

DataGRID

2003 PLANNING FOR WP2

MAY - DEC 2003

Document identifier:	DataGrid-02-PLN-0500-1-5
EDMS id:	
Date:	June 13, 2003
Work package:	WP02: Data Management
Partner(s):	CERN, PPARC, HIP, INFN, ITC, KDC
Lead Partner:	CERN
Document status:	DRAFT 1.5
Author(s):	WP2
File:	plan2003

Abstract: Planning of the work for WP2 in the last three quarters.

CONTENTS

1. REPLICA MANAGEMENT	4
1.1. PRIORITY 1	4
1.2. PRIORITY 2	6
1.3. PRIORITY 3 FOR WP2	9
1.4. PRIORITY 3 FOR WP5	13
1.5. PRIORITY 4	14
1.6. PRIORITY 5	15
1.7. ITEM HISTORY	16
2. TASK DETAILS	17
2.1. REPLICA MANAGER PROXY SERVICE AND NEW CLIENT	17
2.2. VALIDATION TESTS	17
2.3. FUNCTIONAL TESTS	17
2.4. SCALABILITY AND PERFORMANCE TESTS	18
2.5. CONFINED COLLECTIONS	18
2.6. EXTERNAL DEPENDENCIES	19
2.7. INSTALL TARGET	19
2.8. AUTOMATE TAGGING PROCEDURE	20
2.9. UN-CONFIGURE	20
2.10. DEPLOYMENT TESTS	20
3. SPITFIRE	21
3.1. PRIORITY 1	21
3.2. PRIORITY 2	21
3.3. PRIORITY 4	21
4. SECURITY	22
4.1. PRIORITY 1	22
4.2. PRIORITY 2	24
4.3. PRIORITY 3	25



Document Change Record

Issue	Date	Comment	Author
1.0	05/05/03	First draft.	Peter Kunszt
1.1	06/05/03	Additional items and revised dates	Leanne Guy
1.2	07/05/03	Additional items	Heinz Stockinger
1.3	08/05/03	Additional items	Kurt Stockinger
1.4	22/05/03	Added security items	Joni Hahkala
1.5	04/06/03	Added fine grained authz.	kos Frohner

1. REPLICA MANAGEMENT

1.1. PRIORITY 1

All priority 1 items are due 'ASAP'. The planned target dates for completion are nevertheless listed.

P1.1	WP2: Secure Services	<p>Each service needs to be tested in secure mode. This involves</p> <ul style="list-style-type: none"> • Write the Secure Service Installation Guide • Test the Guide by installing each service accordingly • Test the clients in secure mode • Update the installation / user guides if necessary 	<p>William Bell Joni Hahkala Giuseppe Andronico</p>	<p>May 16</p>
P1.2	WP2: Functional Tests	<p>Each service needs a package that can be run on a client to see whether all functionalities are available and run correctly. This test needs to be verbose, and provide a 'cleanup' as well so that it leaves no trace of itself after being run, or having failed. See section 2.3. for more details.</p>	<p>Kurt Stockinger Leanne Guy Peter Kunszt William Bell Giuseppe Andronico Sophie Lemaitre Diana Bosio James Casey Heinz Stockinger Ithzak Ben-Akiva</p>	<p>May 30</p>
P1.3	WP2: Test the RLI	<p>This involves</p> <ul style="list-style-type: none"> • Bloom Filter Tuning • Deployment, setup 	<p>James Casey Levi Lucio</p>	<p>End June</p>

P1.4	WP2: Multiple VO support	Supporting many VOs per service, installation instructions and tests	Gavin McCance Leanne Guy	June 15
P2.1	WP2: Migration Tool	Design and write the migration tool from RC to LRC. <ul style="list-style-type: none">• Design the migration tool• Tool for export• Tool for import• Test and documentation	Leanne Guy Heinz Stockinger	June 15
P2.2	WP5: Migration Tool	Design, write the migration tool for the SE <ul style="list-style-type: none">• Design and implement the migration tool• Test and documentation	Regina Tam Jens Jensen	June 15

1.2. PRIORITY 2

The P2 items are still higher priority than 'normal'.

