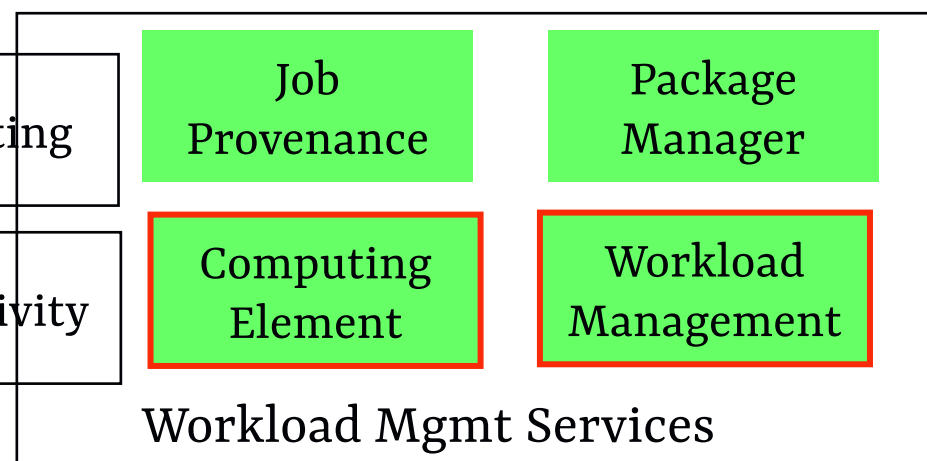
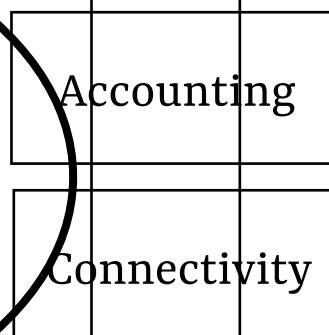
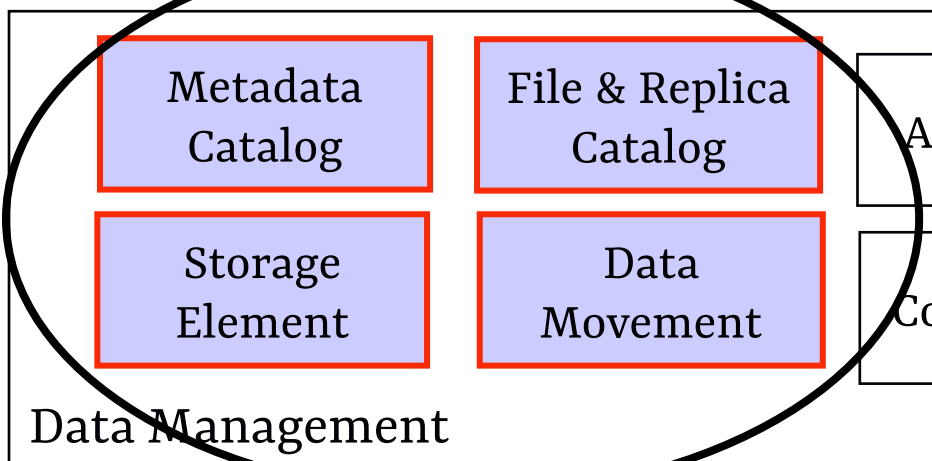
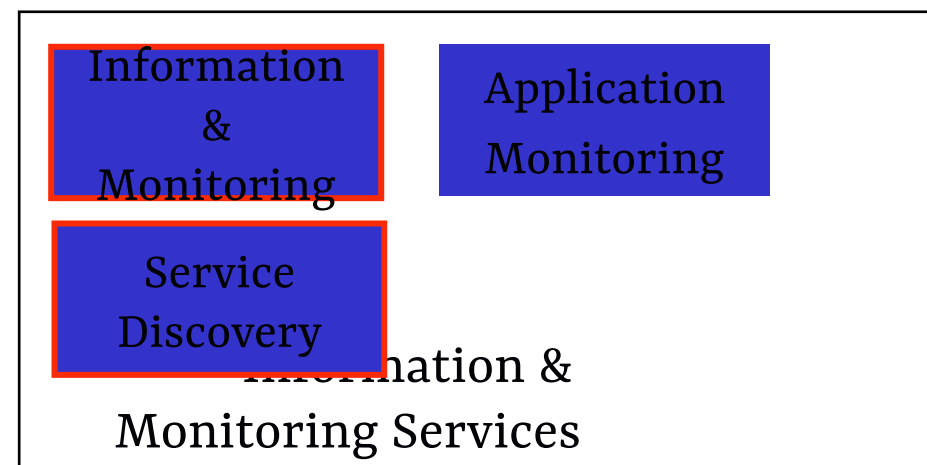
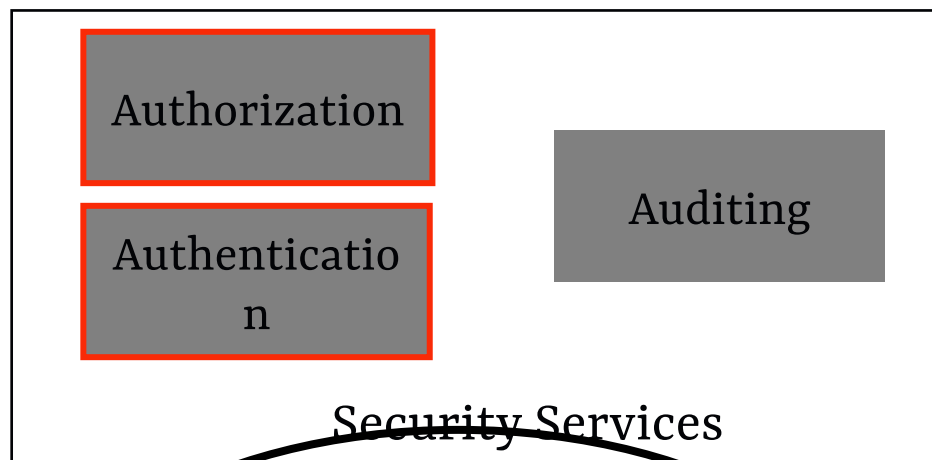



Управление данными в gLite

Белов Сергей belov@jinr.ru
Кутовский Николай

- Предпосылки:
 - пользователи и программы являются источником и потребителем данных
 - основным экземпляром данных принят файл (мы работаем с файлами, а не с объектами или реляционными таблицами)
 - данные = файлы
- Файлы:
 - в основном записываются один раз, читаются многократно
 - размещены на Элементах Хранения - Storage Elements (SEs)
 - могут существовать несколько реплик одного файла на различных сайтах
 - доступны для пользователей Грид “отовсюду”
 - местоположение м.б. определено WMS (data requirements в JDL)
- Также...
 - WMS может пересылать небольшой объём данных с заданием или от выполненного задания: Input and Output Sandbox
 - файлы могут копироваться с локальной файловой системы (WNs, UIs) в Грид (SEs), и наоборот



