

POD

PROOF on Demand
ПРОВОДОНДМЕНД

Anar Manafov, GSI Darmstadt

2010-Oct

PoD on AliEn

Before we start, we have to agree, that

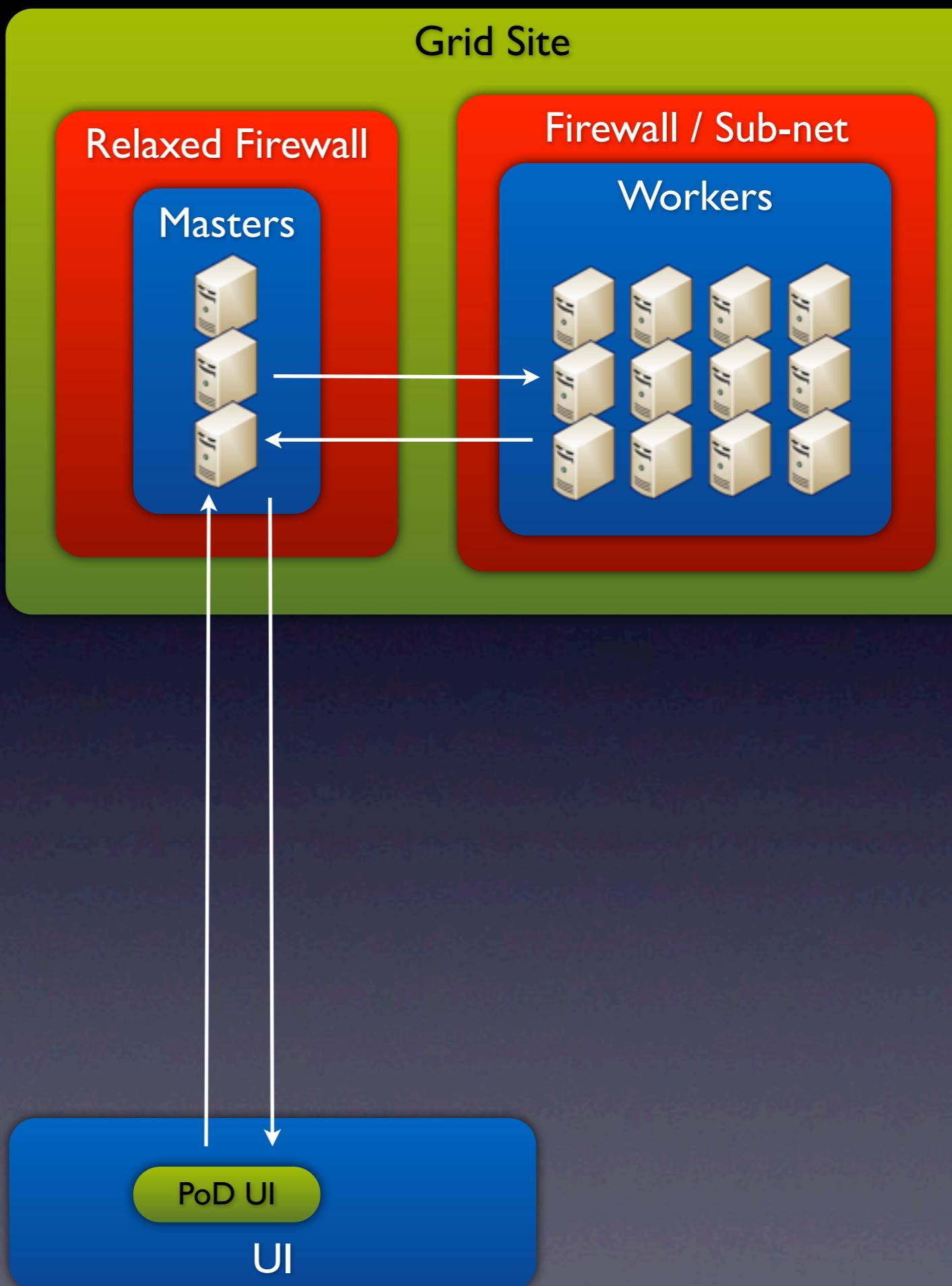
- PoD on AliEn should provide private PROOF clusters.
PROOF is not yet ready to scale for more than ~200WNs.
Additionally, why would we want to have an AliEn PROOF cluster admin? ;)
- AliEn must not know about PoD.
It should just handle special (high priority) jobs. We want a general solution, in case if something other, than PoD comes to the market.
- Running on the Grid may have a job start-up time penalty.
If a user doesn't have a local cluster to run on, than he/she should except this penalty. But the start-up time, must be in a matter of minutes (max) in anyway.

1. Submit a PoD server to the master queue,
2. the PoD server sends info back to PoD UI,
3. submit a job with PoD Workers,
4. WNs connect to the PoD server.

PROOF is ready!

