or to the Network Service Plane. Moreover, it also allows integration with other middleware systems like Globus Toolkit 4 through a dedicated adapter.

We recently started the adaptation of the new adapter interface to the UNCORE 6 middleware after the web-service based version of UNICORE was released in August 2007.

1.2.3 Planned test-bed experiments

Experiments with applications

The MSS will provide support for planned testbed experiments with the PHOSPHORUS applications in M13 – M14. Within the initial experiments the support for job submission in the UNICORE based Grid via the MSS will be in the focus. Further experiments with applications, e.g. reservation of resources and co-allocation of resources for applications execution, will be derived from the results of the first test-bed experiments.

Experiments with the Network Control Plane and the G^2MPLS

In Collaboration with WP1 we will experiment with the new interface for reservation of network resources. Experiments with the PHOSPHORUS G^2MPLS will be made, once an implementation of the G^2MPLS becomes available.

1.3 Automatic Selection of Resources based on Semantic Annotation of Applications

1.3.1 Planned design changes and extensions

This task within WP3 started from scratch thus we will not describe design changes and extensions but the initial design of the framework for automatic selection of resources based on semantic annotation of applications. The overall framework is depicted in Figure 1.4.

The objective of this task is to enhance the capabilities of the MetaScheduling Service (MSS) framework. The MSS framework will be extended with the resource selection service exploiting the knowledge available on the applications' resource requirements and the resources available in the PHOSPHORUS testbed. Thus, for each request to execute one of the PHOSPHORUS applications sent to the MSS in the PHOSPHORUS testbed a set of appropriate resources may be automatically selected by the resource selection service without user intervention.

The state-of-the-art is that the user, who wants to submit a job, has to explicitly specify the resources needed for the execution of an application each time he submits a job. To do this he has to be aware of the requirements of the application. However, most often the developer of the application has a better knowledge of the best-suited resources. Thus, a better approach is to ask the developer to annotate the application once with resource requirement and to use this information each time when an application should be executed later.

To achieve this, the ontology of the resource requirements will be used. This ontology depicts the concepts of the resources (e.g. ComputingSystem) and the applications´ requirement (e.g. ApplicationResource), i.e. the concepts are described in an abstract form, so that they may be processed by an annotation tool in a second step to enrich it with the concrete information gathered for the applications.

For enabling the resource selection service's understanding of the resource demand, it is essential to enrich the ontology with semantic information. We use TUAM, a Tool for Universal Annotating and Mapping [TUAM], which has been developed by the Fraunhofer Institute of Scientific Computing and Algorithms. Using TUAM we can map the abstract concepts with each application's resource requirement. The result is stored in a Ppostgres database and may be exported as a triple-store in a RDF-file format. To retrieve the requirements of an application this file can then be queried by the resource selection service using SPARQL (RDF Query Language) [SPARQL].

The Pellet reasoner [PELLET] of the compute resource selection service will use the output of the SPARQL query interface and match these capabilities with those assured for the testbed resources in the resource. Pellet will reason about the similarity of requested and available capabilities and identify the set of resources

