



RAS - MD TEST REPORT

SOFTWARE TEST AND VALIDATION REPORT

WP4 TASK4 - Verification and Quality Control

Document Filename: **CG-4.4-REP-v1.0-DEMO005-RAS-MD_TestReport.doc**

Work package: **WP4 TASK4 - Verification and Quality Control**

Partner(s): **NCSR "Demokritos"**

Lead Partner: **LIP**

Config ID: **CG-4.4-REP-v1.0-DEMO005-RAS-MD**

Document classification: **PUBLIC**

Abstract: This report describes the validation performed on the package RAS-MD developed by CrossGrid WP 3 Task 1. The tests were performed by Vangelis Floros on behalf of the CrossGrid task 4.4 Testbed, verification and quality control. Migrating Desktop is an advanced user-friendly graphical tool that serves as uniform grid working environment independent on specific grid infrastructure. The Roaming Access Server is a set of modules and plug-ins that provides interfaces to work with grids.

Delivery Slip

	Name	Partner	Date	Signature
From				
Verified by				
Approved by				

Document Log

Version	Date	Summary of changes	Author
1-0-DRAFT-A	7/11/2004	Draft version	Vangelis Floros

CONTENTS

1. CONTEXT.....	4
1.1. TEST REQUEST	4
1.2. TEST TEAM.....	4
1.3. RESOURCES INVOLVED.....	5
2. TEST AND VALIDATION.....	6
2.1. SOFTWARE INSTALLATION.....	6
2.1.1. Tomcat	13
2.1.2. RAS.....	13
2.1.3. LDAP	16
2.1.4. MySQL.....	16
2.2. ADDITIONAL TESTBED MODIFICATIONS	17
2.3. TEST DEVELOPMENTS	17
2.4. USABILITY.....	17
2.5. FUNCTIONALITY	17
2.5.1. Unit tests	18
2.5.2. System tests	18
2.5.3. Stress tests.....	18
2.6. COMPATIBILITY.....	18
2.7. SECURITY AND NETWORKING.....	18
2.8. PREVIOUSLY REPORTED ISSUES	20
3. ISSUES FOUND	21
3.1. ISSUES FOUND IN THE SOFTWARE	21
3.1.1. Issue 001 [bugs #594]	21
3.1.2. Issue 002 [bugs #595]	25
3.1.3. Issue 003 [bugs #596]	25
3.1.4. Issue 004 [bugs #432]	25
3.1.5. Issue 005 [bugs #488]	25
3.2. ISSUES FOUND IN THE DOCUMENTATION.....	25
3.2.1. Issue 001 [bugs #428]	25
3.2.2. Issue 002 [bugs #422]	25
3.2.3. Issue 003 [bugs #597]	26
4. RECOMMENDATION	27
5. REFERENCES	28
6. INTEGRATION/VALIDATION REQUEST	29

1. CONTEXT

Test and validation of the package RAS-MD (Roaming Access Server – Migrating Desktop) developed by CrossGrid WP 3 task 1.

The Roaming Access Server (RAS) offers a well-defined set of web-services that can be used as an interface for accessing Grid systems and services in a common, standardized way. It consists of several independent parts responsible for job submission, job monitoring, user profile management, data management, authorisation, and application information management. RAS may support wide variety of clients including personal computers, laptops, and in the future PDA, and mobile phones. It is specifically targeted for supporting mobile grid users.

Migrating Desktop is an advanced user-friendly graphical shell that serves as uniform grid working environment independent on specific grid infrastructure. It provides a Java based GUI designed especially for mobile users and is independent of OS platform (MS Windows, Linux, Solaris) and hardware (personal computers, laptops, workstations). MD is a complex environment that integrates many tools and allows working with many grids transparently and simultaneously. It exploits capabilities provided as Web Services from RAS and JSS (Job Submission Service).

1.1. TEST REQUEST

The test request was submitted by Miroslaw Kupczyk (pawelw@man.poznan.pl) on 21 September 2004, using the CrossGrid Integration and Validation request form (<http://www.lip.pt/cgi-bin/middlewaretest-request.cgi>) and it was assigned the request ID 109577623283.7149291372956. The package test and validation was immediately assigned by Jorge Gomes (Jorge@lip.pt) to Vangelis Floros (floros@di.uoa.gr).

The initial request concerned version 4.0.0 of RAS-MD nevertheless during the tests additional versions were produced in order to fix bugs that were being discovered. Thus the latest version of the software actually tested was 4.0.10.

The software URL provided in the “Test Request” did not point to CrossGrid’s repository but rather to the CVS source files area at:

http://savannah.fzk.de/cgi-bin/viewcvs.cgi/crossgrid/crossgrid/wp3/wp3_1-portals/

The actual RPMs were located and downloaded from the autobuild area of gridportal:

<https://savannah.fzk.de/distribution/crossgrid/autobuilt/i386-rh7.3-gcc3.2.2/wp3/RPMS/>

The RPMs are also available from the main repository area of gridportal which means that they can be downloaded automatically using an LCFGng based install and the cg-lcfg-go script.

The URL for the Installation and users manual pointed to the documentation area of WP 3.1 in gridportal where the two respective documents can be easily located in Word format.

The URL for the Development manual pointed again to an area in Gridportal where one can navigate and retrieve API documentation in JavaDoc HTML files, for the development JSS plugins. Nevertheless no concrete documentation is provided explaining the process.

For the Software requirements and Software design documentation the URLs provided were incorrect. Apparently for this information the interested reader has to consult the relevant WP 3.1 and 3.2 deliverables.

1.2. TEST TEAM

The tests were performed by task 4.4 members from NCSR “Demokritos”.

- Vangelis Floros (floros@di.uoa.gr)

1.3. RESOURCES INVOLVED

For the purposes of testing a large number of computing resourced were utilized:

- 1 dedicated RAS machine (**cgnode03.di.uoa.gr**) was allocated in the University of Athens for the local deployment of RAS-MD and all the required supporting services (LDAP, MySQL etc).
- All the CrossGrid LCG-2 (tag 2.0.10) production testbed sites and in particular:
 - Host **rb01.lip.pt** was used as the Resource Broker.
 - Host **ii01.lip.pt** was used as the Information Index.
 - Various CEs registered in the LCG-2 sites were used for testing job submission and monitoring facilities provided by RAS-MD.
 - Various SEs in the testbed were used for the replication and file management operations provided by MD.
- A number of Workstations at the University of Athens were used as clients to test Migrating Desktop, running either as Applet or Java WebStart application.

2. TEST AND VALIDATION

2.1. SOFTWARE INSTALLATION

As it has been decided by the CrossGrid integration team, RAS-MD should comprise a standalone machine which would serve a specific VO (typically CG). Yet a process for setting up the software in such a machine did not exist at the beginning of the T&V procedure. Therefore it was decided that one of the outcomes of T&V should also be such an installation process together with the configuration files and any required scripts needed for the automatic system setup. The process should be based on LCFGng and should limit as possible the manual steps required.

