Die Sicherheitsstrukturen bei LCG 2

Dieses Dokument beschreibt die Mechanismen der Authentifizierung und Autorisierung unter LCG 2.6.

Version: 0.1
Datum: 26.01.2006
Autor: Jürgen Glowka FZK

Dank an Ursula Epting, Ingrid Schäffner und Ariel Garcia für ihre hilfreiche Unterstützung.

Forschungszentrum Karlsruhe - IWR

Inhaltsverzeichnis:

1 LCG 2.6 Einleitung

Die Ursprünge der LCG-Software liegen im European Data Grid Projekt (EDG 2001-2004). Entwickelt wurde es auf der Basis der Globus Toolkits (GT). Seit 2004 wird diese Middleware im Rahmen des EGEE-Projekts weiterentwickelt. Die neueste Version LCG 2.7 ist derzeit in Entwicklung und soll Anfang 2006 zum produktiven Einsatz kommen.
Die LCG -Software hat ein breites Anwendungsfeld. Im EGEE–Projekt wird sie von über 70 Partnern in mehr als 30 Ländern genutzt. Sie ist an über 150 Sites mit mehr als 20000 CPUs installiert und wird in vielen Wissenschaften verwendet: Geowissenschaft, Hochenergiephysik, Bioinformatik, Astrophysik etc. (Siehe auch http://eu-egee.org)

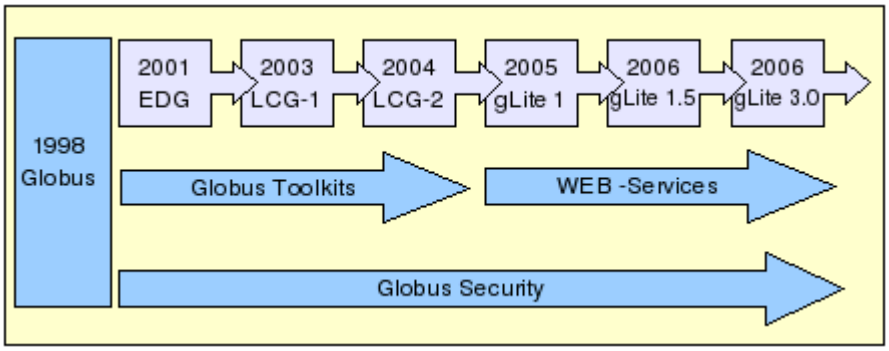


Schaubild: Entwicklungshistorie LCG - gLite

Die hier beschriebenen Komponenten und Sicherheitsmechanismen gelten auch weitgehend für die Nachfolgeversion, die nun als gLite bezeichnet wird.
Der aktuelle Entwicklungsstand bei gLite ist die Version 1.5. Noch im Jahr 2006 sollen LCG 2.7 und gLite 1.5 in der Version gLite 3.0 wieder zusammengeführt werden.
Das folgende Schaubild zeigt die Schichten der LCG-Software auf der Basis von Linux.
Die Architektur der LCG-Software besteht aus mehreren Schichten, die sich wie folgt

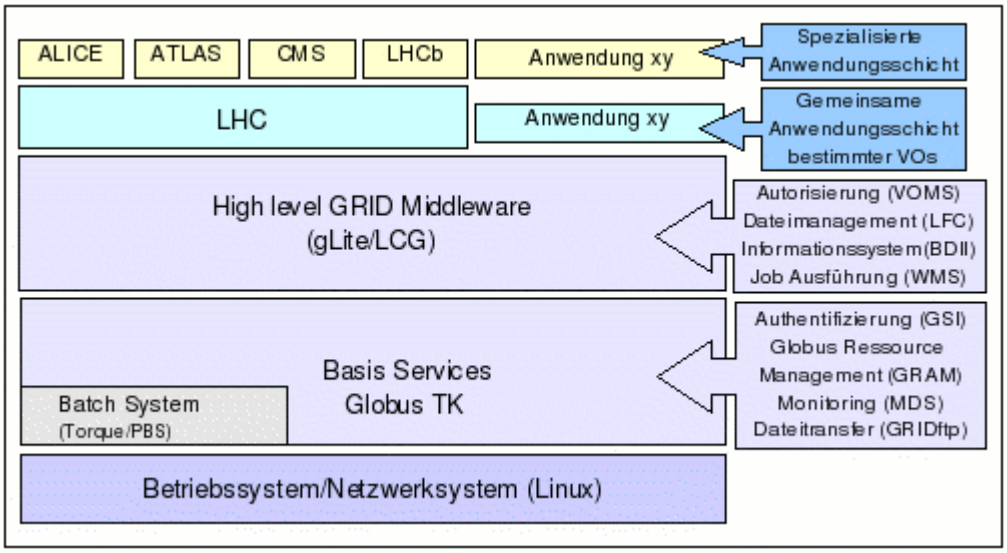


Schaubild: Schichtenmodell von LCG

darstellen lassen. In der untersten Schicht befindet sich das Betriebssystem. Hierauf folgen die Basis Services des Globus Toolkits, die unter anderem Methoden zur Authentifizierung zur Verfügung stellen, sowie ein Batch System. Darauf setzt nun die sog. High level GRID Middleware auf, mit deren Hilfe z.B. die Autorisierung durchgeführt werden kann. Die oberen Ebenen des Modells bilden nun die gemeinsame oder spezialisierte Anwendungsschicht der einzelnen Anwendungsgruppen mit ihrer experimentspezifischen Software.