P2.3	WP2: RLS/RMC Proxy and Client	This involves <ul style="list-style-type: none"> • Write down the design, go through ATF • Implement the proxy server • Implement the clients and update the replica manager 	James Casey Peter Kunszt Gavin McCance Leanne Guy	July 22
P2.4	WP5: GACLs	Integrate Access Control in the SE core <ul style="list-style-type: none"> • Plan and spec • Implementation 	Andrew W Tara Jens	July 18
P2.5	WP5: Asyn- chronous Operations	Make the create, cache calls behave asynchronously <ul style="list-style-type: none"> • Plan and spec (done) • Queueing • Implementation • Unit Tests 	Jens Jensen Owen Synge	August 8
P2.6	WP5: SRM v1	Provide SRM v1 interface. The proper implementation will need to use the asynchronous operations, so an adaptation is needed after Async ops are done.	Glen Johnsson Jens Jensen	July 4/Aug 15

P2.7	WP5: SRM extensions	<p>Provide extensions (subset of SRM v2 interface)</p> <ul style="list-style-type: none"> • Delete and exists calls, error handling • Plan and spec • Implementation • Unit Tests 	Glen Johnsson Regina Tam Jens Jensen	July 11
P2.8	WP5: Setup/Configuration	<p>Provide setup and configuration enhancements</p> <ul style="list-style-type: none"> • non-LCFG installation mechanisms • reorganization of current mechanisms • Scripting 	Owen Synge	End June
P2.9	WP2: Architecture document for release 2.0 functionality	<p>The architecture of release 2.0 is significantly different from previous versions. We have a lot of disparate documents describing various parts of the architecture but no clear overview. Include sequence diagrams and deployment concerns.</p>	Leanne Guy All	End June

P2.10	WP2: Scalability and Performance Tests	Design and write the tests that are to be carried out for scalability and performance. Same pool of people as functionality tests. See section ??. <ul style="list-style-type: none"> • Scalability tests: design • Implementation • Documentation 	Kurt Stockinger Leanne Guy Peter Kunszt William Bell Giuseppe Andronico Sophie Lemaitre Diana Bosio James Casey Heinz Stockinger Ithzak Ben-Akiva	Continuous
P2.11	WP2: Mini Testbed Operations	Operating the mini-testbed is higher-than normal priority. <ul style="list-style-type: none"> • Upgrades • Tests • Website 	Levi Lucio James Casey Giuseppe Andronico	Continuous
P2.12	WP2: Documentation	Documentation on all release 2.0 components.	Many people	Continuous
P2.13	WP5: Documentation	Documentation on all release 2.0 components.	Many people	Continuous



1.3. PRIORITY 3 FOR WP2

These are the 'normal' priority items.

P3.1	RLS GUI	<p>RLS Graphical User Interface, to the EDG implementation</p> <ul style="list-style-type: none"> • Integrate the changes in the existing GUI • Test and documentation 	<p>Livio Salconi James Casey</p>	<p>June 28</p>
P3.2	Security for C++ Clients	<p>Integrate the GSI security for the C++ clients and test it.</p> <ul style="list-style-type: none"> • Design, implement the changes in gsoap • Integrate in the existing modules • Test and documentation 	<p>Mika Silander Joni Hahkala Gavin McCance Marko Niinimäki</p>	<p>End May</p>
P3.3	Replica Subscription Service	<p>The RSS is the GDMP equivalent.</p> <ul style="list-style-type: none"> • Design (done) • Implement • Test and documentation 	<p>Gavin McCance Ithzak Ben-Akiva</p>	<p>R2.1</p>
P3.4	Replica Manager Proxy Service	<p>The RM proxy and new RM client need to be designed based on the delegation model (to be implemented) and the current functionality. See section 2.1..</p> <ul style="list-style-type: none"> • Design • Implement • Test and documentation 	<p>UNASSIGNED</p>	<p>R2.1. The design should be done by mid June</p>
P3.5	Validation tests	<p>Tests to validate the installation of services and clients. The validation tests should be runnable by anyone (usually the sysadmin who installed the components). See section 2.2.</p> <ul style="list-style-type: none"> • Implement • Test and documentation 	<p>Kurt Stockinger Leanne Guy Peter Kunszt William Bell Giuseppe Andronico Sophie Lemaitre Diana Bosio James Casey Heinz Stockinger Ithzak Ben-Akiva</p>	<p>Continuous</p>
P2.14	WP2: Optor-Sim	<p>Testing of new scheduling and replica optimisation algorithms.</p>	<p>David G. Cameron Ruben Carvajal-Schiaffino A. Paul Millar Caitriana Nicholson Kurt Stockinger Floriano Zini</p>	<p>Continuous</p>