The first step was to prepare the rpm lists for the new machine. As a basis the rpmlists of the UI machine was used since RAS-MD developers suggested, in their T&V request, that the software should be installed on the local UI. The software is comprised by five (5) RPMs which contain the main RAS-MD system together with the configuration files needed for preparing LDAP and MySQL databases:

- cg-wp3.1-RAS-services-4.0.10-1
- cg-wp3.1-portals-MD-4.0.10-1
- cg-wp3.1-JSS-client-4.0.10-1
- cg-wp3.1-RAS-ldap-4.0.10-2
- cg-wp3.1-RAS-mysql-4.0.10-1

The software depends on Apache Web Server and Apache Tomcat Servlet container. The following RPMs already available from the CrossGrid repository were used to fulfil these dependencies:

- apache-1.3.27-3
- tomcat4-4.1.24-full.2jpp

For the LDAP software a bundled version of OpenLDAP 2.0.27, prepared by RAS-MD developers was used. This version is comprised by four RPMs:

- openldap-clients-2.0.27-2.7.5multimaster
- openldap-2.0.27-2.7.5multimaster
- openldap-servers-2.0.27-2.7.5multimaster
- openldap-devel-2.0.27-2.7.5multimaster

To satisfy the dependency on MySQL the already available rpmlist *mysql-server-CG-rpm.h* was included. All of the above were edited in file *RoamingAccessServer-CG-rpm.h* located in directory `$COMMON/cg/ng_rpmlist` (note: `$COMMON` points to the path of the local common repository as it is downloaded from the CVS).

```
/*
  RoamingAccessServer-CG-rpm.h
  =====
  Additional packages for CrossGrid UIs
*/

#ifndef _REMOTEACCESSSERVER_CG_RPM_H
#define _REMOTEACCESSSERVER_CG_RPM_H

+apache-1.3.27-3
+tomcat4-4.1.24-full.2jpp

#include "mysql-server-CG-rpm.h"
```

```
+ca_HellasGrid-local-0.1-1
+cg-wp3.2-jss-cpp-srv_gcc3_2_2-2.1.15.2.4-1/i486
-openldap12-1.2.13-8
-openldap-devel-2.0.27-2.7.3

+openldap-clients-2.0.27-2.7.5multimaster
+openldap-2.0.27-2.7.5multimaster
+openldap-servers-2.0.27-2.7.5multimaster
+openldap-devel-2.0.27-2.7.5multimaster

+cg-wp3.1-RAS-services-4.0.10-1
+cg-wp3.1-portals-MD-4.0.10-1
+cg-wp3.1-JSS-client-4.0.10-1
+cg-wp3.1-RAS-ldap-4.0.10-2
+cg-wp3.1-RAS-mysql-4.0.10-1

#endif /* _REMOTEACCESSSERVER_CG_RPM_H */
```

The entry point of the RAS-MD machine RPM lists is the file RAS-rpm which is located in directory */opt/local/linux/7.3/rpmsg*:

```
/*
  RAS-rpm
  =====
  Roaming Access Server high-level RPM list
*/

/* Rpmfile files path definitions */
#include "cfgdir-rpm.h"

/* Include the complete CrossGrid setup */
#include CG_RPMDIR/RAS-CG-rpm>

/* Local extra packages you might want to install */
#include "local-rpm.h"
```

The main configuration file for RAS-MD is *RoamingAccessServer-CG-cfg.h*. This file together with the respective LCFGng object which is being developed by FZK is intended to perform most post-installation configuration steps, mainly the population of file *RAS.conf* with the appropriate values. Presently the file contains the following :

```
/*
  UserInterface-CG-cfg.h
  =====
  Additional configuration for a CrossGrid UI
*/

#ifndef _ROAMINGACCESSSERVERCG_CFG_H
#define _ROAMINGACCESSSERVERCG_CFG_H

/* OBJECTS UPDATE: use Computing Element rpmlist
 * -----
---- */
```

```
+updaterpms.rpmcfg          RAS-rpm

#include "cgcfg-CG-cfg.h"
#include "user-ws-CG-cfg.h"

#define USER_UID_MYSQL      100
#define USER_GID_MYSQL     101

#include LCG2_CFGDIR/mysql-cfg.h"

/* NOTE THAT UNDERSTRIKES ( '_' ) IN THE ARGUMENTS WILL BE REPLACED TO DOTS
 * IN THE CONFIG FILE!!!! Sorry for this quick hack. If it goes wrong
 * contact me. Marcus. */
cgcfg.ras      SE_Protocol SE_Host SE_Port SE_Path SE_Mode \
  LDAP_Host LDAP_DN \
  JSS_Protocol JSS_Host JSS_Port JSS_Path \
  RB_Host RB_Port \
  LB_Host LB_Port \
  FMS_Protocol FMS_Host FMS_Port FMS_Path \
  WS_FM_Protocol WS_FM_Host WS_FM_Port WS_FM_Path \
  SQL_Server SQL_ConnectionType SQL_User SQL_Passwd \
  SQL_Database SQL_NodesTable SQL_UsersTable \
  II_Host II_DN

/* CoG Config */
/*
 * if CA certificates are not in getCaCertLocations set
 * #CoG.Properties.Path to cog.properties file which define
 * cacert property
 * CoG.Properties.Path =
 */

/* SE Config */
cgcfg.value_SE_Protocol      gsiftp
cgcfg.value_SE_Host         SE_HOSTNAME
cgcfg.value_SE_Port        2881
cgcfg.value_SE_Path        CE_CLOSE_SE_MOUNTPOINT/cg
cgcfg.value_SE_Mode        passive

/* LDAP Config */
cgcfg.value_LDAP_Host      cgnode03.di.uoa.gr
cgcfg.value_LDAP_DN       dc=ras, dc=crossgrid, dc=org
cgcfg.value_LDAP_SSL      false
cgcfg.value_LDAP_User     cn=root, dc=ras, dc=crossgrid, dc=org
cgcfg.value_LDAP_Passwd   secret

/* JSS Config */
cgcfg.value_JSS_Protocol   http
cgcfg.value_JSS_Host      cgnode03.di.uoa.gr
cgcfg.value_JSS_Port      8080
cgcfg.value_JSS_Path      /ras/services/JobSubmission
/* RB Config */
cgcfg.value_RB_Host       rb01.lip.pt
cgcfg.value_RB_Port       7772

cgcfg.value_LB_Host       rb01.lip.pt
```

```
cgcfg.value_LB_Port          9000

/* FileManager Config */
cgcfg.value_FMS_Protocol     https
cgcfg.value_FMS_Host        cgnode03.di.uoa.gr
cgcfg.value_FMS_Port        8443
cgcfg.value_FMS_Path        /ras/services/FileManagerWSServicePortType

/* SQL Database settings */
cgcfg.value_SQL_Server      ras.man.poznan.pl/
cgcfg.value_SQL_ConnectionType jdbc:mysql://
cgcfg.value_SQL_User        gridadmin
cgcfg.value_SQL_Passwd      gridadmin
cgcfg.value_SQL_Database    GRID1
cgcfg.value_SQL_NodesTable  nodes
cgcfg.value_SQL_UsersTable  users
cgcfg.value_SQL_SSL         true

/*
 * These creative credentials are included from file in a local
 * Directory on the LCFG server which is not in CVS!
 * Make sure RAS_SQL_USER and RAS_SQL_PASSWD are defined at a
 * place which is not in CVS!
 */
/* cgcfg.value_SQL_User          RAS_SQL_USER
 * cgcfg.value_SQL_Passwd        RAS_SQL_PASSWD
 */

/* II config */
cgcfg.value_II_Host          ii01.lip.pt:2170
cgcfg.value_II_DN            Mds-vo-name=local,o=grid

/* VO Config */
cgcfg.value_VO                cg
cgcfg.value_HOSTLIST_FILE    /var/tomcat4/webapps/ras/WEB-
INF/hostlistfile
cgcfg.value_GRID_MAPFILE     /etc/grid-security/grid-mapfile

/* SQL/SSL Config */
cgcfg.value_SSL_keyStore     cgcfg.value_SSL_keyStorePassword
cgcfg.value_SSL_trustStore   cgcfg.value_SSL_trustStorePassword

/* RM Config */
cgcfg.value_RM_PATH          /opt/edg/bin/edg-replica-manager
cgcfg.value_RM_LD_LIBRARY_PATH /opt/edg/lib
cgcfg.value_RM_TMP           /tmp/
cgcfg.value_RM_CONF_PATH     /opt/edg/etc/edg-replica-manager/edg-
replica-manager.conf

/* Allow connections to LDAP */
EXTRA(xinetd.allow)         slapd
xinetd.allow_slapd          SITE_ALLOWED_NETWORKS

/* Start required services */
```

```
EXTRA(chkconfig.services)      httpd ldap tomcat4
chkconfig.levels_httpd        345
chkconfig.levels_ldap         345
chkconfig.levels_tomcat4      345

#ifdef SE_VO_CG
EXTRA(uicmnconfig.vo)          cg
uicmnconfig.hlr_cg             #HLRLocation = "fake HLR Location"
uicmnconfig.myproxy_cg        MyProxyServer = MY_PROXY_SERVER

uicmnconfig.nslines_cg         cg01
uicmnconfig.nslines_cg01      UI_RESBROKER:7772

uicmnconfig.lblines_cg         cg01
uicmnconfig.lblines_cg01      UI_RESBROKER:9000
#endif

#ifdef SE_VO_EXTCG
EXTRA(uicmnconfig.vo)          extcg
uicmnconfig.hlr_extcg          #HLRLocation = "fake HLR Location"
uicmnconfig.myproxy_extcg     MyProxyServer = MY_PROXY_SERVER

uicmnconfig.nslines_extcg      extcg01
uicmnconfig.nslines_extcg01    UI_RESBROKER:7772

uicmnconfig.lblines_extcg      extcg01
uicmnconfig.lblines_extcg01    UI_RESBROKER:9000
#endif

#ifdef SE_VO_TUTCG
EXTRA(uicmnconfig.vo)          tutcg
uicmnconfig.hlr_tutcg          #HLRLocation = "fake HLR Location"
uicmnconfig.myproxy_tutcg     MyProxyServer = MY_PROXY_SERVER

uicmnconfig.nslines_tutcg      tutcg01
uicmnconfig.nslines_tutcg01    UI_RESBROKER:7772

uicmnconfig.lblines_tutcg      tutcg01
uicmnconfig.lblines_tutcg01    UI_RESBROKER:9000
#endif

#endif /* _REMOTEACCESSSERVERCG_CFG_H */
```