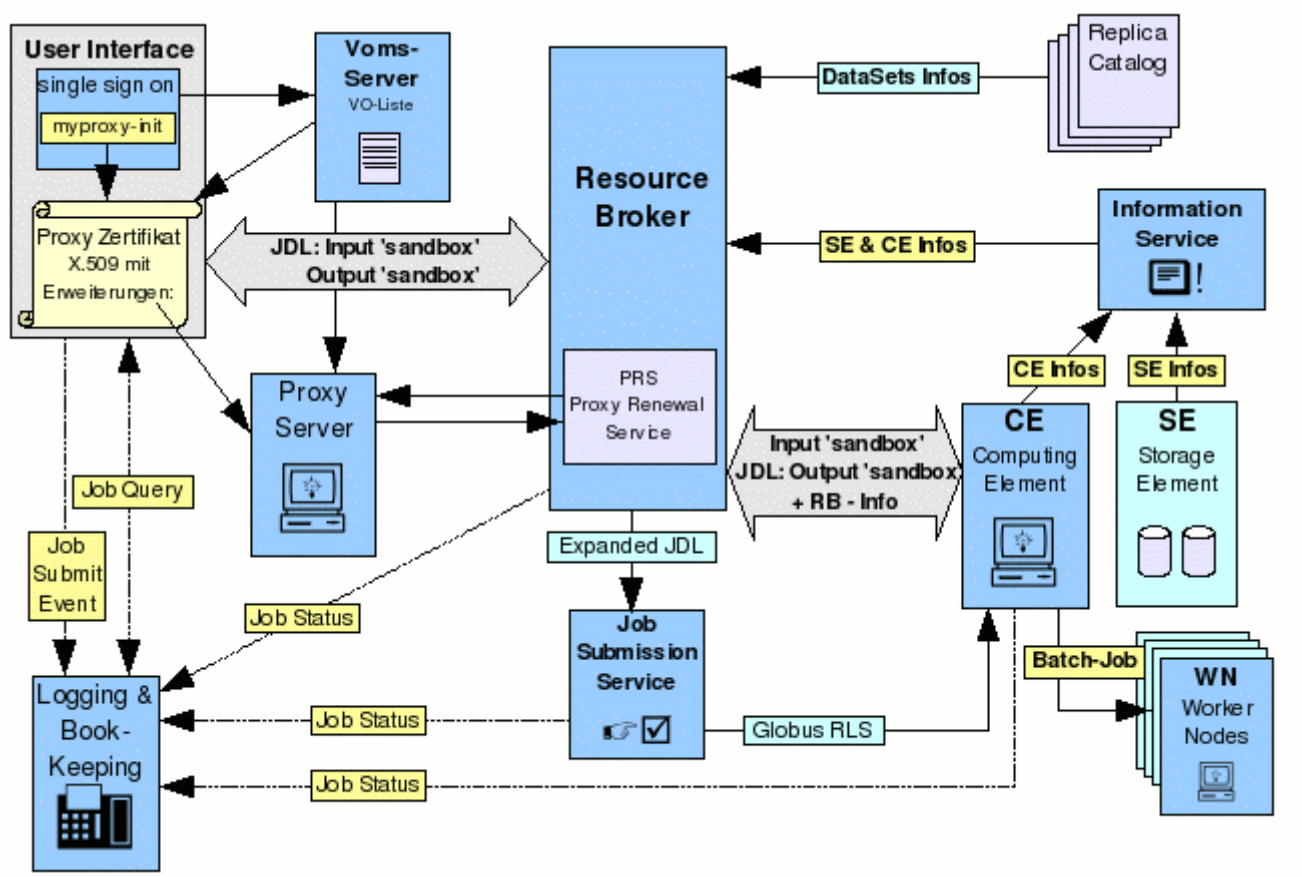


Schaubild: Übersicht LCG-2.6 Jobsubmit

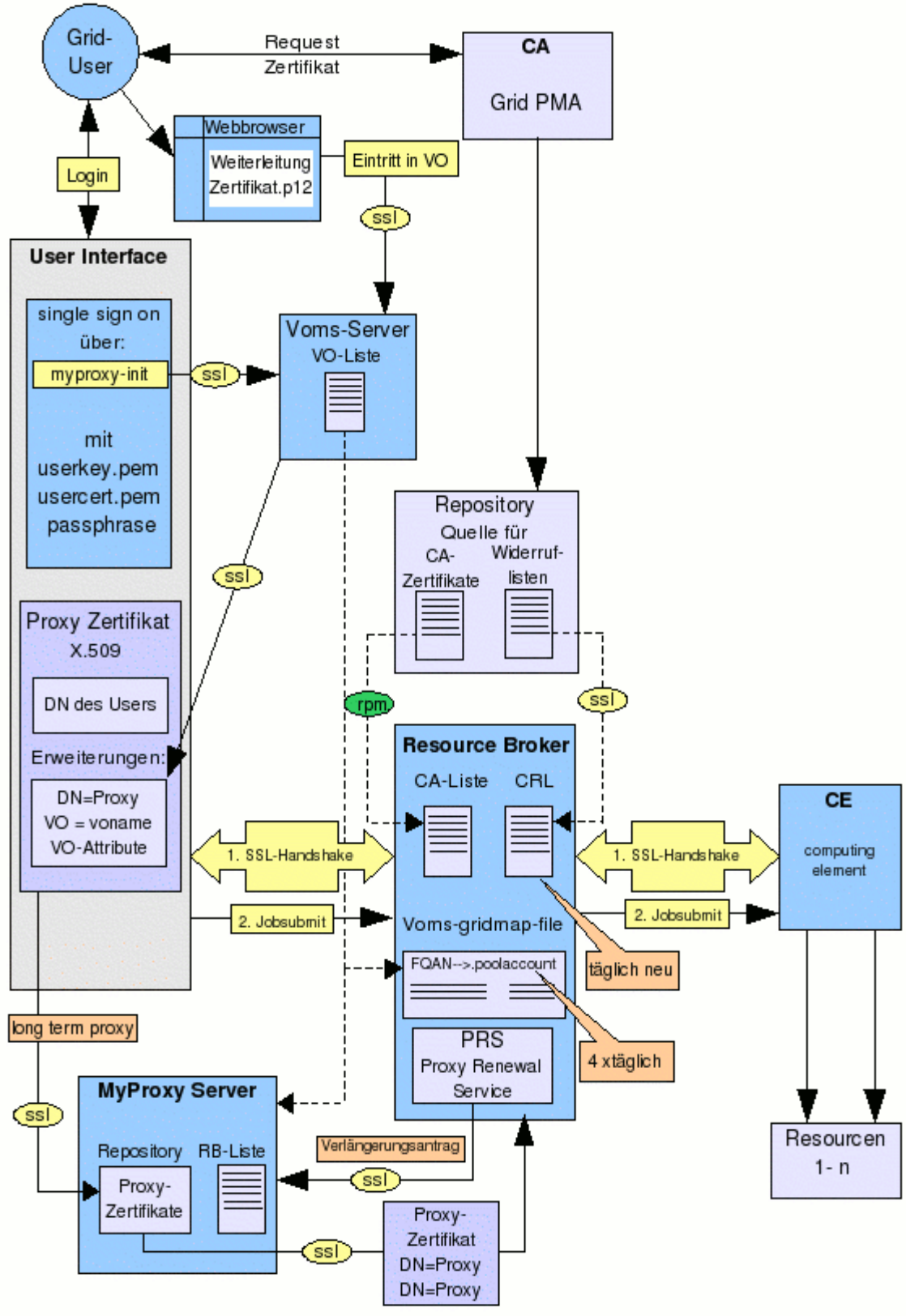
Das Schaubild zeigt eine Übersicht über den Ablauf eines Jobs im Grid unter Einbeziehung der Sicherheitsstrukturen und der Monitorfunktionen von LCG.

Vom UI sendet der Benutzer seine Jobanfrage in Form einer JDL-Datei an den Resource-Broker. Die Ausgaben seines Jobs kann er hier entgegennehmen. Der Resource-Broker ermittelt über den IS die sinnvollsten Ressourcen im Grid und gibt den Job über den Job Submission Service an ein CE weiter. Zur Überwachung und für die Abrechnung des Jobs werden weitere Dienste angestoßen. Die Informationen die sie liefern, werden auf dem LB gesammelt.

Die eigentlichen Rechenaufträge laufen auf den Worker Nodes (WN) wo sie über die Batch-Jobs vom Computing Element gesteuert werden.
Der Replica Catalog ist das Verzeichnis der Daten, die für den Benutzer zugänglich sind.

1.1 Die Mechanismen der Authentifizierung und Autorisierung in LCG

Folgendes Schaubild zeigt das Zusammenwirken der einzelnen Komponenten von LCG bezüglich der Authentifizierung und Autorisierung. Beginnend mit dem EEZ-Request des Benutzers an die CA, gefolgt von dem Eintritt in eine VO. Die Erstellung des PZs auf dem UI (für die Jobsubmission) unter Einbeziehung eines VOMS-Servers und die Ablage auf dem Myproxy-Server, sowie dem Vorgang der automatischen Verlängerung des PZs. Es zeigt auch die notwendige Vertrauensstellung zur CA sowie die Abfrage der CRLs und der Zertifikatslisten von der CA.



1.1.1 Das Verzeichnis `/etc/security` (GSI)

Das Verzeichnis ist die Kernkomponente bei der Authentifizierung des Benutzers. Es befindet sich jeweils auf dem RB, CE und SE und wird zur Überprüfung des EEZ der Benutzer herangezogen. Es entspricht der Policy von GSI und enthält für jede CA, die im Grid akzeptiert ist fünf Dateien mit gleichem Namen und unterschiedlichen Extensionen. Der Dateiname entspricht einem Hashwert, der über den Namen der jeweiligen CA gebildet wird
(`dd4b34ea.* --> '/C=DE/O=GermanGrid/CN=GridKa-CA'`)
Für den Fall, dass die CA eine hierarchische Struktur besitzt, müssen sowohl für die Root-CA als auch für alle untergeordneten CAs jeweils alle o.g. fünf Dateien hinterlegt sein.

Beispieleintrag: `/etc/grid-security/certificates/`

```
dd4b34ea.0
dd4b34ea.crl_url
dd4b34ea.info
dd4b34ea.r0
dd4b34ea.signing_policy
```

`hashwert.0`

In dieser Datei befindet sich das Zertifikat der CA, die über einen vertrauenswürdigen Pfad übermittelt bzw. verteilt wurde.
Beispiel: Zertifikat im Klartext anzeigen

```
openssl x509 -in dd4b34ea.0 -text
```

`hashwert.crl_url`

Hierin befindet sich die URL, an der die entsprechende CA ihre Revokationsliste (CRL) hinterlegt hat.

Beispieleintrag: `/etc/grid-security/certificates/dd4b34ea.crl_url`

```
http://grid.fzk.de/ca/gridka-crl.pem
```

`hashwert.info`

Diese Datei enthält generelle Informationen über die CA.

Beispieleintrag: `/etc/grid-security/certificates/dd4b34ea.info`

```
# @(#)$Id: dd4b34ea.info,v 1.3 2005/10/18 20:26:01 pmacvsdg Exp $
# Information for CA GermanGrid
#   obtained from dd4b34ea in GermanGrid/
alias = GermanGrid
url = http://grid.fzk.de/cgi-bin/welcome_ca.pl
crl_url = http://grid.fzk.de/ca/gridka-crl.pem
email = GridKa-CA@iwr.fzk.de
status = accredited:classic
version = 1.0
sha1fp.0 = 1F:E4:41:02:EC:A7:57:D8:4A:7E:A6:EE:CC:5B:A4:19:10:57:CA:17
```

hashwert.r0

In dieser Datei befindet sich die aktuelle CRL der entsprechenden CA, die mehrfach täglich über die URL aus der Datei `hashwert.crl_url` heruntergeladen wird.
Eine CRL ist maximal vier Wochen gültig. Die CRL ist verschlüsselt und von der jeweiligen CA signiert. Sie kann im Klartext mit dem openssl-Befehl eingesehen werden.

Beispiel: CRL im Klartext

```
openssl crl -in bb130558.r0 -text
```

hashwert.signing_policy

Bestimmt den Namensraum, den die jeweilige CA signieren darf.

```
# EACL GermanGrid CA operated by FZK
# @(#)$Id: dd4b34ea.signing_policy,v 1.2 2005/07/18 pmacvsdg Exp $
#
access_id_CA  X509   '/C=DE/O=GermanGrid/CN=GridKa-CA'
pos_rights    globus CA:sign
cond_subjects globus '"/C=DE/O=GermanGrid/*" "/O=GermanGrid/OU=*"'
```

Zeile 1 bezeichnet den DN der CA sowie die Art der Zertifikate, die ausgestellt werden.
Die zweite Zeile besagt, dass die CA signieren darf.
Die letzte Zeile bezeichnet den Namensraum, den die CA signieren darf. Im Beispiel hier steht der Stern (*) für alle organisatorischen Einheiten 'unterhalb' von GermanGrid.

1.1.2 Proxy und MyProxy-Server

Ein Proxy-Zertifikat ist ein Stellvertreter (Proxy) des Benutzerzertifikats, das zur Authentifizierung und Autorisierung des Benutzers im Grid verwendet wird.

