Glexec overview

Gerben Venekamp
NIKHEF

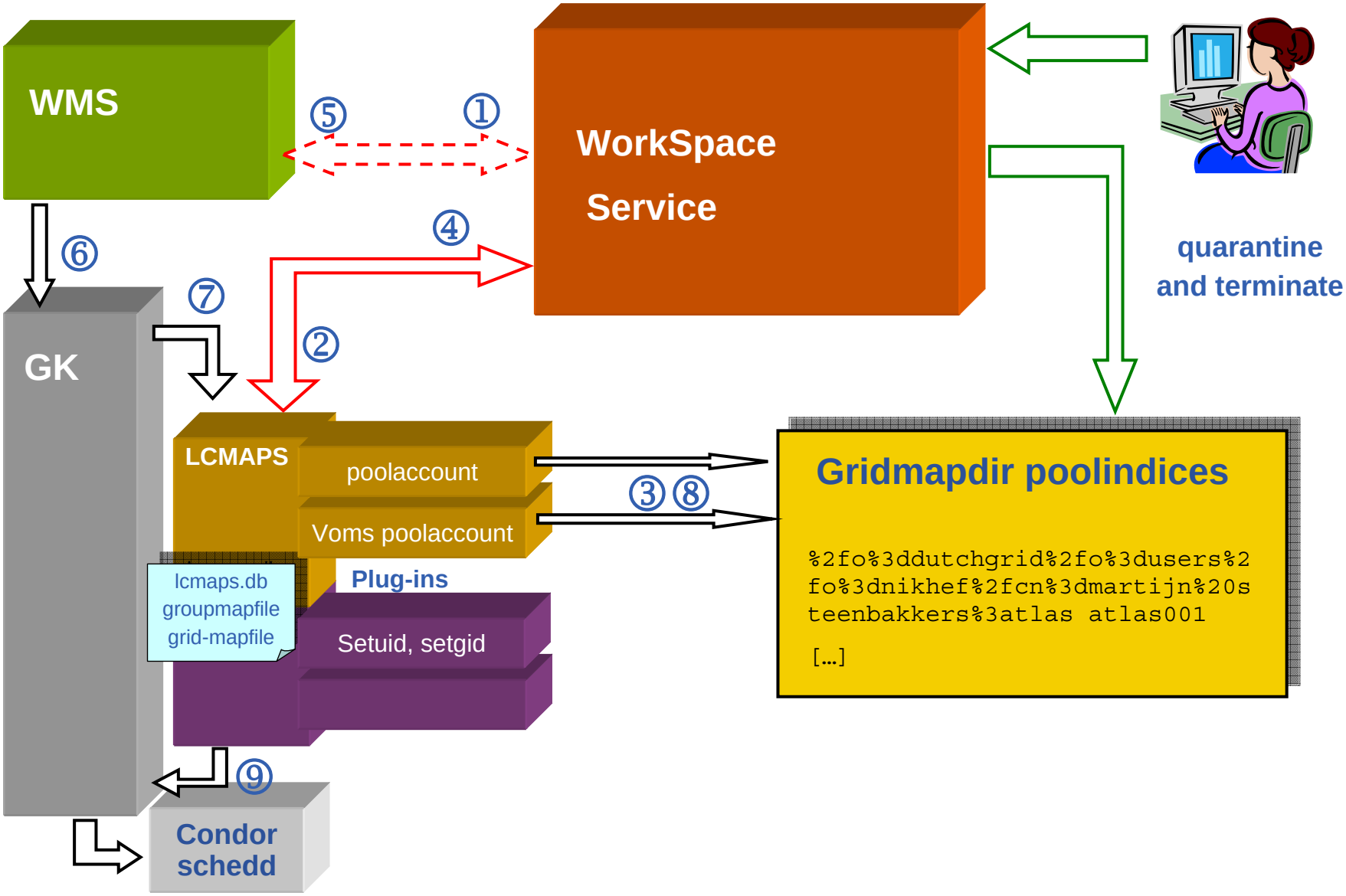
Contents

- Isolation/sandboxing components
- Why do we need glexec (and gsexec)?
- How does glexec fit in the gLite architecture?
- Comparison between glexec and gsexec
- How can glexec be invoked?
- Status

Isolation (sandboxing)

