SEVENTH FRAMEWORK PROGRAMME

Research Infrastructures

Deliverable 4.2

Report on the needs identified and the solutions planned and/or implemented

Grant Agreement number:	RI-261572
Project acronym:	WeNMR
Project title:	WeNMR, A worldwide e-Infrastructure for NMR and structural biology
Deliverable type and status	Report – Public
Due Date	February 1, 2012
Delivery Date	February 3, 2012 (M15)
Project coordinator name, title:	Prof. Alexandre M.J.J. Bonvin
Organization:	Utrecht University, The Netherlands
Tel:	+31 30 2533859
Fax:	+31 30 2537623
E-mail:	a.m.j.j.bonvin@uu.nl
Project website address:	www.wenmr.eu

Table of Contents

Section 1: Summary of Deliverable

Background

The main goal of WP4 is the continuation and improvement of the Grid infrastructure and of the services that were developed within the e-NMR project, both by ensuring software and hardware updates and at the level of the interaction with users. Important for this deliverable is that the aim is also to extend the gateway will as required to meet the specific needs of WeNMR users and other stakeholders, such as software developers willing to contribute additional tools to the portfolio of WeNMR.

Goal

The goal of this deliverable was to identify specific needs of the WeNMR users and other stakeholders in terms of services to be deployed on the project grid.

Section 2: Results

Background

In Task 4.3, the project partners 1-3 have identified specific needs of the WeNMR users and other stakeholders in terms of services to be deployed on the project grid. Each partner has been acting independently in identifying these needs, but coordinated with the others also under the guidance of Partner 4 (INFN) to devise the best corresponding solution(s).

List of tools developed

Consequently, the following tools have been developed and made available to users and other communities / VRCs via the WeNMR portal (http://www.wenmr.eu/wenmr/support/documentation/grid-services/glite):

- Automated grid submission and polling daemons
- Jobs - user friendly utility to manage the users jobs
- perusal option: save your job output before reaching walltime
- eToken with Xen (grid admin)
- JobControl - utility to manage multi jobs submission (for site admin)
- Testing accessibility of all SE from all CE (for grid admin)
- Automatic enmr.eu test job submission on all CEs (for site admin)
- CheckGRID - Utility to test and publish GRID resources (for site admin)
- Grid deployment scripts (for site admin)

As can be seen from the list above, the services developed target both administrators of grid sites and users. The former category has been addressed in order to facilitate the support of WeNMR by additional sites, leading to an easier expansion of the WeNMR computational capacity. Tools targeted at users try to make the submission and/or management of different type of calculations that require more complex submissions than an individual jdl. Below, some of the aforementioned services are briefly described/commented.

Automated grid submission and polling daemons

The “Automated grid submission and polling daemons” page provides scripts that enable the creation of web portals running calculation on the grid through a general mechanism completely transparent to the users that pools all jobs operations. The usefulness of these scripts to the WeNMR community is motivated by the fact that the entire concept of WeNMR service provision is centered around the usage of web portals. This scheme is characterized by a separation of three layers of operation, between which there is no direct communication, namely the web layer, the grid interface layer and the grid itself. Pools are created and used for storage of jobs and of results packages. User service requests are received via the web interface. They are then processed on the server, up to the point of generating a job package that is stored on disk. A daemon job (grid-submission) is running on

the grid interface on a scheduled base scanning the 'job pool' for job packages and submitting these to the Grid when found. Another daemon job (grid-polling) is periodically checking running jobs for their status, retrieving the results when ready and placing these in a result pool (this daemon can also resubmit failed jobs). Finally, results are presented back to the user, possibly after post-processing (results-processing). Submission, polling and retrieval of output are handled using a standard toolbox for Grid operations. Usage of this scheme is particularly advantageous when using robot certificates for Grid submission.

Jobs

The "Jobs" utility is a Python program that allows users to manage multi-jobs submissions. These are particularly common e.g. in calculations where hundreds of different conformations are analyzed to obtain a statistics enabling the interpretation of experimental data for flexible systems such as multi domain proteins or multi subunit biological systems. Such calculations are divided into jobs including tens of conformations and thus tens to hundreds of jobs must be simultaneously submitted, checked and retrieved. This can pose a significant overhead on users for the management of many files. The utility creates for the user an SQLite database where all the jobs are tracked. It additionally takes care of interactions with the grid environment, from proxy initialization to retrieval of results.

Testing accessibility of all SE from all CE

A series of script to systematically and automatically test the accessibility of all Storage Element (SE) from every Computing Element (CE) has been written. Some of the WeNMR jobs, when executed on the worknode of a CE, read and write some data on a SE, sometime located far from the CE. It is therefore important to ensure that this can be done. Possible problems could include a wrongly configured firewall or a failure in the authentication protocol between the CE and SE. This script allows the identification of such a problem so that the concerned site admin can be notified via a GGUS ticket.

