Open Science Grid

Introduction to the Grid and the glideinWMS architecture

Rob Quick <rquick@iu.edu>
OSG Operations Coordinator
Manager High Throughput Computing Group
Indiana University

Slides by Igor Sfiligoi

Open Science Grid

Logistical reminder

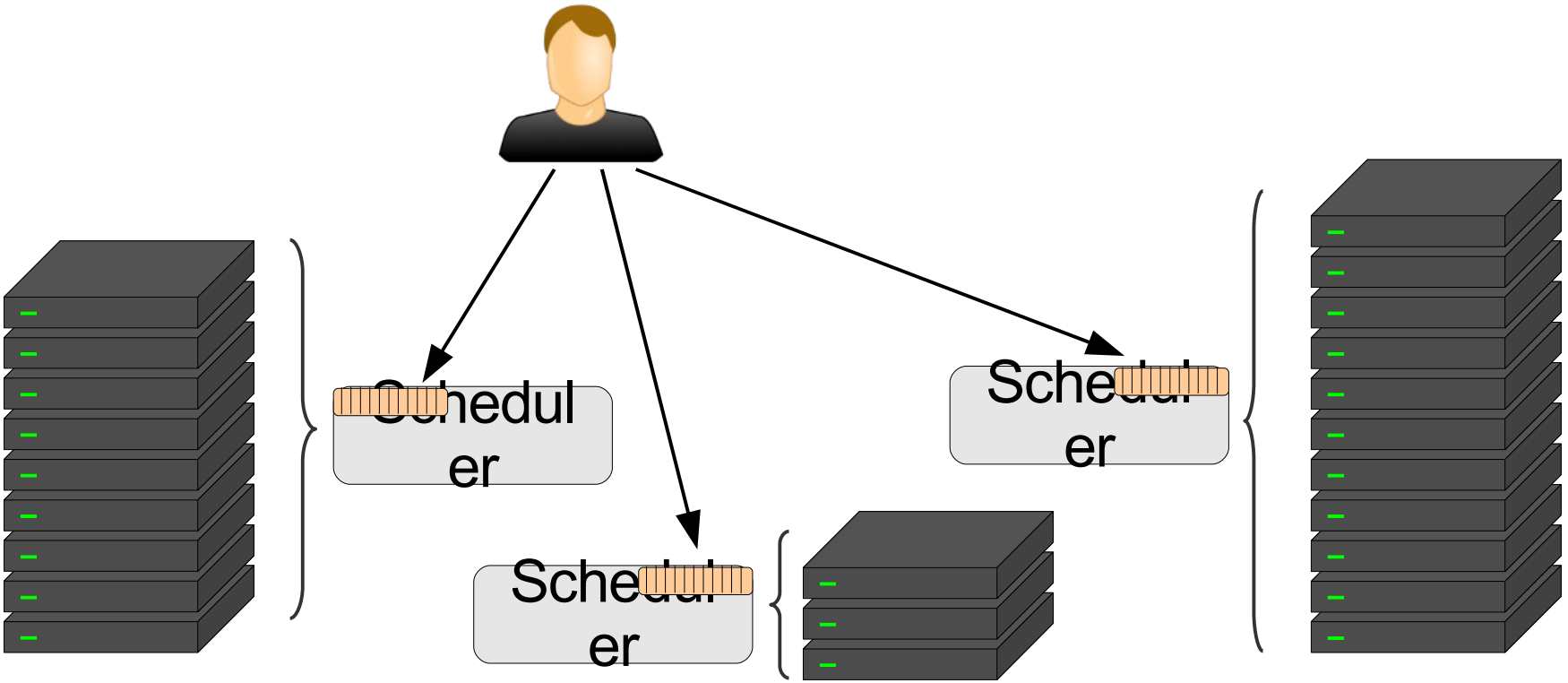
It is OK to ask questions

- During the lecture
- During the demos
- During the exercises
- During the breaks

If I don't know the answer,
I will find someone who likely does

Reminder - DHTC

DHTC is about computing on **more than one** HTC system



This lecture goes into details of DHTC

I just want to do my science.
I will leave the direct DHTC tasks to the Overlay admins.