- **Storage Element** – общий интерфейс к ресурсам памяти
 - Storage Resource Manager** Castor, dCache, DPM, ...
 - Native Access protocols** rfiio, dcap
 - Transfer protocols** gsiftp
- **I/O Servers** – обеспечивает POSIX I/O для пользователя **gLite-I/O**
- **Catalogs** – определение местоположения файлов
 - File Catalog**
 - Replica Catalog**
 - File Authorization Service**
 - Metadata Catalog**
 - LCG File Catalog (LFC)**
 - AMGA Metadata Catalog**
- **File Transfer** – управляемая надёжная передача файлов
 - Data Scheduler** (в разработке)
 - File Transfer Service** **gLite FTS**
 - (обеспечивает физическую передачу)
 - File Placement Service** **gLite FPS**
 - (взаимодействие FTS и каталогов
способом транзакций)

- **Storage Element** – это сервис, который позволяет пользователю или приложению сохранять данные для будущего использования
- Управление локальными ресурсами памяти (диски) и интерфейс к Mass Storage Systems (ленты), таким как
 - HPSS, CASTOR, DiskeXtender (UNITREE), ...
- Способность управлять различными системами хранения данных единым способом и прозрачно для пользователя (обеспечивается через SRM интерфейс)
- Поддержка основных протоколов передачи данных
 - GridFTP обязательно
 - Другие по возможности (RFIO, dCache, DPM, https, ftp, etc...)
- Поддержка “привычного” протокола доступа для ввода/вывода удалённых файлов
 - POSIX (like) I/O client library for direct access of data (GFAL)
- Все данные на SE должны считаться данными “только для чтения”. Поэтому они не могут быть изменены, кроме как удалены или замещены.
- Различные виртуальные организации могут использовать разные политики управления квотами на пространство SE.

- Протоколы передачи и доступа к данным, поддерживаемые в gLite, приведены в таблице ниже.

Протокол	Тип	Безопасность GSI	Описание	Обязательность
GSIFTP	Передача файлов	Да	подобно FTP	Да
Gsidcap	ввод/вывод файла	Да	удаленный доступ к файлу	Нет
Insecure RFIO	ввод/вывод файла	Нет	удаленный доступ к файлу	Нет
Secure RFIO	ввод/вывод файла	Да	удаленный доступ к файлу	Нет

- **GSIFTP** предоставляет функциональность FTP протокола, но с поддержкой Grid Security Infrastructure (GSI). Этот протокол ответственен за быструю, безопасную и эффективную передачу файлов с/на SE. Он дает возможность управлять передачей файлов между двумя удаленными по отношению клиенту элементами хранения данных (**third-party transfers**), также как и передавать данные в несколько параллельных потоков. Каждый SE имеет, как минимум, один GridFTP сервер (т.е. сервер с поддержкой протокола gsiftp).
- Каждый SE в инфраструктуре EGEE имеет как минимум один GridFTP-сервер, т.е. сервер, который поддерживает протокол gsiftp.

- В настоящее время в gLite для прямого доступа к файлам на удаленном SE используются такие протоколы, как **Remote File Input/Output (RFIO)** и **GSI dCache Access Protocol (gsidcap)**. RFIO был разработан для работы с ленточными системами хранения данных (например, CERN Advanced STORage manager, **CASTOR**) и имеет реализации как с поддержкой безопасности (**secure RFIO**), так и без нее (**insecure RFIO**). gsidcap – это оригинальный протокол dCache (**dcap**) с поддержкой безопасности GSI. **dCache** – дисковая система хранения данных.
- **File**-протокол использовался в прошлом для локального доступа к файлам на сетевых файловых системах. В настоящий момент он больше не поддерживается и используется только для указания файла на локальной машине (например, на UI или WN), но не для записи на SE.

- **Classic SE:** состоит из GridFTP сервера и обслуживающего один диск или дисковый массив демона с протоколом insecure RFIO. Скоро этот тип SE перестанет поддерживаться.
- **CASTOR:** система массовой памяти с кэширующими дисками и ленточным носителем. Виртуальная файловая система (пространство имен, *namespace*) избавляет пользователя от всей сложности организации дисков и лент, лежащей в основе таких систем. Миграция файла между диском и лентой происходит под управлением процесса, называемого *stager*. Доступ к данным осуществляется по протоколу insecure RFIO. Т.к. этот протокол является незащищенным, то доступ по нему разрешается только из той же сети, где находится сам SE. С определенными модификациями дисковый буфер CASTOR может использоваться в качестве дисковой системы хранения данных.
- **dCache:** состоит из сервера и одного или нескольких пулов узлов. Сервер - это единственная точка доступа ко всему SE и формирует из файлов на дисковых пулах единое дерево виртуальной файловой системы. Узлы с дисками можно динамически добавлять в пул. Доступ к данным по протоколу *gsidcap* организовыван по типу POSIX. dCache широко используется как в чисто дисковых системах, так и в качестве дискового буфера ко многим ленточным системам хранения данных.

- **LCG Disk pool manager:** является менеджером дисковых пулов (**DPM**), подходящим для относительно малых грид-сайтов (до 10 Тб общего дискового пространства). Диски можно динамически добавлять в пул в любое время. Как в dCache и CASTOR, виртуальная файловая система скрывает от пользователя всю сложность архитектуры дисковых пулов. Защищенный протокол RFIО предоставляет доступ к файлам отовсюду.

- Менеджер ресурсов хранения данных (**Storage Resource Manager, SRM**) является реализацией единого интерфейса (через соответствующий SRM-протокол) для операций с дисковыми и ленточными ресурсами хранения данных.
- Любой тип SE, за исключением морально устаревающего Classic SE, имеет собственный SRM-интерфейс.
- SRM скрывает от пользователя всю сложность организации ресурсов хранения данных и предоставляет ему единый набор команд для операций с данными, практически независимыми от типа SE.
- Последние версии SRM (v2.2) предоставляет возможность запрашивать данные, хранить их на дисковом буфере определенное время, резервировать место для ожидаемых данных и т.д.
- Также с помощью SRM пользователь может передавать данные между двумя удаленными по отношению к клиенту элементами хранения данных
- Важно упомянуть, что SRM - это протокол управления данными на/между SE, а не протокол передачи самих данных. Например, задача не может считывать данные из файла в режиме реального времени с SE по SRM-протоколу.