Das Zertifikat des Benutzers wurde bereits mit dem privaten Schlüssel einer CA signiert, der die LCG-2-Middleware vertraut und ist durch eine Passphrase(Mantra oder Passphrase=längeres Passwort) von einer Mindestlänge von 12 Zeichen geschützt.
Zunächst wird vom Benutzer auf dem UI ein neues Schlüsselpaar erstellt, wobei der neu erstellte private Schlüssel nicht durch eine Passphrase geschützt ist.
Für den Single Sign On-Mechanismus (SSO) und die Job-Delegation werden dem Proxy-Zertifikat über den Voms-Server die non-critical extensions hinzugefügt. In den non-critical extensions ist die Zugehörigkeit zur entsprechenden VO, die Rolle (role) und Fähigkeiten (capabilites) festgehalten.

Mit dem privaten Schlüssel des Benutzers wird nun der neu erstellte öffentliche Schlüssel inklusive der Zusatzinformationen signiert und somit das temporäre Proxy-Zertifikat erstellt.

Beim einfachen Befehl (`grid-proxy-init`) werden keine Informationen vom VOMS-Server abgefragt. Der Benutzer braucht somit einen direkten Eintrag im Grid-mapfile.

```
$grid-proxy-init
$voms-proxy-init -voms xteam
```

Befehle zur Erzeugung eines Proxy-Zertifikats

Das Proxy-Zertifikats gilt von nun an als Beweis für die Identität des Benutzers, enthält auch den privaten Schlüssel und wird nur mit dem Leserecht des Benutzers ausgestattet. Es hat aus Sicherheitsgründen nur eine Gültigkeitsdauer von zwölf Stunden
(als Voreinstellung, ist jedoch veränderbar).

```
-rw------- 1 glowka users 3588 Dec 13 13:46 /tmp/x509up_u1014
```

Speicherort des Proxy-Zertifikat in der Form: `/tmp/x509up_u[user-id]`

```
            X509v3 extensions:
                1.3.6.1.4.1.8005.100.100.5:
                    0...0...0...0..\...0H.F0@.>0<1.0...U.
    GermanGrid1.0
    ..U....FZK1.0...U....Juergen Glowka.....J0H.F0D1.0...U.
    .
    GermanGrid1.0
    ..U....FZK1.0...U....host/dgrid-
    voms.fzk.de0^M..*.H..^M......._:.0"..20060117103910Z..2006
    0117223910Z0[0Y.
    +.....Edd.1K0I.". dgtest://dgrid-
    voms.fzk.de:150000#.!/dgtest/Role=NULL/Capability=NULL0,0.
    ..U.8....0...U.#..0...pjW&.....Gt.7
    }../...0^M..*.H..^M.........E..E......Y{.n........`.~...j.
    J..I.0}....P._..QP.s..VV......C........z.e(.M....Co......F
    ....]..8..
    ..M....7.....I..Lk}n.....:.'..
                X509v3 Key Usage:
                    Digital Signature, Key Encipherment, Data
    Encipherment
                1.3.6.1.4.1.8005.100.100.6:
                    03
```

Erweiterung des X.509-Zertifikats mit non-critical Extensions, ermittelt mit der Befehlszeile:
`$ openssl x509 -in /tmp/x509up_u[user-id] -text`

Die Erweiterung wird eingeleitet mit einer eindeutigen Nummer.
Die 1.3.6.1.4.1. oid steht für Enterprises und wird von iana.org vergeben.
8005 bezeichnet das Physics Department, Queen Mary, University of London und ist registriert für EDG.

Die folgenden Blöcke spiegeln den Namen, die Zugehörigkeit zu einer VO, der Untergruppe einer VO, die Rolle und die Cabability eines Benutzers wider.

Advanced Proxy Management

Dauert ein Job länger als ein Proxy-Zertifikat gültig ist, wird dieser abgebrochen. Um dies zu vermeiden, wird ein dedizierter Proxy-Server eingesetzt.
Für länger andauernde Jobs wird ein PRS (Proxy Renewal Service), der auf dem RB (Resource Broker) läuft, zusammen mit einem PS (Proxy Server) auf einem dedizierten Rechner eingesetzt.
Der Benutzer erzeugt für sich ein long-term proxy mit einer Gültigkeitsdauer von sieben Tagen. Dieses wird abgelegt im Repository des Proxy-Servers. Alle von diesem long-term proxy abgeleiteten Proxy-Zertifikate haben jeweils eine Gültigkeitsdauer von zwölf Stunden.

```
$ myproxy-init -s <proxyserver_name> -d
$ myproxy-info -s <proxyserver_name> -d
```

Befehle zur Erzeugung eines langlebigen Proxy-Zertifikats

Auf folgender Webseite kann der Benutzer nachschauen, welche Proxy-Server für seine VO verfügbar sind:

`https://goc.grid-support.ac.uk/gridsite/db/index.php`

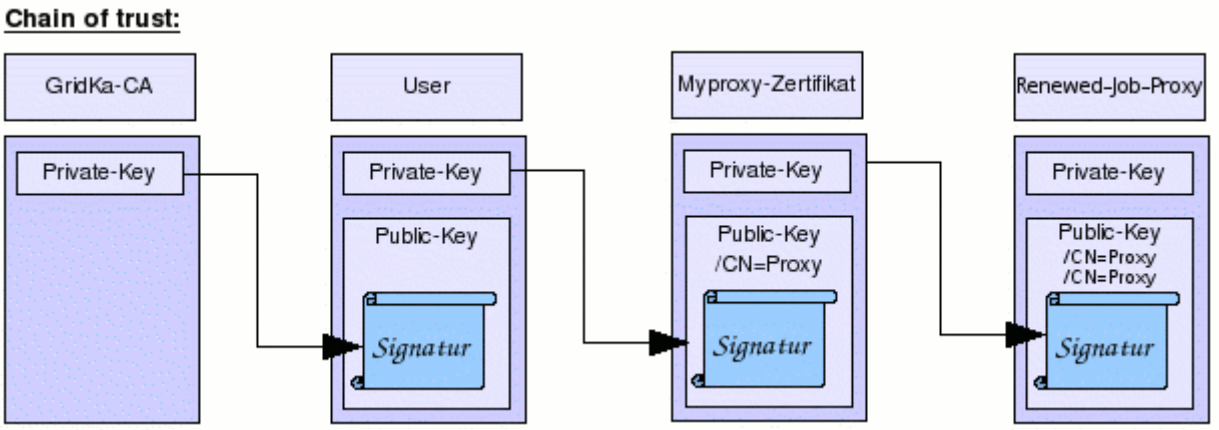


Schaubild: MyProxy renewal

Der Benutzer speichert die langlebigen User-Proxys auf dem Proxyserver. Bei Bedarf werden diese Proxys auf Antrag des PRS immer wieder erneuert. Dabei wird immer mit dem 'alten' Proxy-Zertifikat ein neues Proxy-Zertifikat erstellt. Es entsteht so eine Kette des Vertrauens. Dies geschieht so lange, bis der Job abgearbeitet ist. Die VOMS-Erweiterungen werden dabei jedesmal aktualisiert.

Das advanced proxy management wird auf dem UI in Form der myproxy-Befehlsgruppe zur Verfügung gestellt.
Damit das WMS (Workload Management System) weiß, welchen Proxyserver es für den Erneuerungsprozess verwenden muss, wird hierzu der Name des Servers in der JDL-Datei des Benutzers angegeben.

1.1.3 Gridmapfile und LCAS/LCMAPS

Autorisierung über das grid-mapfile (Globus)

Sobald die Gültigkeit eines Benutzerzertifikates überprüft wurde (Authentifizierung), wird im grid-mapfile nach dem Subject gesucht. Das grid-mapfile enthält Zeilen der Form:
`"/O=GermanGrid/OU=FZK/CN=Heinz Mann" team001`
Wenn eine Übereinstimmung gefunden wurde, wird der Job des Benutzers unter dem lokalen Benutzerkonto am Ende der Zeile (`team001`) ausgeführt. Das Benutzerkonto wird hier statisch einem Subject zugewiesen.

Der EDG Autorisierungsmechanismus

Die EDG/LCG Middleware erweitert das Globus Schema um das Konzept der Virtual Organization (VO). Das VO System stellt verteilte Informationsquellen für den Globus Mechanismus zur Verfügung, aus denen die grid-mapfiles automatisch generiert werden.

Technisch gesehen gibt es zu jeder VO ein LDAP-Server, der die Subjects (DN) der zugehörigen Benutzer enthält. Diese Informationen werden automatisch und regelmäßig mehrmals täglich abgerufen, und es wird ein grid-mapfile generiert. Zur Anwendung kommt das Perl-Skript edg-mkgridmap, das die Informationen über die abzurufenden VOs und deren LDAP-Server aus einem Konfigurationsfile bezieht (`/opt/edg/etc/edg-mkgridmap.conf`).
Da die Mitglieder einer VO im allgemeinen nicht auf allen am Grid teilnehmenden Systemen lokale Benutzerkonten haben, kann auch keine statische Zuweisung zu diesen erfolgen.
Stattdessen werden die Subjects einer VO einem Konto aus einem Pool zugewiesen, der zum Zeitpunkt der Ausführung des Jobs nur einmal vergeben wird. Die tatsächlichen Konten würden im Beispiel einer VO=FZK mit dem Pool .xteam etwa
xteam001, xteam002, usw. heissen. Das grid-mapfile hat dann Zeilen der Form:

```
"/O=GermanGrid/OU=FZK/CN=Heinz Mann" .xteam
```

