

# Grid Security

## gLEexec

Outline for an upcoming implementation

[steffen.schreiner@cern.ch](mailto:steffen.schreiner@cern.ch)

March 18<sup>th</sup> 2010



# Motivation



Sites in the Grid run jobs on behalf of ALICE users.

Currently, **all jobs run as one system user** on each site.

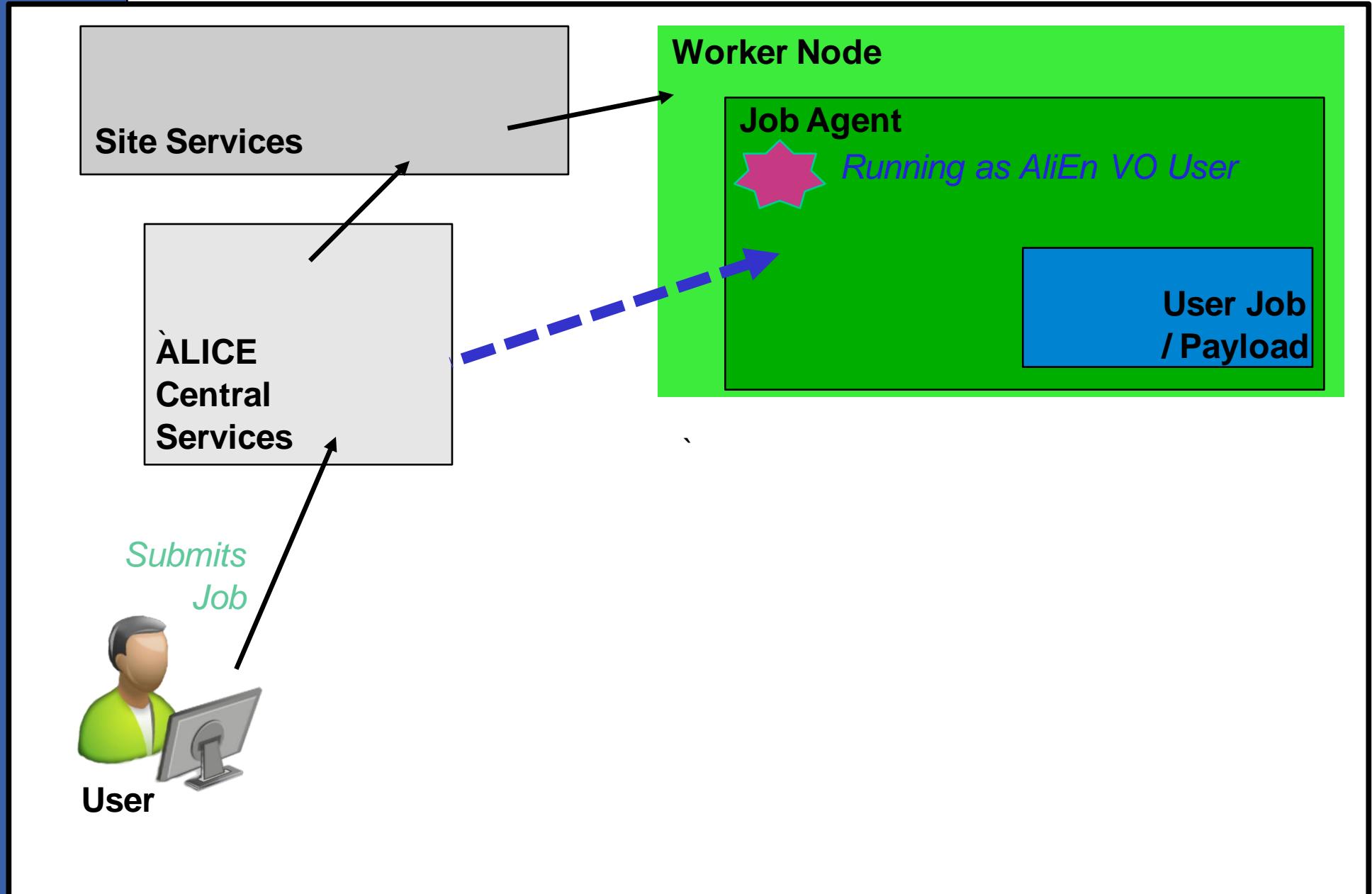
**Sites want to see the jobs running by users the jobs belong to.**

Primary motivation: Accountability / Traceability

Then, they could e.g. ban or restrain certain users.

Currently, they could only affect the whole VO.

# AliEn Job Execution





# Challenge



**How to run a job with a user representing the owner of the job ?**

A site knows the VO ALICE, but not all its users.

How to ensure a job runs with the right user ?