Since RAS-MD depends on MySQL, LDAP, Apache and Tomcat, four respective users should be present in the machine. These users are controlled by LCFGng and are declared in file user-ws-CG-cfg.h:

```
/*
 user-ws-CG-cfg.h
 =====
 User definition for the webservices users
 */
```

```
#ifndef _USER_WS_CG_CFG_H
#define _USER_WS_CG_CFG_H

EXTRA(auth.groups) tomcat4
auth.groupgid_tomcat4 USER_GID_TOMCAT4

EXTRA(auth.users) tomcat4
+auth.usercomment_tomcat4 Tomcat4
+auth.userhome_tomcat4 /var/tomcat4
+auth.usergroup_tomcat4 tomcat4
+auth.useruid_tomcat4 USER_UID_TOMCAT4
+auth.usershell_tomcat4 /bin/bash

EXTRA(auth.groups) apache
auth.groupgid_apache USER_GID_APACHE

EXTRA(auth.users) apache
+auth.usercomment_apache Apache
+auth.userhome_apache /var/www
+auth.usergroup_apache apache
+auth.useruid_apache USER_UID_APACHE
+auth.usershell_apache /bin/false

EXTRA(auth.groups) ldap
auth.groupgid_ldap USER_GID_LDAP

EXTRA(auth.users) ldap
+auth.usercomment_ldap Ldap
+auth.userhome_ldap /var/lib/ldap
+auth.usergroup_ldap ldap
+auth.useruid_ldap USER_UID_LDAP
+auth.usershell_ldap /bin/false

#endif /* _USER_WS_CG_CFG_H */
```

Note that the user *mysql* is declared and control from file *LCG2_CFGDIR/mysql-cfg.h*. The above list contains directives that are defined in file *site-CG-cfg.h* as follows:

```
#ifndef USER_UID_MYSQL
#define USER_UID_MYSQL          998
#endif
#ifndef USER_GID_MYSQL
#define USER_GID_MYSQL          998
#endif

#ifndef USER_UID_APACHE
#define USER_UID_APACHE        996
#endif
#ifndef USER_GID_APACHE
#define USER_GID_APACHE        996
#endif

#ifndef USER_UID_TOMCAT4
```

```
#define USER_UID_TOMCAT4      91
#endif
#ifndef USER_GID_TOMCAT4
#define USER_GID_TOMCAT4      91
#endif

#ifndef USER_UID_LDAP
#define USER_UID_LDAP        55
#endif
#ifndef USER_GID_LDAP
#define USER_GID_LDAP        55
#endif
```

The profile of the RAS node was defined in file `cgnode03` located under `/var/obj/conf/server/source`:

```
/*
 * Template file for RAS configuration
 */

/* Host specific definitions */
#define HOSTNAME cgnode03

/* Configuration files path definitions */
#include "cfgdir-cfg.h"

/* Some useful macros */
#include LCG2_CFGDIR/macros-cfg.h"

/* Site specific definitions */
#include LOCAL_CFGDIR/site-cfg.h"

/* Linux default resources */
#include LCG2_CFGDIR/redhat73-cfg.h"

/* Linux local resources */
#include LOCAL_CFGDIR/redhat73-local-cfg.h"

/* Ethernet module to load
 * Uncomment and set appropriately if the global Ethernet module
 * defined in redhat73-cfg.h file is not suitable for the specific host
 */
/* +hardware.mod_eth0    alias eth0 e1000 */

/* Include the complete CrossGrid setup */
#include CG_CFGDIR/RAS-CG-cfg"

/* Local extra configuration for you site */
#include LOCAL_CFGDIR/local-cfg.h"
```

After all the files have been prepared we used `cg-lcfg-go` script to update the local repository with all the files needed and compile the profile for the machine. Then the machine was setup from scratch using the LCFGng boot disk. When the installation finished we proceed with the manual configuration steps:

2.1.1. Tomcat

The first manual step is to setup Apache Tomcat. Tomcat is installed under `/var/tomcat4` (Note: in the rest of the document we refer to this path as `CATALINA_HOME`). RAS is deployed as an axis service under `$CATALINA_HOME/webapps/ras`. The entry point for running MD is a web page and an applet installed under `/var/www/html/crossgrid`. Migrating Desktop is then available from `http://<hostname>/crossgrid`.

