

# PoD

PROOF on Demand

PROOF on Demand

## PoD Development Status Report

# Propaganda

# 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

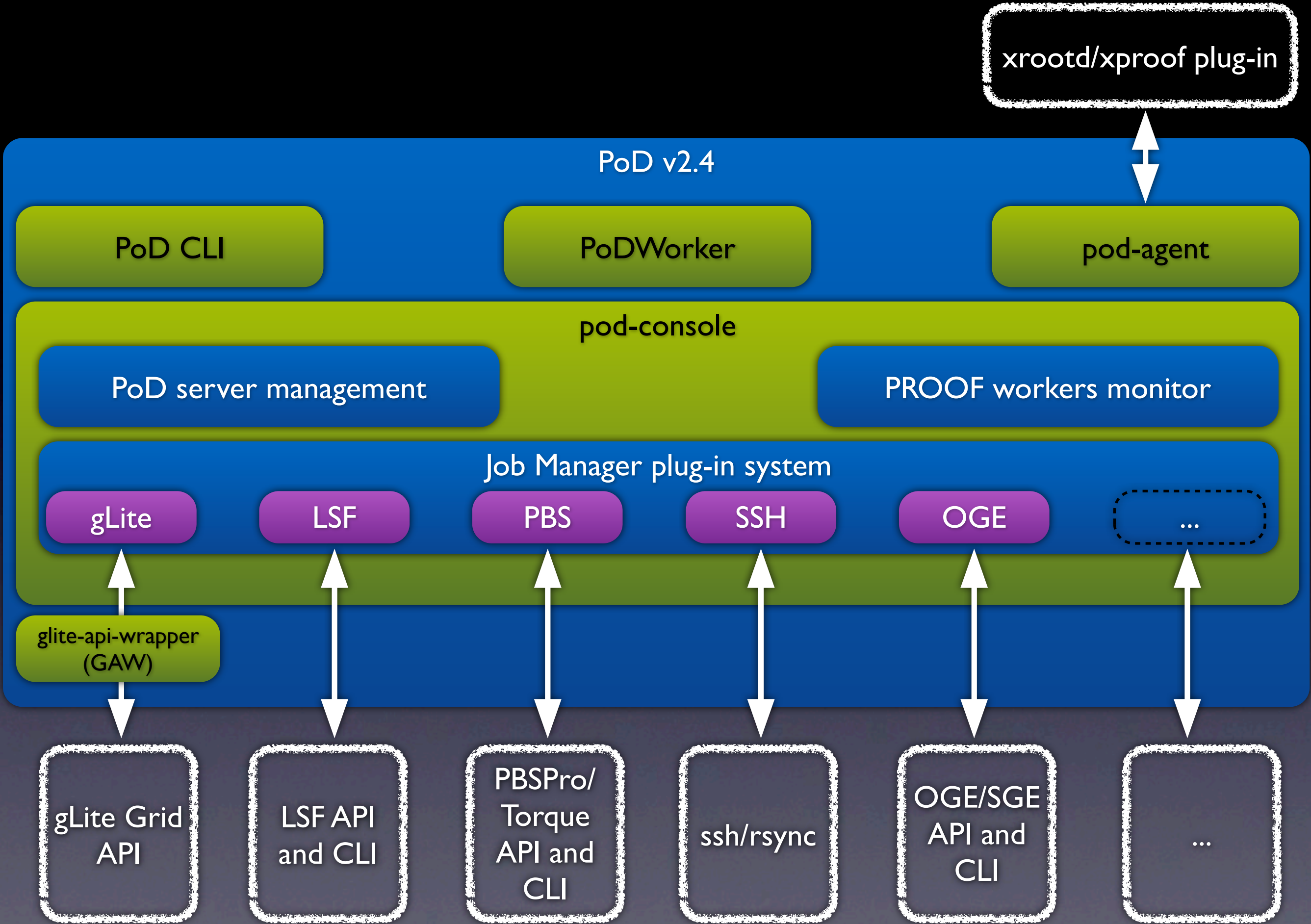
## User

- 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.