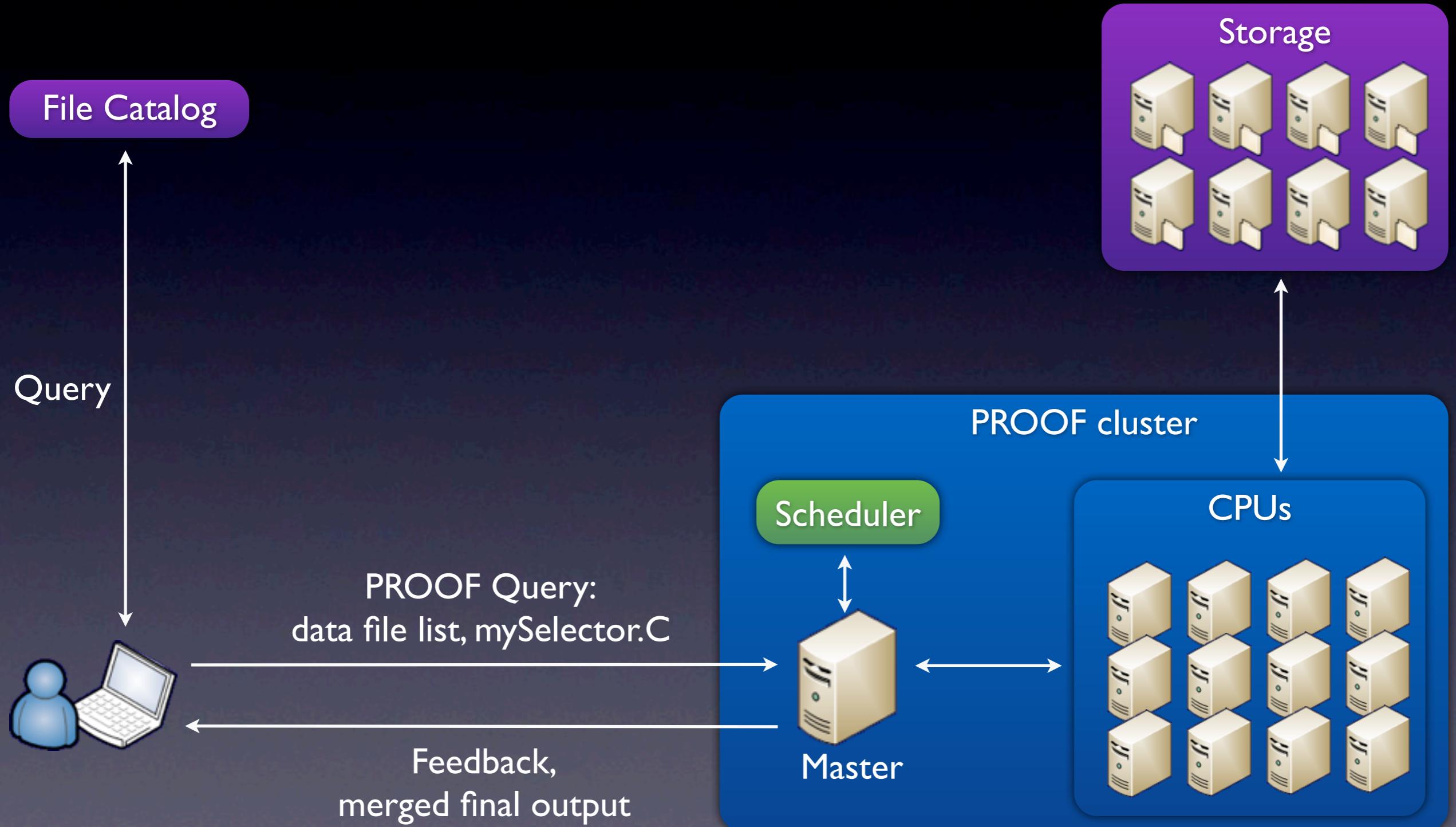
Requirements

1. a dedicated queue for masters,
2. the maximum priority for this kind of jobs,
3. TCP Ports (small ranges):
 1. WNs: incoming from Masters (required only for native PROOF connections, otherwise PoD will use packet-forwarding),
 2. WNs: outgoing to Masters,
 3. Ms: incoming from WNs,
 4. Ms: outgoing to UI,
 5. Ms: incoming from UI.



Back up slides

PROOF



What is PoD?

PROOF on Demand (PoD) is a tool-set, which sets up a PROOF cluster on any resource management system.

PoD is NOT a substitution of PROOF!
It is rather a helper tool for PROOF.

“static”/pre-installed PROOF cluster

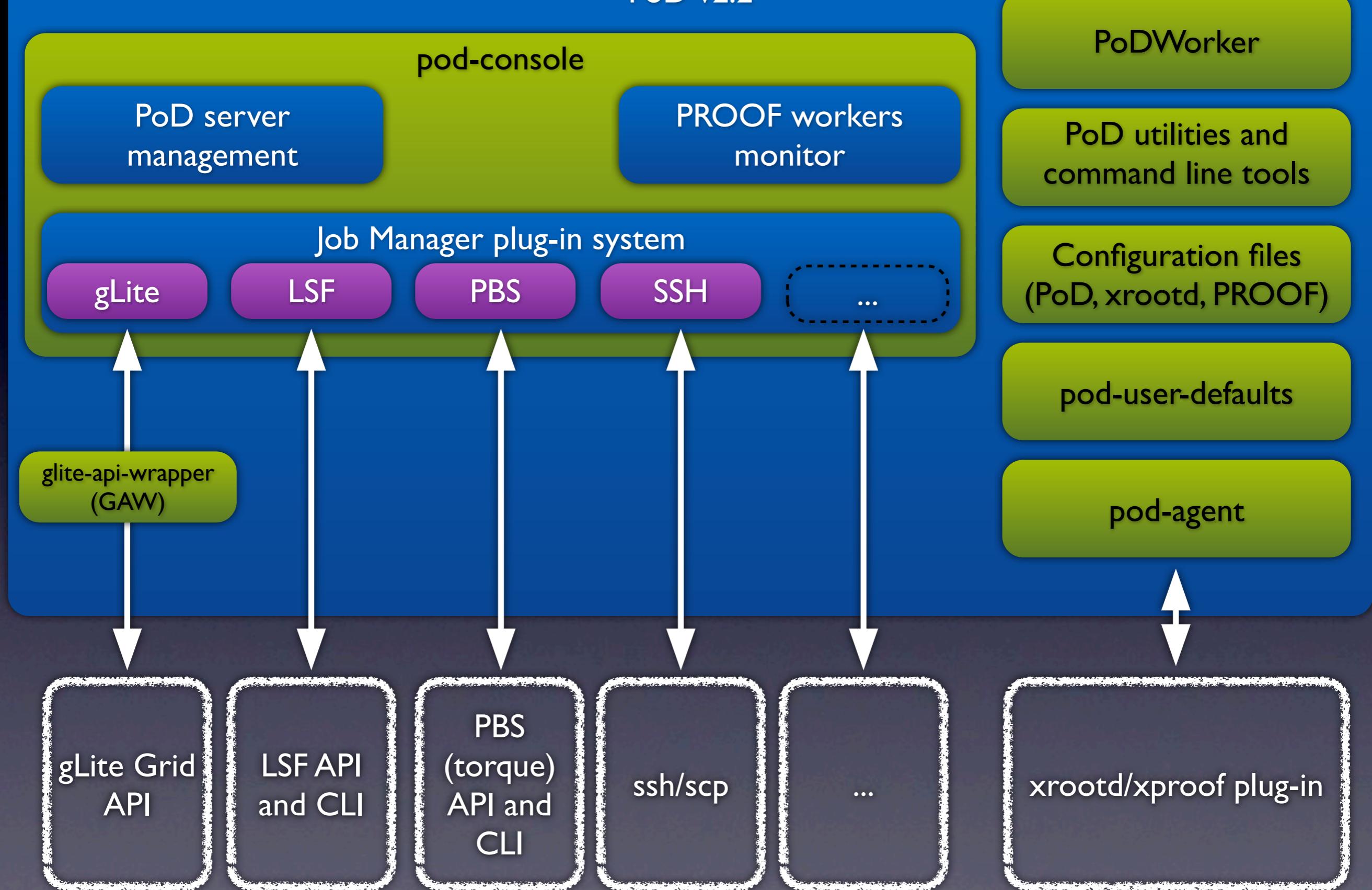
- One user can disturb other users.
- From time to time admin. interventions are needed.
- There is only one ROOT/xrootd version for PROOF services.
- There is a master node limitation.

PROOF via PoD

User

- can entirely control his/her dynamic cluster,
- can setup and use it on demand,
- can dynamically change an amount of workers,
- can select a preferable master host,
- doesn't need admins to take an action,
- doesn't disturb other users,
- is free to choose a ROOT version for services.