RAS services are accessed from MD via SSL. Therefore Tomcat must be configured to use SSL. To do this the administrator has to

- a. Generate a keystore
- b. Edit file `$CATALINA_HOME/conf/sever.xml` to declare the above generated keystore and activate SSL security.

To generate the keystore the following command was issued:

```
$ keytool -genkey -keyalg RSA \
-dname "C=GR, O=Crossgrid, O=UoA, ou=UoA, cn=cgnode03.di.uoa.gr" \
-alias tomcat -keystore /var/tomcat4/conf/ras_keystore.pkcs12
```

The above command generates an RSA key-pair, produces a self signed certificate and stores them in file `ras_keystore.pkcs12`. File `server.xml` was modified to contain the following directives:

```
.
.
.
<!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
  port="8443" minProcessors="5" maxProcessors="75"
  enableLookups="true"
  acceptCount="100" debug="0" scheme="https" secure="true"
  useURIVValidationHack="false" disableUploadTimeout="true">
  <Factory className="org.apache.coyote.tomcat4.CoyoteServerSocketFactory"
    clientAuth="false" protocol="TLS"
    keystoreFile="/var/tomcat4/conf/ras_keystore.pkcs12"
    keystorePass="secret" connectionTimeout="50000" />
</Connector>
.
.
.
```

2.1.2. RAS

Next the file `$CATALINA_HOME/webapps/ras/WEB-INF/RAS.conf` was edited to define the computing resources that would be used during the test and validation procedure. As mentioned this file will be populated by the appropriate LCFGng object in the future. The `RAS.conf` used for the tests is the following:

```
#####
###                                     ###
###           Server Section           ###
###                                     ###
#####
#####
##### CoG Config #####
```

```
#####  
# if CA certificates are not in getCaCertLocations set  
#CoG.Properties.Path to cog.properties file which define  
#cacert property  
#CoG.Properties.Path =  
  
#####  
##### SE Config #####  
#####  
  
SE.Protocol = gsiftp  
SE.Host      = cgnode04.di.uoa.gr  
SE.Port      = 2811  
SE.Path      = /flatfiles/SE00/cg  
SE.Mode      = passive  
  
#####  
##### LDAP Config #####  
#####  
  
#LDAP.Host = willow.crossgrid.man.poznan.pl  
#LDAP.DN   = dc=ras, dc=crossgrid, dc=org  
  
LDAP.Host = cgnode03.di.uoa.gr  
LDAP.DN   = dc=ras, dc=crossgrid, dc=org  
  
# true - connect to LDAP over SSL (port 636)  
# false - connect to LDAP without SSL (port 389)  
LDAP.SSL  = false  
  
LDAP.User = cn=root, dc=ras, dc=crossgrid, dc=org  
LDAP.Passwd = secret  
  
#LDAP.Host = zues21.cyf-kr.edu.pl  
#LDAP.DN   = dc=zeus21, dc=cyf-kr, dc=edu, dc=pl  
  
#####  
##### JSS Config #####  
#####  
  
JSS.Protocol = http  
JSS.Host      = cgnode03.di.uoa.gr  
JSS.Port      = 8080  
JSS.Path      = /ras/services/JobSubmission  
  
#####  
##### RB Config #####  
#####  
  
RB.Host = rb01.lip.pt  
RB.Port = 7772  
  
LB.Host = rb01.lip.pt  
LB.Port = 9000  
  
#####  
##### FileManager Config #####  
#####  
FMS.Protocol = https  
FMS.Host      = cgnode03.di.uoa.gr
```

```

FMS.Port      = 8443
FMS.Path      = /ras/services/FileManagerWSServicePortType

#####
#####  SQL Database settings  #####
#####
#SQL.Server   = ras.man.poznan.pl/
SQL.Server    = localhost/

SQL.ConnectionType = jdbc:mysql://
SQL.User       = gridadmin
SQL.Passwd    = gridadmin
SQL.Database   = GRID1
SQL.NodesTable = nodes
SQL.UsersTable = users
SQL.SSL       = false

#####
#####  II Config  #####
#####

II.Host = ii01.lip.pt:2170
II.DN   = Mds-vo-name=local,o=grid

#####
#####  VO Config  #####
#####

VO = cg
HOSTLIST_FILE = /var/tomcat4/webapps/ras/WEB-INF/hostlistfile
GRID_MAPFILE = /etc/grid-security/grid-mapfile

#####
#####  SQL/SSL Config  #####
#####

SSL.keyStore=
SSL.keyStorePassword=
SSL.trustStore=
SSL.trustStorePassword=

#####
#####  RM Config  #####
#####

RM.PATH = /opt/edg/bin/edg-replica-manager
RM.LD_LIBRARY_PATH = /opt/edg/lib
RM.TMP = /tmp/
RM.CONF_PATH = /opt/edg/etc/edg-replica-manager/edg-replica-manager.conf

```

Also the file *hostlistfile* located in the same directory with RAS.conf was edited. This file contains the RAS machines that serve respective VOs. In our case this file has the following contents:

```
.cg https://cnode03.di.uoa.gr:8443
```

RAS requires the presence of two general purpose LCG-2 files: The grid-mapfile and the edg-rsl-setup-cfg file. Both were copied from the CE machine. In the production phase of the system these files have to be prepared and updated automatically as it is done in the other cluster machines.