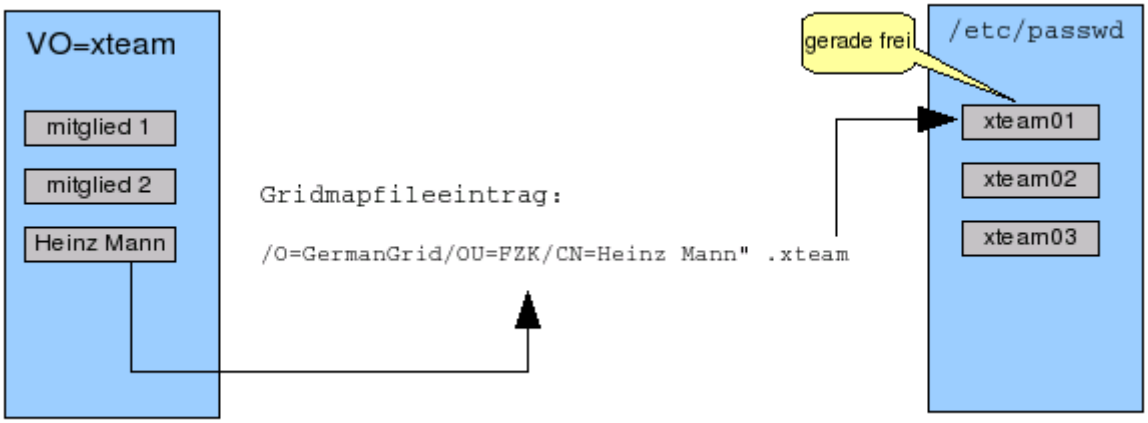


Schaubild: Mapping eines VO-Mitgliedes über das Gridmapfile auf einem lokalen Benutzer

Wird keine entsprechende Zeile gefunden, wird der Job abglehnt.
Dieses Verfahren beruht auf dem momentan auslaufenden LDAP-basierten Berechtigungssystem.

VO Management System mit VOMS-Server:

Die Benutzerautorisierung über VOs beruht auf zentralen Datenbanken (VOMS). Pro VO gibt es eine Datenbank. Die Datenbanken werden von den Komponenten RB, CE und SE genutzt, um eine lokale Liste von autorisierten Benutzern zu erzeugen.
Für die Autorisierung werden hierbei die VOMS-Erweiterungen im Proxyzertifikat herangezogen. Die lokalen Berechtigungen werden über LCAS/LCMAPS bzw. das Gridmapdir realisiert. Dieser Mechanismus soll hier genauer erläutert werden.

Erreicht eine Grid-Anfrage, wie etwa ein Job oder ein Dateitransfer, eine der Komponenten RB, CE oder SE führt das LCAS/LCMAPS-System eine Berechtigungsprüfung durch.
Bei diesem Verfahren werden über LCMAPS die Berechtigungen des Benutzers auf ein lokales Unix-Benutzerkonto dynamisch gemappt. So versucht z.B. ein entsprechendes LCMAPS-Plugin auf einem CE bei einer Job-Anfrage einen Eintrag des Benutzers über seinen FQAN im Grid-Mapfile zu finden.

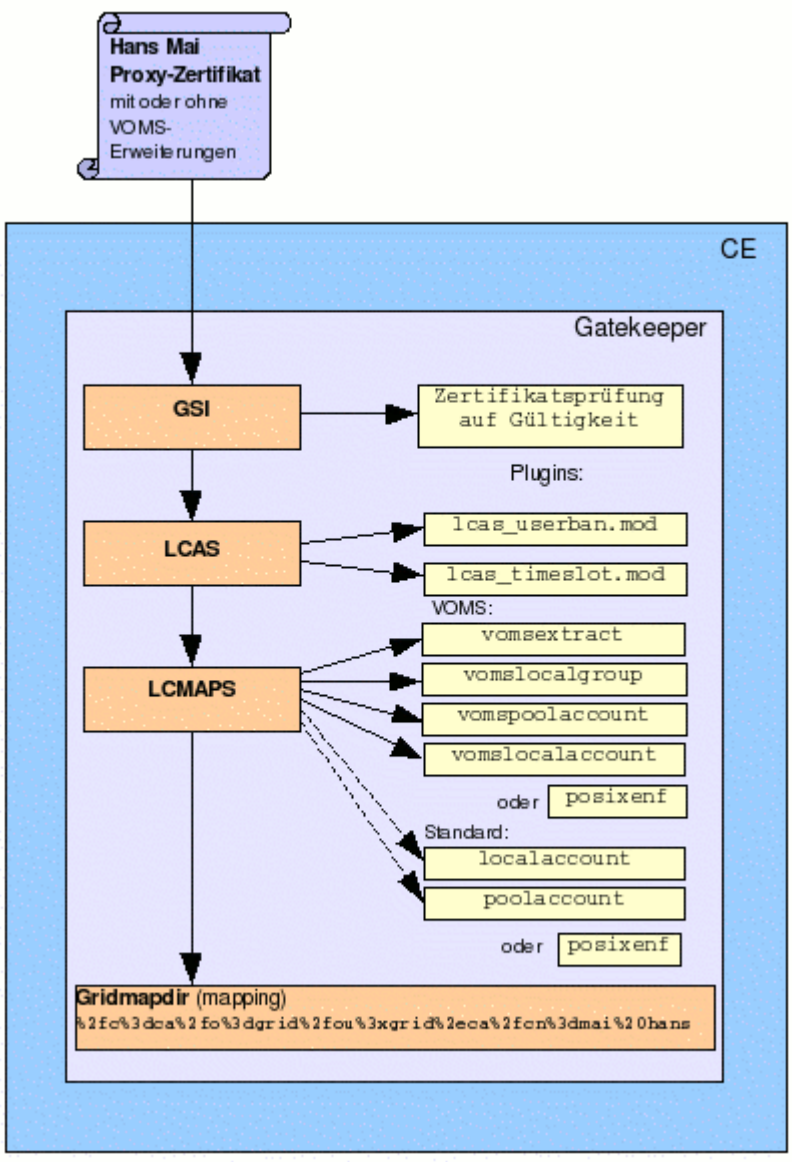


Schaubild: Authentifizierung und Autorisierung über LCAS/LCMAPS

LCAS

Bei einer Anfrage aus dem Grid steuert der Gatekeeper die Autorisierung des Benutzers. Zunächst wird nach den GSI-Richtlinien das Zertifikat des Benutzers überprüft. Nachdem die Authentifizierung des Benutzers abgeschlossen ist, startet der Gatekeeper die LCAS-Module (plugins), die in der Datei `/opt/edg/etc/lcas/lcas.db` verzeichnet sind.

Das Plugin `lcas_userban.mod` erlaubt es, Benutzer, die in der Datei `ban_users.db` mit ihrem DN-Namen eingetragen sind für das System zu sperren. Dies bietet die Möglichkeit, zusätzlich zur CRL-Liste, auf Zertifikatsmissbrauch lokal schnell zu reagieren.
Dies erscheint notwendig, da die Sperrung eines Benutzers über die CRL bis zu 12 Stunden dauern kann und nach der Sperrung das PZ dennoch weiterhin gültig ist. Zudem muss ein Benutzer, der z.B. gesperrt wird, da sein Job eine Endlosschleife erzeugt, kein neues Zertifikat bei der CA beantragen.

Durch das Plugin `lcas_timeslots.mod` können Zeitintervalle bestimmt werden, in denen der Rechner für Grid-Dienste zur Verfügung steht.
Weitere LCAS-Plugins können hier, je nach Bedarf, einfach durch Hinzufügen weiterer Dateien, wie z.B. `/opt/edg/etc/lcas/lcas_voms.in` integriert werden. Dieses Plugin vergleicht VOMS-Attribute mit einer Site-spezifischen ACL.

LCMAPS

Nun startet der Gatekeeper die LCMAPS-Module, die in der Datei `/opt/edg/etc/lcmaps/lcmaps.db` verzeichnet sind. Nach der lokalen Policy werden diese Module in einer bestimmten Reihenfolge abgearbeitet.

```
# policies
voms:
vomsextract      -> vomslocalgroup
vomslocalgroup   -> vomspoolaccount
vomspoolaccount  -> posixenf | vomslocalaccount
vomslocalaccount -> posixenf

standard:
localaccount -> posixenf | poolaccount
poolaccount  -> posixenf
```

Hat ein Benutzer ein Zertifikat mit VOMS-Erweiterungen, greift der Abschnitt `policies`.
Die einzelnen `voms*`-Module überprüfen z.B., welcher VO der Benutzer angehört, es für diese entsprechende Poolaccounts gibt, welche VOMS-Gruppen auf welche Unix-Gruppen gemappt werden sollen, usw..
Schließlich wird vom Modul `posixenf` die Unix UID/GID zugeordnet und ein entsprechender Eintrag im Gridmapdir erzeugt, falls noch keiner vorhanden ist.
Der Policy-Abschitt `standard` stellt die Kompatiblität zu älteren Standards (siehe oben, globus bzw. edg-Autorisierung) her und gilt für den Fall, dass sich der Benutzer ein Proxy-Zertifikat ohne VOMS-Erweiterung erzeugt hat. Er benötigt somit einen direkten Eintrag (z.B. von root) im Grid-mapfile.

Gridmapdir

Der Gridmapdir-Mechanismus beruht auf der Globusrichtlinie 1.1.3, welche eine dynamische Bereitstellung von lokalen Unix-Benutzerkonten ermöglicht. Diese Konten werden in gewohnter

Manier vom lokalen Unix-Administrator erstellt. Sie werden je nach Bedarf an die anfragenden Benutzer für die Dauer des Jobs verteilt. Der Mechanismus ähnelt in gewisser Weise DHCP, wo IP-Adressen dynamisch an Benutzer verteilt werden.

Kennzeichnend für das Verfahren ist der Punkt, jeweils vor dem Namen des lokalen Kontos im Gridmapfile. Ist er vorhanden, greift die gridmapdir_userid-Funktion.