- **Roadmap**
  - virtualization of resources (VM) *or* assigning of local credentials
  - should be transparent for the user
- **EGEE-I Architecture**
  - only based on credential mapping (although VM solution is coming close!)
  - do as little as possible with ‘root’ privileges: “suexec” wrapper functionality is needed.
  - minimizing local management: poolaccounts & poolgroups
  - credential mapping and manipulation: LCMAPS
  - management capabilities on these accounts: WorkSpace Service (WSS)

LCMAPS and the WSS in gLite-1



‘suexec’ requirements

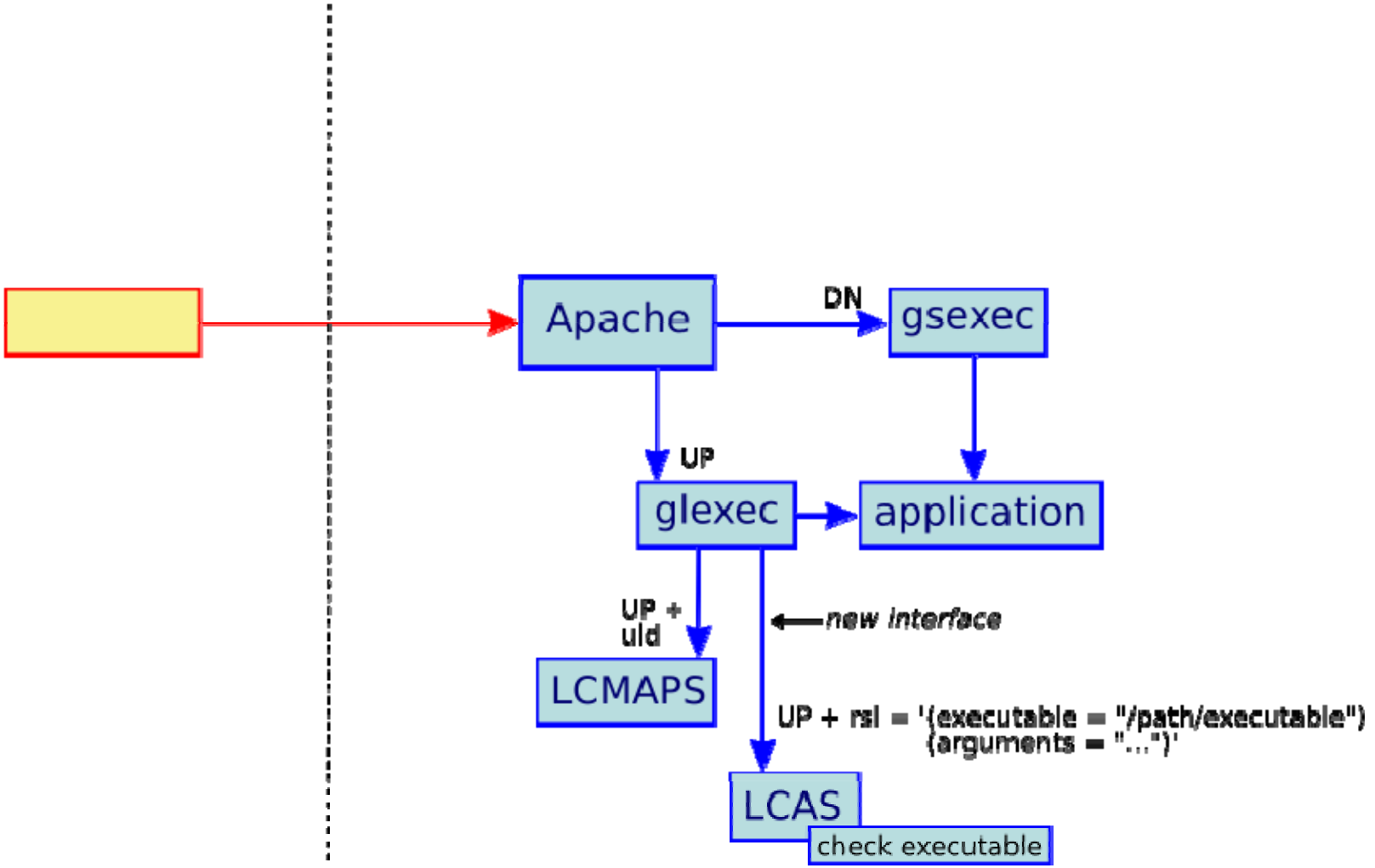
- **Do as little as possible with ‘root’ privileges**
- **Allow *VO-services* to use (pool)accounts for each user job**
- **In- and outgoing pipes, file descriptors should be preserved as much as possible.**
- **Should be usable by C/C++ and java services: program executable.**
- **Sites should still be able to control the access to their resources**
- **Apache’s suEXEC fulfills many of these requirements**
  - Safe code, has proven itself
  - Works with apache
  - 2 clones from apache’s suEXEC: gsexec (part of gridsite) and glexec (part of gLite)