PoD v2.2

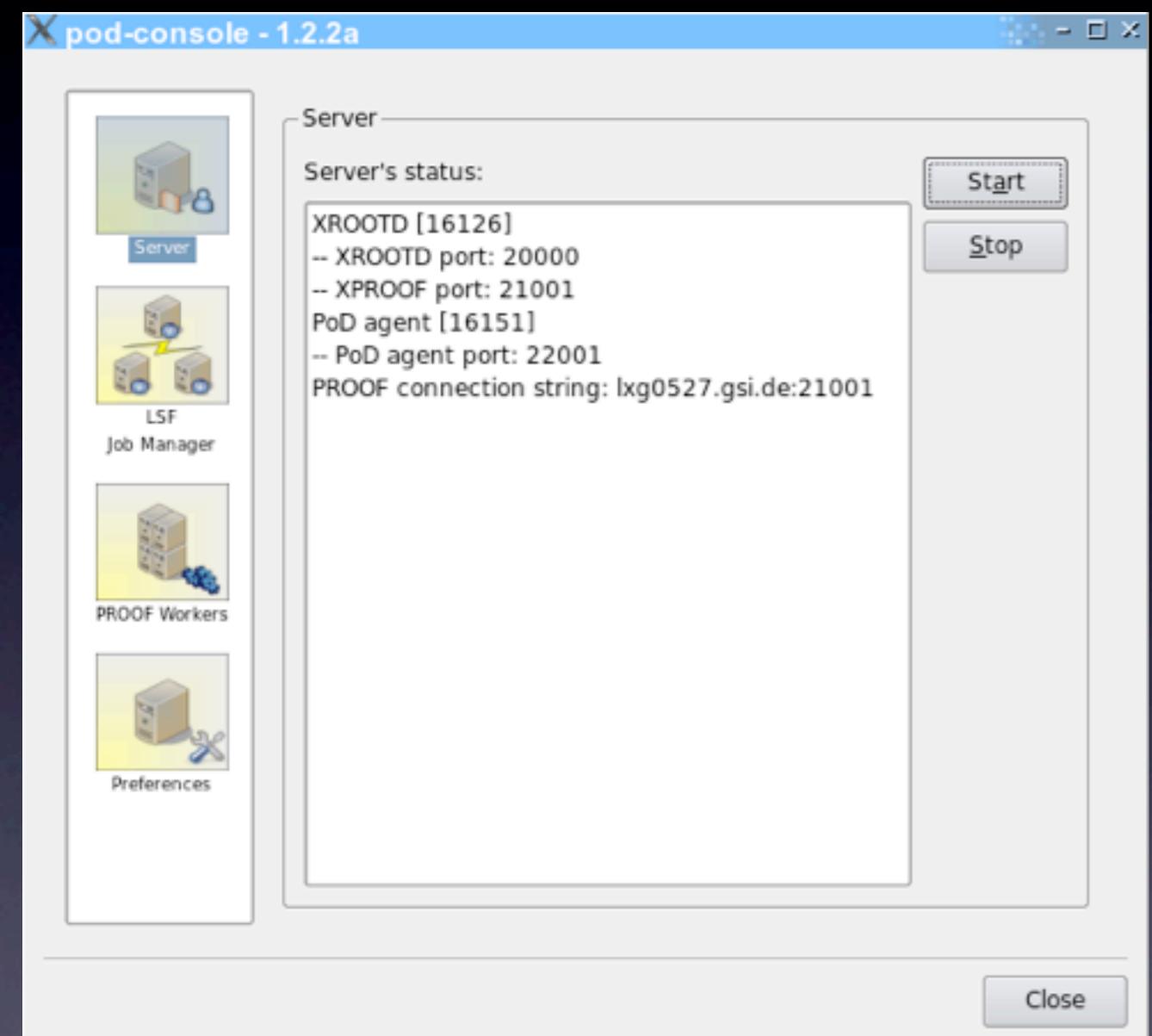


Key features

- Easy to use
- GUI & Command-line
- Different job managers (gLITE, LSF, PBS, SSH)
- Multiuser/-core environment
- Native PROOF connections
- Packet-forwarding
- User defaults - configuration

3 steps to set your private PROOF cluster up

PoD server



3 steps to set your private PROOF cluster up

PoD server



Job Manager (gLITE, PBS, LSF, SSH)