# Key features

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





# Current status

v2.2 was released (2010-09-17)

v2.3 was released (2010-10-03)

v2.4 was released (2010-11-14)

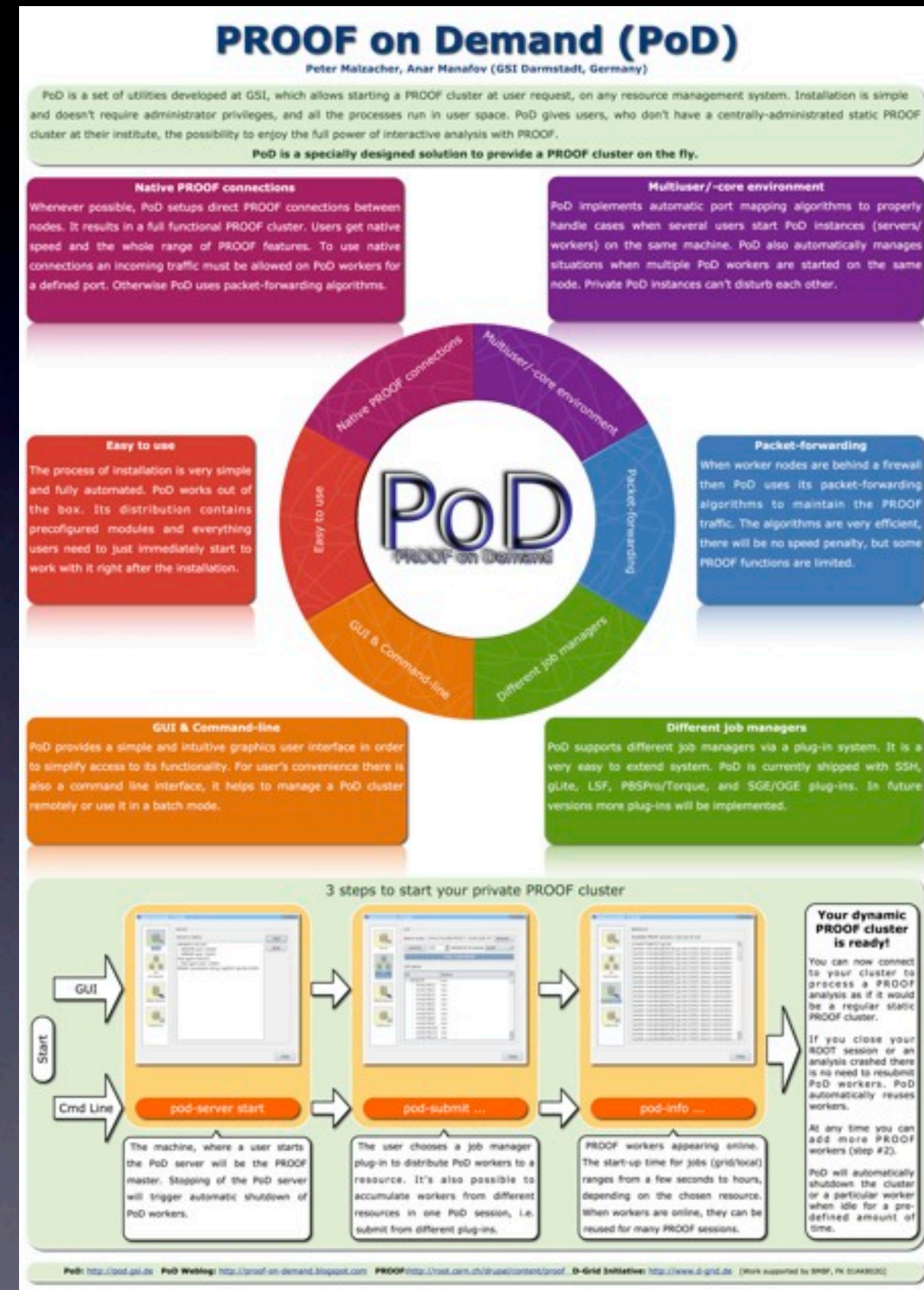
v3.x - is under development



# CHEP 2010 was a very successful event for PoD

PoD was a leader in its field :)