# gLExec

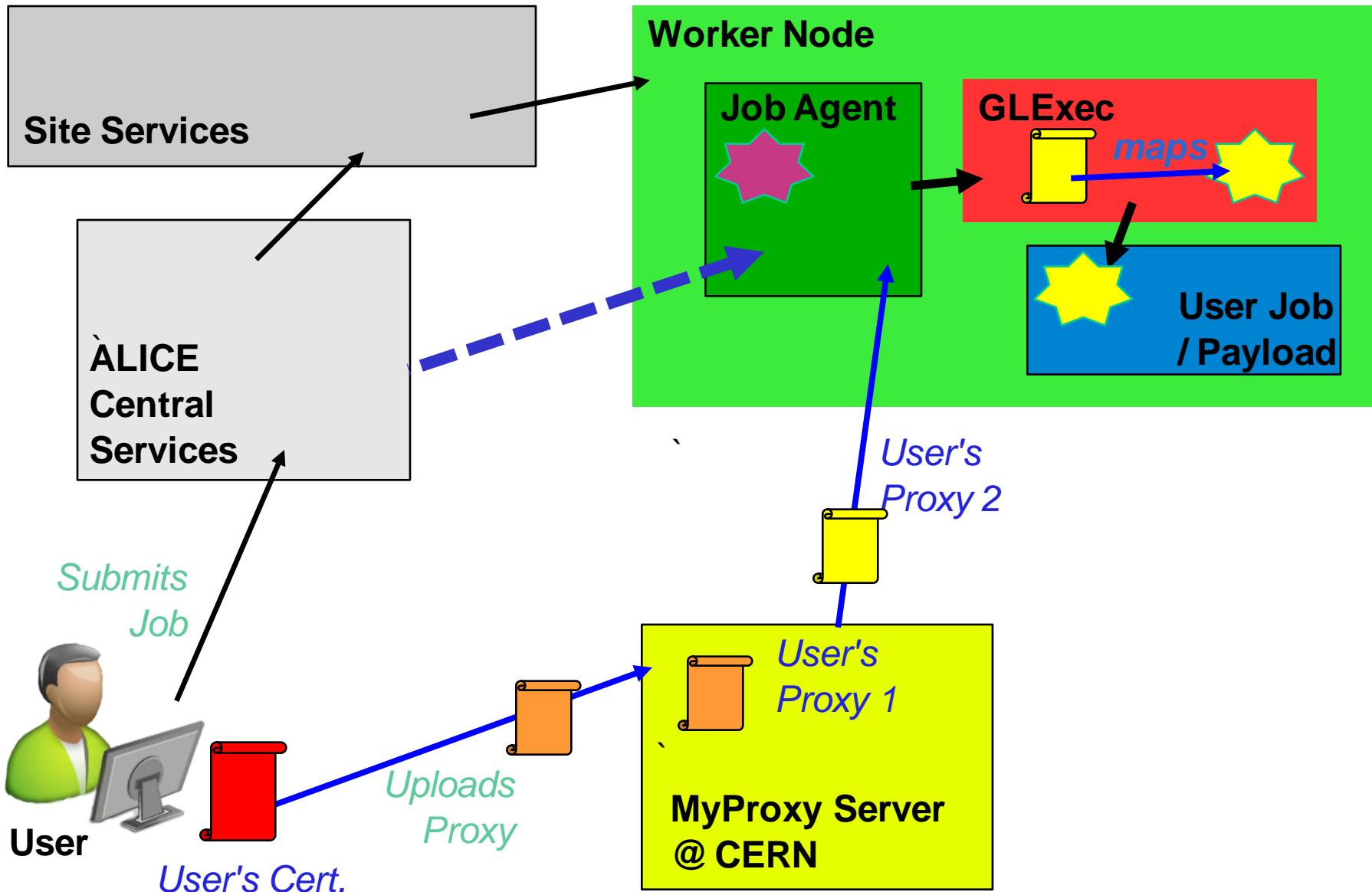


gLExec is a program that can switch users  
... just like *su* or *sudo*

1. It takes the command to run (a job) and a proxy certificate.
2. The proxy certificate is mapped to a local user, based on rules.
3. The command is executed as the mapped user.

**... nothing more !**

# AliEn Job Execution - gLEexec





# It's not that easy !

gLEc does only the user switch, how you implement the delivery of the user proxy is your business. How to protect/secure it, as well!

Just uploading the user proxy to MyProxy is not enough!

Worker Nodes have no right to download proxies from MyProxy. Thank goodness!

**We need to find a good way to introduce the permission for the Worker Node to download one user's proxy for a job.**



# Why that ?

The motivation was to make user's **accountable** for their actions and get **traceability** ...