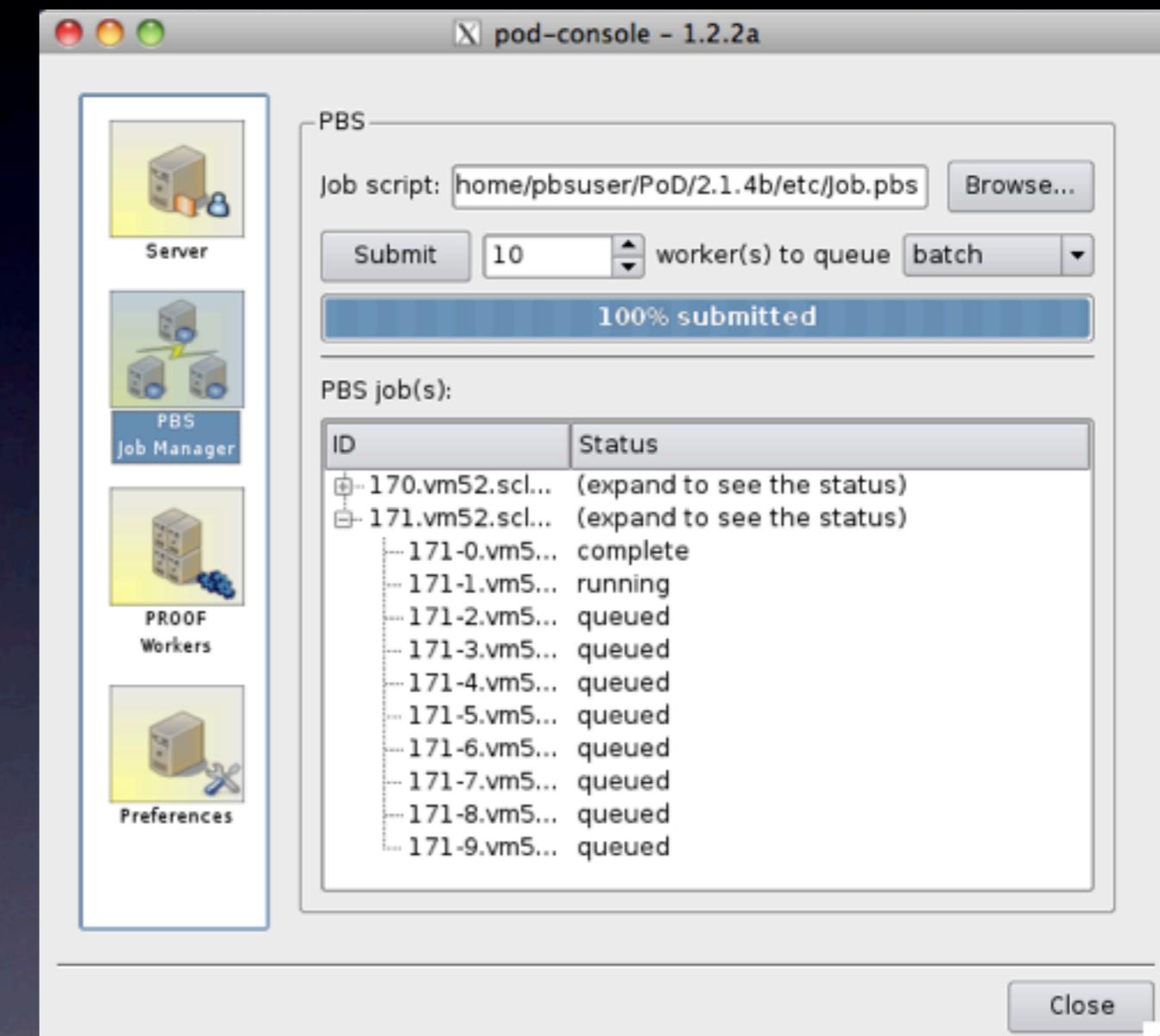
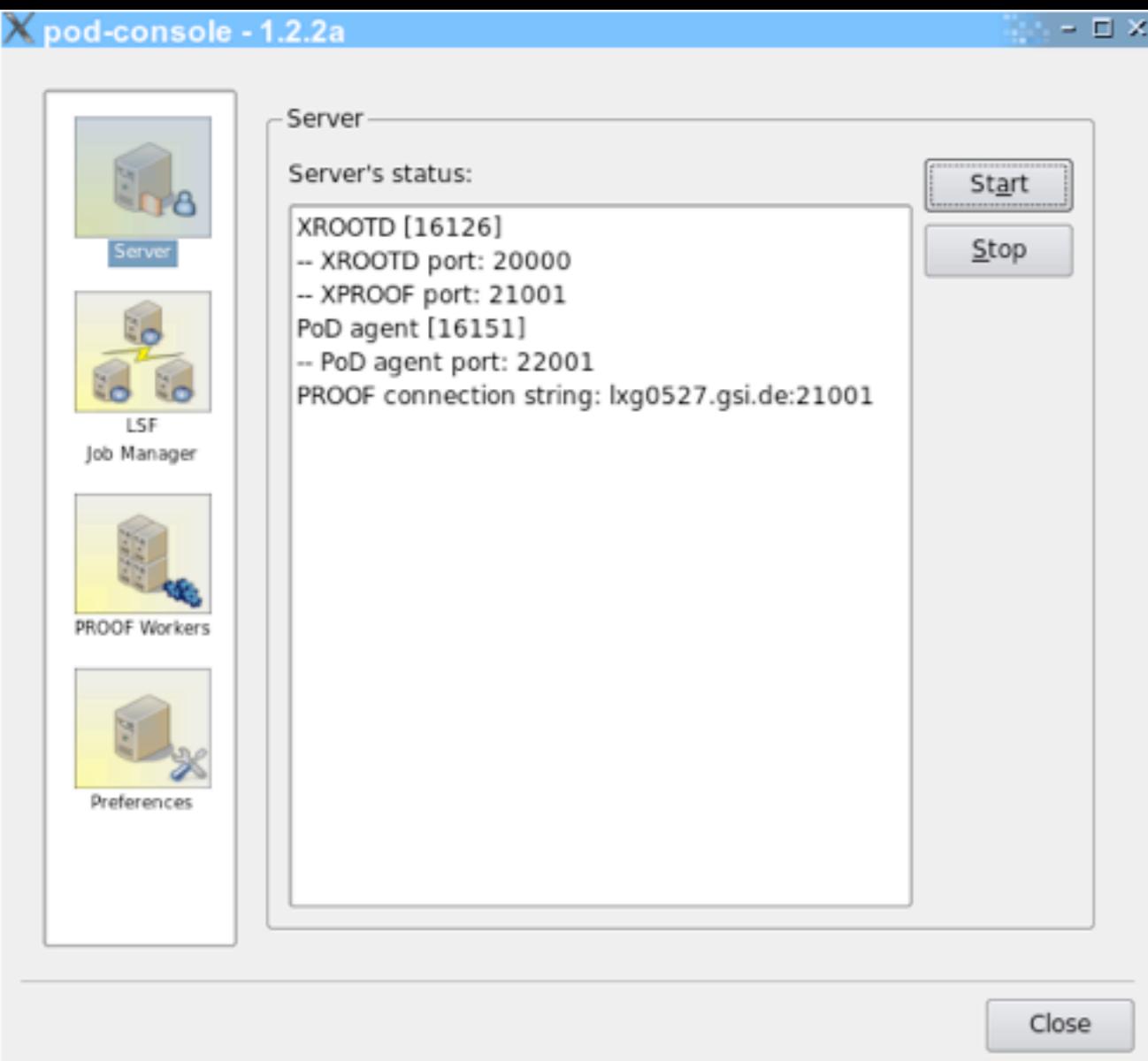
The image shows two windows side-by-side. The left window is titled "pod-console - 1.2.2a" and contains a sidebar with icons for Server, LSF Job Manager, PROOF Workers, and Preferences. The main area displays server status: XROOTD [16126] (port 20000), XPROOF port 21001, PoD agent [16151] (port 22001), and a PROOF connection string: lxg0527.gsi.de:21001. It features "Start" and "Stop" buttons. The right window is titled "PROOFAgent Management Console - v 1.0.5" and shows a sidebar with icons for Server, gLite Job Manager, Workers, and Preferences. The main area is titled "Grid" and includes fields for "JDL File" (set to PROOFpackage/etc/gLitePROOF_FZK1.jdl) and "Endpoint" (set to https://dgrid-rb.fzk.de:7443/glite_wms_wmproxy). It has a "Submit" button, a dropdown for "worker(s)" (set to 4), and a progress bar showing "0% submitted". A table below lists submitted jobs with IDs like https://dgrid-rb.fzk.de:9000/h-T3LW... and https://dgrid-rb.fzk.de:9000/nGV..., all marked as "Running". A context menu is open over one of the job entries, listing options: Copy JobID (Ctrl+C), Get output (Ctrl+O), Get logging info (Ctrl+L), Remove Job (Ctrl+R), and Cancel Job (Ctrl+E).

3 steps to set your private PROOF cluster up

PoD server



Job Manager (gLITE, PBS, LSF, SSH)