suexec wrapper

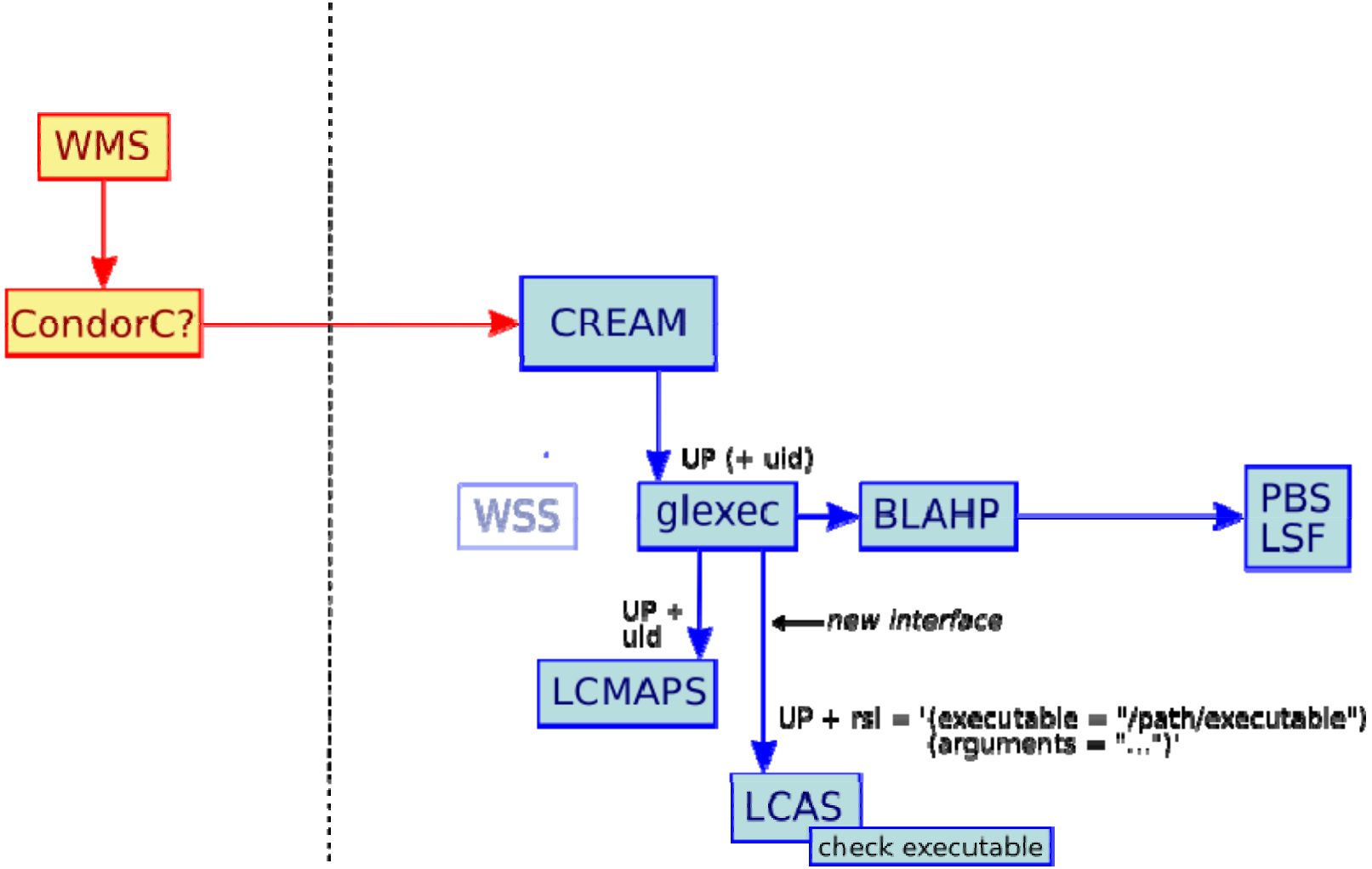
Thin layer with root privileges will replace gatekeeper

- **Intended for identity-switching services:**
  - *condor, gridsite, globus gram, cream.*
- **Internal**
  - Uses LCMAPS/Workspace service as credential mapping mechanism
  - Executes the requested command with local credentials
- **External interface**
  - Should be usable by C, java (, perl?) services: program executable.
  - A (user) credential should be passed to ***suexec***.
  - In- and outgoing pipes, file descriptors should be preserved as much as possible.