Она запускает задачу, которой нужны:

- данные реконструкции физического события
- данные симуляции
- некоторые файлы с данными анализа

Результаты также должны быть где-то сохранены

В JINR
на dCache

В PNPI
на classic SE

В Fermilab
на дисковом массиве



dCache

Собственная система, свой протокол и параметры

gLite DPM

Система, независимая ни от dCache ни от Castor

Castor

Нет связи с dCache или classic SE

SRM

Я общаюсь с ними от
вашего имени

Я буду выделять место для
ваших файлов

И я буду использовать
протоколы передачи
данных, чтобы пересылать
ваши файлы туда

- Данные хранятся на **disk pool servers** или **Mass Storage Systems**
- Управление этими ресурсами должно обеспечивать:
 - Прозрачный доступ к файлам (migration to/from disk pool)**
 - Выделение места для файлов (Space reservation)**
 - Получение информации о статусе файлов (File status notification)**
 - Управление временем жизни файлов (Life time management)**
- **SRM (Storage Resource Manager)** сервис реализует все эти требования:
 - SRM это Грид сервис, который реализует взаимодействие с локальными ресурсами хранения данных и обеспечивает Грид-интерфейс для внешнего мира**
 - SRM – это протокол управления ресурсами хранения данных, а не протокол доступа к файлам или протокол передачи файлов.**
- SRM разработан, чтобы служить единым интерфейсом для управления дисковыми (или ленточными) ресурсами.
- В gLite взаимодействие с SRM обычно скрыто за сервисами более высокого уровня (DM tools и APIs)

Протоколы доступа к файлам в gLite SE 3.1:

Протокол	Тип	GSI	Описание
GSIFTP(GridFTP)	Передача файлов	Да	Аналог FTP
gsidcap (GSI dCache Access Protocol)	Ввод/вывод	Да	Удалённый доступ
insecure RFIO (Remote File Input/Output Protocol)	Ввод/вывод	Нет	Удалённый доступ
secured RFIO (gsirfio)	Ввод/вывод	Да	Удалённый доступ

* Протокол **file** сейчас используется только для доступа к файлам на локальном компьютере (т.е. на UI или WN), но не к файлам на Грид SE

** GridFTP сейчас является обязательным для каждого из типов SE, поддерживаемых в gLite и основным для передачи файлов в Грид.

- Grid Unique ID (GUID)** однозначно определяет файл в Грид:
`guid:<36_bytes_unique_string>`
 Например:
`guid:38ed3f60-c402-11d7-a6b0-f53ee5a37e1d`
- Logical File Name (LFN)** – логическое имя файла, которое можно использовать для ссылки на файл вместо GUID (и которое, как правило, является более распространенным способом ссылки на файл). Оно имеет следующий формат:
`lfn:<any_string>`
 Например:
`lfn:importantResults/Test1240.dat`
 В файловом каталоге (**LCG File Catalog, LFC**) логические имена файлов организованы в иерархическую структуру, на подобие дерева папок на диске. В этом случае формат будет такой:
`lfn:/grid/<MyVO>/<MyDirs>/<MyFile>`

- **Storage URL (SURL)**, также известный как **Physical File Name (PFN)**, определяет реплику (копию) файла на элементе хранения данных. Общий формат:

`<sfn|srm>://<SE_hostname>/<some_string>`

где префикс *sfn* используется для файлов, которые находятся на SE, не имеющем SRM-интерфейса, а префикс *srm* – для имеющих таковой.

В случае *sfn*-префикса, строка, идущая за именем машины, указывает на местонахождение файла и может быть разделена на 1) путь к области хранения данных на самом SE (т.н. точка доступа), 2) относительный путь к виртуальной организации владельца файла и 3) относительный путь к файлу.

`sfn://<SE_hostname><SE_Accesspoint><VO_path><filename>`

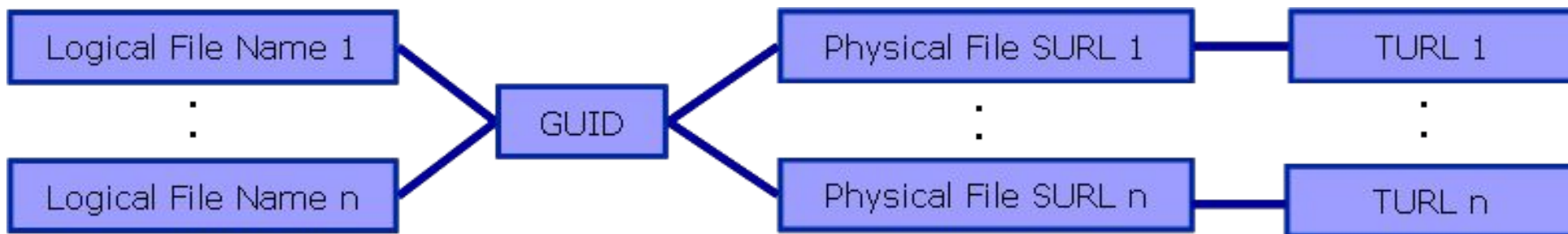
Например:

`sfn://tbed0101.cern.ch/data/dteam/doe/file1`

Для SE, имеющих SRM-интерфейс, SURL не будет иметь какой-то конкретный формат, отличный от *srm*-префикса и имени машины. В общем случае, такие элементы хранения данных могут использовать виртуальную файловую систему и имя, которое получает файл, может не иметь ничего общего с его физическим местоположением на SE, которое тоже может меняться со временем. Например:

`srm://srm.cern.ch/castor/cern.ch/grid/dteam/doe/file1`

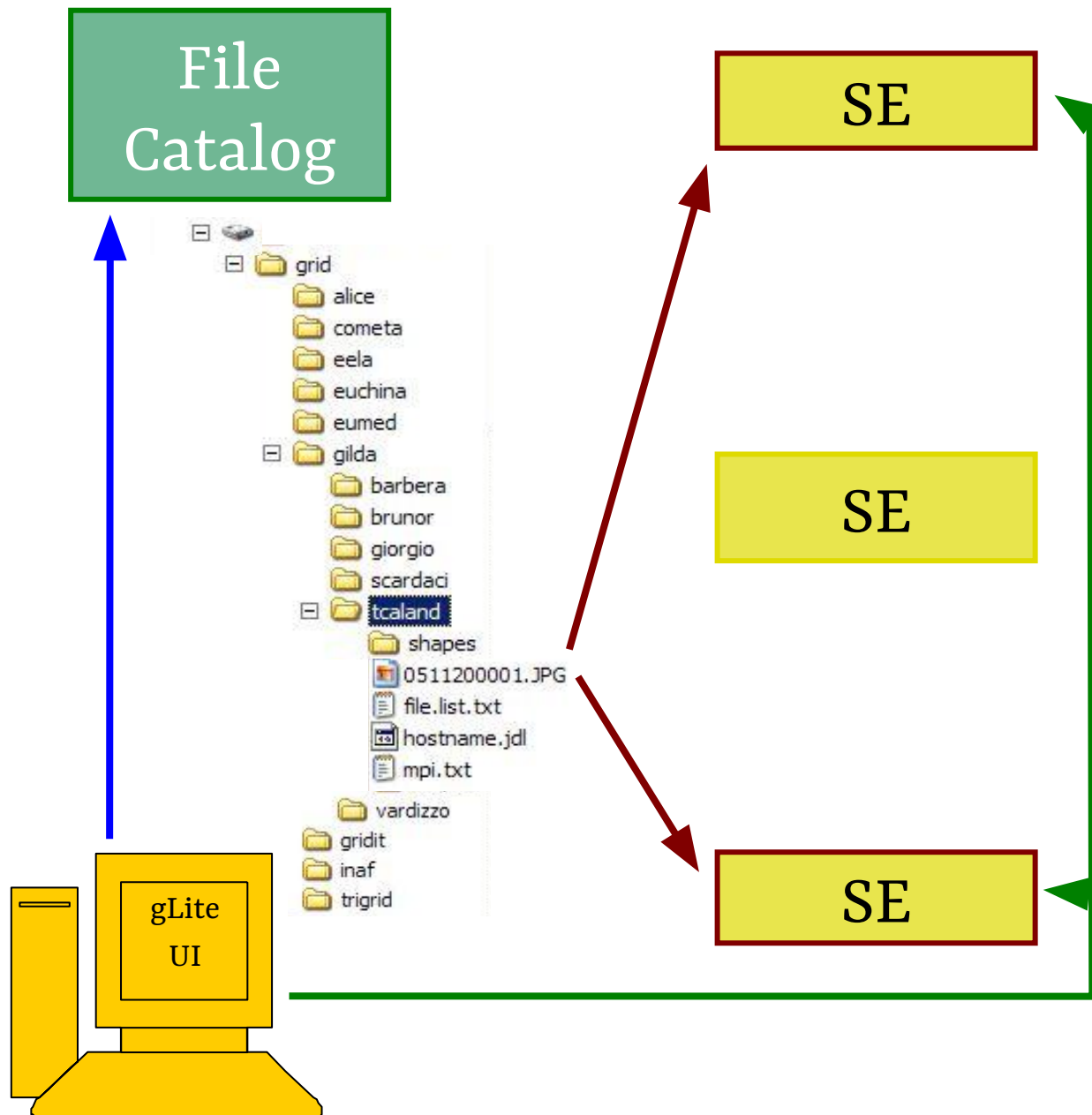
- Transport URL (TURL)** – действительный единообразный идентификатор ресурса (Uniform Resource Identifier, URI) с необходимой информацией для доступа к файлу на SE. Формат:
`<protocol>://<some_string>`
 Например:
`gsiftp://tbed0101.cern.ch/data/dteam/doe/file1`
 где `<protocol>` - действительный протокол, поддерживаемый SE, для доступа к содержимому файла (GSIFTP, RFIO, gsidcap); строка после двойной косой черты может иметь любой формат, распознаваемый элементом хранения данных, обсуживающим данный файл.
- В то время, как SURL, в принципе, является неизменными (это поля в каталоге), TURL динамически получается из SURL через информационную систему или SRM-интерфейс (для тех SE, у которых он есть). TURL может меняться с течением времени и должен считаться действительным только относительно небольшой промежуток времени с момента его получения.