P3.6	External Dependencies	Currently most external jars are being held in edg-java-tools. To update the dependencies is tedious and error-prone work, the jars can get easily out-of-date. This should be automated. See section 2.6.. <ul style="list-style-type: none"> • Implement • Test and documentation 	UNASSIGNED	June 13
P3.7	Install Target	Deploy the install ant target so that the jars get installed on Marianne's buildbox in \$EDG_LOCATION/share/java. See section 2.7.. <ul style="list-style-type: none"> • Implement • Test and documentation 	Ákos Frohner +1	May 30
P3.8	Tagging procedure automation	Write scripts to automate the tagging procedure. This can be very helpful when tagging all the software at once or just to avoid errors in tagging. See section 2.8..	UNASSIGNED	End July
P3.9	Un-configure script	There is a configure script but no un-configure script if a service needs to be removed. See section 2.9..	UNASSIGNED	End July
P3.10	Deployment Tests	Deploying the RPM, configuring the service and running the functional tests. See section 2.10..	UNASSIGNED	End August
P3.11	Confined Collections	Collections of GUIDs that are kept in the LRC and are replicated as one. See section 2.5.. <ul style="list-style-type: none"> • Design • Implement • Test and documentation 	UNASSIGNED	End July
P3.12	VOMS Integration	VOMS needs to be integrated into the secure services.	Olle Mulmo	R2.1?
P3.13	CVS Server	Run the CVS server.	Paul Millar	Continuous
P3.14	Build Box	Run the Build Box.	Levi Lucio	Continuous
P3.15	ITeam Rep	ITeam representative. Provides feedback to/from ITeam, is up-to-date on service status.	Paul Millar	Continuous

P3.16	ATF Rep	Architecture Task Force representative. Provides feedback to/from ATF, produces control-flow diagrams for all services, organizes 'brainstorm' sessions on hot items.	Leanne Guy	Continuous
P3.17	QA Rep & Enforcement	Represent WP2 in the QA group and check whether the QA requirements are actually fulfilled by the WP2 modules.	Diana Bosio +?	Continuous
P3.18	Bugzilla Roster	Accept and assign bugs.	Leanne Guy Coordination of weekly rotation for all.	Continuous
P3.19	Website Maintenance	Maintain website such that the information is up-to-date, new information is available. Each task manager is responsible for the sub-contents of all task sites, everyone has write-access to the website. So the names are just the responsables, but everyone should feel free to update the site for the best.	Peter Kunszt(main) Joni Hahkala (sec) Marko Niinimäki (spitfire) Kurt Stockinger (optimization) Leanne Guy (ATF & RLS & Bugzilla) Levi Lucio (bob) Heinz Stockinger (replication)	Continuous
P3.20	LCFG	Coordination of LCFG testing and deployment. Note that writing the LCFG objects is the responsibility of all.	Leanne Guy Paul Millar(?)	Continuous
P3.21	SE Access Estimation for ROS	Integrate SE estimator of WP5 and Cross Grid in Replica Optimization Service.	Kurt Stockinger	continuous

1.4. PRIORITY 3 FOR WP5

These are the 'normal' priority items.