You still should learn.

Good.
This is the spirit.

But then it is up to me to fix your screw ups!

Open Science Grid

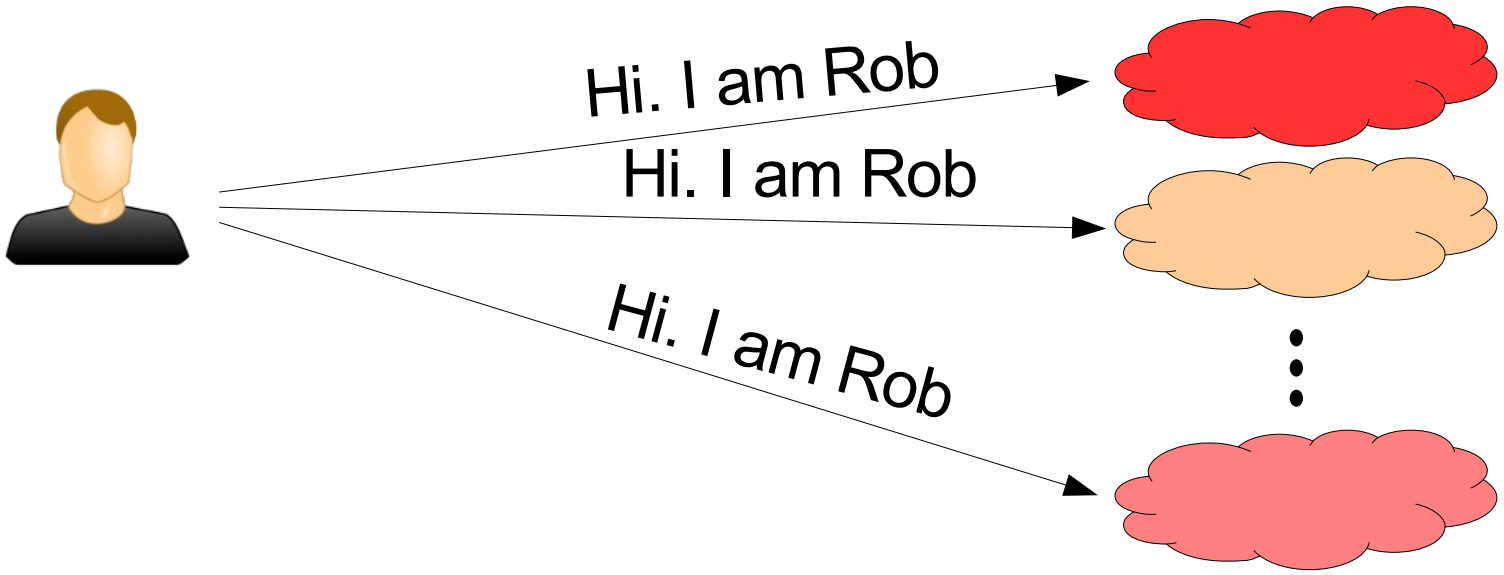
The Grid

- One instance of DHTC
- The idea behind the Grid is to provide a single interface to any HTC system
  - No matter where it is located
  - No matter who operates it
  - No matter what technology it uses
- Based on two principles
  - Single sign-on
  - An abstraction layer for job submission

Single sign-on

The idea is simple

The user should use the same mechanism to submit jobs to any site
(and there can be 100s of them!)



OSG uses Certificates

- Think of it as a passport
  - It is issued once to you
  - You present it for inspection when doing immigration
  - The immigration officer uses the information in the passport to let you in
- In OSG it is essentially a file

OSG uses Certificates

Think of it as a passport

It is issu[illegible] on

You [illegible]

In [illegible]