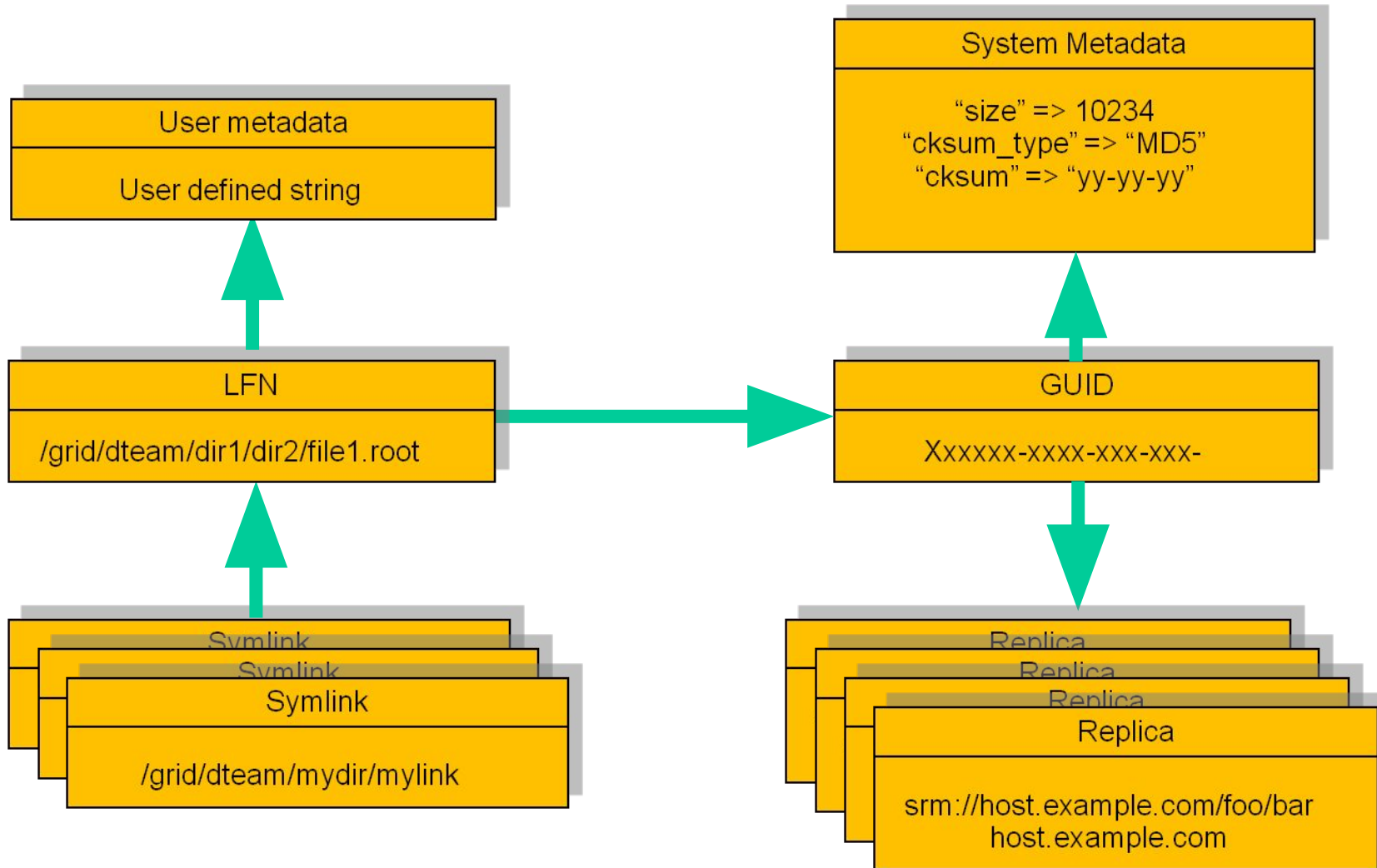


- Пользователям и приложениям необходимо иметь возможность находить файлы в Грид. Файловый каталог (FC) – это сервис, который хранит соответствие между LFN, GUID и SURL. **LCG File Catalog (LFC)** – это реализация FC в gLite 3.
- Каталог публикует свой определитель местонахождения (синонимы: единообразный локатор, Uniform Resource Locator, URL) в информационной системе таким образом, что сервис может быть найден инструментами управления данными и другими сервисами (например, WMS).
- LFC может использоваться как локальный файловый каталог, хранящий только реплики определенного набора сайтов, так и в качестве Глобального файлового каталога, содержащего информацию о всех файлах в Грид. Последний может иметь множество своих копий “только для чтения”, которые содержат идентичную информацию и могут быть разнесены по основным компьютерным центрам, тем самым обеспечивая децентрализацию и разгрузку Глобального файлового каталога.
- LFC предоставляет такие функциональные возможности, как транзакции, сессии, обработку большого количества запросов как одной операции, иерархическое пространство имен для LFN.

- **Примечание:** файл считается грид-файлом, если он физически присутствует на каком-либо элементе хранения данных И зарегистрирован в файловом каталоге. Существуют определенные *высокоуровневые инструменты*, которые проверяют согласованность между файлами на SE и соответствующими записями в каталоге. Однако использование *низкоуровневых инструментов* управления данными может привести к рассогласованию между физическими именами и записями в каталоге, что искажает информацию в Грид. Поэтому использование низкоуровневых инструментов настоятельно НЕ РЕКОМЕНДУЕТСЯ, за исключением случаев крайней необходимости.

- Поддержка курсоров для больших запросов
- Таймауты и повтор запросов от пользователя
- Транзакции с использованием API
- Иерархическое пространство имён
- Единый каталог, где LFN – основной ключ
- Дальнейшие LFN – как символические ссылки на основное
- Интегрированная GSI авторизация и аутентификация
- Поддержка ACL
- Интеграция с VOMS
- Поддержка системных метаданных (размер файла, дата создания,...) + поле для пользовательских метаданных
- База данных: Oracle или MySQL





- Структура каталогов пространства имён LFC имеет следующий формат:

`/grid/<VO>/<subpaths>`

Пользователь конкретной виртуальной организации (ВО) будет иметь права записи и чтения только в соответствующей <VO> поддиректории. Политика доступа к поддиректориям более глубокого уровня (<subpaths>) реализуется в рамках каждой ВО.

- Для всех lfc-команд доступны страницы руководства, которые можно просмотреть, выполнив команду

\$ man <lfc-command>

где <lfc-command> - одна из команд LFC.

- Большинство этих команд работают подобно их Unix-эквивалентам, но применяются для операций над каталогами и файлами пространства имён LFC.
- При работе с командами LFC пользователь должен помнить, что операции, которые они выполняют, относятся только к файловому каталогу, а не физическим файлам, представленными в нем соответствующими записями.

lfc-ls	вывод содержимого LFC директории
lfc-mkdir	создание директории в LFC
lfc-rename	переименование файла/директории
lfc-rm	удаление файла/директории
lfc-ln	создание символической ссылки на файл/директорию
lfc-chmod	изменение режима доступа к файлу или директории в LFC
lfc-chown	изменение владельца файла/директории
lfc-setcomment	добавление/изменение комментария о файле/директории
lfc-delcomment	удаление комментария о файле/директории
lfc-getacl	получение списка контроля доступа к файлу/директории
lfc-setacl	установка списка контроля доступа к файлу/директории
lfc-setacl	установка списка контроля доступа к файлу/директории

Полный список команд LFC приведён в источнике [2] в секции “LFC commands”.

- **lfc-ls** - команда просмотра содержимого директории в LFC . Она выводит список всех логических имен файлов (LFNs) в указанной директории.

ВНИМАНИЕ! -R опцию для вывода рекурсивного списка по всем поддиректориям НЕ рекомендуется часто использовать, т.к. это очень ресурсоёмкая операция для каталога и поэтому её следует избегать.

Пример вызова команды lfc-ls:

```
$ lfc-ls /grid/edu
```

```
generated
```

- **lfc-mkdir** - команда создания директорий в LFC:

```
$ lfc-mkdir /grid/edu/kut
```

```
$ lfc-ls -l /grid/edu
```

```
drwxrwxr-x 3 101 101 0 Mar 25 16:11 generated
```

```
drwxrwxr-x 0 101 101 0 Mar 25 16:18 kut
```


- **lfc-ln** - команда создания символической ссылки на файл или директорию. Таким образом, два LFN могут указывать на один и тот же объект (файл или директорию). В приведённом ниже примере создается символическая ссылка `/grid/edu/kut_symlink` на LFN `/grid/edu/kut`:

```
$ lfc-ln -s /grid/edu/kut /grid/edu/kut_symlink
```

Проверка успешности выполнения предыдущей команды:

```
$ lfc-ls -l /grid/edu/
```

```
drwxrwxr-x 3 101 101          0 Mar 25 16:11 generated
drwxrwxr-x 0 101 101          0 Mar 25 16:18 kut
lrwxrwxrwx 1 101 101          0 Mar 26 15:09 kut_symlink -> /grid/edu/kut
```

- **`lfc-setcomment`** и **`lfc-delcomment`** – команды добавления/удаления метаданных (комментариев) к записям в LFC. Например:

```
$ lfc-setcomment /grid/edu/kut "<|kut home LFC directory|>"
```

```
$ lfc-ls --comment /grid/edu/
```

```
generated
```

```
kut <|kut home LFC directory|>
```

```
$ lfc-delcomment /grid/edu/kut
```

```
$ lfc-ls --comment /grid/edu/
```

```
generated
```

```
kut
```

- **lfc-rm** - команда удаления LFN из файлового каталога. Эта команда может использоваться для удаления файла или директории с пространства имён LFC, но с двумя ограничениями:
 - файл может быть удалён только при условии, что в файловом каталоге нет связанных с ним SURL'ов. Если SURL существует, тогда для удаления LFN нужно использовать команды из набора `lcg_utils` (будут описаны дальше);
 - директория может быть удалена (опция `-r`) только в том случае, если она пустая.

```
$ lfc-rm /grid/edu/kut_symlink
```