P3.22	VOMS integration	SE webservice needs upgrades to integrate VOMS implementation <ul style="list-style-type: none"> • Integrate the changes in the existing GUI • Test and documentation 	Glen Johnsson Akos Frohner	July 22
P3.23	Disk Cache Management	SE core development, involving <ul style="list-style-type: none"> • Needs to be fully spec'd out • Allow for pinning files in the cache • Cleanup on pin expiration • Depends on async ops • Test and documentation 	Jens Jensen Owen Synge	Mid August
P3.24	Log Management	Clean out and simplify logging	Tara Regina	June 28
P3.25	Interaction with WP9/WP10	help WP9/10 to get the SE on their MSS	Andrew Regina	End July

1.5. PRIORITY 4

Not essential, but useful.

P4.1	Statistics on Build Box	Add statistics information to build page (total code size, total classes).	Paul's students?	June 27
P4.2	Script to find out EDG Variables	The scripts already exist (Joni wrote them) and what remains to be done is to install them and make all the packages which need the variables resolved refer to them.	Joni Hahkala?	May 30
P4.3	Logical Collections in RMC	Collections on LFNs in the RMC. This is depending on further input from the ATF, HEPICAL and LCG, hence the low priority. Might never happen.	UNASSIGNED	R2.1?
P4.4	Publications at International Conferences, Journals etc.	Present data management software and activities in international conferences, journals etc. in order to make the activities of EDG public and participate to the international Grid community.	Heinz Stockinger, all	Continuous
P4.5	Access History for ROS	Provide access history for optimization algorithms.	Kurt Stockinger	continuous
P4.6	Optimization Algorithms for ROS	Support additional replica optimization algorithms already tested in OptorSim.	avid G. Cameron Ruben Carvajal-Schiaffino A. Paul Millar Caitriana Nicholson Kurt Stockinger Floriano Zini	continuous

1.6. PRIORITY 5

Items listed here as priority 5 are items we have considered but have currently not received any request to implement from the EDG applications. Currently no dates are assigned. Unless we receive any feedback on these items they will not be implemented before the end of EDG

P5.0	Consistency Service	Consistency service as indicated in Reptor design papers as well as in various conference papers. A consistency service takes care of updates in replicas: either simple versioning or real updates. Note that this activity is mainly a <i>research activity</i> to provide a <i>first prototype for possible follow up projects</i> . There is not aim to turn this into a production service within the lifetime of EDG.	Heinz Stockinger, Kurt Stockinger, possibly others	no deadline
------	----------------------------	---	--	-------------

1.7. ITEM HISTORY

Items here have already been completed and are listed to keep them in the document for tracking purposes.

-	WP2: LCR/RMC Native Attributes	<p>Currently the LRC only takes attributes of type string. The functionality needs to be extended in order to provide attributes with types of integer, float and date as well.</p> <ul style="list-style-type: none"> • Design • Implementation • Testing and Documentation 	Sophie Lemaitre	May 30
-	WP2: SRM Castor Integration	<p>Integrate the Castor SRM into the replica manager.</p> <ul style="list-style-type: none"> • Implementation • Testing 	Peter Kunszt Emil Knezo	May 16
-	WP2: Write the RLI	<p>This involves</p> <ul style="list-style-type: none"> • Bloom Filters (write) • Sending updates from LRC • Receiving updates at RLI, storing filters • Admin methods at LRC to add/remove RLIs to send filters to 	James Casey Levi Lucio	End May

2. TASK DETAILS

2.1. REPLICA MANAGER PROXY SERVICE AND NEW CLIENT

Based on the delegation model, the RM proxy service needs to serve all requests that are 'remote', i.e. it needs to enable the local WN to be shielded from external connectivity, acting through the RM proxy only.

The design should be driven by KISS and performance requirements. All of the calls that can be handled using local services should be kept in the existing client (like interactions with the LRC).