- *there was obvious attention on PoD,*
- *big interest in SSH, OGE and Condor plug-ins.*
- *I've shown more than 20 live demos,*
- *during CHEP's week PoD download counts raised,*
- *we got a number of new customers (not single users, but institutions),*
- *there were many new suggestions, feature requests.*



# 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@lxcg27.gsi.de	, -p24	, /tmp/test	, 10
a2	, user@lxi001	,	, ~/pod_wn	, 8
125	, doom@host.my	, -p22	, /opt/pod	, 16

the plug-in is in PoD  
CLI only, so-far





# 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.



# OGE plug-in

- supports OGE v6.0 or higher,
- uses DRMAA v1,
- is implemented in PoD GUI (DRMAA),
- is implemented in PoD CLI (OGE CLI),
- supports so far only a shared home FS.

# PoD WNs

- Dependencies have been revised in order to lighten WNs.  
Removed dependencies on perl modules and python.
- Now PoD supports a custom user environment script.

# PoD WNs

- WNs don't download PoD pre-compiled binaries anymore.
- Automatically detect of a shared home FS.
- Improved start-up time of the workers.



# Mac OS X

PoD UI/Server supports Mac OS X

# xproofd

PoD uses now xproofd instead of xrootd.

It helps users to run PoD and xrootd (for data)  
on the same machines.

# PoD plug-in system

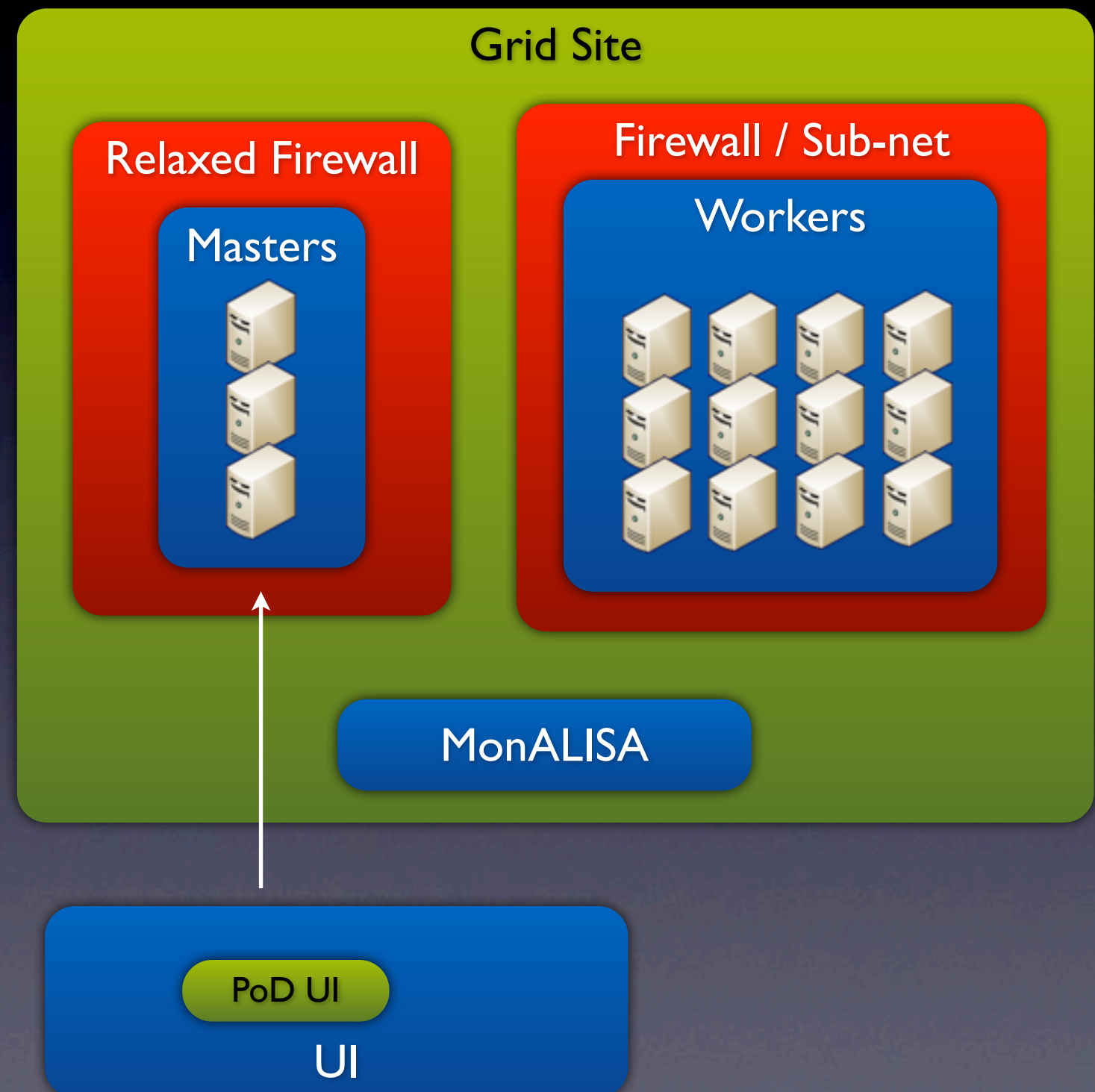
The CLI plug-in system was slightly updated in order to make it much easier to extend with new plug-ins and to support old ones.