Проверка успешности выполнения предыдущей команды:

```
$ lfc-ls -l /grid/edu/kut_symlink
```

```
/grid/edu/kut_symlink: No such file or directory
```

- LFC позволяет присоединять к файлу или директории **список прав доступа** (access control list – ACL), т.е. такой список, где указано, кому можно иметь доступ к объекту или модифицировать его. Указание прав очень похоже на то, как это реализовано в Unix-системах: r – read (читать), w – write (записывать), x – execute (исполнять). Комбинация таких прав может быть применена к
 - 1) пользователю (user);
 - 2) группам пользователей (groups);
 - 3) любым другим пользователям (other).
- В LFC пользователи и группы определяются с помощью **числовых виртуальных идентификаторов пользователя или группы** (user id – uid, group id – gid), которые виртуальны в том смысле, что они существуют только внутри пространства имён LFC.
- Пользователя можно указать через его имя, виртуальный идентификатор пользователя (uid) или отличительное имя (Distinguished name, DN). Аналогично для групп пользователей.
- Каждая директория в LFC имеет ACL-файл, используемый по умолчанию, права в котором применяются для любого файла или поддиректории, создаваемых внутри данной (т.е. права наследуются). После создания объекта внутри директории списки доступа к нему можно изменять.

- **`lfc-getacl`** – команда получения списка доступа.

```
$ lfc-getacl /grid/edu
```

```
# file: /grid/edu
```

```
# owner: root
```

```
# group: edu
```

```
user::rwx
```

```
group::rwx      #effective:rwx
```

```
other::r-x
```

```
default:user::rwx
```

```
default:group::rwx
```

```
default:other::r-x
```

```
$ lfc-getacl /grid/edu/kut
```

Результат выполнения команды предоставляет информацию об отличительном имени (Distinguished name, DN) и группе владельца директории/файла, а также о текущем и изначальном списках прав доступа.

- Программное обеспечение промежуточного уровня gLite 3.1 предоставляет широкий набор средств для загрузки/скачивания файлов в/из Грид, репликации (копирования) файлов и взаимодействий с файловым каталогом. Эти средства скрывают от пользователя всю сложность взаимодействия файлового каталога с элементами хранения данных, а также минимизируют риск искажения грид-файлов.
- Помимо этого, есть низкоуровневые средства для операций с файлами (команды `edg-gridftp-*`, `globus-url-copy`, `srm-*` команды). В некоторых случаях они бывают довольно полезны, но их настоятельно НЕ рекомендуется использовать малоопытным пользователям, т.к. эти низкоуровневые инструменты не проверяют соответствие между файлом на элементе хранения данных и записью в файловом каталоге.

- Прежде, чем выполнять какие-либо операции с данными, необходимо знать, какие сервисы для управления ими (элементы хранения данных и файловый каталог) доступны для вашей виртуальной организации. В gLite 3.1 это можно сделать при помощи утилиты `lcg-infosites`.

```
$ lcg-infosites --vo edu se
```

```
Avail Space(Kb) Used Space(Kb) Type SEs
```

```
-----
20630000      n.a           n.a    vps106.jinr.ru
20400000      1             n.a    vps108.jinr.ru
```

```
$ lcg-infosites --vo edu lfcLocal
```

```
vps104.jinr.ru
```

где `--vo` – имя виртуальной организации, соответствующая информация о которой запрашивается,

`se` указывает на то, какие данные должны быть выведены (в приведённом выше примере запрашивается информация об элементах хранения данных).

`lfcLocal` обозначает, что информация запрашивается о локальном по отношению к данному интерфейсу пользователя файловому каталогу, обслуживающему указанную ВО.

edg-gridftp-ls	список файлов и директорий на SE
edg-gridftp-mkdir	создание директории на SE
edg-gridftp-rmdir	удаление директории на SE
edg-gridftp-rm	удаление файла на SE
edg-gridftp-rename	переименование файла или директории на SE
edg-gridftp-exists	определение существования на SE файла или директории
globus-url-copy	копирование файла с одного SE на другой (в том числе, с UI на SE и наоборот)

Перед созданием директории на SE, можно убедиться, что она ещё не существует:

```
$ edg-gridftp-exists gsiftp://vps106.jinr.ru/dpm/jinr.ru/home/edu/kut
```

```
error globus_ftp_client: the server responded with an error
```

Эта ошибка означает, что директория не существует.

Создание директории на SE:

```
$ edg-gridftp-mkdir gsiftp://vps106.jinr.ru/dpm/jinr.ru/home/edu/kut
```

Проверка успешности выполнения предыдущей операции:

```
$ edg-gridftp-exists gsiftp://vps106.jinr.ru/dpm/jinr.ru/home/edu/kut
```

Отсутствие ошибок свидетельствует о том, что директория была создана.

Также можно вывести список файлов и директорий:

```
$ edg-gridftp-ls gsiftp://vps106.jinr.ru/dpm/jinr.ru/home/edu
```

```
kut
```

Копирование файла с локального диска UI на SE:

```
$ globus-url-copy -vb file:///etc/group \  
gsiftp://vps106.jinr.ru/dpm/jinr.ru/home/edu/kut/test_file1.dat
```

```
$ edg-gridftp-ls gsiftp://vps106.jinr.ru/dpm/jinr.ru/home/edu/kut
```

```
test_file1.dat
```

lsg-cr	копирование файла из Грид на локальный диск (скачивание)
lsg-cr	копирование файла на SE и его регистрация в файловом каталоге (загрузка в Грид)
lsg-del	удаление файла, одной или всех его реплик
lsg-rep	копирование файла с одного SE на другой и регистрация его в файловом каталоге
lsg-gt	получение TURL для данного SURL и указанного протокола
lsg-sd	присвоение файлу статуса "Done" для указанного SURL в SRM-запросе

lcg-aa	Получить список узлов (состояние узлов) для указанного GUID
lcg-ra	Получить список узлов для указанного GUID
lcg-rf	Получить список узлов для указанного GUID, с указанием SE
lcg-uf	Получить список узлов для указанного GUID, с указанием SE
lcg-la	Получить список узлов для указанного GUID, LFN, GUID и SURL
lcg-lg	Получить список узлов для указанного GUID, LFN и SURL
lcg-lr	Получить список узлов для указанного GUID, LFN, GUID и SURL

- Для загрузки файла в Грид (т.е. передачи его с локальной машины на элемент хранения данных с регистрацией этого файла в файловом каталоге) используется команда **lcg-cr** (`cr` – сокращение от `copy®ister`), которая после успешного завершения выдаст GUID файла:

```
$ lcg-cr --vo edu -d vps108.jinr.ru file:/etc/group
```

где аргументом является полный путь к файлу на локальной машине, а в опциях `--vo <vo>` указывается виртуальная организация пользователя, `-d <destination>` - элемент хранения данных, на который должен быть скопирован файл.

