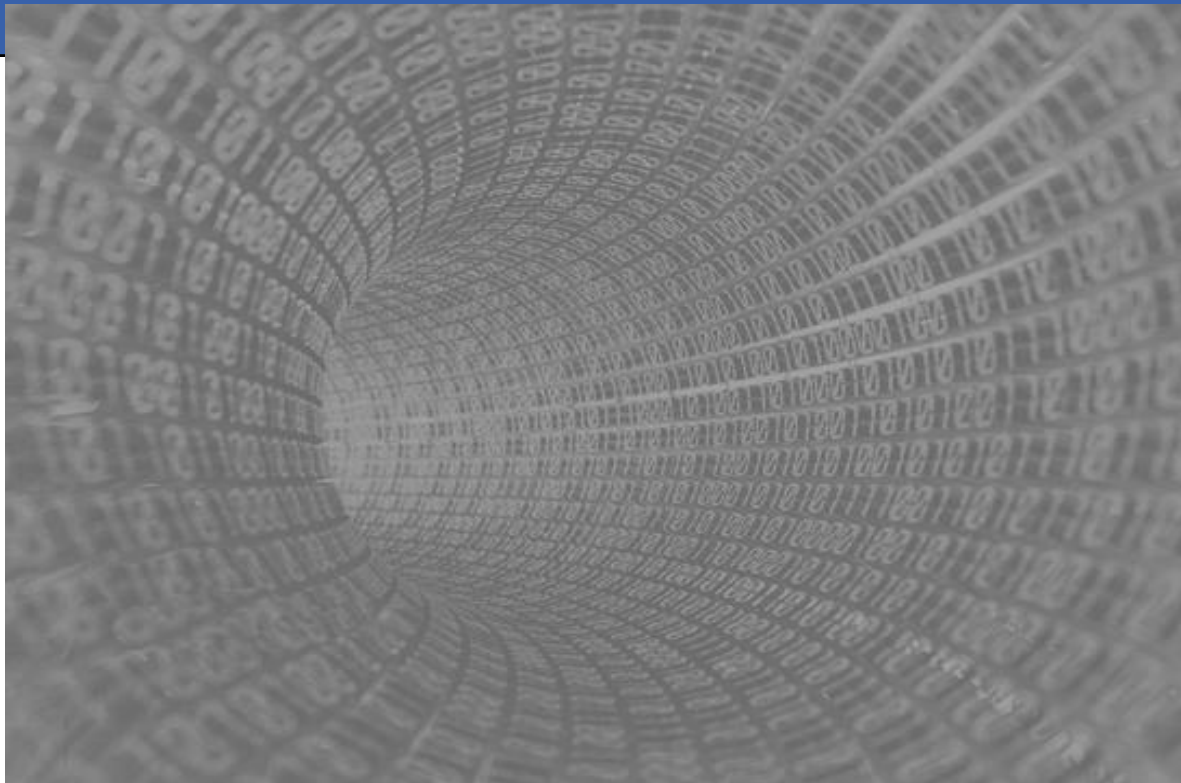


AliEn and gLExec

ALICE Offline Week July 2010



gLExec is a program that can switch users
... just like *su* or *sudo*

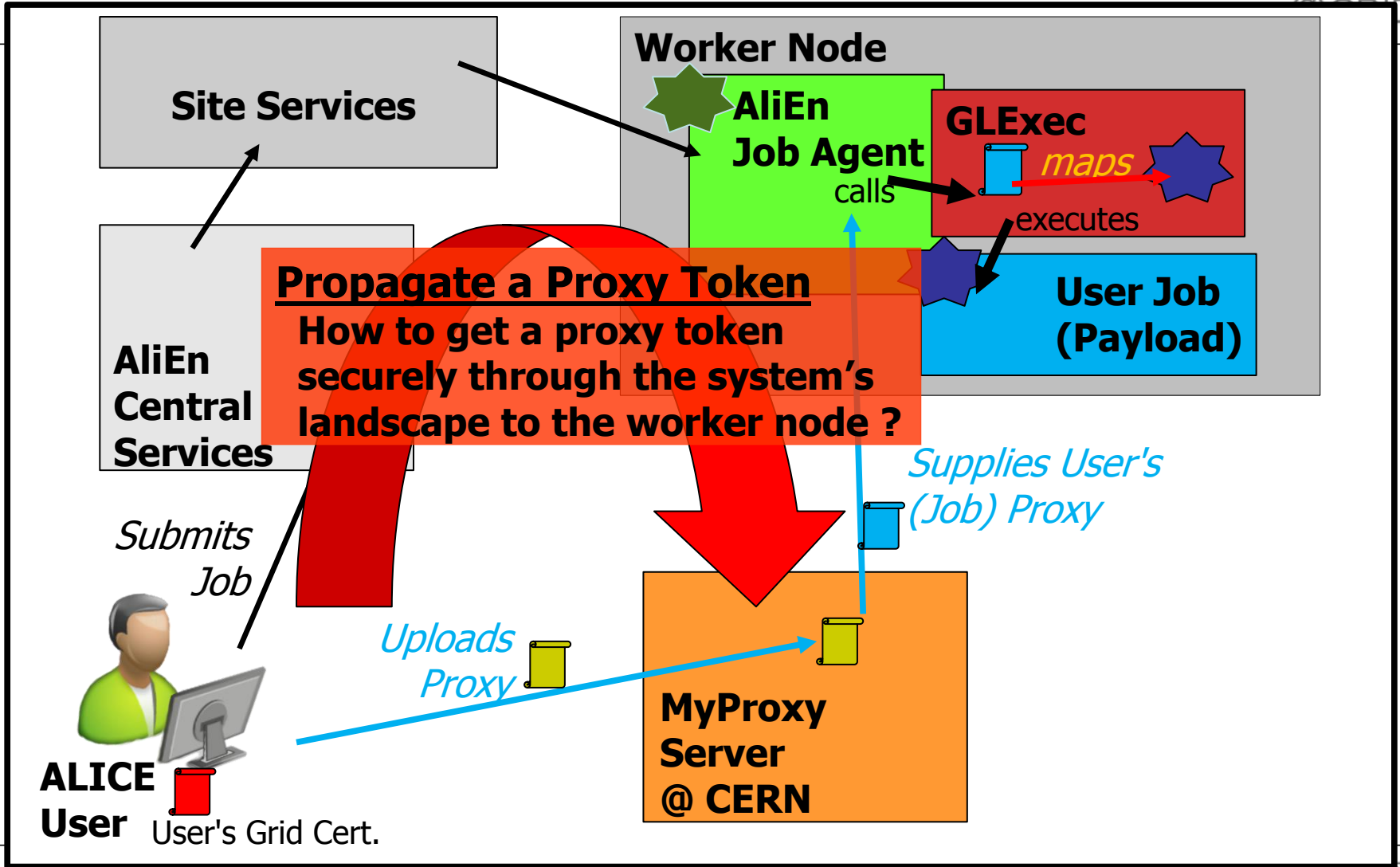
- Takes the command to run (the payload) and a proxy certificate
- Proxy certificate is mapped to a local user, based on rule sets
- The command is executed by using the mapped user account

... nothing more !

scope of implementation

- gLExec is only the user switch. Implementation of the **proxy delivery and its protection/security is not provided.**
- We need to implement a security subsystem to get the proxy on the WNs
- In order to really gain accountability/transparency we need to ensure the job has **both the right proxy and payload**
- If the we just shoot around proxies, we make the system a lot worse and less secure in terms of accountability!

scenario and problem



first level mile stones



- Enabling the AliEn JobAgent/Pilot to use gLExec
-> Let gLExec use the pilot job's proxy to run jobs
- The user's proxy needs to be uploaded to MyProxy
- Introduce the functionality for the Worker Node to download the user's proxy for a job

current status – calling it



FZK and CNAF provided third VO Boxes (CREAM) for gLExec

We found a bug in the temp. folder preparation (mkgltmpdir)