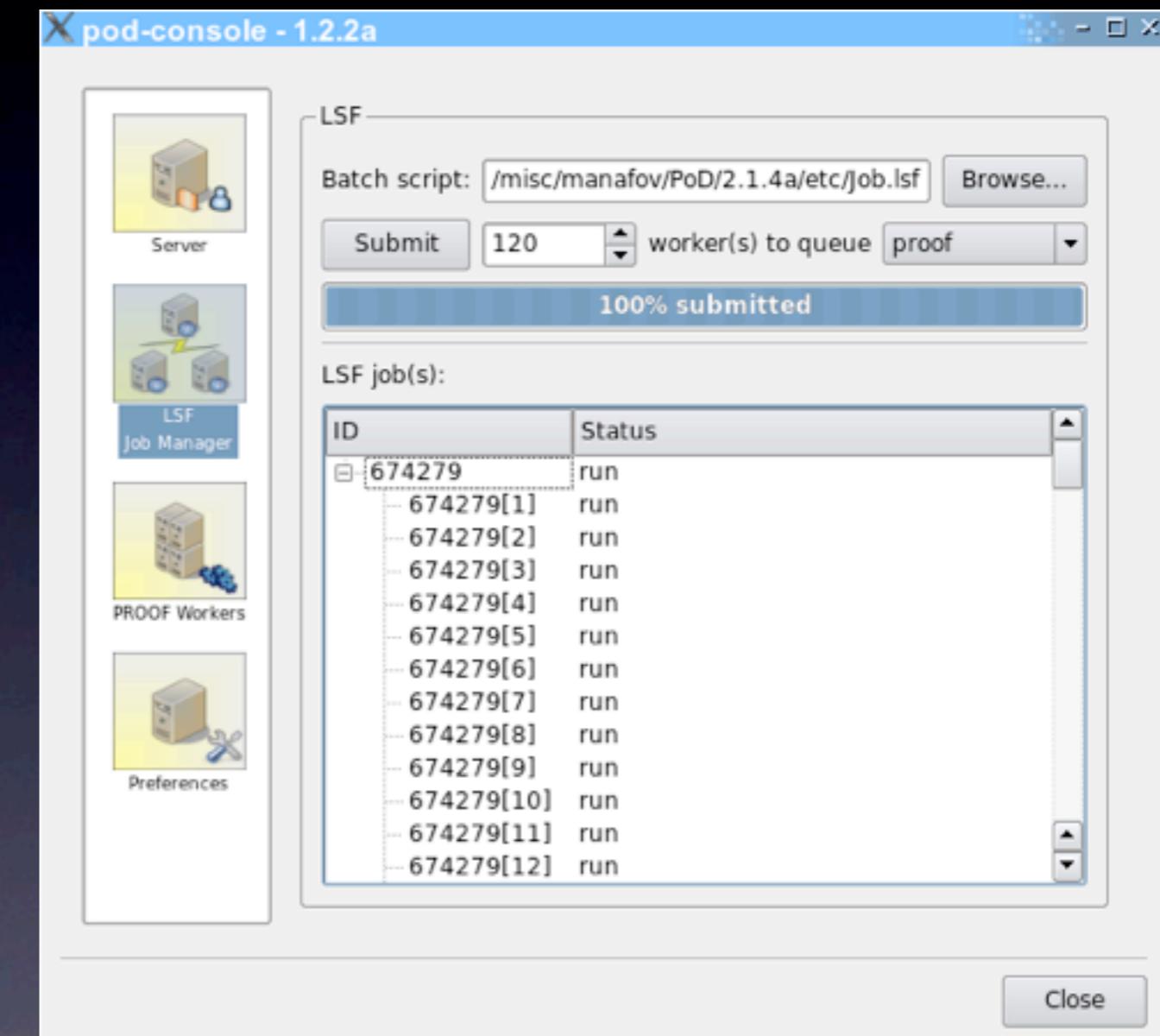
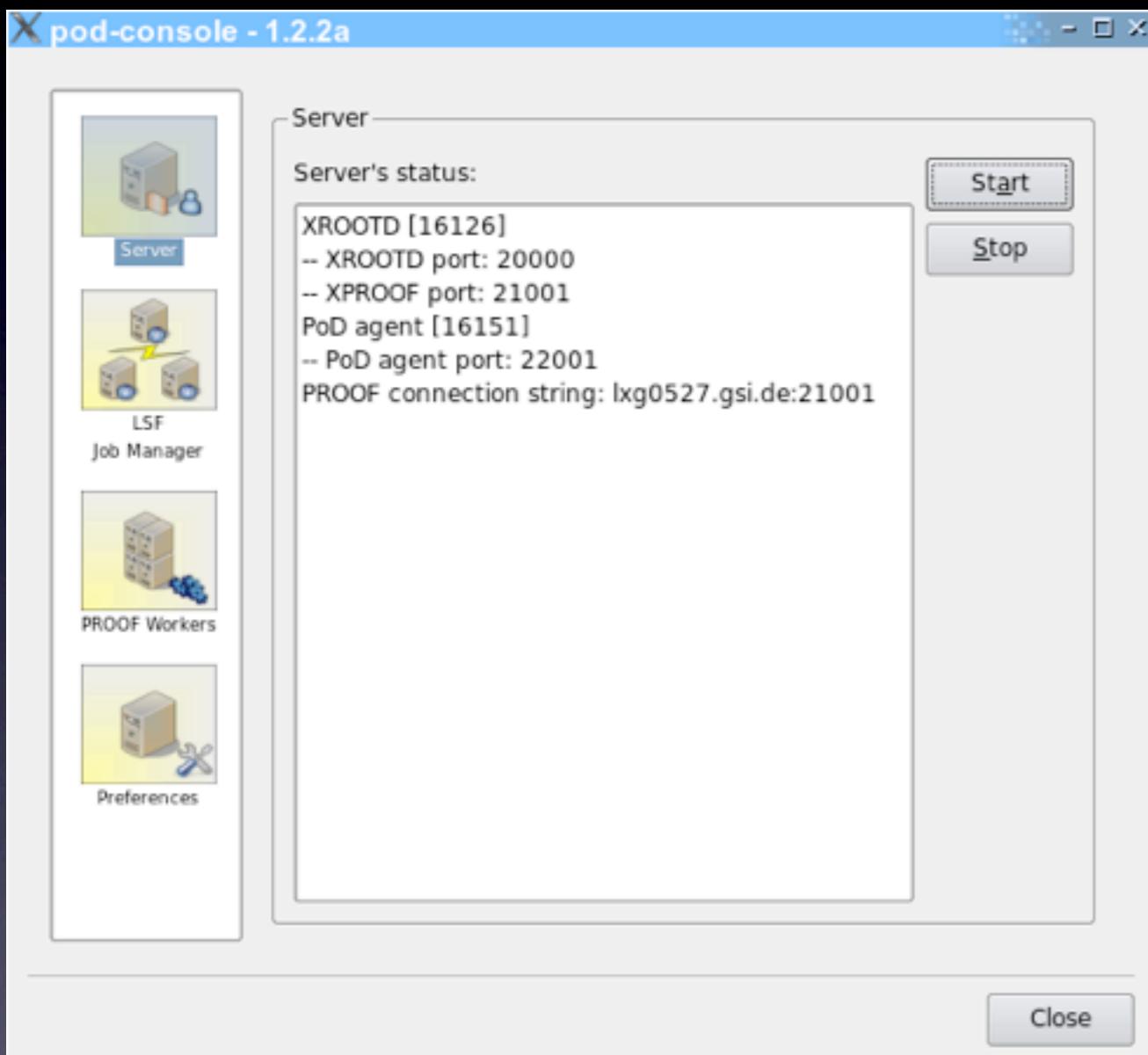


3 steps to set your private PROOF cluster up

PoD server



Job Manager (gLITE, PBS, LSF, SSH)