Для указания относительного пути на SE, куда пользователь желает сохранить файл, можно использовать опцию `-P`:

```
-d <destination> -P my_dir/my_file
```

Если опция `-P` не указана, то путь будет сгенерирован автоматически.

Также можно указать полный SURL, включая имя машины SE, путь и желаемое имя файла, но путь должен быть указан в соответствии с виртуальной организацией пользователя.

- Примеры вызова команды с разным указанием места назначения:

```
$ lcg-cr --vo edu -d srm://vps108.jinr.ru/dpm/jinr.ru/home/edu/kut/test1 file:/etc/group
```

Проверка успешной записи файла:

```
$ edg-gridftp-exists gsiftp://vps108.jinr.ru/dpm/jinr.ru/home/edu/kut/test1
```

Указание относительного пути на SE для записи файла:

```
$ lcg-cr --vo edu -d vps108.jinr.ru -P kut/test_file2.dat file:/etc/group
```

Проверка успешной записи файла:

```
$ edg-gridftp-ls gsiftp://vps108.jinr.ru/dpm/jinr.ru/home/edu/kut
```

Используя опцию -l можно указывать LFN в LFC для копируемого файла.

```
$ lcg-cr --vo edu -d vps108.jinr.ru -P kut/test_file3.dat -l lfn:/grid/edu/kut/test3.lfn  
file:/etc/group
```

Примечание: Как уже упоминалось выше, LFN в LFC представлены в виде иерархической структуры, наподобие дерева папок в Unix. Т.е. LFN имеет вид /grid/<vo>/<dir1>. Поддиректории в пространстве имён НЕ создаются автоматически при выполнении команды lcg-cr и пользователь должен позаботиться самостоятельно о создании нужных поддиректорий заранее (команды lfc-mkdir и lfc-rmdir).

- После того, как файл был сохранен на SE и зарегистрирован в файловом каталоге, при помощи команды **lcg-rep** на него можно создать реплики :

```
$ lcg-rep -v --vo edu -d vps106.jinr.ru guid:fa2907ca-f81c-468d-b1c0-4126718f709f
```

где файл для репликации можно указать через LFN, GUID или SURL, в опции -d указывается SE, на котором должна быть создана копия файла. В качестве места назначения может быть как просто имя машины SE, так и полный SURL, записанный в таком же формате, как и для команды `lcg-cr`. Команда репликации также допускает использование опции -P для указания относительного пути места назначения.

Примечание: Каждый SE может хранить только одну реплику для конкретного GUID. Если пользователь попытается использовать **lcg-rep** для создания ещё одной копии файла на SE, который уже содержит на него реплику, то команда выполниться успешно, но новая копия файла создана не будет.

- Команда **lcg-lr** (list replicas) позволяет пользователю получать список всех реплик для файла, зарегистрированного в файловом каталоге.

```
$ lcg-lr guid:fa2907ca-f81c-468d-b1c0-4126718f709f
```

В качестве аргумента этой команды можно использовать LFN, GUID или SURL.

- Команда **lcg-lg** (list GUID) возвращает GUID файла, связанный с указанным LFN или SURL.

```
$ lcg-lg lfn:/grid/edu/kut/test3.lfn
```

```
guid:fa2907ca-f81c-468d-b1c0-4126718f709f
```

- Команда **lcg-la** (list aliases) выводит список всех LFN, относящихся к указанному файлу, который может быть определён через GUID, любое из логических имён файла или SURL одной из его реплик.

```
$ lcg-la srm://vps108.jinr.ru/dpm/jinr.ru/home/edu/kut/test_file3.dat
```

```
lfn:/grid/edu/kut/test3.lfn
```

- Команда **lcg-cp** (copy) предназначена для копирования файла из Грид на ресурс вне Грида. Первый аргумент (исходный файл) может быть LFN, GUID или SURL. Вторым аргументом может быть как локальное имя файла либо действительный TURL.

```
$ lcg-cp -v lfn:/grid/edu/kut/test3.lfn file:/tmp/test_kut.dat
```

```
$ ll /tmp/test_kut.dat
```

```
-rw-rw-r-- 1 kut kut 1092 Apr 2 16:29 /tmp/test1.kut
```

Примечание: хотя эта команда была разработана для копирования файлов из Грид на локальные ресурсы, при правильном указании TURL она может быть использована как для передачи файлов между двумя SE, так и для копирования файла с локальных ресурсов в Грид. Этого **НЕ следует делать**, потому что будет иметь место такой же эффект, как и в случае с командой lcg-fer, но **без регистрации** в файловом каталоге, что сделает эту реплику “невидимой” для грид-пользователей.

Примечание: если копирование файла происходит с ленточных запоминающих устройств большой ёмкости (Mass Storage System, MSS), то запрашиваемый файл может отсутствовать на дисковом буфере, а находиться на ленте. По этой причине, копирование файла может закончиться неудачно из-за истечения времени ожидания при выполнении команды, т.к. файл должен быть помещён с ленты в дисковый буфер. Опция -t позволяет менять время ожидания.

- Команда **lcg-gt** (get TURL) позволяет получить TURL из SURL и поддерживаемый протокол. Результат выполнения этой команды зависит от того, имеет ли SE SRM интерфейс или нет. Но в любом случае, результатом являются три строки, первая из которых всегда TURL, а остальные две (**requestID** и **fileID**) имеют значение только, если SE поддерживает SRM. Более подробную информацию можно узнать в [2].

```
$ lcg-gt srm://vps108.jinr.ru/dpm/jinr.ru/home/edu/kut/test_file3.dat gsiftp
```

```
gsiftp://vps108.jinr.ru/vps108.jinr.ru:/storage/edu/2008-04-03/test_file3.dat.86.0
```

```
88
```

```
0
```

- В случае SRM интерфейса, TURL возвращается самим SRM. Если SE является ленточным накопителем, то прежде, чем TURL будет возвращён команде, файл будет скопирован на дисковый буфер, если в момент запроса он на нём отсутствовал. Поэтому выполнение команды **lcg-gt** может занять довольно продолжительное время, но зато последующий вызов команды **lcg-cr** с данным TURL начнёт копирование файла немедленно, т.к. он уже будет находиться на дисковом буфере. Это одна из причин, почему SRM интерфейс желательно использовать для всех MSS.