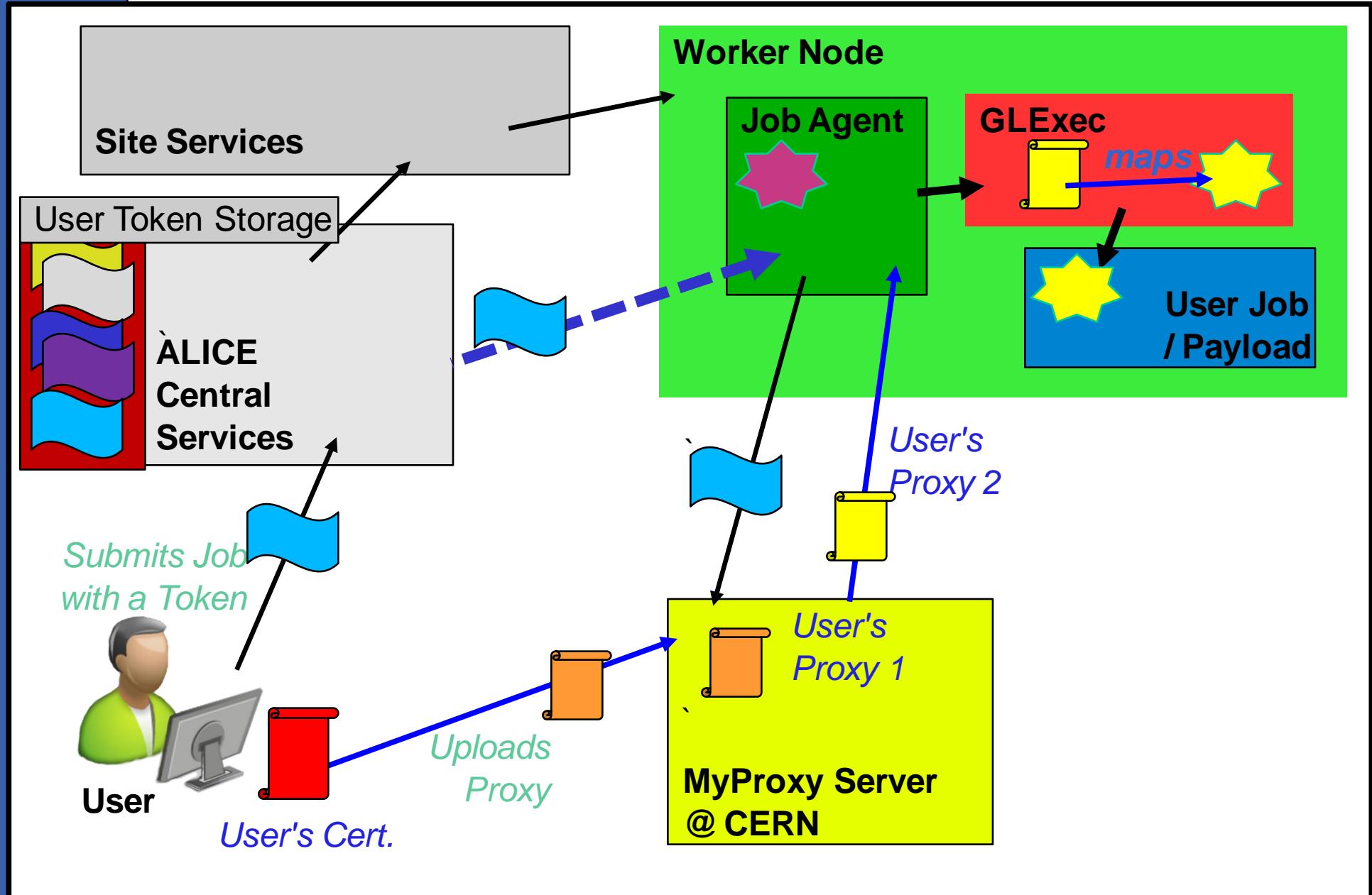
If a Worker Node can get any proxy, we are in big trouble anyway!

If we can't ensure a job runs with the right user (proxy), what we need gLExec and all the implementation effort for ?