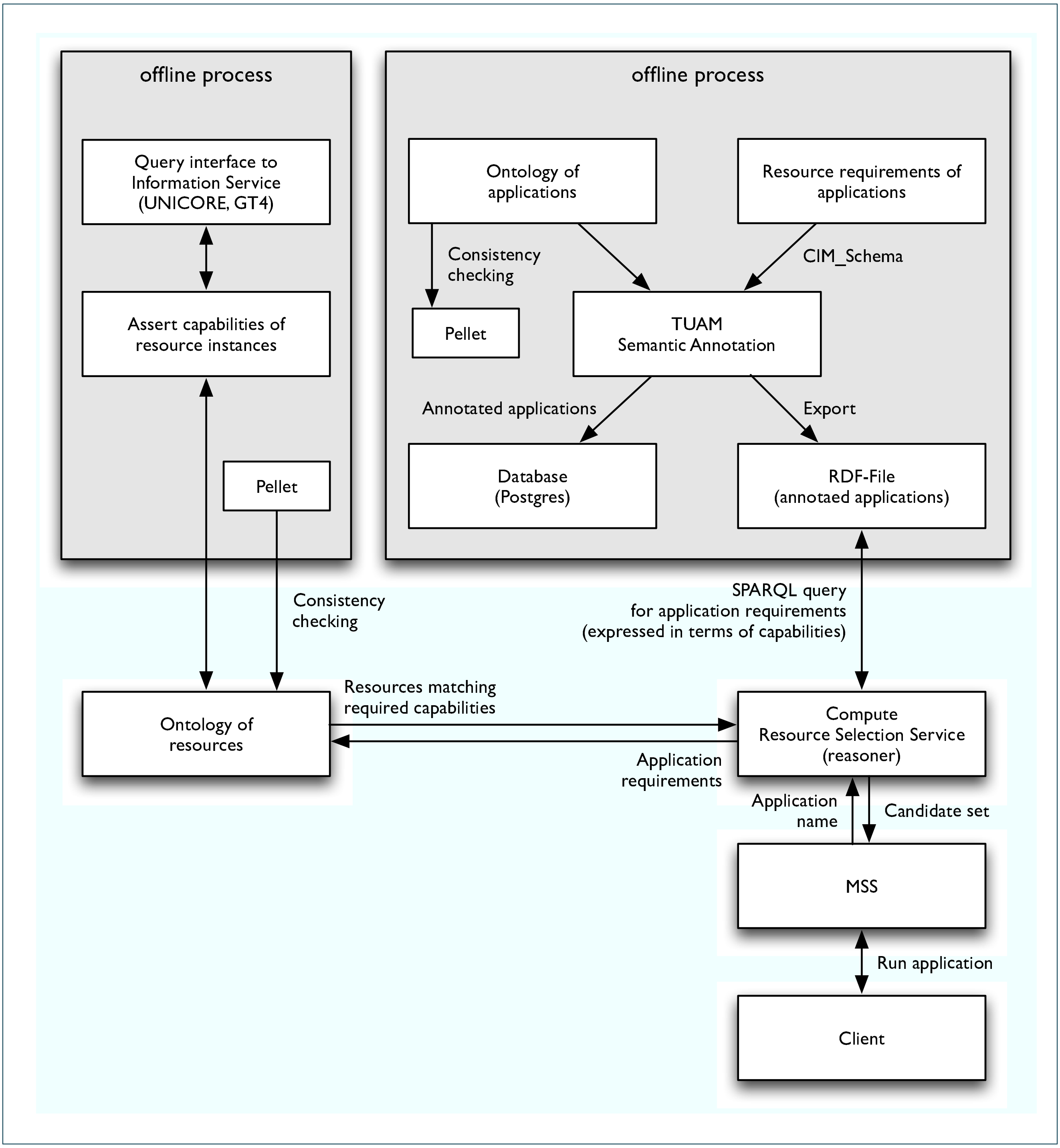

Figure 1.4: Framework for the automatic selection of resources based on semantic annotation of applications

fulfilling at least the requirements of the application to be run. This list – probably empty if no resources of the testbed can fulfil the requirements - will be returned to the MSS using the schema of the resource selection service currently under review at the Open Grid Forum [OGSA-RSS].

1.3.2 Status of implementation and deployment

We started the task identifying the software prerequisites:

- Tool for creating and managing ontologies, to be used for
  - the ontology of the application resource requirements,
  - the resource ontology,
- the ontologies themselves to be used for application resource requirements and resources,
- the reasoning tool for
  - checking the consistency of the ontologies,
  - the resource selection service,
- the tool for the semantic annotation.
- the tool for querying resource requirements of an application from the output of the semantic annotation.

As **tool for creating and managing ontologies** we selected and installed Protégé [PROTÉGÉ], a freely available ontology editor.

As **ontology for the applications' requirements** we decided to create a new ontology that fits the needs of the framework. As an initial step we gathered the requirements of all applications to be used in the PHOSPHORUS testbed. This was done by creating and distributing a questionnaire, which was returned by all partners in WP3. Based on these replies the layout of the ontology was defined.

Moreover, the information received is stored using a CIM-Schema [CIM] as an XML document, which is besides the ontology the essential second source of input for the annotation with TUAM.

For **the resource ontology** we decided to start with an existing ontology developed in the framework of the FP6 project UniGrids. This ontology is currently inspected to check whether there is a need for modifications or extensions.

As **reasoning tool** we selected and implemented the freely available Pellet reasoner, which is used for both checking the integrity of the two ontologies and – as part of the resource selection service – to select a set of appropriate resources based on the resource requirements of an applications (the required capabilities extracted from the RDF-file) and the capabilities of the resources in the PHOSPHORUS testbed.