3 steps to set your private PROOF cluster up

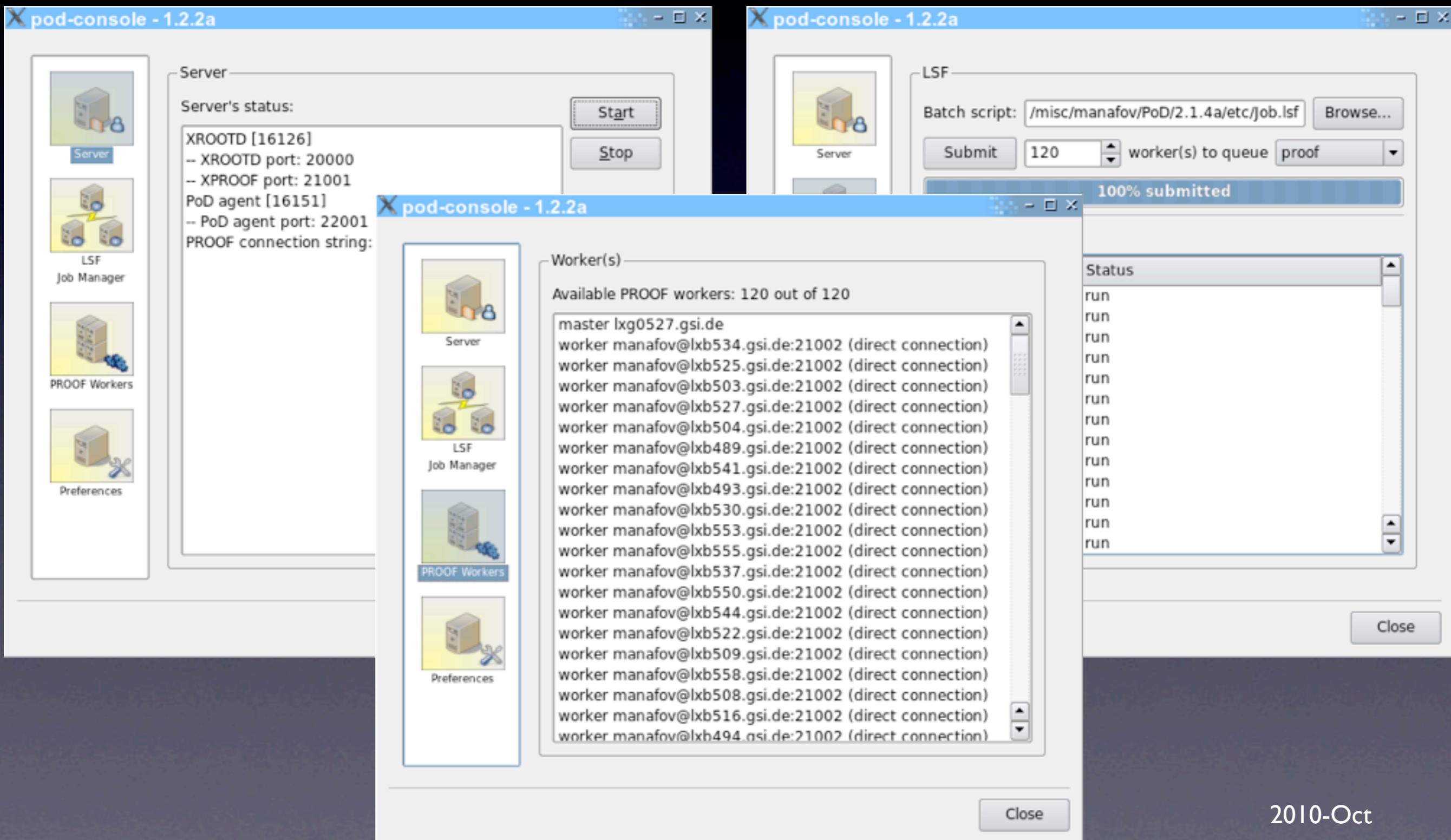
PoD server



Job Manager (gLITE, PBS, LSF, SSH)



your
PROOF
cluster



CLI

- pod-server
- pod-info
- pod-submit
- pod-user-defaults
- pod-prep-worker

CLI

- pod-server
- pod-info
- pod-submit
- pod-user-defaults
- pod-prep-worker

Example

```
$ pod-server start (status)
$ pod-submit -r lsf -q queue -n 150
$ pod-info -n (-l)
$ pod-server stop
```

CLI

- pod-server
- pod-info
- pod-submit
- pod-user-defaults
- pod-prep-worker

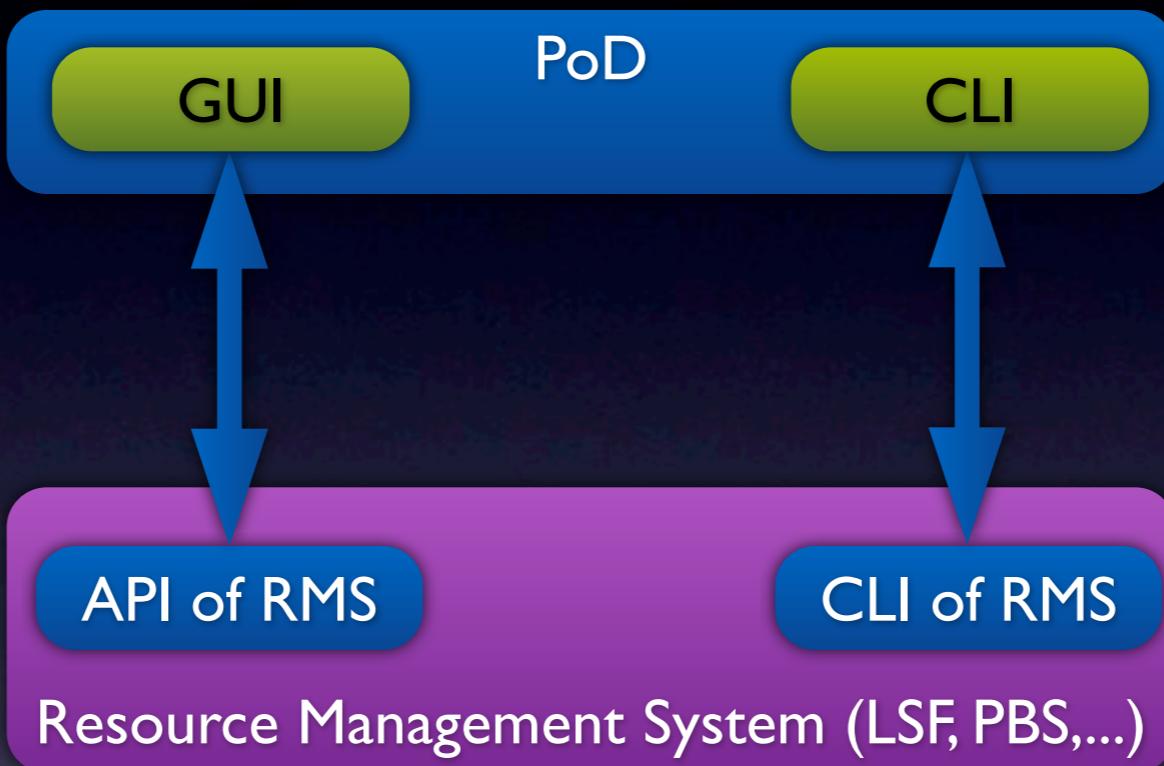
Example

```
$ pod-server start (status)
$ pod-submit -r lsf -q queue -n 150
$ pod-info -n (-l)
$ pod-server stop
```

can be used in a PROOF script

```
TProof::Open( gSystem->GetFromPipe("pod-info -c") )
```