Automatic enmr.eu test job submission on all CEs

A series of scripts to submit a test jobs everyday at 1am, 2am and 3am, on every Computing Element (CE) that support enmr.eu, and to retrieve the results 23 hours later has been written. We noticed in the past that often, some grid site got misconfigured. This usually includes some problem with the VOMS server certificate, some misconfigured environment variables, or some disks/partitions that were not correctly mounted. We want to be able to detect those issues as soon as they appear, and to ask the site admin to repair it as soon as possible, in order to ensure continuously a smooth execution of all the enmr.eu jobs. The series of scripts is composed of :

(i) a script to generate one jdl file per CEs supporting enmr.eu VO;
(ii) a script to submit all the created jdl files to the grid;
(iii) a script to check the status of all submitted job;
(iv) a script to retrieve the output of all jobs that finished successfully;

(v) a script to retrieve the output of all jobs that terminated with an error; and
(vi) a script to retrieve the logging-info of all jobs, with maximal verbosity.

Finally, a 'master script' launches the scripts (i) and (ii) at 1am, 2am and 3am via a cron job, while another master script launch the scripts (iii) (iv) (v) and (vi) to retrieve the results 23 hours later. A quick inspection of the results allows the identification of faulty grid site. The scripts log all relevant information to be able to open a GGUS ticket if necessary.

CheckGRID

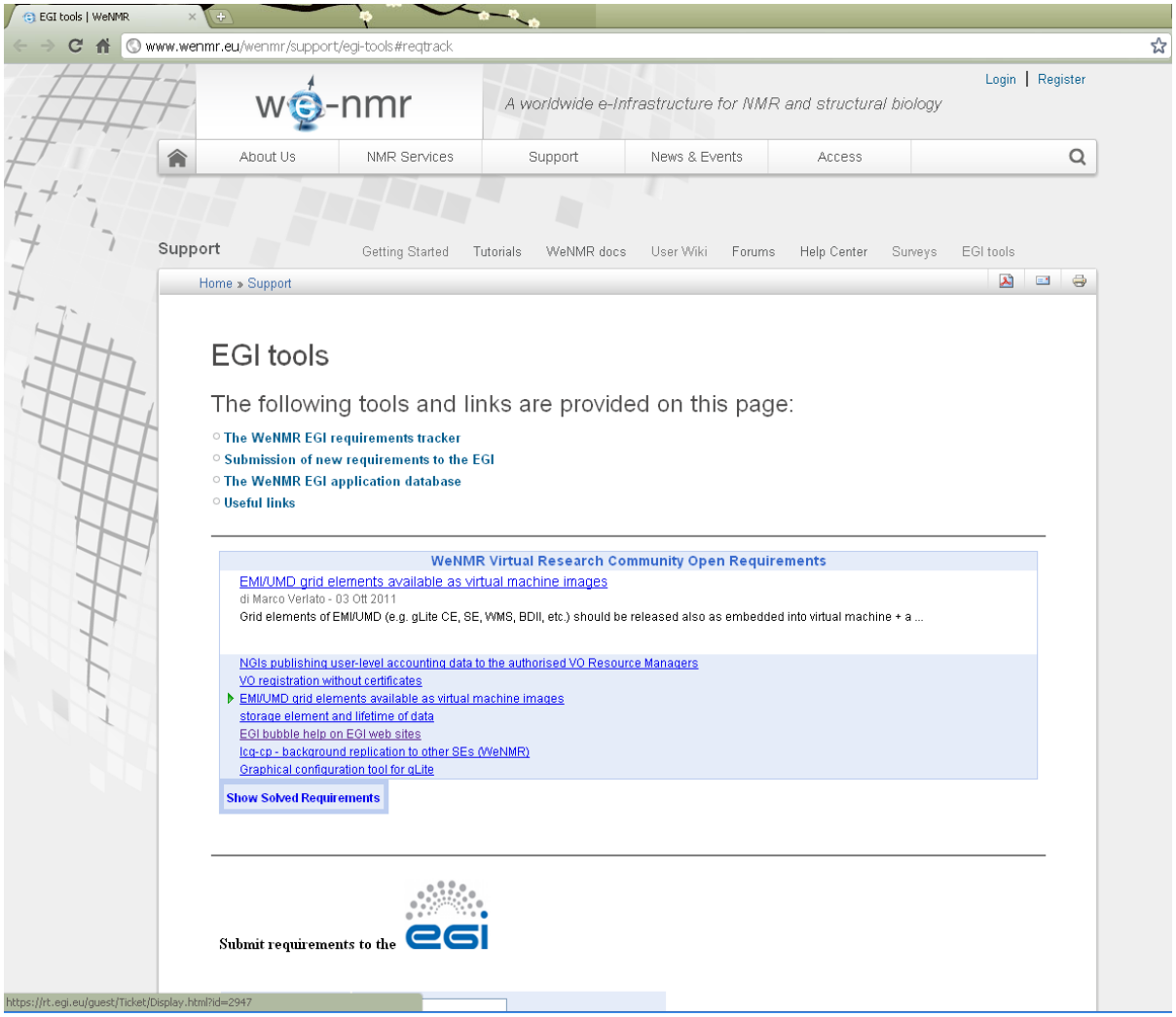
The "CheckGRID" tool is another Python program that allows one to check the availability of grid resources; this can be run a regular schedule (e.g. every two hours) to maintain the information always updated. The results are published in two dictionary files: one containing the history of the results, the other one containing the current queue statistics including the waiting time for the jobs execution. These data can be used to select the CE where to submit the jobs in order to guarantee that jobs will always start in a reasonable time. In this context, it is particularly useful to embed the analysis of the data provided by the tool just ahead of jobs submission to guarantee the fastest execution. SE's are also checked.