2.1.3. LDAP

The RAS-ldap package creates the basic schema and the root user entry. In order to be used by MD ldap has to be populated with the basic application and plugin information. This was performed manually using an ldif file (*dump.ldif*) that was provided by the RAS-MD programmers. The file was loaded in the database using the *slapadd* command as root:

```
$ slapadd -b "dc=ras,dc=crossgrid,dc=org" -l "./dump.ldif"
```

2.1.4. MySQL

As with LDAP, the MySQL database used by RAS-MD has to be configured manually. This is done using the *mysql* command and the following SQL script that was provided by RAS-MD programmers.

```
CREATE DATABASE IF NOT EXISTS GRID1 DEFAULT CHARACTER SET utf8;

GRANT ALL PRIVILEGES ON GRID1.* TO gridadmin@"%" IDENTIFIED BY 'gridadmin';
GRANT ALL PRIVILEGES ON GRID1.* TO gridadmin@localhost IDENTIFIED BY
'gridadmin';

SET CHARACTER SET utf8;

use GRID1;

CREATE TABLE IF NOT EXISTS users
(
    id INTEGER AUTO_INCREMENT PRIMARY KEY,
    owner_id VARCHAR(254) NOT NULL
) TYPE=InnoDB ;

CREATE TABLE IF NOT EXISTS nodes
(
    id INTEGER AUTO_INCREMENT PRIMARY KEY,
    parent_id INTEGER,
    owner_id INTEGER NOT NULL,
    pfn TEXT,
    lfn TEXT,
    name VARCHAR(254),
    type VARCHAR(254),
    size BIGINT,
    creation_date VARCHAR(254),
    modification_date VARCHAR(254),
    state VARCHAR(254),
    guid VARCHAR(254),
    INDEX owner_ind (owner_id),
    INDEX parent_id (parent_id),
    FOREIGN KEY (owner_id) REFERENCES users(id),
    FOREIGN KEY (parent_id) REFERENCES nodes(id)
) TYPE=InnoDB ;
```

The above script creates a database named *GRID1*, a user *gridadmin* and two tables owned by this user, namely *users* and *nodes*. Note that this information is used in the SQL section of the RAS.conf file.

When the above steps are completed RAS is activated by starting all the required services:

```
$ service ldap start
$ service mysql start
$ service httpd start
$ service tomcat4 start
```

During the tests RAS-MD packages were upgraded or downgraded numerous times. The setup of the above services is not influenced with the exception of the files RAS.conf and hostname which are overwritten and have to be re-edited by the administrator to bring back the old values.

2.2. ADDITIONAL TESTBED MODIFICATIONS

No additional testbed modifications were required

2.3. TEST DEVELOPMENTS

The LCFGng based installation requires the presence of an LCFGng object that handles the population of the RAS configuration files after the packages have been installed. Such an object is currently being developed by *Marcus Hardt* at FZK (marcus.hardt@iwr.fzk.de). An early version of the object was used during this Test and Validation.

2.4. USABILITY

From the installation point of view it is already evident that the process is not simple and requires quite a few manual steps and some ad hoc solutions (e.g. coping files from other machines in the cluster). This is more or less expected since the installation process has not been LCFG-ized yet. It is expected when this is completed and the procedure is fully documented the installation of a new RAS machine to be trivial and straightforward.

For the user, Migrating Desktop offers a simple and friendly environment, which he/she can learn to use and exploit in a very short time. Still the user has to have some basic knowledge of the EDG/LCG-based Grid Testbed concepts. MD assumes that the user is familiar with concepts like Resource Brokers, Storage Elements, GridFTP, job submission etc. The application comes with sufficient online help and a simple tutorial that will help a grid-aware user to quickly get start exploiting Grid resources.

The response time of distributed operations is average. This is more or less expected since MD depends on various components and remote services. The more time consuming operations are those of Job Submission, Job Status retrieval and replication operations using Virtual Directory (see next paragraph). These operations require the communication and cooperation of various remote services (e.g. MD invoking a JSS operation, which communicates with the Resource Broker, which consults the Information Index and so on). Many of the MD GUI operations (e.g. creating a new Grid Desktop) are also performed by calling time consuming operations (LDAP updates) a fact that delays response times for these actions also.

2.5. FUNCTIONALITY

Migrating Desktop provides functionality that enables two major grid activities: Job Management (Submission and Monitoring) and Data handling. It also serves as a place-holder for third-party tools developed in CrossGrid.

Job management utilizes the Job Submission Services (JSS) toolkit and in turn the Job submission and monitoring infrastructure of CrossGrid (RB, CEs, WNs etc). The user is given a set of preconfigured GUI dialogs (known as Application plug-ins) that can be used to define the submission details of

either general purpose jobs (using for instance the Command Line Application plug-in) or special purpose jobs tailored for the execution of applications developed in the context of CrossGrid (Flood, ANN HEP, Sleuth, DataMining etc).

Data handling operations utilize Datagrid facilities like GridFTP and Replica Location Services. They are performed using two similar tools called “Grid Commander” and “Grid Explorer”. They both offer a Windows Explorer like environment for accessing local and remote file systems as well as a special file system called Virtual Directory. The Virtual Directory uses remote disk space provided by Storage Elements together with Replication Services to present the user with a globally accessed limitless storage area.

Migrating Desktop provides a plug-in extensibility framework that enables the integration of third-party tools in the Graphical Environment. In this way the user utilizes MD as a single entry point for accessing Grid Capabilities. Currently five such tools have been integrated: a VNC client, an SSH client, PPC (Performance Prediction Component), GridBench and the Flood Application Plug-in.

2.5.1. Unit tests

RAS-MD cooperates with various components in order to provide the job submission and monitoring services, data transfer and replication services, and tool plug-ins, thus it makes little sense to perform extensive unit tests, isolating the system from its context.

The following operations do not require external interaction and were completed successfully:

- Creation of New Local File Shortcut
- Creation of New Local Application Shortcut
- Editing of Local File from shortcut
- Execution of Local Application from Shortcut
- On-line help invocation and browsing
- Various simple GUI activities (Windows manipulation, Shortcut rename, Shortcut deletion, Drag ‘n drop of icons between different Desktops etc)

2.5.2. System tests

Migrating Desktop was tested extensively over a period of four weeks in order to track any potential problems, evaluate the application performance and validate that the announced functionality is present. Unfortunately due to a number of critical problems we never really succeeded in setting up a stable RAS-MD installation, thus no safe assumptions can be made regarding the functionality of the system. For a complete discussion of the problems see the Issues section and the recommendations at the end of this document.

2.5.3. Stress tests

No stress tests were performed.

2.6. COMPATIBILITY

No safe assumptions can be made regarding compatibility problems of RAS-MD. As it was already described in the installation process, the software depends on a number of external packages mainly Tomcat, Apache, OpenLDAP and MySQL. During the tests and due to the severe problems that we were faced with, it wasn’t possible to identify any compatibility issues.

2.7. SECURITY AND NETWORKING

Users are identified by their CrossGrid certificate. The user certificate together with the corresponding private and public keys have to be present in the client machine from where the MD is executed. It should be noted that this is rather restrictive in the sense that the user has to copy his/her credentials in the machine before accessing MD. This may cause a potential security risk in the case that the local machine is not controlled by the user but belongs to someone else. A work-around for this would be for MD to support proxy certificates stored in a MyProxy server (see 3.1.5). Currently MyProxy server support is not supported by MD and apparently there are not plans for implementing such support in the near future.

The required proxy certificate is created every time MD is started. The password used for generating the proxy is entered by the user the first time that he/she logs in MD. It is then saved in the local client host in an encrypted form under the user's home directory (in location **\$HOME/.md**) and is used for all subsequent logins.

Credentials and data travel over the wire, between MD and RAS, in an encrypted form using SSL and https. That is the reason why during the setup of RAS-MD a host certificate is generated for Tomcat. Although we have used a self signed certificate for the test and validation of the software, we suggest that for machines that will host RAS, the certificates should be issued by the established Certification Authorities as with Computing Elements and Storage Elements. If RAS will be installed in one of the previous mentioned roles, the existing host certificates can be used.