Beispieleinträge: `/etc/grid-security/grid-mapfile`

Einzelner Benutzer:

```
"/O=GermanGrid/OU=FZK/CN=Heinz Mann" .xteam
```

VOMS-Benutzergruppe:

bei einem Mapping basierend auf VOMS-Attributen

```
"VO=Atlas/GROUP=/Atlas/MonteCarlo/ROLE=admin/CAPABILITY=none" .xteam
```

Neue Schreibweise:

```
"/Atlas/MonteCarlo/Role=admin/Capability=none" .xteam
```

D.h. entweder wird ein Benutzerkonto aus der Liste der freien Konten zugeordnet oder es erfolgt eine Rückgabe des Benutzernamens, der dem anfragenden Benutzer bereits zugeteilt ist.
Die Informationen über bereits vergebene oder belegte lokale Benutzerkonten werden im Verzeichnis `/etc/grid-security/gridmapdir` (auch ein anderer Pfad wäre möglich) festgehalten.

Für jedes Benutzerkonto im Pool, ob vergeben oder nicht, existiert eine leere Datei mit dem Namen des Benutzers in diesem Verzeichnis.

Ist ein Benutzerkonto vergeben, existiert auch ein hard link mit dem DN-Namen des Benutzers.

Beispieleintrag: `/etc/grid-security/gridmapdir/`

```
Inode      Dateiname

1867958
%2fc%3dca%2fo%3dgrid%2fou%3outdoorgrid%2eca%2fcn%3dmeier%20hans
1867958    atlas003
```

Diese DN-Namen (subject names) werden URL-encoded gespeichert. D.h. Buchstaben und Zahlen werden normal, andere Zeichen werden im hexadezimalen Wert mit vorangestelltem Prozentzeichen dargestellt. (= -> %3d, / -> %2f)

Bsp.: `/c=ca/o=dgrid/ou=outdoorgrid  ->`
`%2fc%3dca%2fo%3dgrid%2fou%3doutdoorgrid`

Dies ist notwendig, da der Slash unter Unix auf der Kommandozeile eine Sonderbedeutung hat.
Um herauszufinden, welches lokale Konto mit welchem DN-Namen korreliert, muss man die Inode-Einträge der Dateien vergleichen. Haben zwei Dateien die gleiche Inode, ist das lokale Konto das im Dateinamen festgehalten ist, vergeben

Beispiel: Kommandozeile zum Auslesen belegter Benutzerkonten

```
ls -i| grep ^"` ls -i| grep "^[1-9]"|cut -d" " -f1|sort -t" " -k1| uniq -d`"|sort
```

Versucht ein Prozess einen bereits vergebenen Account noch einmal zu vergeben erhält er von LCMAPS eine Fehlermeldung. (Die Bibliotheksfunktion gridmapdir_newlease bedient sich des Systemaufrufs link(2))
Für den (ungünstigsten) Fall, dass in der Zeit zwischen den Systemaufrufen stat(2) und link(2) eines Prozesses ein zweiter Prozess denselben Account zuordnen will, wird durch die Funktion gridmapdir_newlease nochmals ein stat(2) auf die Datei mit dem lokalen Benutzernamen abgesetzt, nachdem der Link gesetzt wurde. Ist nun der über stat(2) ermittelte Linkcount nun höher als 2, wird der Prozess verworfen und nach einem anderen, freien lokalen Account gesucht.

Sobald feiner differenzierte VOMS-Attribute zur Anwendung kommen, kann auch ein groupmapfile eingesetzt werden (`/etc/grid-security/groupmapfile`). Dabei wird ein FQAN (Fully Qualified Attribute Name) auf eine Unix-Gruppe gemappt.

Beispiel: FQAN(Fully Qualified Attribute Name)

```
<Gruppenname>/Role=[Rollenname][/Capability=<Capability-Name>]

        EGEE/Role=Administrator/Capability=Accounting
```

Beispiel: Groupmapping in `/etc/grid-security/groupmapfile`

```
"/EGEE/*" egee
```

Dies bedeutet, dass alle Benutzer mit dem VOMS-Attribut `/EGEE/*` auf die Unixgruppe mit dem Namen `egee` gemappt werden.

Der Mechanismus arbeitet auch auf einem verteilten, NFS-basierten Dateisystem, so dass auch hier eindeutige Zuordungen innerhalb eines Clusters möglich sind. Auch wenn verschiedene Rechner über ihren eigenen Gatekeeper, ssh-gsi oder gsi-ftpd verfügen.

Es ist so möglich mehrere Benutzer-Pools einzurichten, beispielsweise mit unterschiedlichen Rechten auf das Dateisystem.

Lösen der lokalen Benutzerkonten

Das Lösen der lokalen Benutzerkonten von den DN-Namen muss jedoch durch einen weiteren Mechanismus geschehen. Etwa durch einen Cronjob, der stündlich die zeitlich festgelegten Vergaben überprüft und für eine erneute Freigabe bereitstellt.

Kritisches:

Falls das Heimatverzeichnis bei der Lösung des Mappings gelöscht wird, verliert der Grid-Benutzer eventuell wichtige Daten. Wird es nicht gelöscht, kann der Nachfolger, der das Konto aus einem Pool erhält, die Daten des Vorgängers einsehen. Hier ist der Benutzer selbst verantwortlich, seine Daten auf ein SE zu übertragen.

Zur Lösung wird von LCG das RPM `lcg-expiregridmapdir-1.0.0-2.rpm` bereitgestellt. Es besteht im Wesentlichen aus dem Perl-Skript `/opt/edg/sbin/lcg-expiregridmapdir.pl`, das von einem gleichnamigen Cronjob gestartet wird.

Das Skript löscht Hardlinks im Verzeichnis `/etc/grid-security/gridmapdir/` die nicht mehr gebraucht werden. Kriterien dafür sind, dass das entsprechende Zertifikat, das auf einen lokalen Benutzer gemappt ist, keine laufenden Jobs mehr in Arbeit hat und dass der letzte Job vor mehr als 48 Stunden abgeschickt wurde (default-Wert).
Weiterhin werden von diesem Skript auch das Heimatverzeichnis bereinigt, temporäre Dateien gelöscht und im Hauptspeicher verbliebene Prozesse entfernt.
Je nach verwendetem Batchsystem bedient sich das Skript verschiedener Befehle, um den Jobstatus zu ermitteln. Bei dem System PBS führt das Skript den Befehl `/usr/bin/qstat` aus. Zudem wird ein Logfile `/var/log/lcg-expiregridmapdir.log` erzeugt.

RPMs die für LCAS/LCMAPS auf einem CE erforderlich sind:

```
edg-lcas_gcc3_2_2-1.1.22-1_sl3
edg-lcas_gcc3_2_2-interface-1.0.3-1_sl3
edg-lcas_gcc3_2_2-voms_plugins-1.1.22-1_sl3
edg-lcmaps_gcc3_2_2-0.0.30-1_sl3
edg-lcmaps_gcc3_2_2-basic_plugins-0.0.30-1_sl3
edg-lcmaps_gcc3_2_2-dummy_plugins-0.0.30-1_sl3
edg-lcmaps_gcc3_2_2-interface-0.0.1-1_sl3
edg-lcmaps_gcc3_2_2-voms_plugins-0.0.30-1_sl3
lcg-lcas-lcmaps-1.1.1-1
```

1.2 Komponenten von LCG

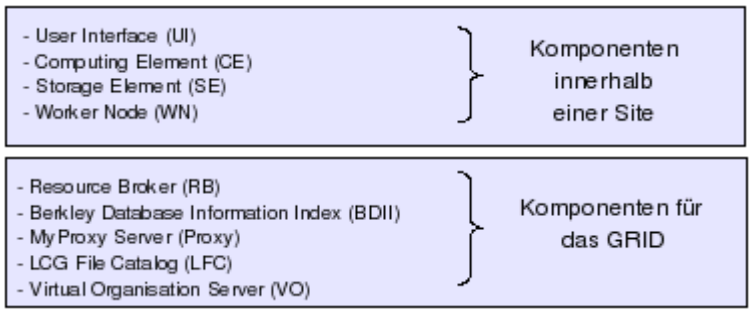


Schaubild: Komponenten von LCG

Das Schaubild zeigt eine Aufteilung der notwendigen LCG-Komponenten, die zum Aufbau einer Site erforderlich sind, sowie derer, die zum Betrieb der Site innerhalb eines Grid notwendig sind. Die Funktionen der Komponenten werden hier im Einzelnen erläutert.

1.2.1 Das User Interface (UI)

Das User Interface ist die Komponente, über die der Benutzer Zugriff auf die Ressourcen des Grid erhält. Im UI hat der Benutzer alle Befehle zur Verfügung, die er benötigt um sich über das Grid zu informieren, auf das Grid zuzugreifen, seine abgeschickten Jobs zu überwachen und die Ergebnisse seiner Berechnungen entgegen zu nehmen.

Folgende Voraussetzungen muss der Benutzer erfüllt haben:

- persönliches X.509-Zertifikat (EEZ) einer Certificate Authority (CA)
- Mitgliedschaft in einer Virtuellen Organisation (VO)

Das UI ist in der Regel der Rechner auf dem die Benutzer ein persönliches Konto haben. Die Sites stellen zwar als Service meist ein UI bereit, der Benutzer sollte sich jedoch auf dem eigenen Rechner ein UI einrichten.
Unter den Betriebssystemen Redhat, Debian oder Scientific Linux ist dies möglich.
Zu diesem Zweck hat Italian ROC eine PlugAndPlay-Schnittstelle entwickelt, die die Einrichtung automatisiert (http://grid-it.cnaf.infn.it/index.php?id=639).