Additional functionalities to consider:

- Buffering of Info-provider queries (for speed, RGMA may be slow..)
- Fail-safe write operation to RMC: Buffer write requests if the RMC is not reachable, returning a 'queued' status. The job may continue and the RM-proxy queues the write requests to the RMC and executes them once the RMC is up again. If the operation times out / fails, a message should be sent to a valid email address.
- Book-keeping of ongoing replications to local SRM. This is a pre-requisite to be able to implement the non-blocking 'prefetch' capability of getBestFile.

2.2. VALIDATION TESTS

Installation validation of the services, clients and modules can be achieved by a set of scripts that perform:

- Check all dependencies, installed components
- Check configuration of service
- Check accessibility and configuration of database
- Check whether the system is responding, up and running
- Check version
- Check security

2.3. FUNCTIONAL TESTS

Functional tests are client-side tests of the services (RLS, RMC, ROS) and test-suites for the replica manager. The functional tests should be runnable by anyone with enough privileges, usually the sysadmin who has installed the package.

A complete functional test suite has the following characteristics:

- Func-tests come with a separate RPM to be installed on a client (for services) or where the given module is available (ERM).
- The test services are configurable through a configuration file (i.e. location of the service endpoint, names of the test files, etc)
- The test package comes with a detailed documentation describing each test and indications for detecting failures/success.

- There is one command that runs all tests.
- The output to stdout is like junit's (one dot per success). The logfile should contain very verbose messages.
- All commands are called in a mode where they are expected to succeed.
- All commands are called erroneously where they are expected to return the correct error message.
- If the tests fail, the service(s) might be in a state where temporary test data is stored (like registered test-files). There is a mechanism to clean this temporary data. This cleanup is called also when the test is restarted.

2.4. SCALABILITY AND PERFORMANCE TESTS

Scalability and performance tests are supposed to give numbers on the limits of the system. There are documents that can be taken as starting point what to test for, like the CMS requirements for replica catalogs written by Koen Holtman a year ago.

The following needs to be tested, i.e. limits need to be given:

- Behaviour of the server while having many concurrent requests for different operations.
- Timed numbers for all operations, depending on server load and on amount of data in the database.

[Please think of more..]

2.5. CONFINED COLLECTIONS

Confined collections are sets of GUIDs. Each confined collection gets a GUID of its own, hence it's possible to have sub-collections. They are implemented on the LRC, since each member of a confined collection has to be available at a given site.

The following operations need to be present in the collection API of the LRC:

newCollection Create a new collection, returns the GUID. There has to be an array of GUIDs passed in to define the initial set, an empty collection cannot exist on the LRC.

addMember Add one (or many) members to the collection. Members are always given as GUIDs. The difference in semantics to newCollection is only that this method checks whether the collection is already there and whether the members are not already set (in which case they are ignored, no error is thrown).

deleteMember Delete one (or many) members from a collection.

listMembers Returns all GUIDs that are member of the give collection.

isMemberOf Returns all collection GUIDs that a given GUID is member of. This does not recurse, i.e. if a collection is a subcollection of another collection, its members are not found to be members of the higher-level collection.

isMember Returns true or false, checking whether a given (set of) GUID(s) is actually member of a given collection.

deleteCollection Remove the collection.

The proposed first-order design for the tables necessary to store collections is simple: Just add a single table having the columns `Collection_GUID` and `Member_GUID`. All of the above methods can be implemented as simple select and insert statements in this table.

Another possibility to look at (similarly to the design by AliEn) is to store the collections in several tables, grouped by collections in order to enable really large numbers of collections. The bookkeeping of the collection tables is in yet another table.

2.6. EXTERNAL DEPENDENCIES

Getting rid of the external jars in `edg-java-tools` or at least make them always up to date: It is possible to periodically gather the necessary external jars automatically in the following fashion. The idea is to have the external jars in a cvs module (e.g. "`edg-jar-cache`") and to update the periodically in the following way: Tests to validate the installation of services and clients. The validation tests should be runnable by anyone (usually the sysadmin who installed the components).