GUI vs CLI

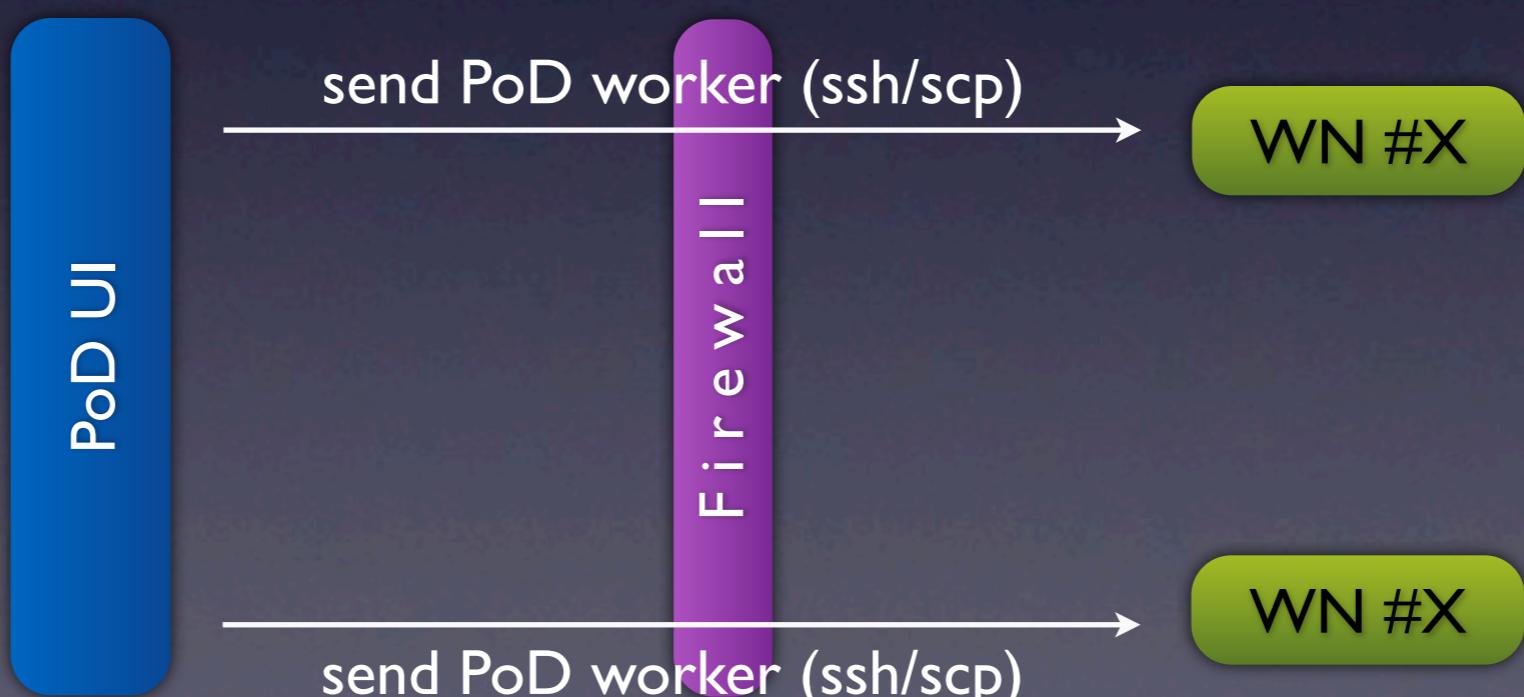


This strategy helps to be more flexible and better integrated into different setups and environments.

SSH plug-in

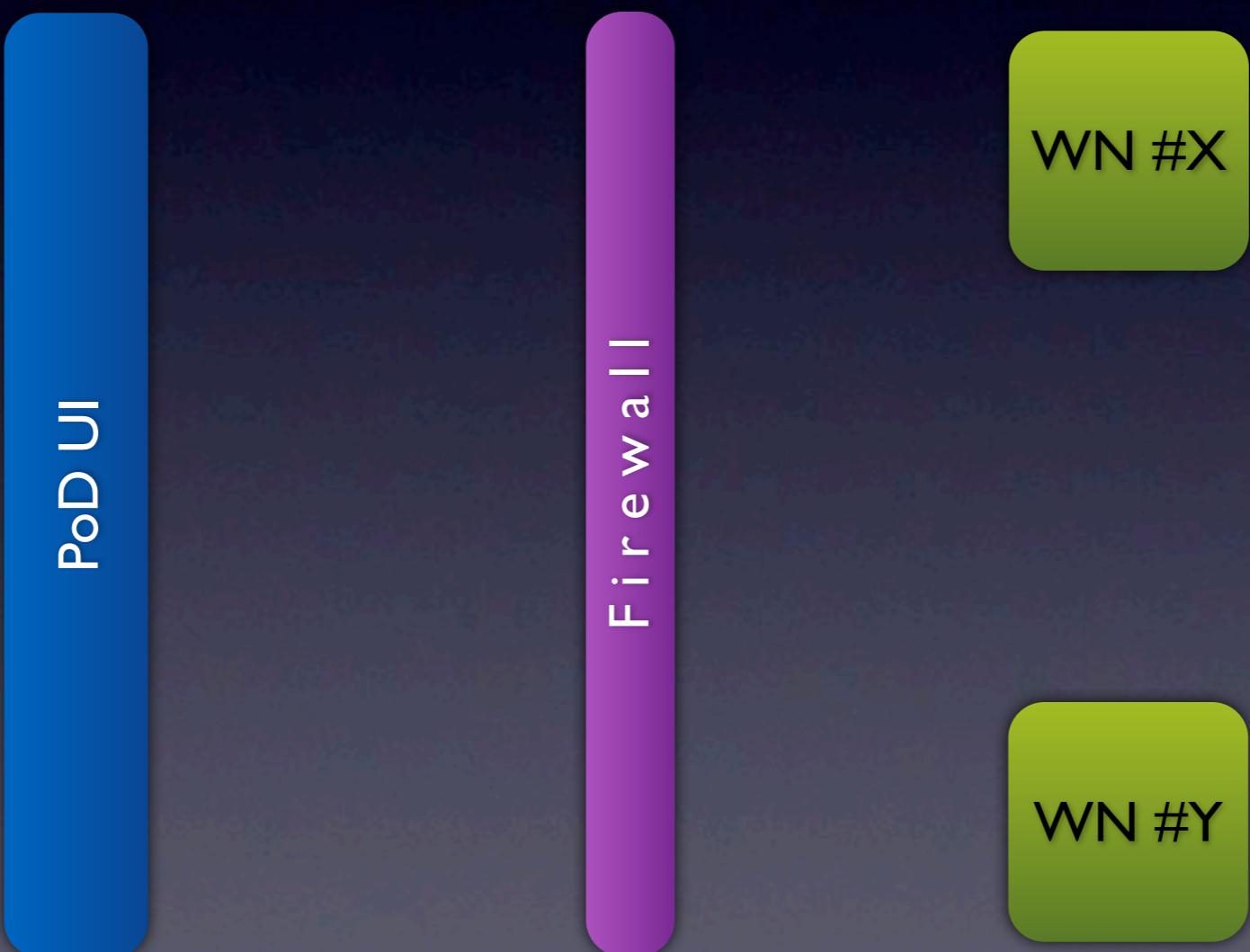
a simple CSV file as an input to the plug-in:

<i>id</i>	<i>login@host</i>	<i>ssh_params</i>	<i>wn_dir</i>	<i>num_of_workers</i>
r1	anar@lxg27.gsi.de	-p24	/tmp/test	10
a2	user@lx1i001	,	~/pod_wn	8
125	doom@host.my	, -p22	/opt/pod	16