Autorisierung/Authentifizierung
Zunächst meldet sich der Benutzer mit seinem Benutzernamen und seinem Passwort an seinem Rechner, der als UI konfiguriert ist, an. Nutzt er jedoch die UI auf einer Site, so wird die Übertragung mit ssh gesichert und erfordert dort ebenfalls eine Authentifizierung mit Benutzername und Passwort.
Vom UI aus führt der Benutzer eine Authentifizierung und Autorisierung in der Form eines Single Sign On (SSO) durch und kann dann die Dienste nutzen, die vom Informations-, Workload- und Data Managementsystem angeboten werden.
Dies geschieht durch die Erzeugung eines Stellvertreter-Zertifikats (Proxy-Zertifikat = PZ) mit dem entsprechenden Befehl `voms-proxy-init` bzw. `grid-proxy-init`.
Weitere Anmeldungen oder Überprüfungen des Benutzers sind fortan nicht mehr notwendig. Diese werden über das PZ im Hintergrund, ohne weiteres Zutun des Benutzers durchgeführt.

Im jeweiligen Heimatverzeichnis (`$HOME/.globus`) liegen der private Schlüssel und das Zertifikat des Benutzers (`usercert.pem` und `userkey.pem`), die gebraucht werden, um ein Proxy-Zertifikat zu erzeugen. Das Mantra des privaten Schlüssels soll mindestens 12 Zeichen umfassen.
Das Proxy-Zertifikat selbst wird unter dem Pfad `/tmp/x509up_u[user-id]` abgelegt.
Der private Schlüssel und das Proxy-Zertifikat sind also für den lokalen Administrator (root) zugänglich. Hier wird auf die lokalen Policies aufgesetzt (vertrauenswürdiger Administrator, sicherer Standort des Rechners, etc.).

CLI-tools
An der Befehlszeile des UIs (command line interface, CLI) hat der Benutzer die Möglichkeit, folgende Basisoperationen im Grid durchzuführen:

- Jobs abschicken, die auf einem Computing Element gerechnet werden
- Auflistung der passenden Ressourcen zu dem entsprechenden Job
- Dateien kopieren oder replizieren
- einen oder mehrere Jobs wieder löschen
- das Ergebnis einer Berechnung entgegennehmen
- den aktuellen Status eines abgeschickten Jobs einsehen

Folgende Befehlsgruppen können vom User Interface aus benutzt werden

Globus tools (`/opt/globus/bin/`)
z.B. `globus-job-run, globus-job-submit, globus-job-status, globus-job-get-output`

EDG tools – Job-Verwaltung (`/opt/edg/bin/`)

z.B. `edg-job-submit, edg-job-status, edg-job-get-logging-info, edg-job-get-output`

Information System queries (`/opt/lcg/bin/`)
`lcg-infosites, lcg-info, ldapsearch`

Data Management
`/opt/lcg/bin`/
`lcg-cr, lcg-aa, lcg-cp, lcg-del, lcg-la, lcg-lr etc.`

`/opt/edg/bin/`
`edg-gridftp-ls, edg-gridftp-exists, edg-gridftp-mkdir, edg-gridftp-rename` etc.

`/opt/glite/bin/`
`glite-catalog-ls, glite-catalog-mkdir, glite-catalog-mv, glite-catalog-rm` etc.

`/opt/lcg/bin/`
`dpns-ls, dpns-mkdir, dpns-rm, globus-url-copy, srmcp, srm-get-metatdata` etc.

Im folgenden Schaubild sehen Sie eine schematische Darstellung der Funktionsweise des UI.

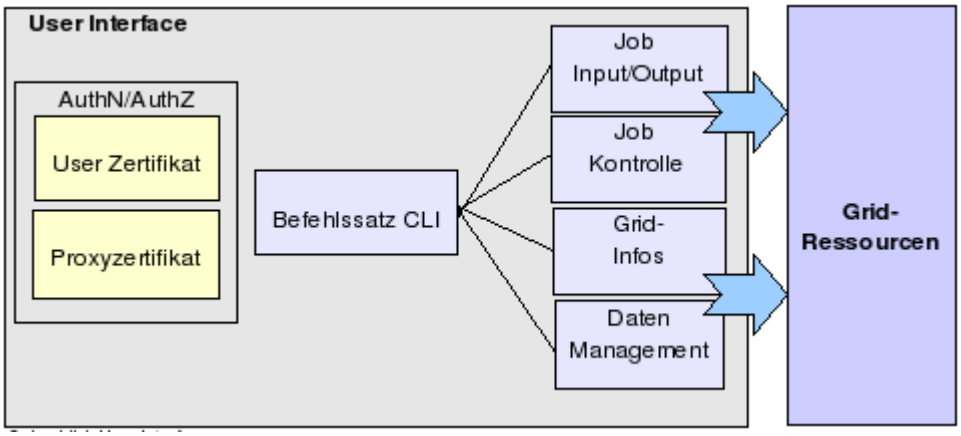


Schaubild: User Interface

1.2.2 Der Resource Broker (RB)

Der Ressource Broker (RB) ist ein etabliertes Grid Datenmanagementsystem zur Handhabung verteilter, gegebenenfalls replizierter Daten. Die Organisation der Daten erfolgt auf Datei-Basis. Der RB wurde auf der Basis des Storage Resorce Brokers (SRB) vom San Diego Supercomputer Center entwickelt und wird von vielen wissenschaftlichen Projekten eingesetzt.

Der RB hat die Aufgabe Ressourcen, die vom Benutzer benötigt werden, aufzufinden und zur Verfügung zu stellen.

Auf einem RB laufen normalerweise folgende Services:

Der Network Server (NS) bearbeitet eingehende Anfragen vom UI, er authentifiziert den Benutzer, kopiert die Input und Output Sandbox (Eingaben und Ausgaben des Benutzers) von oder zu dem

UI. Er registriert die Benutzer, die ein long term proxy benutzen wollen beim Proxy Renewal Service, der ebenfalls auf dem RB läuft.
Der RB leitet die Anfragen bzw. Jobs weiter zum Workload Manager (WM). Ist ein Job angenommen, ruft der WM den Matchmaker auf, der in Zusammenarbeit mit dem Information Service (IS) und dem Replica Location Service (RLS) die am besten zum Job passenden Ressourcen auffindet (z.B. bezüglich Hauptspeicherausbau, Prozessoranzahl, der Nähe zum SE, etc.) Danach wird der Job Controller (JC) aufgerufen um den Job an die Condor-G Komponente weiterzugeben.

Die Condor-G Komponente reicht die Jobs dann an das CE weiter. Zusätzlich generiert sie einen weiteren Job, genannt Grid Monitor, pro User und pro CE, die gebraucht werden um Jobinformationen zu sammeln und zur Verfügung zu stellen.
Ein Condor-Service (GAHP) fungiert als ein GRAM-Client (GRAM - Globus Resource Allocation Manager) für alle Jobs.
Ein GASS-Server (GASS - Global Access to Secondary Storage) nimmt die Ergebnisse der Grid Monitor Jobs entgegen.
Der Log Monitor (LM) überprüft kontinuierlich das Condor-G Logfile auf Ereignisse bezüglich der gerade aktiven Jobs. Wird ein Job, ausgeführt durch das Batchsystem, abgebrochen, informiert der LM den WM ob der Job auch auf einem anderen CE wieder aufgenommen werden kann.
Der RB erzeugt auch ein wrapper script, das auf den WNs ausgeführt wird.
Dieses enthält Befehle um die Logging-Informationen zu erzeugen, die Umgebungsvariablen passend zu setzen und benötigte Dateien zu kopieren.

Authentifizierung

Die Authentifizierung auf dem RB wird durchgeführt vom NS (Network Server) mit Funktionen aus der Globus Library. Zur Überprüfung wird das Verzeichnis `/etc/grid-security/certificates` herangezogen.

Überprüft wird:

- Ist das Benutzerzertifikat noch gültig oder abgelaufen?
- Gibt es eine Vertrauensstellung zu CA die das Benutzerzertifikat signiert hat?
- Steht das Benutzerzertifikat in der CRL?

Autorisierung

Alle Prozesse auf dem RB laufen unter dem Benutzer `edguser`.
Eine Ausnahme bildet der Dateitransfer mit GSIFTP, das die Übertragungen der Input und Output Sandbox vornimmt. Hierzu wird ein User aus den Poolaccounts benutzt.
Gridftp nimmt diese Zuordnung jedoch selbst anhand von Globus-Funktionen vor.
Es existiert auch noch kein LCAS/ LCMAPS (erst bei der Folgeversion gLite ist LCMAPS implementiert).

Problematik der RB-Architektur

Die Job-Kommunikation ist nur über den RB möglich (submit, status, get-output…).
Ist der Broker nicht erreichbar, gehen Jobs nach einer festgelegten Zeit verloren (timeout) und Proxy-Zertifikate werden ungültig.
Die Job-Submission ist statisch und somit ist kein Wechsel des CEs nach der Job-submission möglich.

1.2.3 Das Computing Element (CE)

Das CE verwaltet die angeschlossenen Worker Nodes über ein Batch-System und stellt somit die Schnittstelle zu einem Cluster dar. So verteilt das CE über das Batch-System die Jobs an die WNs. Der Ausdruck CE bezeichnet neben der Rechnerkomponente selbst auch die einzelnen Warteschlangen des Batchsystems auf dem CE.
Ein Batchsystem kann auch verschiedene Warteschlangen enthalten, z.B. für kurze oder für längere Jobs oder für verschiedene VOs.