Grid deployment scripts

The "Grid deployment scripts" are an ensemble of scripts that allow an administrator to deploy software in a standardized manner across a multitude of sites. The scripts address all steps in this endeavor, from the creation of jdl's to perform the deployment to checking the results of the procedure and optionally tagging the various CE's. The scripts are useful both to add a new software to the existing grid infrastructure and to deploy the already available software programs to newly added CE's.

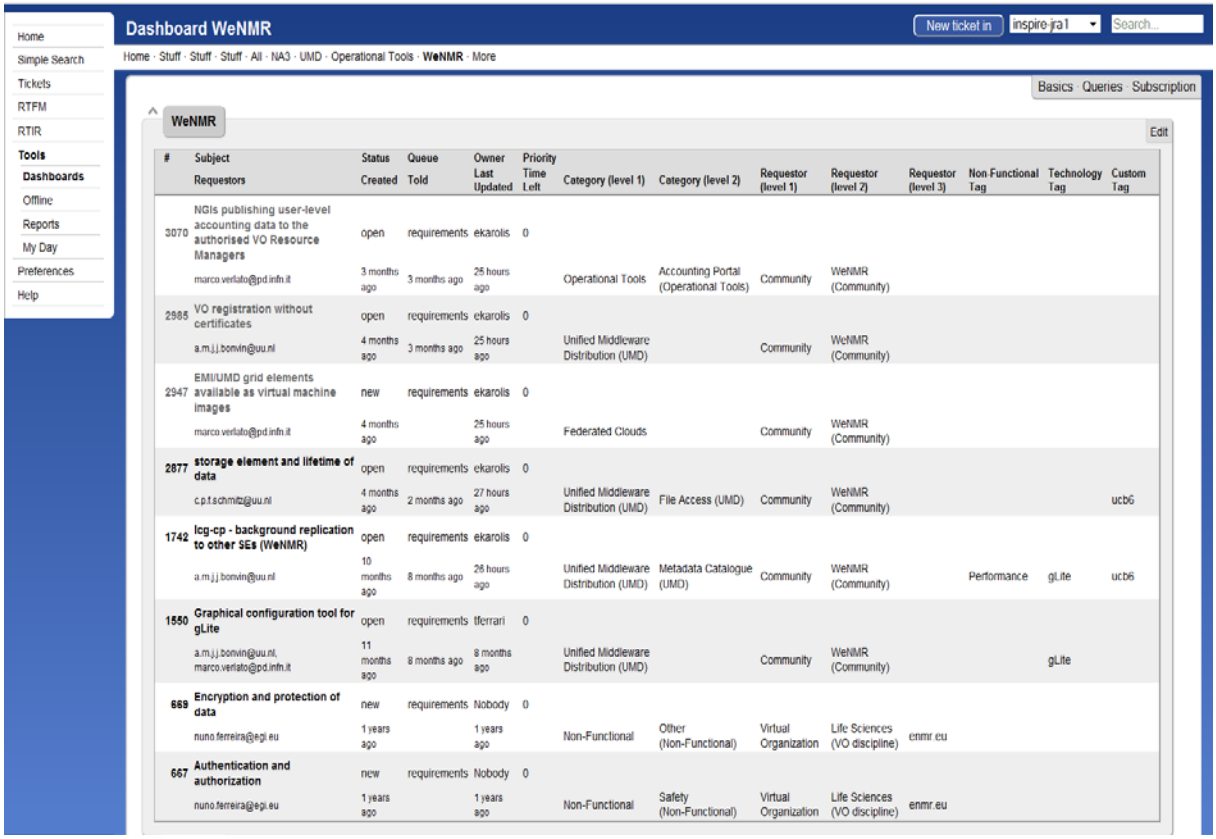
Requirements to EGI

WeNMR has also been active in sending requirements to EGI. The WeNMR gateway hosts a page dedicated to both following existing requirements (the requirement tracker) and to submit new requirements (at http://www.wenmr.eu/wenmr/support/egi-tools#reqtrack). A snapshot of the page is below:



The requirements submitted were about 20 in total. These ranged from signaling issues with various functionalities to, more commonly, requesting new tools or new features to be added in existing tools or gLite.

At present, there are eight open requirements, listed below:



Dashboard WeNMR

New ticket in inspire-jira1 Search...

Home · Stuff · Stuff · Stuff · All · NA3 · UMD · Operational Tools · WeNMR · More

Basics · Queries · Subscription

WeNMR

Edit

#	Subject / Requestors	Status / Created	Queue / Told	Owner / Last Updated	Priority / Time Left	Category (level 1)	Category (level 2)	Requestor (level 1)	Requestor (level 2)	Requestor (level 3)	Non Functional Tag	Technology Tag	Custom Tag
3070	NGIs publishing user-level accounting data to the authorised VO Resource Managers marco.verlato@pd.infn.it	open 3 months ago	requirements 3 months ago	ekarolis 25 hours ago	0	Operational Tools	Accounting Portal (Operational Tools)	Community	WeNMR (Community)				
2985	VO registration without certificates a.m.j.j.bonvin@uu.nl	open 4 months ago	requirements 3 months ago	ekarolis 25 hours ago	0	Unified Middleware Distribution (UMD)		Community	WeNMR (Community)				
2947	EMI/UMD grid elements available as virtual machine images marco.verlato@pd.infn.it	new 4 months ago	requirements	ekarolis 25 hours ago	0	Federated Clouds		Community	WeNMR (Community)				
2877	storage element and lifetime of data c.p.f.schmitz@uu.nl	open 4 months ago	requirements 2 months ago	ekarolis 27 hours ago	0	Unified Middleware Distribution (UMD)	File Access (UMD)	Community	WeNMR (Community)				ucb6
1742	lcg-cp - background replication to other SEs (WeNMR) a.m.j.j.bonvin@uu.nl	open 10 months ago	requirements 8 months ago	ekarolis 26 hours ago	0	Unified Middleware Distribution (UMD)	Metadata Catalogue (UMD)	Community	WeNMR (Community)		Performance	gLite	ucb6
1550	Graphical configuration tool for gLite a.m.j.j.bonvin@uu.nl, marco.verlato@pd.infn.it	open 11 months ago	requirements 8 months ago	lferrari 8 months ago	0	Unified Middleware Distribution (UMD)		Community	WeNMR (Community)			gLite	
669	Encryption and protection of data nuno.ferreira@egi.eu	new 1 years ago	requirements	Nobody 1 years ago	0	Non-Functional	Other (Non-Functional)	Virtual Organization	Life Sciences (VO discipline)	enmr.eu			
667	Authentication and authorization nuno.ferreira@egi.eu	new 1 years ago	requirements	Nobody 1 years ago	0	Non-Functional	Safety (Non-Functional)	Virtual Organization	Life Sciences (VO discipline)	enmr.eu			

The requirements fell into a variety of categories, but related most commonly to middleware aspects. The requirement submitted most recently involved the publication of UserDN in the accounting systems. This is important for WeNMR as we are using the VO FQAN in the usage records for application accounting purposes. However, many NGIs having Resource Centres that support the enmr.eu VO do not publish this information into the EGI Accounting Portal. Among the solved tickets, ticket 1241 (https://rt.egi.eu/guest/Ticket/Display.html?id=1241), addressing MPI support, was particularly noteworthy and was solved in the EMI-1 release of April 2011.

Section 3: Summary

Nine tools have developed that directly target specific needs of the WeNMR community, both at the administrative level and at the users level. All these tools are documented in the WeNMR site and thus made available to other communities. The majority of them are targeted to administrators and software providers; this is not unexpected as the philosophy of WeNMR is to hide the complexity of grid operations to users by means of its gateway and web portals for services. We have mainly worked on facilitating the porting of new applications to the grid, the implementation of new portals, and the extension of the WeNMR grid itself. In the future, we will continue to monitor the needs of the user community as well as of all other relevant stakeholders, which will trigger the development of new tools.