# AliEn integration

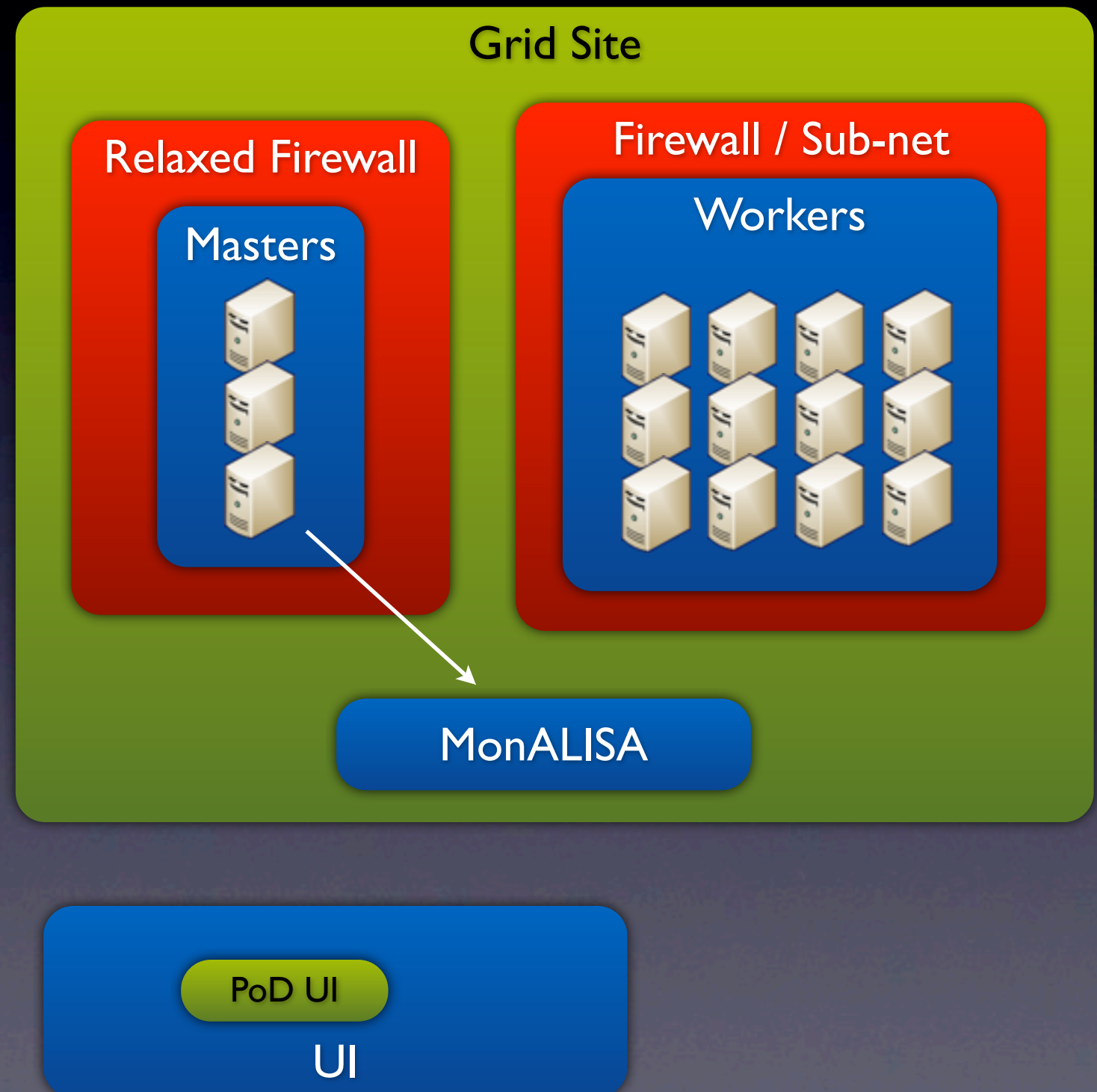
# Step #1 User request a PoD server

- User's request is only a configuration file for PoD server. No binaries are transferred.
- AliEn using master queue starts a PoD server on the PoD-aware master machines, using user's config file.
- User could specify, a minimum and a maximum number of workers and for how long (s)he can wait for the max. number of workers.