Beispiel: Warteschlangen:

```
<hostname>:<port>/<batch_warteschlange_name>
```

```
globus-run-jgg-13.fzk.de:2119/jobmanager-lcgpbs-long
globus-run-jgg-13.fzk.de:2119/jobmanager-lcgpbs-long
```

Es werden derzeit verschiedene Batchsysteme unterstützt, wie z.B. Portable Batch System (PBS), Load Sharing Facility (LSF), Torque und Condor.

Auf dem CE läuft ein Gatekeeper, der die Benutzer überprüft, die Jobs annimmt und für jeden einzelnen Job einen job manager erzeugt. Der job manager bildet die Schnittstelle zum lokalen Batchsystem und wird gebraucht um Jobs weiterzuleiten oder zu stornieren. Der Status eines Jobs wird durch eine Monitorsystem überwacht, welches eine Instanz pro Benutzer erzeugt und die Statusabfragen durchführt und an einen BDII weiterleitet.

Authentifizierung:
Die Authentifizierung findet wie auf dem RB statt.
Zur Authentifizierung wird GSI benutzt. Der Mechanismus stützt sich im wesentlichen auf SSL/TSL und das Verzeichnis `/etc/grid-security/certificates.`
Die Authentifizierung auf dem CE wird durchgeführt vom gatekeeper mit Funktionen aus der Globus Library.

Schaubild: Watching the Grid

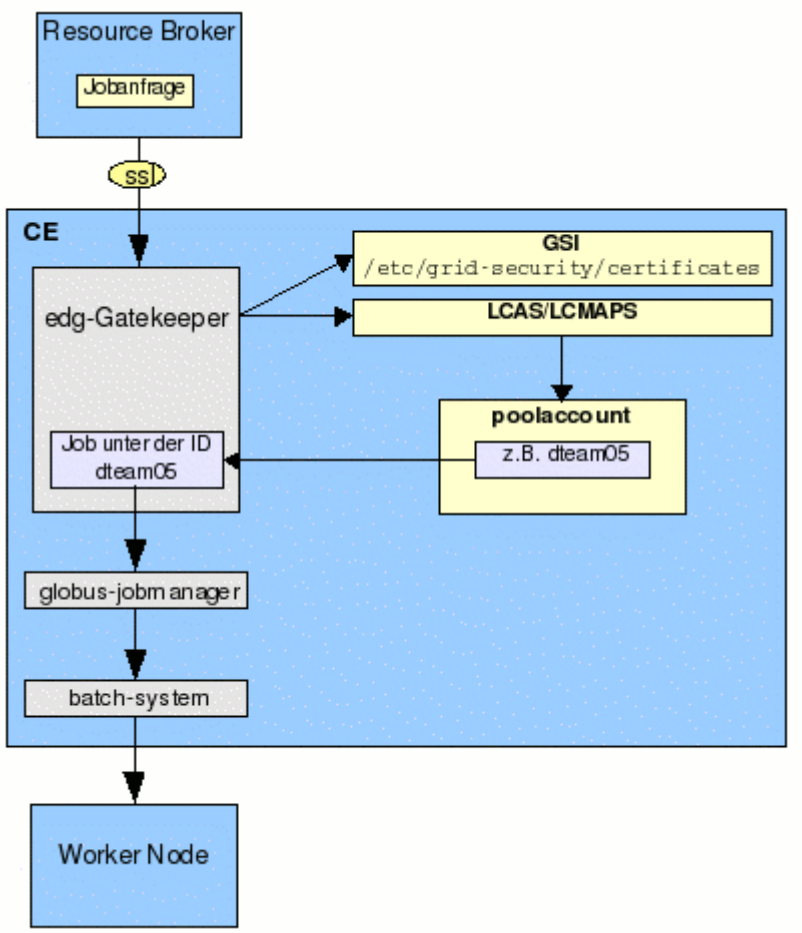


Abbildung: Authentifizierung und Autorisierung auf einem CE

Autorisierung:

Die Autorisierung geschieht auf dem CE und dem SE in gleicher Weise und wird über die Mechanismen LCAS/LCMAPS bzw. Gridmapfile durchgeführt.
Derzeit ist die Autorisierung über VOMS noch nicht überall implementiert. Das ältere LDAP-basierte Berechtigungssystem soll jedoch künftig ganz durch VOMS ersetzt werden

1.2.4 Das Storage Element (SE)

Das SE stellt einen einheitlichen Zugriff auf große (Massen-) Speichersysteme zur Verfügung und ist der zentrale Speicherplatz für Grid-Dateien.
Auf dem SE werde die Quelldaten für die Jobs abgelegt und es kann auch die Ergebnisdaten der Jobs entgegennehmen.
Das klassische SE besteht im wesentlichen nur aus einem GSIFTP-Server und einem Speichermedium.
Neuere Entwicklungen unterstützen über einen SRM (Storage Ressource Manager) den Zugriff auf Massenspeichersysteme wie DPM (Disk Pool Manager), Castor oder dCache.
Mit diesen können auch Bandlaufwerke eingebunden werden.

Die Art des Speichersystems ist jedoch für den Benutzer transparent.
Er verfügt auf der UI über die Befehle aus der Gruppe `lcg*` und `lfc*` zum Datenmanagement und kann damit einen sicheren Dateitransfer, beruhend auf Mechanismen des GridFTP durchführen.

Authentifizierung/Autorisierung

Die Authentifizierung auf dem SE wird über X.509-Zertifikate, wie beim RB und CE durchgeführt. Die Autorisierung wird ebenfalls wie beim RB und CE über die Mechanismen LCAS/LCMAPS und das Gridmapfile durchgeführt.

Künftig soll das GSIFTP durch Web-Services ersetzt werden. Für GSIFTP ist es notwendig, auf der Firewall einen großen Port-Bereich offen zu halten. Zusätzlich zum Kontroll-Port 2811 liegen die Daten-Ports im Bereich von 20000-60000 offen, was eine zusätzliche Angriffsfläche darstellt. Wogegen bei den Webservices lediglich der Port 443 zur gesicherten Übertragung offen gehalten werden muss.

Die folgende Tabelle zeigt die Datenzugriffsprotokolle, die derzeit von LCG unterstützt werden.

Protokoll	Typ	GSI-Sicherheit	Beschreibung	Optional
GSIFTP	File Transfer	Ja	FTP- ähnlich	Nein
gsidcap	File I/O	Ja	Remote file access	Ja
insecure RFIO	File I/O	Nein	Remote file access	Ja
secure RFIO (gsirfio)	File I/O	Ja	Remote file access	Ja

In der aktuellen LCG-2 Version muss jedes SE über einen GSIFTP-Server verfügen.
Es vereint die Basisfunktionalität von FTP mit der GSI-Sicherheit und dient zum effizienten Transfer von Dateien vom oder zum SE.
Zum Fernzugriff auf die Dateien, also nicht zur Übertragung, werden die Protokolle Remote File Input/Output Protocol (RFIO) und GSI dCache Access Protocol (gsidcap) benutzt.
RFIO wurde zum Zugriff auf Bandlaufwerke entwickelt, wie z.B. CASTOR (CERN Advanced STORage manager).
Die unsichere Version (insecure RFIO) unterstützt kein GSI und kann somit nur für den Datenzugriff aus dem lokalen Netzwerk und für den Zugriff der WNs auf die SEs benutzt werden. Jedoch nicht von einer UI aus. Die Authentifizierung findet nur über die Unix UID und GID statt.
Das secure RFIO hingegen ist komplett GSI-fähig, verwendet Zertifikate (PZ) und kann problemlos auch für den Datentransfer auf eine entfernte Site benutzt werden.

1.2.5 Die Worker Node (WN)

Ein WN ist der Rechner, auf dem die Jobs der Benutzer ausgeführt und die eigentlichen Berechnungen durchgeführt werden. Eine Reihe von WNs die von einem CE gesteuert werden, bezeichnet man als einen einzelnen Cluster.
Auf einem WN laufen keine LCG-Services. Eine WN benötigt auch nur ein Mindestmaß an Middleware-Software um in einem Grid betrieben zu werden, wie z.B. eine Client-API für den Zugriff des Informationssystems oder der LCG-Services.

Die Skripte oder Programme des Benutzers laufen hier nicht direkt ab, sondern sie werden vom WMS in eigene Skripte eingebettet.
Das WMS führt auch die Kopiervorgänge der input/output sandbox von und zum RB aus.

Autorisierung/Authentifizierung

Für die Ausführung eines Jobs ist bei einer WN keine lokale Zugangsberechtigung eines

Benutzers erforderlich.
Eine Authentifizierung, bzw. Autorisierung des Benutzers muss somit auf einer WN nicht durchgeführt werden. Diese wird im Vorfeld auf dem CE durchgeführt.
So befindet sich auf den einzelnen Rechenknoten (WNs) normalerweise auch weder ein Grid-mapfile noch Komponenten der GSI, da sie einfache Klienten eines Batchsystems (z.B. PBS oder torque) des CE sind.

1.2.6 LCG-File Catalog (LFC)

Mit dem LCG File Catalog werden die im Grid gespeicherten Dateien und ihre Replikas (Kopien) verwaltet und für Benutzer, sowie Ressourcen auffindbar gemacht. Er wird von der CERN IT Grid Deployment Gruppe zur Verfügung gestellt. Er ersetzt und verbessert den älteren Replica Location Service (RLS).