The information on the requirements of the applications was then used to perform the **semantic annotation** of the applications with their proper resource demands. We selected and installed TUAM (a Tool for Universal Annotation and Mediation developed at the Fraunhofer Institute SCAI) for the semantic annotation. The resulting annotation is stored in a postgres database and exported as RDF-file for the resource selection process at later stage.

For **querying resource requirements** of an application from the output of the semantic annotation we selected the SPARQL, which is a Candidate Recommendation of the W3C for a RDF Query Language.

The results of the first 12 months:

- Software selected and installed at FHG:
    - Protégé ontology tool
    - TUAM tool for semantic annotation
    - Pellet installed as reasoner tool
    - Postgres database for annotation output
- Ontology of resource requirements of the applications
- Annotation of the applications with their proper requirements
- Resource ontology of the UniGrids project

There is no need for further deployment of the implemented infrastructure as the resource selection service is needed only at the sites where a MetaScheduling Service is running and in the PHOSPHORUS testbed currently there is only one instance of the MetaScheduling Service running at FHG serving the entire testbed.

1.3.3 Planned test-bed experiments

There will be tests of the functionality of the components of the framework for the automatic selection of resources based on semantic annotation of applications in the first phase of the testbed experiments. The implementation of the resource selection service and its integration into the MetaScheduling Service will be done in the second phase of the project as planned. Once the prototype of the resource selection service will be available we will integrate it into the MetaScheduling Service and perform experiments in the testbed.

1.4 Globus Toolkit 4

1.4.1 Planned design changes and extensions

Initially the Globus Toolkit 4 [GT4] environment is required for the DDSS experiments, which use GridFTP for the data-transfer from and to the storage elements. For this purpose no changes have to be made in the Globus Toolkit 4 environment.

In the second phase of the project an adapter between the MSS and the Globus Toolkit 4 will enable the user to use the MSS functionality also in Globus Toolkit 4 environments. This will not only allow to do reservations and co-allocations of compute and network resources in a Globus Toolkit 4 environment but also across UNICORE and Globus Toolkit 4 controlled resources. Modifications will affect the WS-GRAM service and the interfaces between WS-GRAM and the local scheduling or batch queuing systems.

1.4.2 Status of implementation and deployment

Globus Toolkit 4 has been installed – at least - at those sites of the PHOSPHORUS testbed, participating in the DDSS experiments using GridFTP.

1.4.3 Planned test-bed experiments

Currently no specific experiments are planned on the middleware layer with the Globus Toolkit 4. Some functionality of the Globus Toolkit will be used for the DDSS experiments.

1.5 INCA

1.5.1 Planned design changes and extensions

There are several extensions and changes planned for the INCA deployment over the PHOSPHORUS test-bed, mainly oriented to become more user-friendly and easily exploitable by external applications.

1.5.2 Status of implementation and deployment

INCA is currently deployed only at PSNC site. We are currently negotiating with UESSEX and CESNET for an INCA deployment on their sites.

The PSNC deployment is impossible to test without a working deployment over other sites.

1.5.3 Planned test-bed experiments

The following use cases have to be tested in PHOSPHORUS testbed:

- Sharing of Text Documents
- Sharing of Complex Media
- Backup and Restore
- System Imaging
- System Restore
- Concurrent file Insertion
- Concurrent file retrieval
- Concurrent Mixed Operations

The following test steps have to be performed in PHOSPHORUS testbed in order to validate the previously listed use cases:

- 1.CLI interface/small file insertion (PU)
- 2.CLI interface/small file retrivial (GE)
- 3.CLI interface/big file insertion (PU)
- 4.CLI interface/big file retrivial (GE)
- 5.Backup application/number of small files (PU)
- 6.Restore application/number of small files (GE)
- 7.Backup application/heterogeneous files (PU)
- 8.Restore application/heterogeneous files (GE)
- 9.System Imaging (PU)
- 10.System Restore (GE)
- 11.Parallel automated file insertion from multiple hosts (PU)
- 12.Parallel automated file retrivial from multiple hosts (GE)
- 13.Parallel automated file insertion and retrivial from multiple hosts (PU and GE)

In the case of the steps marked as (PU) each file would be added to the storage overlay, in case of a big file it will be splitted in fixed size chunks and each chunk will be submitted to the overlay as a unique file. Additionally it maintains a distributed self-organized database (balanced and fault-tolerant) with the metadata of the files and their network locations.