# Step #2 PoD server registers in MonALISA

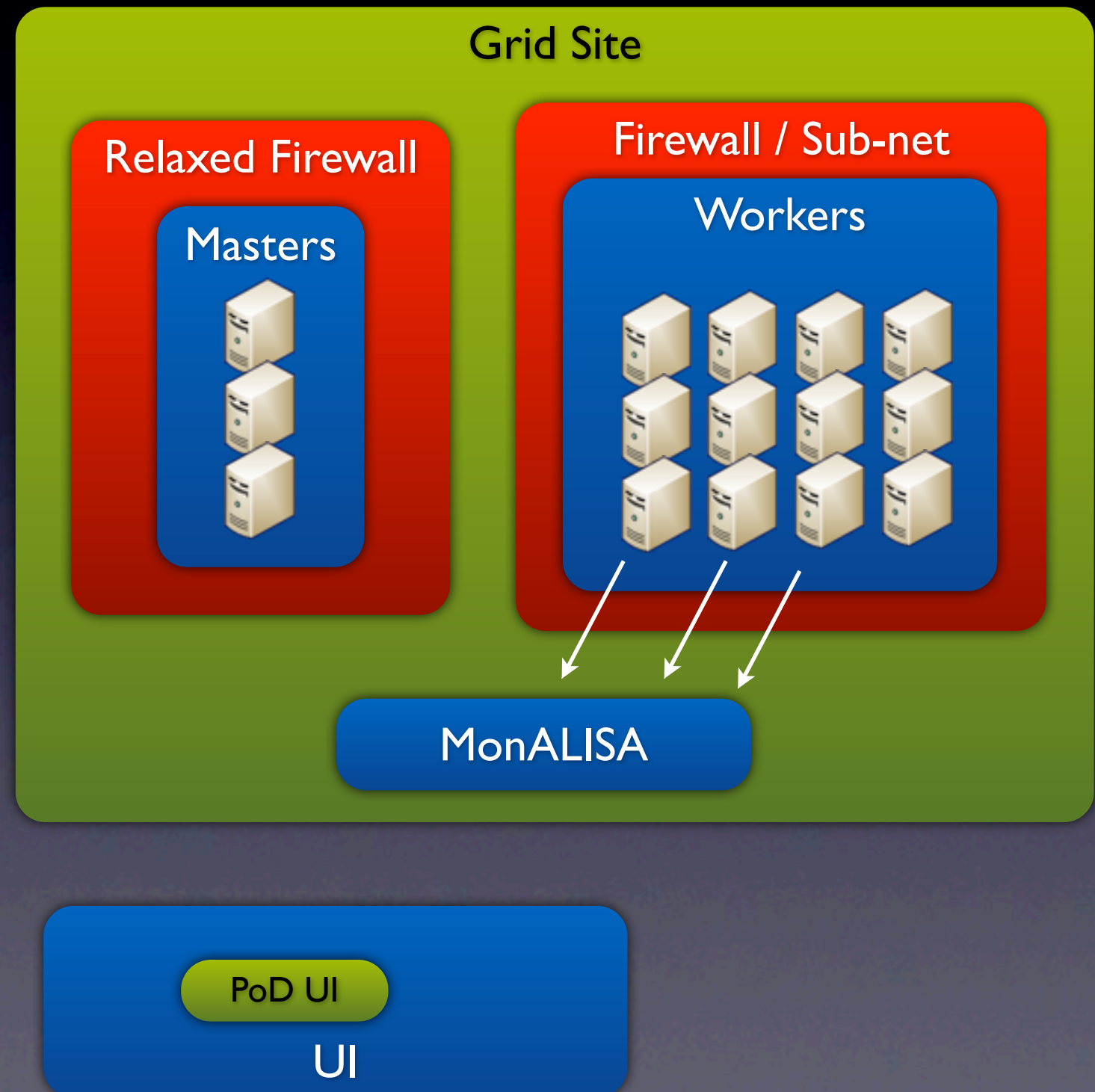
- Server sends info to MonALISA every X sec.
- Server stops sending info, if it:
  - got enough workers or
  - can't wait longer (as user requested) or
  - shut down.