Der LFC stellt u.a. folgende Funktionen zur Verfügung

- einen hierarchischen Namensraum und Operationen auf diesem
- integrierte GSI- Authentifizierung und Autorisierung
- Access Control Listen (Unix Berechtigungen und POSIX ACLs)
- Aufbau von Sessions
- Checksummen für Dateien
- Unterstützung von Oracle und Mysql Datenbanken

Authentifizierung und Autorisierung:
Die sichere Version des LFC bietet sowohl Kerberos 5 als auch GSI zur Authentifizierung an.

Wird GSI verwendet, so wird die VO des Benutzers local via grid-mapfile auf ein uid/gid Paar gemappt, welches dann zur Autorisierung verwendet wird. Jeder Benutzer der VO erhält dabei die gleichen Rechte, d.h. jeder Benutzer kann die Einträge der anderen VO-Mitglieder verändern oder löschen.
Ebenfalls ist die VOMS-Unterstützung implementiert, dabei werden die VOMS-Rollen feingranlular auf verschiedene Gruppen-IDs im LFC abgebildet. Wiederum werden die UID/GID Paare zur Autorisierung verwendet mittels der Datei-Eigentümer-Rechte, die im Katalog als System Metadaten auf dem Logical File Name (LFN) gespeichert werden. Die Standard Unix-Berechtigungen und POSIX-compliant ACLs für jeden Katalog Eintrag werden unterstützt.

1.2.7 Berkeley Database Information Index (BDII)

BDII fungiert als eine Art Informations-Cache in Verbindung mit MDS2 (Monitoring and Discovery Service). Er dient der Speicherung von Daten über die Art und den Zustand von Ressourcen des Grid in sogenannten Verzeichnissen. Seine Datenbank und wird auch vom RB abgefragt, um geeignete Ressourcen für die eingehenden Jobs zu finden.
Der Site-BDII sammelt seine Informationen über Abfragen des GRIS (Grid Resource Information Service, der auf den Komponenten CE und SE läuft. Die Informationen werden in einer LDAP-Datenbank nach dem GLUE-Schema (Grid Laboratory for a Uniform Environment) hinterlegt, die wiederum vom zentralen BDII höherer Ebene eingesammelt werden.
Pro Grid existiert ein zentraler BDII, jedoch durchaus auch in mehrfacher Ausführung, d.h. mit gleicher Konfiguration.

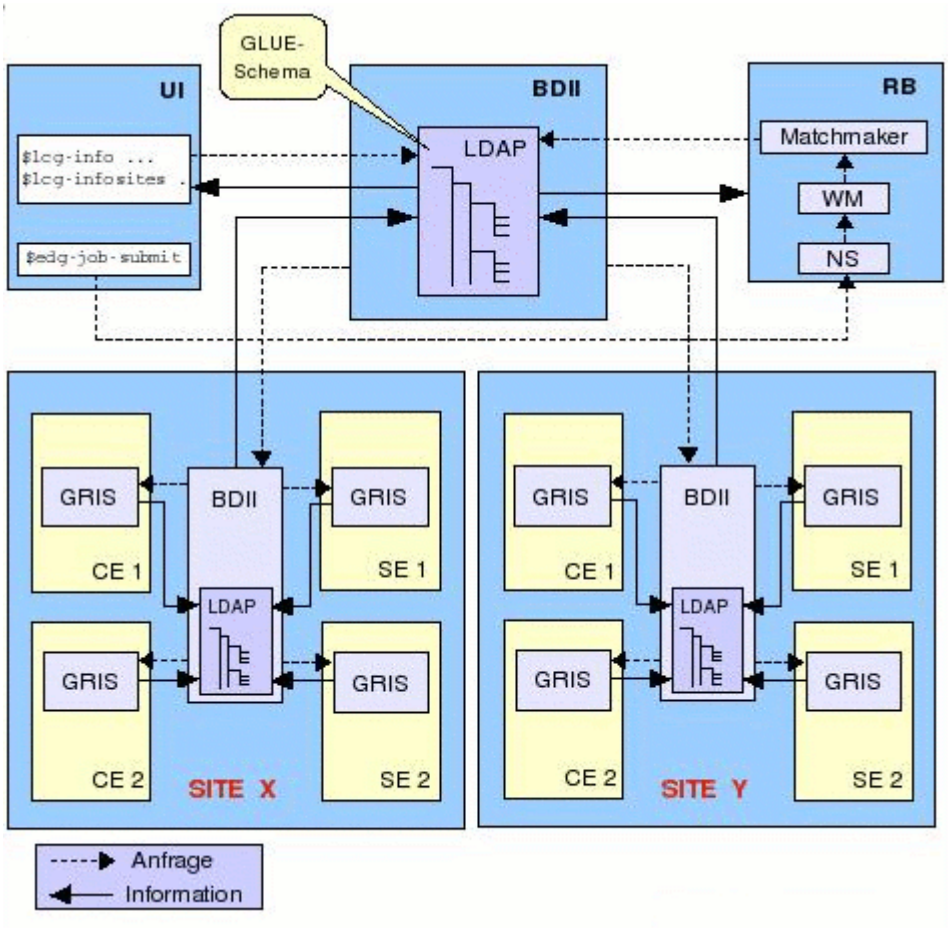


Abbildung: BDII

Über die `lcg-info*` Befehle kann der Benutzer über das UI Informationen über die Ressourcen des Grid und deren Status abfragen.
Die Befehle greifen ihrerseits auf LDAP-Abfragebefehle zurück.

Die in den LDAP-Datenbanken gespeicherten Ressourcen sind nach einer bestimmten Hierarchie, dem GLUE Schema gespeichert (Grid Laboratory for a Uniform Environment).

Authentifizierung/Autorisierung

Die LDAP-Anfragen können ohne Zertifikat, Proxy-Zertifikat oder End-Entity Zertifikat durchgeführt werden. Zu einer Abfrage muss man nur die Adresse, bzw. den URL des Hosts wissen, sowie den Port, auf dem er antwortet. Es ist auch kein Login erforderlich.
Dies ist zwar keine Sicherheitslücke, es kann jedoch für einen potentiellen Angreifer nützlich sein, Zugriff auf diese Informationen zu haben.

Künftig soll der BDII ganz durch RGMA (Relational Grid Monitoring Architecture) ersetzt werden, welches bei LCG 2.6 bereits für das Monitoring verwendet wird. RGMA, verwaltet die Statusinformationen von Ressourcen in einer relationalen Datenbank und kann mit gängigen SQL-Kommandos abgefragt werden.
Zudem sind bei RGMA bereits Sicherheitsmechanismen (X.509) Zertifikate) implementiert, so dass Unbefugte keinen Zugriff auf die Informationen über die Grid-Ressourcen erhalten können.

Quellen:

- LCG-2 User Guide Document identifier:CERN-LCG-GDEIS-454439
- The LHC Grid - Peter Kacsuk MTA SZTAKI
- http://www.gridsite.org/gridmapdir/ - Pool Accounts patch for Globus
- http://www.nikhef.nl/grid/lcaslcmaps/installation_notes/INSTALL_WITH_WORKSPACE_SERVICE
- "managing authorization in a Grid environment" http://grid-auth.infn.it/docs/voms-FGCS.pdf.
- http://edg-wp2.web.cern.ch/edg-wp2/security/voms/edg-voms-credential.pdf
- LHC COMPUTINGGRID LCG-2 MIDDLEWAREOVERVIEW CERN-LCG-GDEIS-498079
- D-Grid – AK2 Middleware und ServicesD-Grid – AK2 Anhang Bestandsaufnahme 2004

Abkürzungsverzeichnis

ACL - Access Control List
API - Application Programmers Interface
BDII - Berkeley Database Information Index
CA - Certificat Authority
CASTOR - CERN Advanced STORage manager
CLI - Command Line Interface
CN - Common Name
CRL - Certificat Revokation List
DN - Distinguished Name
DPM - Disk Pool Manager
EDG - European Data Grid
EEZ - End Entity Certificate
EGEE - Enabling Grids for E-Science in Europe
FQAN - Fully Qualified Attribute Name
GAHP - Grid ASCII Helper Protocol
GASS - Global Access to Secondary Storage
GLUE - Grid Laboratory for a Uniform Environment
GRAM - Globus Resource Allocation Manager
gsidcap - GSI dCache Access Protocol
GSIFTP - FTP-Protokoll erweitert mit GSI-Funktionalitäten
GSI - Globus Security Infrastructur
GT - Globus Toolkits
JC - Job Controller
JDL - Job Description Language
LB - Logging and Bookkeeping Service
LCAS - Local Centre Authorization System
LCG - LHC Computing Grid
LCMAPS - Local Credential Mapping Service
LDAP - Lightweight Directory Access Protocol
LFC - LCG File Catalog
LFN - Local File Name
LFN - Logical File Name
LHC - Large Hadron Collider
LM - Log Monitor
LRC - Local Replica Catalog

LRMS - Local Resource Management System
LSF - Load Sharing Facility
NFS - Network File System
NS - Network Server
OU - Organisation Unit
PBS - Portable Batch System
PRS - Proxy Renewal Service
PS - Proxy Server
PZ - Proxy-Zertifikat
RB - Ressource Broker
RFIO - Remote File Input/Output protocol
RGMA - Relational Grid Monitoring Architecture
RLS - Replica Location Service
ROC - Regional Operation Center
SE - Storage Element
SRB - Storage Resorce Brokers
SSH - Secure Shell
SSO - Single Sign On
UI - User Interface
URL - Uniform Ressource Locater
VOMS - Virtual Organization Membership Service
VO - Virtual Organisation
WMS - Workload Management System
WM - Workload Manager
WN - Worker Node