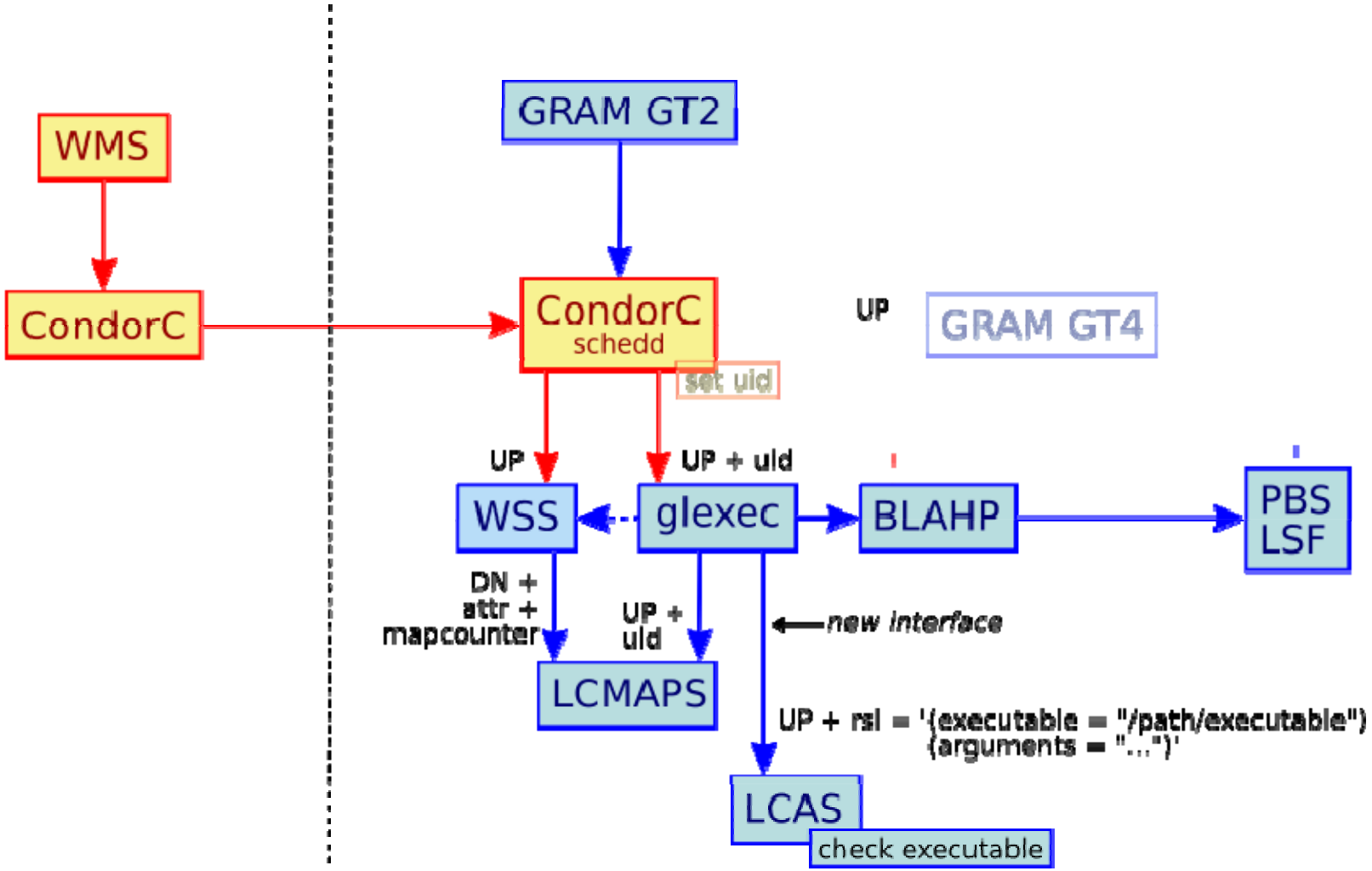
Scenario I: Apache



Scenario II: Cream



Scenario III: Condor/WSS



gsexec ‘versus’ glexec

Glexec

- **Can be used by VO services**
- **Input: user proxy, executable + args**
- **Optional input in “verify” mode: uid, gid**
- **Uses LCAS to**
  - check the user ban/white list
  - verify the user proxy (validity, lifetime, full/limited, CRL)
  - Hold the executable against a whitelist.
- **Uses LCMAPS to do the account mapping**
  - Dry-run is used in “verify” mode
- **Copies and chowns the delegated user proxy**
- **Code in org.glite.security.glexec**