A similar policy should be applied to signed Applets. In order to run MD the user has to accept the certificate of PSNC which has been used to sign the MD applet. Ideally these code-signing certificates should be also issued by the CrossGrid CAs using an appropriate Certificate Policy.

Also SSL enabled are all communications with LDAP and MySQL. During the tests SSL was not actually activated for the above since no installation instructions were available. Nevertheless it is obvious that having separated keys and certificates for each service is an administrative burden. For this reason it is recommended to evaluate the possibility that a single certificate (probably the machine certificate) could be used for all services.

One connection that is still not secured is the one between RAS and JSS which runs on non-secure port 8080 of Tomcat. To our knowledge this feature is currently being tested by JSS developers and it is expected to be available soon.

As far as firewall configuration is concerned, the following ports should be open in the machine hosting RAS:

- **80** (HTTP – inbound) for Apache Web Server. This is the entry point for the MD applet and the required jars.
- **8443** (HTTP – inbound) for Tomcat. This is where RAS Web Services are deployed.
- **2811** (GSIFTP - outbound) for third party transfer during file replication operations.
- All ports required for communicating with the Testbed's RB, LB and II : **7772** (Resource Broker - outbound), **9000** (Logging & Book keeping - outbound) and **2170** (Information Index - outbound).

The following ports are required to be open in the client host where MD is executed.

- **8443** (HTTP – outbound) for connecting with the RAS machine. If tunnelled mode is also selected, for communicating with SEs and file transfer, this port is also used for this reason.
- **2811** (GSIFTP - outbound) port and GLOBUS_TCP_PORT_RANGE (inbound) ports for GridFTP file transfer with unless tunnelled mode is selected.

2.8. PREVIOUSLY REPORTED ISSUES

The following issues reported in the last RAS-MD T&V reported were already resolved in the previous version of the software:

Issue 001 [bugs #425]: RAS cannot locate client GridFTP dynamic load library

Issue 002 [bugs #426]: RAS cannot locate file edg-replica-manager.conf

Issue 003 [bugs #429]: Job Submission does not work when JSS changes location

Issue 004 [bugs #424]: GridBench plug-in does not work

Issue 005 [bugs #423]: MD Freezes/Crashes when changing the Desktop name or background

Issue 006 [bugs #421]: MD Freezes when creating new DeskTop

The following issues are presumably resolved and have been removed from the bugtracker. Nevertheless we haven't been able to fully test and validate them. For the time being they will be removed from the issues list of the present report but we do not exclude the possibility to re-introduce them if required in the next validation reports.

Issue 009 [bugs #433]: An indication when MD is ready after startup should be implemented

Issue 010 [bugs #427]: Windows Menu is disabled when Desktop Window is maximized

Issue 012 [bugs #170]: Cannot cancel Job Submission operation

Issue 014 [bugs #255]: Hidden windows are not properly stored in the profile.

The following issues are still in the Bugtracker but to the extent that we were able to test them, have been resolved:

Issue 007 [bugs #434]: SEs do not respond correctly when using FTP active mode

Issue 008 [bugs #430]: MD runs out of memory

3. ISSUES FOUND

3.1. ISSUES FOUND IN THE SOFTWARE

3.1.1. Issue 001 [bugs #594]

(Severity: **Critical**, Priority: **Immediate**)

“SSLException: Unknown record version” after submitting a job

This is the main and most serious problem that appeared during the validation of the software and was the reason for its failure. The problem appeared in various occasions and the actual source of it is still under investigation by the RAS-MD developers. From the early evidence we have during the integration meeting it seems that after LDAP database is being populated with job information (either after job submissions, or when loading the history of submitted jobs using the *getuserjobs* operation of JSS). This breaks the connection between tomcat and ldap and causes the following message to appear in the client Java console every time MD tries to access LDAP:

```
AxisFault
  faultCode: {http://schemas.xmlsoap.org/soap/envelope/}Server.userException
  faultSubcode:
  faultString: javax.net.ssl.SSLException: Unsupported record version
Unknown 101.116
  faultActor:
  faultNode:
  faultDetail:
    {http://xml.apache.org/axis/}stackTrace:javax.net.ssl.SSLException:
Unsupported record version Unknown 101.116
  at com.sun.net.ssl.internal.ssl.InputRecord.a(Unknown Source)
  at com.sun.net.ssl.internal.ssl.InputRecord.read(Unknown Source)
  at com.sun.net.ssl.internal.ssl.SSLSocketImpl.a(Unknown Source)
  at com.sun.net.ssl.internal.ssl.SSLSocketImpl.a(Unknown Source)
  at com.sun.net.ssl.internal.ssl.AppInputStream.read(Unknown Source)
  at java.io.BufferedInputStream.fill(Unknown Source)
  at java.io.BufferedInputStream.read(Unknown Source)
  at
org.apache.axis.transport.http.HTTPSender.readHeadersFromSocket (HTTPSender.
java:509)
  at
org.apache.axis.transport.http.HTTPSender.invoke (HTTPSender.java:133)
  at
org.apache.axis.strategies.InvocationStrategy.visit (InvocationStrategy.java
:71)
  at org.apache.axis.SimpleChain.doVisiting (SimpleChain.java:157)
  at org.apache.axis.SimpleChain.invoke (SimpleChain.java:122)
  at org.apache.axis.client.AxisClient.invoke (AxisClient.java:183)
  at org.apache.axis.client.Call.invokeEngine (Call.java:2682)
  at org.apache.axis.client.Call.invoke (Call.java:2665)
  at org.apache.axis.client.Call.invoke (Call.java:2355)
  at org.apache.axis.client.Call.invoke (Call.java:2278)
  at org.apache.axis.client.Call.invoke (Call.java:1772)
  at
org.crossgrid.wp3.portals.roamingaccessserver.applicationinfows.Application
ManagerWSServicePortSoapBindingStub.getAllApplicationInfo (Unknown Source)
```