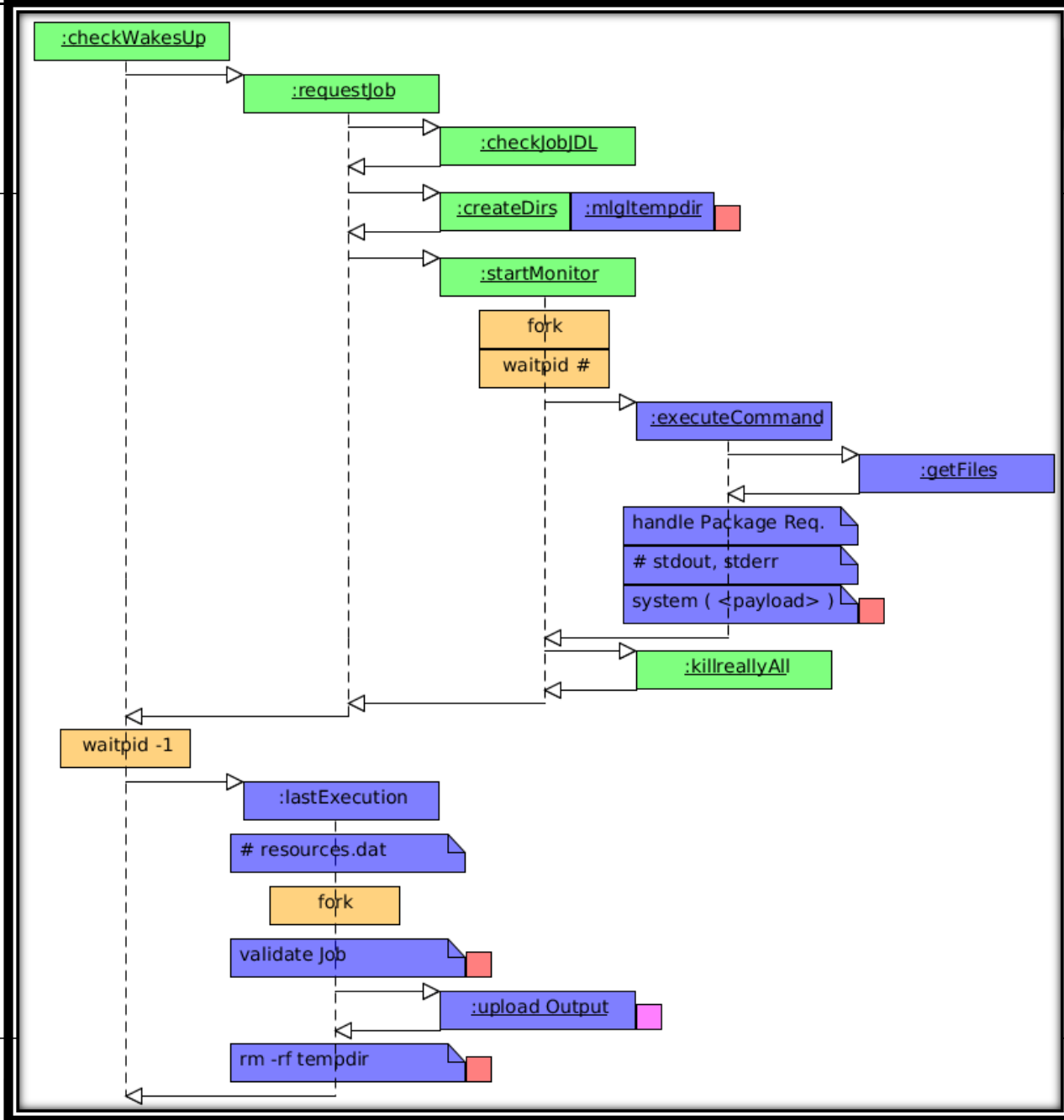
-> Maarten Litmaath provided us with a preliminary workaround

- Calling JDL's executable by gLExec within an AliEN Job works
- Executable runs perfectly by another user (gLYourself + X)
- Output (stderr, stdout) can be captured+registered by AliEn
- Problem with the interaction of temp folder creation and gLExec

modifications in the AliEn JobAgent



1. set up environment variables and proxy location
 2. set up a sub directory for gLExec
 3. move the job data into the sub directory
 4. call gLExec with prepared arguments
 5. get the job's output data
 6. Run the validation script
 7. clean up the sub directory after job execution
-
- x1 JobAgent needs to assess Job's state
- x2 the <spy> needs to be able to access the job's workdir