- Файл, сохранённый на SE и зарегистрированный в файловом каталоге, может быть удалён командой **lcg-del**. Если в качестве аргумента указан SURL, то конкретно эта реплика будет удалена. Если указан GUID или LFN, то нужно указать опцию **-s <SE>**, которая показывает, какая из реплик должна быть удалена. В противном случае используется опция **-a**, что приводит к удалению всех реплик файла с элементов хранения данных и в файловом каталоге. Если GUID-LFN не имеет реплик в каталоге, такая запись тоже удаляется из файлового каталога.

```
$ lcg-lr lfn:/grid/edu/kut/test3.lfn
```

```
srm://vps106.jinr.ru/dpm/jinr.ru/home/edu/generated/2008-04-03/filecff22288-59f1-4478-a4f4-593b3b9ac1bc
```

```
srm://vps108.jinr.ru/dpm/jinr.ru/home/edu/kut/test_file3.dat
```

```
$ lcg-del -s vps106.jinr.ru lfn:/grid/edu/kut/test3.lfn
```

```
$ lcg-lr lfn:/grid/edu/kut/test3.lfn
```

```
srm://vps108.jinr.ru/dpm/jinr.ru/home/edu/kut/test_file3.dat
```

```
$ lcg-del -a lfn:/grid/edu/kut/test3.lfn
```

```
$ lcg-lr lfn:/grid/edu/kut/test3.lfn
```

```
vps104.jinr.ru: /grid/edu/kut/test3.lfn: No such file or directory
```

```
lcg_lr: No such file or directory
```

Последнее сообщение об ошибке свидетельствует о том, что LFN больше не зарегистрирован в каталоге, т.к. все реплики были удалены.

- Команда **lcg-rf** (register file) позволяет зарегистрировать в файловом каталоге физически существующий на SE файл, создавая пару GUID-SURL. Опция -g даёт возможность задать GUID, иначе он будет автоматически сгенерирован. Проверка существования файла на SE:

```
$ edg-gridftp-exists gsiftp://vps106.jinr.ru/dpm/jinr.ru/home/edu/kut/test_file1.dat
```

Если файл не существует, то скопируйте его (см. команду globus-url-copy).

Проверка, что файл с данным SURL незарегистрирован в файловом каталоге:

```
$ lcg-lr srm://vps106.jinr.ru/dpm/jinr.ru/home/edu/kut/test_file1.dat
```

```
vps104.jinr.ru: srm://vps106.jinr.ru/dpm/jinr.ru/home/edu/kut/test_file1.dat: No such file or directory
```

```
lcg_lr: No such file or directory
```

Регистрация файла в каталоге:

```
$ lcg-rf -l /grid/edu/kut/test_data.lfn
```

```
srm://vps106.jinr.ru/dpm/jinr.ru/home/edu/kut/test_file1.dat
```

```
guid:63760075-40b6-4916-a9e3-358b18fd0020
```

Проверка успешной регистрации файла в каталоге и его LFN:

```
$ lcg-lr guid:63760075-40b6-4916-a9e3-358b18fd0020
```

```
srm://vps106.jinr.ru/dpm/jinr.ru/home/edu/kut/test_file1.dat
```

```
$ lcg-la guid:63760075-40b6-4916-a9e3-358b18fd0020
```

```
lfn:/grid/edu/kut/test_data.lfn
```

- Команда **lcg-aa** (add alias) позволяет пользователю добавить новый LFN к существующему GUID.

Получение GUID, к которому хотим добавить новый LFN:

```
$ lcg-lg lfn:/grid/edu/kut/test_data.lfn  
guid:63760075-40b6-4916-a9e3-358b18fd0020
```

Проверка существующих LFN для данного GUID:

```
$ lcg-la guid:63760075-40b6-4916-a9e3-358b18fd0020  
lfn:/grid/edu/kut/test_data.lfn
```

Добавление нового LFN:

```
$ lcg-aa guid:63760075-40b6-4916-a9e3-358b18fd0020 lfn:/grid/edu/kut/test_lfn2
```

Проверка существующих LFN для данного GUID:

```
$ $ lcg-la guid:63760075-40b6-4916-a9e3-358b18fd0020  
lfn:/grid/edu/kut/test_data.lfn  
lfn:/grid/edu/kut/test_lfn2
```

- Команда **lcg-ra** (remove alias) удаляет LFN для существующего GUID.

Проверка существующих LFN для данного GUID:

```
$ lcg-la guid:63760075-40b6-4916-a9e3-358b18fd0020
```

```
lfn:/grid/edu/kut/test_data.lfn
```

```
lfn:/grid/edu/kut/test_lfn2
```

Удаление одного LFN:

```
$ lcg-ra guid:63760075-40b6-4916-a9e3-358b18fd0020 lfn:/grid/edu/kut/test_lfn2
```

Проверка существующих LFN для данного GUID:

```
$ lcg-la guid:63760075-40b6-4916-a9e3-358b18fd0020
```

```
lfn:/grid/edu/kut/test_data.lfn
```

- Команда **lcg-uf** (unregister file) удаляет пару GUID-SURL (которым соответствуют первый и второй аргументы команды) в файловом каталоге. Если была удалена последняя реплика файла, то также удаляется и сама пара GUID-LFN.

Примечание: данная команда удаляет только соответствующую запись в файловом каталоге, но не удаляет файл с SE физически.

Удаление реплики из каталога:

```
$ lcg-uf guid:63760075-40b6-4916-a9e3-358b18fd0020  
srm://vps106.jinr.ru/dpm/jinr.ru/home/edu/kut/test_file1.dat
```

```
63760075-40b6-4916-a9e3-358b18fd0020: No such GUID
```

Проверка, что файл с данным GUID незарегистрирован в файловом каталоге:

```
$ lcg-la guid:63760075-40b6-4916-a9e3-358b18fd0020
```

```
vps104.jinr.ru: 63760075-40b6-4916-a9e3-358b18fd0020: No such file or directory
```

```
lcg_la: No such file or directory
```

Проверка, что файл по-прежнему существует на SE:

```
$ edg-gridftp-exists gsiftp://vps106.jinr.ru/dpm/jinr.ru/home/edu/kut/test_file1.dat
```

Удаление файла и директории с SE:

```
$ edg-gridftp-rm gsiftp://vps106.jinr.ru/dpm/jinr.ru/home/edu/kut/test_file1.dat
```

```
$ edg-gridftp-rmdir gsiftp://vps106.jinr.ru/dpm/jinr.ru/home/edu/kut/
```

srm-advisory-delete

srm_v1

srmmv

srm-get-metadata

srmcp

srmrm

srm-get-request-status

srmls

srmrmdir

srm-storage-element-info

srmmkdir

srmstage

1. Веб-сайт [gLite](http://glite.web.cern.ch) Веб-сайт gLite Веб-сайт gLite <http://glite.web.cern.ch>
2. Руководства пользователя gLite
<http://glite.web.cern.ch/glite/documentation/userguide.asp>
3. <https://srm.fnal.gov>