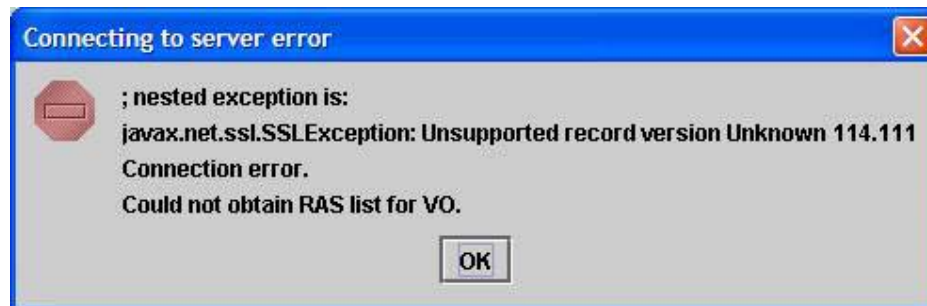
```
    at org.apache.axis.client.Call.invokeEngine(Call.java:2682)
    at org.apache.axis.client.Call.invoke(Call.java:2665)
    at org.apache.axis.client.Call.invoke(Call.java:2355)
    at org.apache.axis.client.Call.invoke(Call.java:2278)
    at org.apache.axis.client.Call.invoke(Call.java:1772)
    at
org.crossgrid.wp3.portals.roamingaccessserver.applicationinfows.Application
ManagerWSServicePortSoapBindingStub.getAllApplicationInfo(Unknown Source)
    at
org.crossgrid.wp3.portals.migratingdesktop.servicemgmt.AppInfoServiceManage
r.getAllApplicationInfo(Unknown Source)
    at
org.crossgrid.wp3.portals.migratingdesktop.jobsubmission.ApplicationChooser
Dialog.setApplications(Unknown Source)
    at
org.crossgrid.wp3.portals.migratingdesktop.jobsubmission.ApplicationChooser
Dialog.jbInit(Unknown Source)
    at
org.crossgrid.wp3.portals.migratingdesktop.jobsubmission.ApplicationChooser
Dialog.<init>(Unknown Source)
    at
org.crossgrid.wp3.portals.migratingdesktop.actions.mdactions.ShowJobWizard.
actionPerformed(Unknown Source)
    at javax.swing.AbstractButton.fireActionPerformed(Unknown Source)
    at
javax.swing.AbstractButton$ForwardActionEvents.actionPerformed(Unknown
Source)
    at javax.swing.DefaultButtonModel.fireActionPerformed(Unknown Source)
    at javax.swing.DefaultButtonModel.setPressed(Unknown Source)
    at javax.swing.plaf.basic.BasicButtonListener.mouseReleased(Unknown
Source)
    at java.awt.AWTEventMulticaster.mouseReleased(Unknown Source)
    at java.awt.Component.processMouseEvent(Unknown Source)
    at java.awt.Component.processEvent(Unknown Source)
    at java.awt.Container.processEvent(Unknown Source)
    at java.awt.Component.dispatchEventImpl(Unknown Source)
    at java.awt.Container.dispatchEventImpl(Unknown Source)
    at java.awt.Component.dispatchEvent(Unknown Source)
    at java.awt.LightweightDispatcher.retargetMouseEvent(Unknown Source)
    at java.awt.LightweightDispatcher.processMouseEvent(Unknown Source)
    at java.awt.LightweightDispatcher.dispatchEvent(Unknown Source)
    at java.awt.Container.dispatchEventImpl(Unknown Source)
    at java.awt.Window.dispatchEventImpl(Unknown Source)
    at java.awt.Component.dispatchEvent(Unknown Source)
    at java.awt.EventQueue.dispatchEvent(Unknown Source)
    at java.awt.EventQueue.dispatchEvent(Unknown Source)
    at java.awt.EventDispatchThread.pumpOneEventForHierarchy(Unknown
Source)
    at java.awt.EventDispatchThread.pumpEventsForHierarchy(Unknown
Source)
    at java.awt.EventDispatchThread.pumpEvents(Unknown Source)
    at java.awt.EventDispatchThread.pumpEvents(Unknown Source)
    at java.awt.EventDispatchThread.run(Unknown Source)
Caused by: javax.net.ssl.SSLException: Unsupported record version Unknown
101.116
    at com.sun.net.ssl.internal.ssl.InputRecord.a(Unknown Source)
    at com.sun.net.ssl.internal.ssl.InputRecord.read(Unknown Source)
```

```
at com.sun.net.ssl.internal.ssl.SSLSocketImpl.a(Unknown Source)
at com.sun.net.ssl.internal.ssl.SSLSocketImpl.a(Unknown Source)
at com.sun.net.ssl.internal.ssl.AppInputStream.read(Unknown Source)
at java.io.BufferedInputStream.fill(Unknown Source)
at java.io.BufferedInputStream.read(Unknown Source)
at
org.apache.axis.transport.http.HTTPSender.readHeadersFromSocket(HTTPSender.
java:509)
at
org.apache.axis.transport.http.HTTPSender.invoke(HTTPSender.java:133)
... 39 more
```

Note that the actual message and code number (e.g. Unkown 101.116) would vary depending on the action that triggered the exception to be thrown. For instance when the user would try to use Grid Commander a message like the one below would appear:



Tomcat remains unstable and no one can connect to RAS until tomcat is restarted. If somebody tries to connect the following error dialog appears:



During the tests we did our best to locate the source of the error and if possible fix it. Two different RAS machines were setup one in UoA and the second in FZK, both of them from scratch. Since the development RAS machine in PSNC (**willow**) has proved to be more stable we compared the RPM lists and the configuration between the original machine and the one in FZK. No major differences were located and the configuration was in most parts identical. Still the problem appeared and many times in a non-deterministic manner.

Finally in the integration meeting in the Integration Meeting at Lisbon it was tested to setup LDAP without SSL support. This prevented the SSL exception error but probably it doesn't reveal the real cause of the problem. According to RAS-MD developers the problem appears when JSS is deployed in the same machine with the rest of RAS Web Services.

3.1.2. Issue 002 [bugs #595]

(Severity: Medium, Priority: Medium)

MD does not free the memory after exiting when is run as a Java WebStart application

This is a quite important issue since if the user starts and stops many times the MD instance the client system will eventually run out of resources. This behaviour is intentional since when MD is started as an applet, killing the java process at the end brings down Internet Explorer. We suggest that this issue should be reconsidered by the Integration Team.

3.1.3. Issue 003 [bugs #596]

(Severity: High, Priority: High)

VNC client plug-in does not start

When trying to start the VNC client plug-in an error appears informing the user that the relative class cannot be loaded.

3.1.4. Issue 004 [bugs #432]

(Severity: medium Priority: low)

Remote File shortcuts pointing to GridFTP and FTP storage

Currently Remote File Shortcuts can point only to a file in the Virtual Directory. It would be useful to extend this functionality in order shortcuts can point to files accessible by GridFTP or FTP outside the Virtual Directory Context.

3.1.5. Issue 005 [bugs #488]

(Severity: medium Priority: medium)

MD should include support for MyProxy credentials.

The user certificate together with the corresponding private and public keys have to be present in the client machine from where the MD is executed. To reduce the risk of private keys being stolen or forgotten in foreign machines, Migrating Desktop should be extended to support proxy certificates stored in MyProxy servers.

3.2. ISSUES FOUND IN THE DOCUMENTATION

3.2.1. Issue 001 [bugs #428]

(Severity: medium Priority: medium)

Help for integrated Application and Tool plug-ins

It would be useful to include in the Migrating Desktop online Help, specific information and instructions regarding the available Applications and Tools that have been integrated in MD. These help material can be prepared by the Application and Tool developers themselves.

Note: Currently on-line help is provided only for the PPC plug-in.

3.2.2. Issue 002 [bugs #422]

(Severity: medium Priority: medium)

Documentation regarding editing of RAS.conf should be provided

RAS.conf contains lots of details regarding remote services that RAS depends on. A comprehensive list explaining the configuration file contents, available options and examples will help site administrators to correctly setup the service.