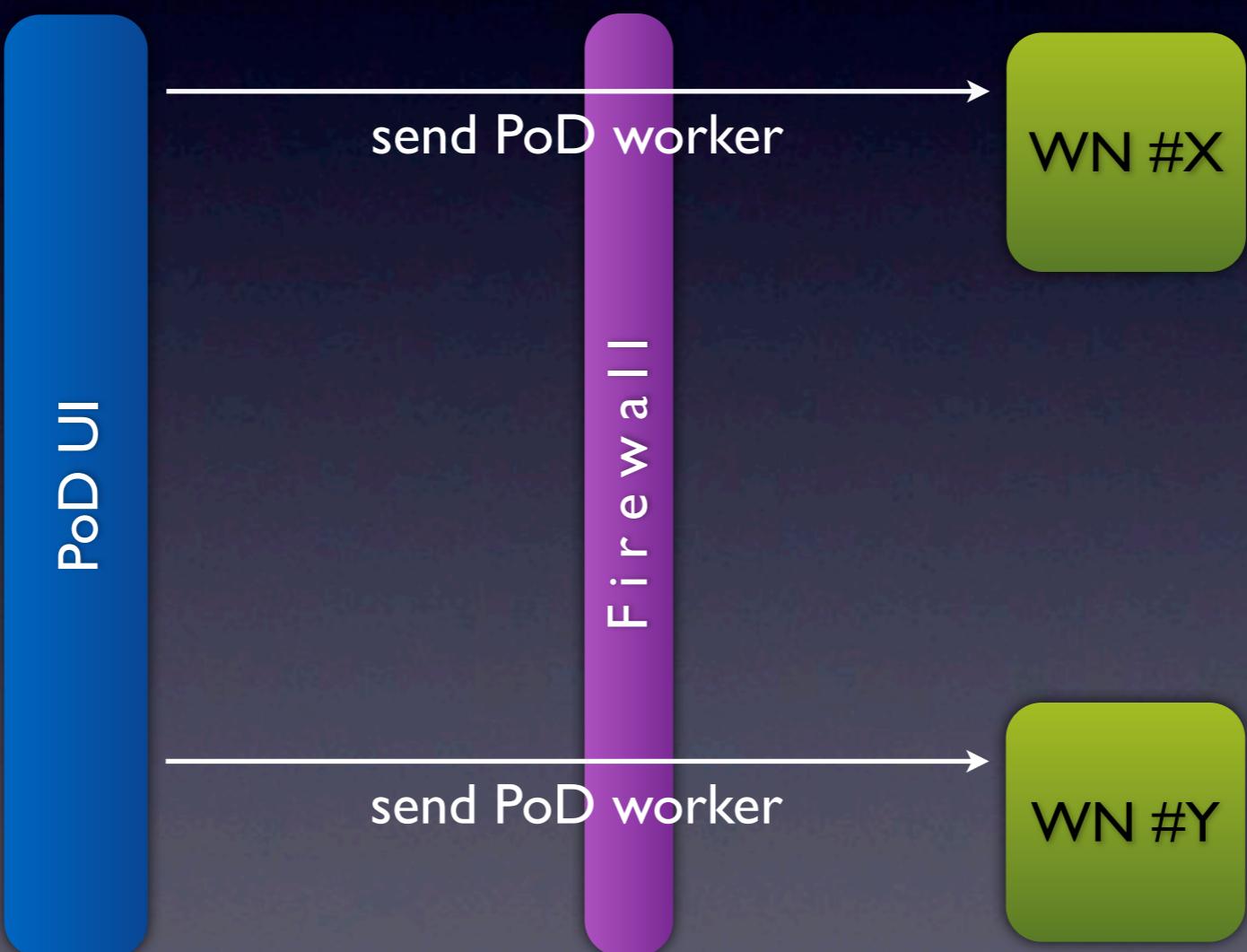
SSH plug-in & tunnel

PoD will optionally create ssh tunnels for nodes behind a firewall.



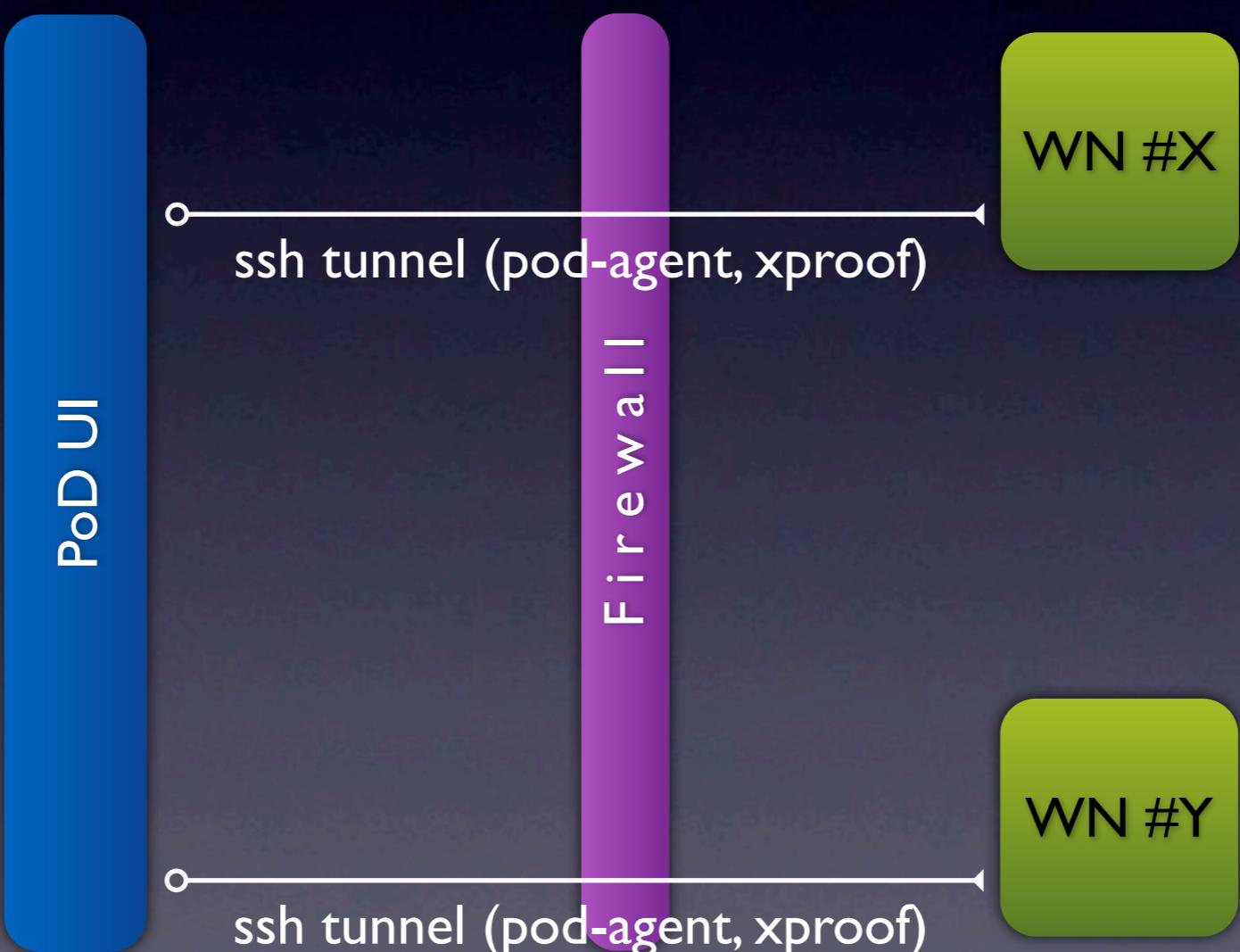
SSH plug-in & tunnel

PoD will optionally create ssh tunnels for nodes behind a firewall.



SSH plug-in & tunnel

PoD will optionally create ssh tunnels for nodes behind a firewall.



SSH plug-in & cloud

Possible workflow:

- distribute data files,
- prepare an OS image, which includes PoD's worker package (made by pod-prep-worker),
- send the image to a cloud provider, requesting an ssh access to nodes,
- pass the list of workers to PoD ssh plug-in.

Enjoy your cloud based PROOF cluster.

ToDo

- “out of server” UI,
- a native Mac OS X implementation of UI,
- an AliEn plug-in,
- an SGE/OGE plug-in.

<http://pod.gsi.de>