Gsexec

- **Only for site-controlled services**
- **Input: user DN, executable + args**
- **Uses traditional (non-VOMS, DN-based) poolaccount mapping**
- **Code resides in org.gridsite.core**
- **Not very configurable (on purpose)**

Glexec (cont’d)

- More configuration options
- Includes fork() wrapper library
  - returns the pid to the calling application
  - C api

Usage of *glexec* (1)

Two run-modes depending on setuid bit settings:

1. *glexec* **is setuid-root: setuid()/setgid() to local user in** *glexec* **code and execute the program**

```
-r-s--x--- 1 root apache /usr/sbin/glexec
```

In the next release:

1. *glexec* **runs as special user:** *glexec* **uses** ***sudo*** **for identity switching and program execution:**

```
-r-s--x--- 1 glexec glexec /opt/glite/sbin/glexec
```

   - ***sudo*** preserves only stdin, stdout, stderr
   - ***sudo*** can be configured to allow the user "**glexec**" to run a predefined set of programs (**blahp, qsub**)

Usage of *glexec* (2)

- **Environment variables to be set before calling glexec**
  - GLEXEC_MODE:
    - "lcmaps_verify_account": glexec <uid> <gid> <command+args>
    - "lcmaps_get_account": glexec <command+args>
  - SSL_CLIENT_CERT and SSL_CLIENT_CERT_<n>: PEM-encoded strings containing the proxy cert and chain components.
  - GLEXEC_SOURCE_PROXY: location of the (delegated) proxy to be used by the user job..
  - GLEXEC_TARGET_PROXY: location where the proxy should be copied to. If not specified ~/.glexec/proxy is used.
  - GLEXEC_ID (optional): unique job id to be used as an index for the jobrepository:
- **All other environment variables are cleared**

Usage of *glexec* (3)

Configuration:

- **At compilation time the default defines in the apache headers (`/usr/include/httpd/*.h`) are overridden by `glexec.h`**
  - Contains at the moment hardcoded locations of lcas and lcmaps config files
  - Next version will use a configuration file instead.
- **lcas and lcmaps use the following config files:**
  - **`/opt/glite/etc/lcas/lcas-glexec.db`**
  - **`/opt/glite/etc/lcmaps/lcmaps-glexec.db`**

Usage of *glexec* (4)

- **Fork wrapper C api:**

```
typedef struct {
   char *target_user;
   char *target_group;
   char *command; } glexec_args_t;
pid_t glexec_fork(glexec_args_t *args);
```

- **Forks off the user process and returns the process id**

Status

Status

- **gsexec available in gridsite version 1.1.14**

- ***glexec*:**
  - waiting for bugs in lcmaps to be fixed and the proxy verification plugin.
  - Needs further debugging and testing

Plans

- **Integration on prototype with CREAM and Condor**
- **Replace callouts to LCAS and LCMAPS by a callout to Workspace service (a.k.a. the full scenario).**
  - Workspace bind service
- **Use of sudo**
- **Use of a glexec configuration file**
- **Interoperability: use the common standardized authZ/mapping callout interface (OSG collaboration)**

Questions

- **Questions?**

Goals

- **Generic access control to services**
  - Authentication
  - Authorization
  - for legacy applications & file access, networks, …
- **Sites are always in control of their resources**
- **Flexibility, scalability**
- **Allow for central control in a site**
- **Converge to a single policy format**
- **Standardization of configuration**
- **Address requirements from NA4, SAAA-RG, and others (incorporated in MJRA3.1 “user requirements”)**

Workspace Service and LCMAPS

Workspace Service (WSS) is part of GT4-core and in gLite-1 (preview)

- *Account creation and account management*

- **Provides lifetime management and *will* provide quota management**
- **Account clean-up mechanism:**
  - Possibility to put account in quarantine first
  - Example clean-up script provided
- **Access control to Workspace Service based on DN and VOMS attr.**
- **Access control to account**
  - Currently based on DN
  - Will provide ACLs on VOMS attributes (?)
- **Support of poolaccounts**
  - Clean-up of poolaccounts
  - Uses LCMAPS as a back-end to manage gridmapdir (poolindex)

LCMAPS and the WSS in gLite-1