-> **We need non-repudiation !**

**If we risk having jobs running with users not representing their owner's proxies, we made the system a lot worse and lose accountability!**

# First Idea





# Problem with that ?



**We would store user tokens in the Central Services from every ALICE user running jobs.**

**WE DON'T WANT TO GO THERE !**

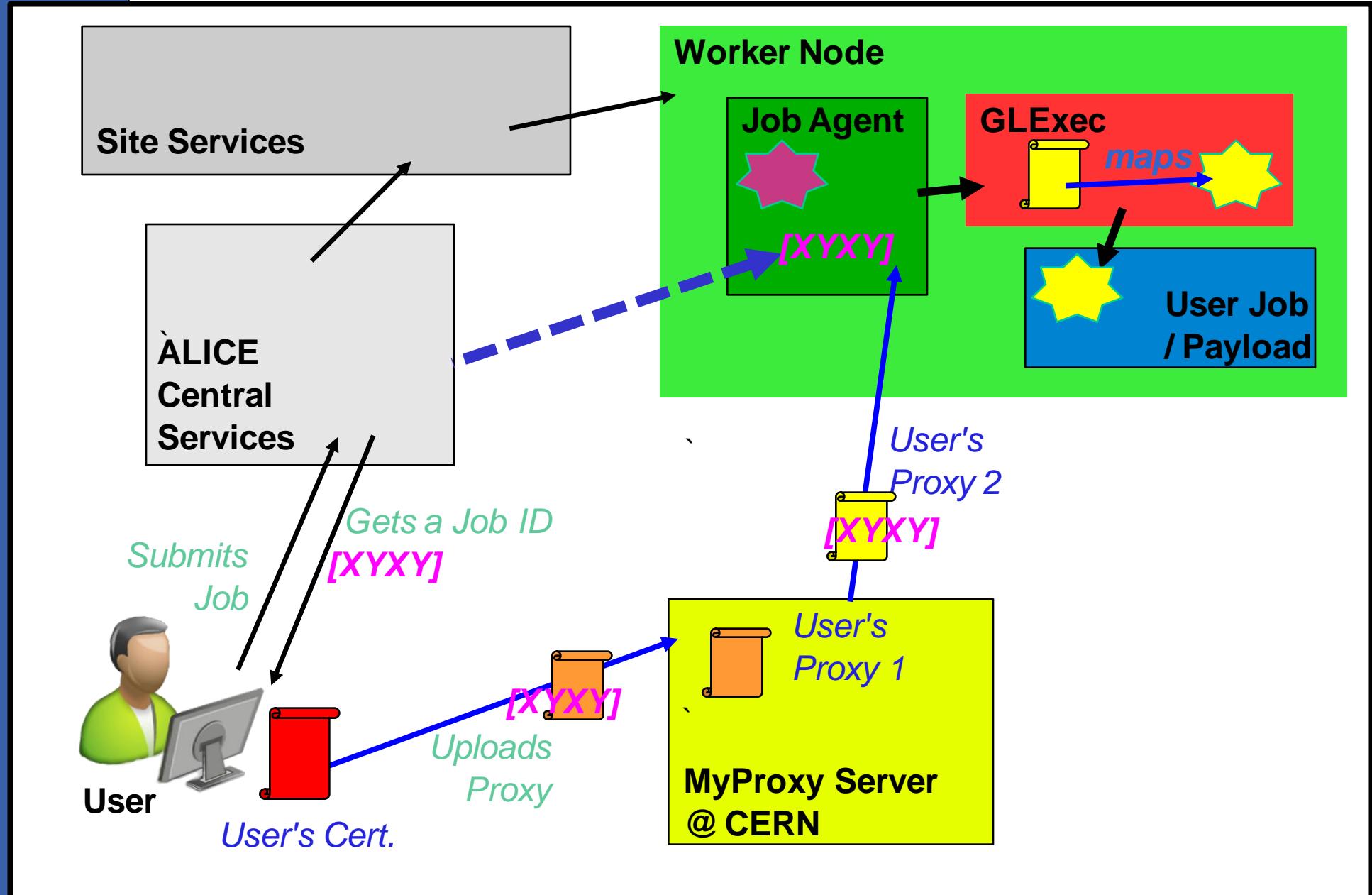
Speaking polemically, this sabotages the hole thing:

With the tokens you get user proxies, but they are maybe not only used for ALICE! Bad,bad,bad!

Let's forget MyProxy and store directly the proxies in the Central Services!

Why all the pain. Why we need the user switch then at all? Forget about it and let's just run the jobs by the VO. Don't we have that already?

# Outline for a Better Idea





# Implementation steps



## 1. The JA needs to be able to call and deal with gLExec.

--> Make gLExec use the pilot job's proxy (already there) to run jobs.

This is basically nonsens but a logical first implementation step.

Once we are there ...

2. The user's proxy needs to be uploaded to MyProxy.
3. Introduce the permission for the Worker Node to download one user's proxy for a job.



# Implementation – Step 1



## AliEn JobAgent needs to ...

- I. set up environment variables and proxy location, while provisionally using the pilots proxy cert.
- II. set up sub directory for gexec user sandbox
- III. move job data into the sub directory
- IV. call gLExec with the prepared arguments  
    << let gLExec run the user job >>
- V. clean up sub directory after job execution



# Thanks for your attention!