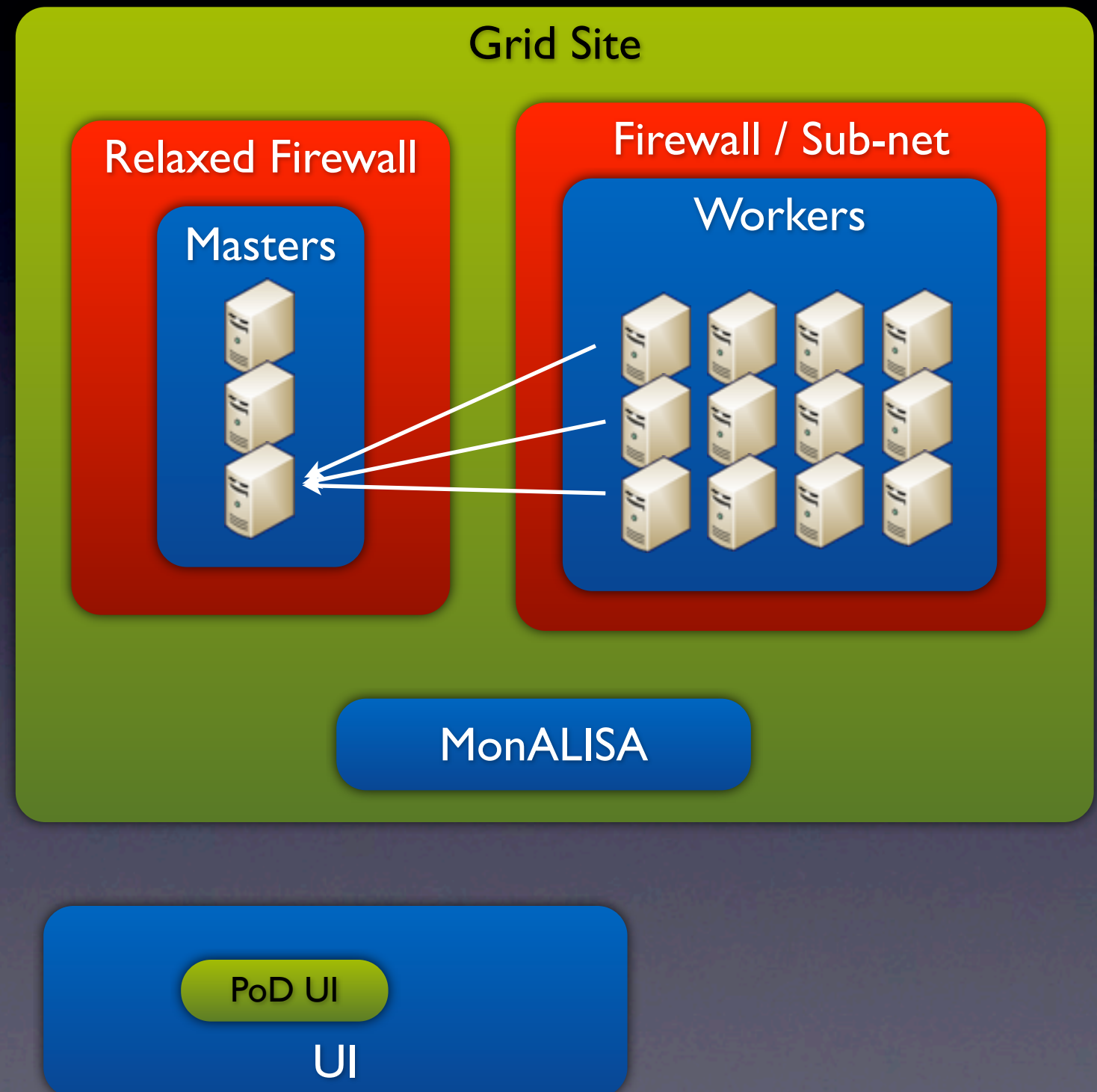
# Step #3 AliEn constantly starts PoD workers whenever there are free job slots

- This is a constant process.
- Before starting a regular job on a free slot, AliEn starts a PoD worker.
- PoD worker checks MonALISA for available PoD servers.
- If there is no servers, PoD worker exits after a define amount of time, so that AliEn job slot could take a normal job.



# Step #4 PoD workers connect to available PoD servers

- As soon as a PoD server got enough workers, it will stop publishing to MonALISA and will stop to except anymore workers.



# Step #5 PoD will process its usual routines in order to setup PROOF

## Site Requirements

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 on PROOF ports.



# Questions to AliEn

Server: Is it possible to fork PoD server under individual users accounts?

WN: Is it possible to start PoD processes under individual users accounts?

# AliEn integration: steps for PoD to accomplish

- implement a central installation feature,
- disentangle PoD UI and PoD Server,
- teach PoD WN and Server to understand PoD's user defaults configuration via PoD protocol,
- teach PoD Server and WNs to talk to MonALISA.

# Short-term ToDo

- a “central installation” feature,
- a Condor plug-in,
- PoD UI and Server disentangling.

<http://pod.gsi.de>