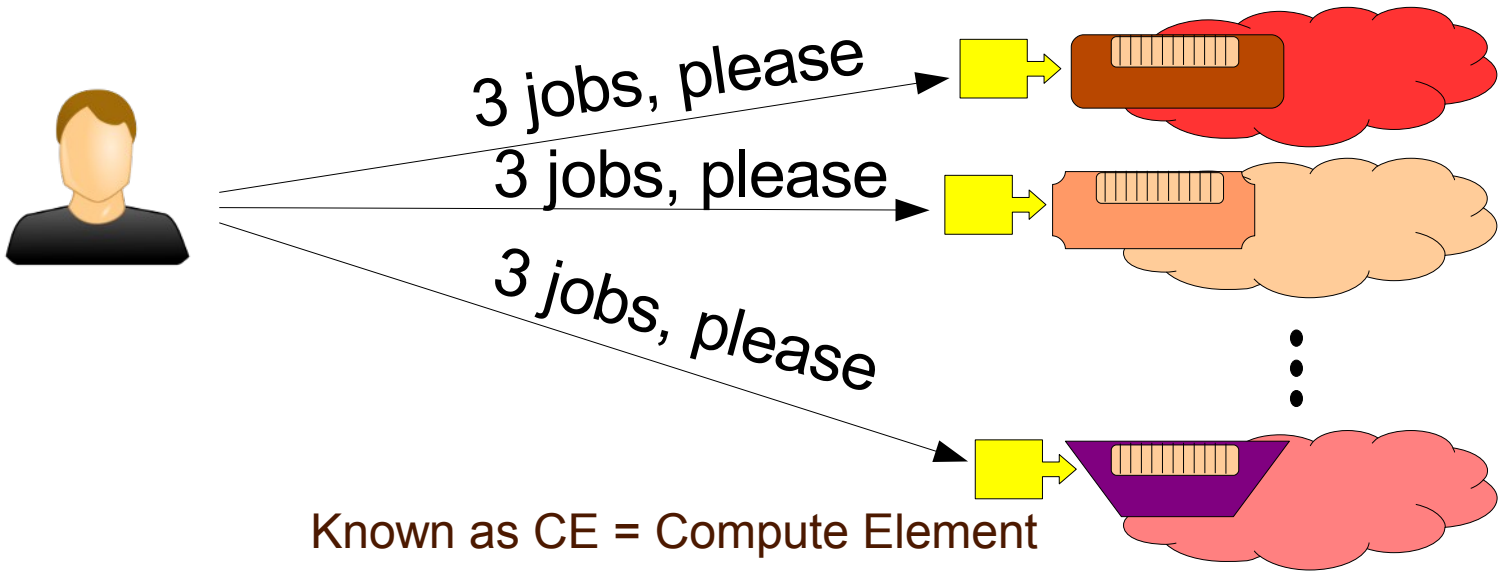
Make sure you get one today!
You will need it for the storage exercises later today.
(We have no Grid exercises)

More details in the afternoon

Abstraction layer for submission

Again, you want the same mechanism to submit jobs to any site

We put an abstraction layer between the user and the site-specific technology



Open Science Grid

Theory and practice

- In practice, no single abstraction layer
  - Several products: GRAM, CREAM, ARC
- Although OSG mostly uses GRAM
  - But even this is under discussion

Theory and practice

In practice, no single abstraction layer

- Several product[illegible] GRAM, CREAM, ARC

Although[illegible]

But e[illegible]

Need a flexible submission tool

Open Science Grid

Enter Condor-G

- Condor happens to be the best, most flexible submit tool
  - Indeed, the recommended tool in OSG
- Condor-G is just a name for the components handling “Grid universe” jobs
  - You would still be using condor_submit and condor_q