- Download all the external rpms specified in a given configuration file in "`edg-jar-cache`";
- Install them to a writable area;
- Retrieve from the installations all the required jars (the ones that are required may also be specified in the configuration file);
- Do a diff with the older jars which are already in "`edg-jar-cache`" and replace the ones that have been updated;
- Commit the changes if necessary.

This algorithm can be ran as a script by a cron job, say, every week. It is also presupposed that during the WP2 build `$EDG_LOCATION` will point to "`edg-jar-cache`" which will make this system compatible with marianne's build. Another requirement here is that the changes to "`edg-jar-cache`" will be limited to some administrator, e.g., the user from whose account the above script will be run.

2.7. INSTALL TARGET

The idea here is to put all the install code in the install target passing it a location parameter that would specify the root of the install. In this way it would be possible to use the install target for:

- Installation in marianne's build, following the dist target. This would do a direct installation to `$EDG_LOCATION`;
- Installation in the release/package area of the build, with the purpose of creating tarball that will be used to generate the rpm.

This needs to be well thought of given that it will imply probably reviewing the the products created by the dist target so that they are already ready for install before the release target is ran. We will probably have to create the "`war`" and "`ear`" in dist, which is OK given that now the difference of responsibilities between the dist and the release target is not clear.

However, this is a lot of work since it needs to be done for all the packages. Akos has volunteered to do a first prototype of the `edg-voms-admin` package which can be used in the future as a base for all other packages.

2.8. AUTOMATE TAGGING PROCEDURE

A script to automatically increase either patch, minor, major number or a combination of them seems to be useful. A way to deal with the changelogs here seems to be just adding automatically the tag comment at the top of the CHANGES file and leave up to the developers the task of adding each individual change.

The script would be able to work in two modes: passive, meaning that the commands would be generated to the standard output but not executed - this would allow the developer to copy/paste/execute in order to check the procedure; active, meaning that the script would execute all the necessary steps.

2.9. UN-CONFIGURE

The configure script creates several other files, registers services in the info service, etc. The RPM uninstallation does not take care of these things, there needs to be a script for each service that will

- Unregister the service from the info provider
- Remove all files that have been created by the configure script
- Clean the databases if any (operation to be confirmed using the db root password).

2.10. DEPLOYMENT TESTS

Enable the deployment of the functional tests during the build. This needs to be addressed carefully given that we need to understand how to deploy the tests outside the box of the developer, specially in what concerns services that involve writing to databases.

The following algorithm specifies a way of executing the functional tests for a given package:

- Install the package, client and func-test rpm;
- Run the configure script for the package;
- Run the functional tests (see section 2.3.;
- Run some unconfigure script (see section 2.9.). The unconfigure script will do the opposite of the configure one;
- Uninstall the package rpm.

This algorithm will work as long as the basic services (Tomcat, MySQL, RGMA) are installed in the machine where the tests will be performed. One also has to take care of having the correct write permissions to do the rpm install as well as the configure step. A possible workaround would be to use an \$EDG_LOCATION different from /opt/edg, but this requires the servlet to be picked up by tomcat from the correct place.

This way of solving the problem looks fairly promising, given that with this sort of test would go through all the steps, including the configure one. This would hopefully avoid configure bugs that we only find after releasing.

3. SPITFIRE

3.1. PRIORITY 1

P1.5	Secure Service client	Service C++	enable queries etc to run using gSoap	Gavin McGance, Marko Niinimaki	June 15
P1.6	Functional testing rpm		prepare an rpm that adds functional tests in /sbin/test. Mainly: a “ping” that the service is present, query Spitfire with a credential/policy/role that is ok, query Spitfire with a cred/policy/role that is NOT ok (should fail gracefully), create/update/remove a table with a cred/policy/role that is ok.	Unassigned	June 15

3.2. PRIORITY 2

P2.15	SpitfireBrowser without Oracle XSQL		modify SpitfireBrowser so that xSQL access config is not needed, i.e., it can use the same configuration as Spitfire	John White, Marko Niinimaki	June 15
-------	--	--	--	-----------------------------	---------

3.3. PRIORITY 4

P4.7	Distributed queries		Interface Spitfire with an agent that can access several Spitfire installations and combine the results. Footnote: actually, a prototype exists already, but should be packaged in some “contrib” rpm.	HIP people	Sep 1
P4.8	OGSA compliance		What would it mean in practice?	Gavin McGance	?