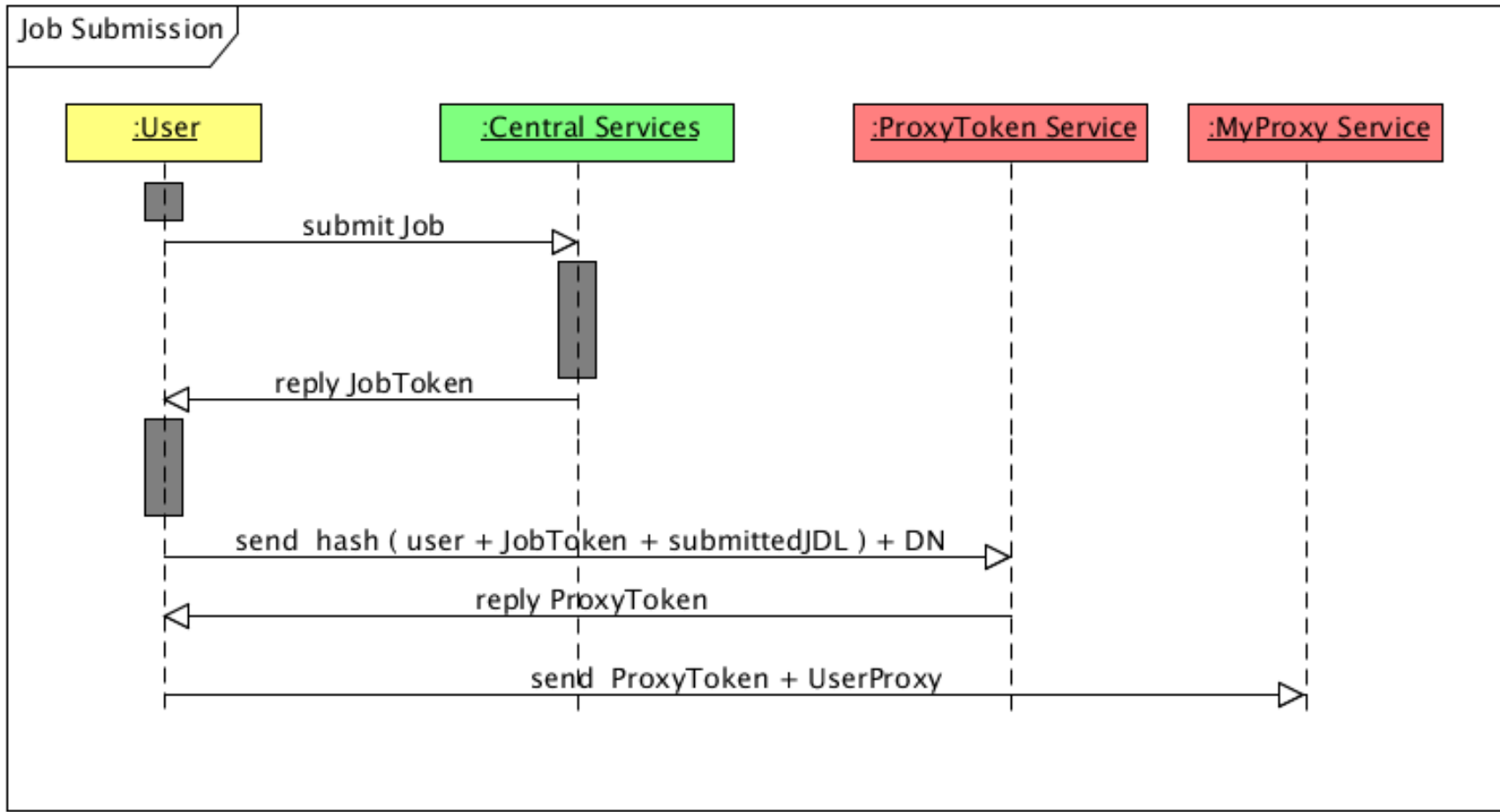


current status– exploiting the AliEn JA with sudo

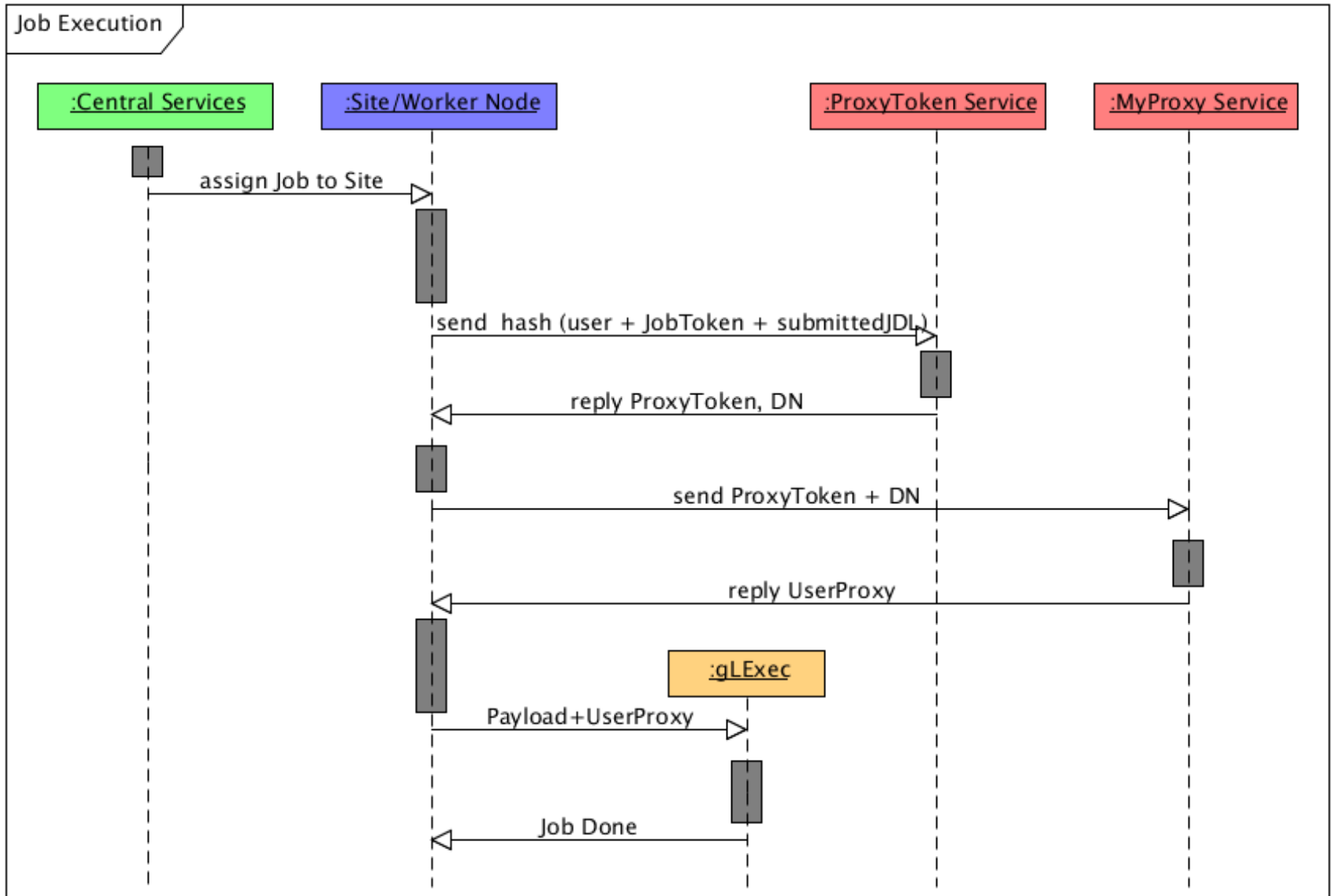


- Local development is done by replacing gLExec with sudo
- Possible to create the layout/changes of the JA and set stubs
- Stubs will be replaced later by gLExec for real testing
- **Two basic szenarios** for a gLExec enabled AliEn JobAgent:
 - A. Cut the JobAgent** into Agent and Job section and call gLExec once
 - B. Make several gLExec calls** within the JobAgent and keep structure

proxy propagation proposals



proxv propagation proposal



Questions ?

aliensh SE discovery

on behalf of Andreas Joachim Peters

ALICE Offline Week July 2010



Storage Selection in aliensh



- Since xalienfs-1.0.14 (current Alien HEAD)
 - Deprecated 'alien_CLOSE_SE' definition
 - However still works
 - Automatic discovery of closest site via *pcalimonitor* during 'alien-token-init'
 - Simplified target selection

```
"@ALICE::CERN::SE"  
  "@disk=2"      - 2 replicas on disk  
  "@tape=2"     - 2 replicas on tape  
  DEFAULT: "@disk=2"
```
- Server part (+alien 2.18-25) deployed on pcapiserv01/04/07

Storage Selection in aliensh



- `alien cp`: GRID download mode
 - tries to read a file from all possible replicas in the order given by MonaLisa service
- `alien cp`: GRID upload mode
 - tries to store replicas until the user defined storage policy (e.g. `disk=2`) is matched
- Server stays backward compatible with previous version
 - Only new client use new functionality
 - Old clients still work

Storage Selection in ROOT



- Mimic same behaviour like aliensh
 - `TFile::Open("alien://.....?alien.disk=2","RECREATE");`
 - Sync. replica creation on close
 - Same closest site auto configuration
 - `TFile::Open("alien://.....?");`
 - tries to open all possible replicas until success in the order given by ML service
- Code not yet available in SVN !!!