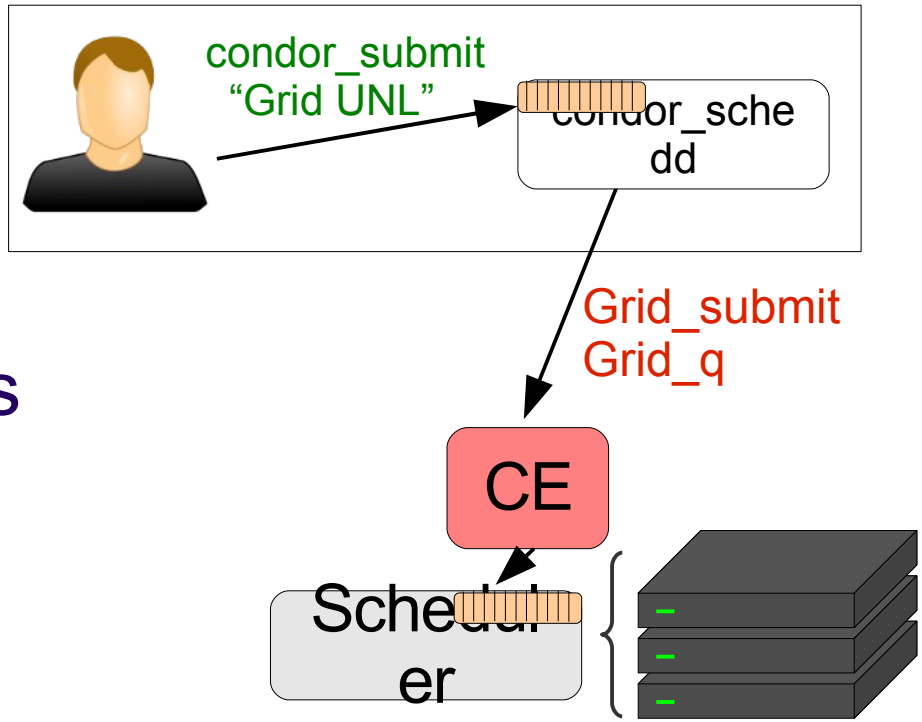
Condor-G details

Condor-G doesn't manage remote resources

- It just forwards and monitors jobs sent to remote HTC systems
- No condor_status

No matchmaking

- User explicitly specifies API and site to use



CE as a black box

- Practically all CE implementations provide only minimal functionality
  - Job submission
  - Basic job monitoring
  - Job removal
- If anything goes wrong, very hard to discover the core reason
  - Not always, but way too often
  - Requires contact with the remote admins

CE as a black box

Practically all CE implementations provide only mi[illegible]

- Job [illegible]
- [illegible]
- [illegible]
- very [illegible]son
- Not alwa[illegible]
- Requires contact with the remote admins

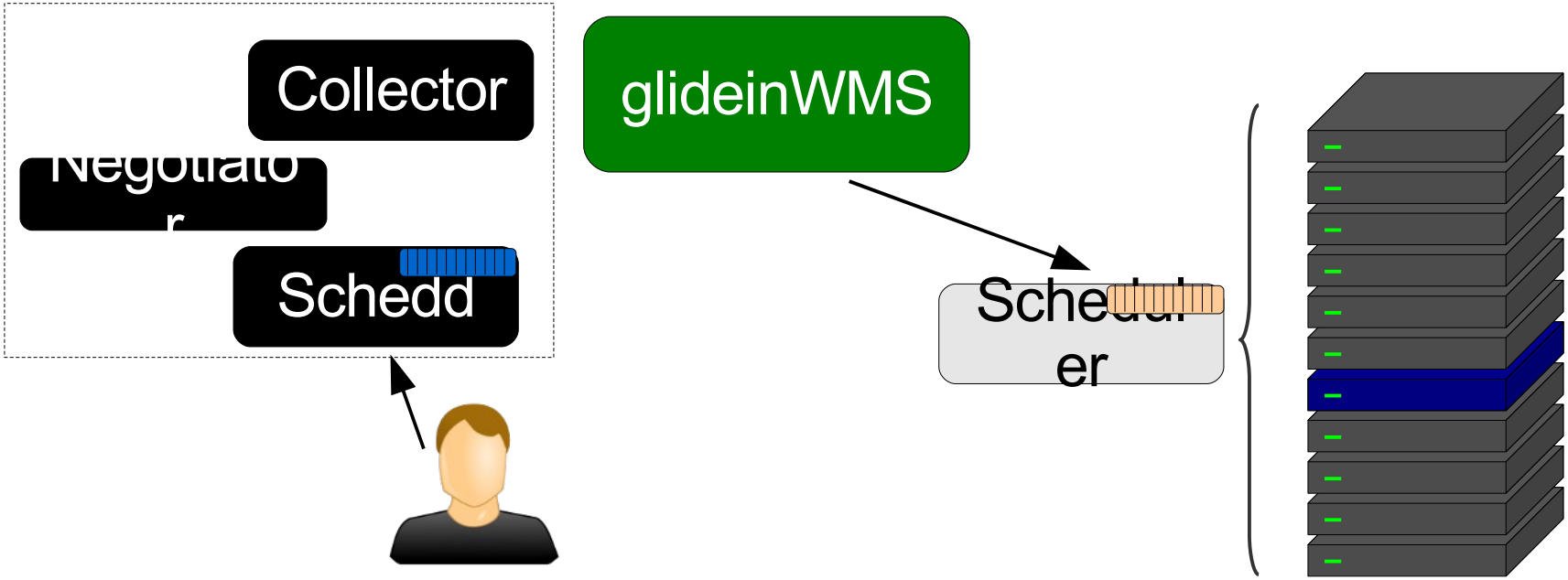
Avoid direct use of the Grid,
if you can.
Find someone else who does it for you.