4. SECURITY

4.1. PRIORITY 1

All priority 1 items are due 'ASAP'. The planned target dates for completion are nevertheless listed.

P1.7	Authentication Packaging	<p>Finish the packaging of authentication</p> <ul style="list-style-type: none"> • Test if browsers need the actual server credentials • If browsers need the server credentials, add the section for them to install scripts and to documentation • Test LCFG and make it work • Update the installation guide if necessary 	<p>Joni Hahkala Gavin McCance Leanne Guy</p>	<p>May 30</p>
P1.8	Test class	<p>Finish writing a simple class for authentication testing that connects the server and checks that the handshake works and prints all the CAs from both ends and any other info available.</p>	<p>Joni Hahkala</p>	<p>Jun 15</p>
P1.9	Developer's release of authz	<p>Package the Authorization framework into a developer's release</p> <ul style="list-style-type: none"> • Collect the stuff from wikihip into a developer's doc adding text where needed. Covers http and soap stuff • Collect the stuff from wikihip into a install appendix for services adding text where needed. • put necessary jars in • examples of all the configuration files affected • scripts if needed • the LCFG parts for configuring the authz system for services. This needs help from others and can be done a bit later. 	<p>GianLuca Volpato</p>	<p>May 30</p>

P1.10	First cut at delegation	Implement a working delegation system that can be tested and played with to get experience.	Niklas Karlsson Ville Nenonen Henri Mikkonen Tuomas Nissi	Jun 15
P1.11	Secure gSOAP	Finish the gSOAP hacking to make it accept proxies from both sides. <ul style="list-style-type: none">• make it work• combine with gSOAP-base• Developer's doc	Mika Silander	Jun 15

4.2. PRIORITY 2

Things that are not critical, but more important than normal.

P2.16	Build system	Fix ant stuff to use install target and remove corresponding functionality from the spec file.	Joni Hahkala	May 30
P2.17	Delegation developer's release	<p>Package the preliminary delegation package into developer's release.</p> <ul style="list-style-type: none"> • Write a developer's doc • write a install appendix for services • put necessary jars in • examples of all the configuration files affected • scripts if needed • the LCFG parts for configuring the authz system for services. This needs help from others and can be done a bit later. 	Niklas Karlsson Ville Nenonen Henri Mikkonen Tuomas Nissi	Jun 30
P2.18	Fine grained authorization	Put fine grained authorization in the replica management tools to be able to satisfy BioMed security requirements.	Ákos Frohner	Sep 1

4.3. PRIORITY 3

Things that are of normal importance.

P3.26	Finish Trust-Manager tests	Add missing Junits.	Joni Hahkala	Jun 15
P3.27	Authz func tests	Complete the authorization manager functional test system with automatic server etc setup	Gian Luca Volpato	Jun 30
P3.28	Other delegation storages	Write the filesystem and/or database storage for delegations	Niklas Karlsson Ville Nenonen	Jul 30
P3.29	SOAP delegation protocol	Write the SOAP version of the delegation protocol. If at all possible make it compatible with OGSA/WS-security	Henri Mikkonen Tuomas Nissi	Jul 30
P3.30	Developer's guidify security	Finish making the security conform to EDG developer's guide where possible.	Joni Hahkala and everybody else	Jul 30
P3.31	Service proxies	Develop the current single-proxy system into service-proxy system. Also provide install appendix, developer's doc and templates for services.	Gavin McCance Akos Frohner James Casey Joni Hahkala	Jul 15