In the case of the steps marked as (GE) each file, or chunk, would be located in a distributed way and retrivied from the storage overlay. As the files could be stored on different host the middleware could make use of parallel streams.

2 Conclusions

The work on middleware design changes and extensions described in this document is on track and in line with the schedules set up in the description of work.

During the first year the partners in WP3 focussed the work on the following four areas of the PHOSPHORUS infrastructure:

- Installation and adaptation of UNICORE 6 as the initial middleware for the experiments with the selected applications

- Installation and adaptation of the MetaScheduling Service to provide a new interface with UNICORE 6 that will allow a similar integration with other middleware like GT4 as well

- Definition of the architecture and the components of the semantic resource selection service, selection and test of the necessary tools, creation of the ontology of application requirements, evaluation of an existing resource ontology

- Deployment and installation of INCA at the PSNC site. Negotiations with UESSEX and CESNET for an INCA deployment on their sites.

3 References

[ARGON] Allocation and Reservation in Grid-enabled Optic Networks
http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0051.pdf

[EASY] Extensible Argonne Scheduling system
http://www.springerlink.com/content/pn740068375677wt

[CIM] Common Information Model
http://www.dmtf.org/standards/cim/

[GT4] Globus Toolkit 4
http://www.globus.org

[OASIS] Organization for the Advancement of Structured Information Standards
http://www.oasis-open.org

[OGSA-RSS] Open Grid Service Architecture Resource Selection Services Working Group
http://www.ogf.org/gf/group_info/view.php?group=ogsa-rss-wg

[MSS] MetaScheduling Service
http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0010.pdf

[OGF] Open Grid Forum
http://www.ogf.org

[PELLET] http://pellet.owldl.com/

[PROTÉGÉ] http://protege.stanford.edu/

[SPARQL] http://www.w3.org/TR/rdf-sparql-query/

[TORQUE] http://www.clusterresources.com/pages/products/torque-resource-manager.php

[TUAM] Tool for Universal Annotation and Mapping
http://www.scai.fraunhofer.de/index.php?id=2384&L=1

[U@SOURCEFORGE] UNICORE Sourceforge Project
http://sourceforge.net/projects/unicore/

[unicore] **Uniform INterface to COmputing REsources**
http://www.unicore.eu

[VIOLA] Vertically Integrated Optical Testbed for Large Applications in DFN
http://www.viola-testbed.de/

[WS-RF] Web Services Resource Framework
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf - announcements

4 Acronyms

AAA	Authentication, Authorisation, Accounting
DDSS	Distributed Data Storage Systems
e2e	end to end
EGEE	Enabling Grids for E-sciencE (European Grid Project)
FC	Fibre Channel
FC-SATA	Fibre Channel to SATA technology (mixed technology used in disk matrices: disk matrix have Fibre Channel ports for hosts connectivity, but contains SATA disk drives)
GEANT2	Pan-European Gigabit Research Network
GEANT+	the point-to-point service in GEANT2
GMPLS	Generalized MPLS (MultiProtocol Label Switching)
G^{2}MPLS	Grid-GMPLS (enhancements to GMPLS for Grid support)
GT4	Globus Toolkit Version 4 (Web-Service based)
KoDaVis	Tool for Distributed Collaborative Visualisation
MSS	MetaScheduling Service
NREN	National Research and Education Network
NRMS	Network Resource Management System
NRPS	Network Resource Provisioning System
PoP	Point of Presence
Protégé	Ontology Editor and Knowledge Acquisition System
QoS	Quality of Service
SNMP	Simple Network Management Protocol
TOPS	Technology for Optical Pixel-Streaming
TPD	Tiled Panel Display
TUAM	Tool for Universal Annotation and Mediation
UNI	User to Network Interface
UNICORE	European Grid Middleware (UNIiform Access to COmpute REsources)
VLAN	Virtual LAN (as specified in IEEE 802.1p)
VIOLA	A German project funded by the German Federal Ministry of Education and Research (Vertically Integrated Optical Testbed for Large Applications in DFN)
VPN	Virtual Private Network
WISDOM	Wide In Silicio Dockong On Malaria