3.2.3. Issue 003 [bugs #597]

(Severity: High Priority: High)

Installation guide should provide detail instructions on how to SSL-enable LDAP and MySQL servers.

4. RECOMMENDATION

Considering the severity of Issue #001 and the fact that no stable deployment of the software could be achieved during the test and validation process, the T&V team has no other option than to propose the following:

- The overall recommendation is “FAIL”.
- The software CANNOT be deployed in the Production Testbed.

Moreover considering the key role of RAS-MD in CrossGrid and that the project is coming to a conclusion and there are plans for further exploitation of RAS-MD, the T&V team would like to make the following commends recommendations regarding the software:

- We suggest that RAS-MD developers should make an effort to deal with the complexity of the system. One of the first design decisions that raise scepticism is the usage of two different databases for information storage. We believe that such solution is overkill and that either an LDAP or a MySQL based approach should be adopted. Moreover considering the nature of the data stored in these databases and the amount and type of actions performed on them by MD we believe that the LDAP solution is not optimal. LDAP databases are best suited for storing hierarchical data that are frequently read and only occasionally updated. The adoption of a different solution we believe that would increase the performance, efficiency and stability of the system. The later is especially important considering the problems we were faced with during this validation for which there is strong evidence that are related with LDAP.
- Since RAS-MD will be deployed as a standalone machine using LCFGng, the developers should provide all the required configuration files (rpm lists, machine profiles, LCFGng objects) and the documentation for installing it. Towards this it is recommended that RAS-MD developers work closely with WP4 people in order to prepare the above components and provide them in the next Test and Validation request. Additionally the process should be made as automatic as possible, reducing the required manual interventions, and where such interventions are required they should be clearly documented.
- The system is build around SSL enabled services. Since SSL impairs the response of the system we suggest that the need for SSL connectivity should be re-evaluated. More specifically it should be reassessed which services have to be accessed remotely and which run internally. In the later case a good firewall configuration would suffice.
- Parallel to the previous point there should be an effort to reduce the number of certificates needed. Ideally only one host certificate should be used for all SSL enabled services.
- Finally we consider the MyProxy support quite important. Crossgrid management and RAS developers should reconsider the plans regarding the implementation this feature.

5. REFERENCES

- [1]. <http://wp3.crossgrid.org/CG3.1-D3.6-v2.1-PSNC-MDUserGuide.doc>
- [2]. http://savannah.fzk.de/cgi-bin/viewcvs.cgi/crossgrid/crossgrid/wp3/wp3_1-portals/INSTALL?rev=1.2&content-type=text/vnd.viewcvs-markup
- [3]. http://www.lip.pt/computing/projects/crossgrid/task4/softvalidation/10875626052.89278163845701/request_form.html

6. INTEGRATION/VALIDATION REQUEST

The test request can be found in:

http://www.lip.pt/computing/projects/crossgrid/task4/softvalidation/109577623283.7149291372956/request_form.html

Request id: 109577623283.7149291372956

Component name: RAS and MD

Version (CVS tag): v4_0_0

Request priority:

Package brief description:

Migrating Desktop is an advanced user-friendly environment that serves as uniform grid working environment independent on specific grid infrastructure. Java based GUI is designed especially for mobile users and is independent on platform (MS Windows, Linux, Solaris) and hardware (personal computers, laptops, workstations). It is a complex environment that integrates many tools and allows working with many grids transparently and simultaneously. The main functionality concerns local and interactive grid application support, local and grid file management, security assurance, authorisation of access to resources and applications, and single sign-on technology based on X509 certificates.

The Roaming Access Server (RAS) offers a well-defined set of web-services that can be used as an interface for accessing HPC systems and services (based on various technologies) in a common, standardized way. All communication bases on web services technology. This way we may support wide variety of client including personal computers, laptops, and in the future PDA, and mobile phones. It is one of the infrastructure features for supporting mobile users. The Roaming Access Server is a set of modules and plug-ins that provides interfaces to work with grids. It consists of several independent parts responsible for job submission, job monitoring, user profile management, data management, authorisation, and application information management.

Code:

Source code in X# CVS ? : Y

Autobuild generates RPMs ? : Y

Software download URL:

http://savannah.fzk.de/cgi-bin/viewevs.cgi/crossgrid/crossgrid/wp3/wp3_1-portals

List of RPMs produced:

cg-wp3.1-portals-MD

cg-wp3.1-RAS-services

Changes:

List of all bugs fixed by this release: -

List of backwards compatibility issues (installation, configuration or run-time): -

Documentation:

Installation manual URL: http://savannah.fzk.de/cgi-bin/viewcvs.cgi/crossgrid/crossgrid/wp3/wp3_1-portals/resources/html/crossgrid/docs/

Users manual URL: http://savannah.fzk.de/cgi-bin/viewcvs.cgi/crossgrid/crossgrid/wp3/wp3_1-portals/resources/html/crossgrid/docs/

Development manual URL: http://savannah.fzk.de/cgi-bin/viewcvs.cgi/crossgrid/crossgrid/wp3/wp3_1-portals/resources/html/crossgrid/plugins/

Software requirements URL: [D3.1](#)

Software design URL: [D3.2](#)

Files:

List of all configuration files (with full path):

`$CATALINA_HOME/webapps/ras/WEB-INF/RAS.conf`

List of all log files (with full path):

List of LCFG configuration objects (and versions):

List of daemons provided:

List of init.d scripts and supported directives (start, stop, restart, etc.):

Deployment:

Affected machine types (UI, WN, CE, SE, etc) and packages to be deployed on each:

UI

Component dependencies (required libraries, packages, etc.):

webservice
tomcat4 >= 4.1.12

Credentials (if any) used by the service:

java keystore generated from machine certificate

List of service ports (inbound,outbound):

Who communicates with the service and from where:

RAS is accessed by MD from users machine around the world

Range of temporary ports used by the service (inbound,outbound):

configured by admin
typically 80 for MD and 8443 for RAS

Testing and Validation:

Unit tests that have been performed on the package:

n/a

Features to be tested:

Mechanisms of load/save job from/to file
Extended GUI for Job Monitoring
Redesign of Job Submission Dialog,
Saving/Loading MD "on demand"
Availability of Replica Manager functions - extension of Grid Commander
functionality
VO support (RAS for one VO)
Authentication of MD plug-ins
SSL connection authentication MD/RAS
Multi-master LDAP replication using SSL
SSL connection authentication RAS/LDAP
Interactive job submission - use case with legacy application
Job format in XML
Improved icon handling mechanism (icon auto-arrange, etc)
Usage of Replica Manager API - all VD operations on files

Features not to be tested:

Test programs download URL: [n/a](#)

Other considerations:

Contacts:

Test requester:

Name:	Miroslaw Kupczyk		
WP:	3	Partner:	PSNC
Task:	1	E-mail	miron@man.poznan.pl

Developer/origin:

Name:	PSNC		
Project:	crossgrid		
WP:	3	Partner:	PSNC

Task: 1 E-mail: miron@man.poznan.pl