Questions so far?

Reminder - glideinWMS

A Condor based overlay system

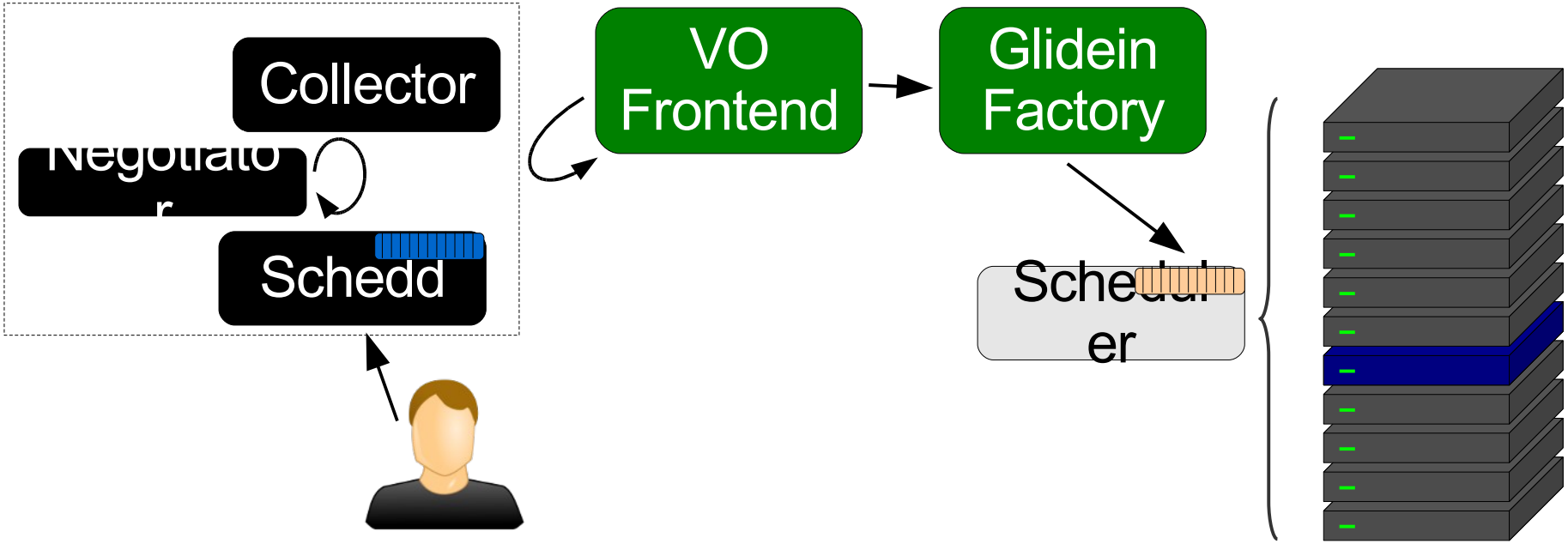
- i.e. looks like a regular Condor system to the users
- Adds a resource provisioning service
  (i.e. the lease manager)



The inner structure

The glideinWMS is really composed of two components

- A VO Frontend – The matchmaker
- A Glidein Factory – The pilot submitter



The Glidein Factory

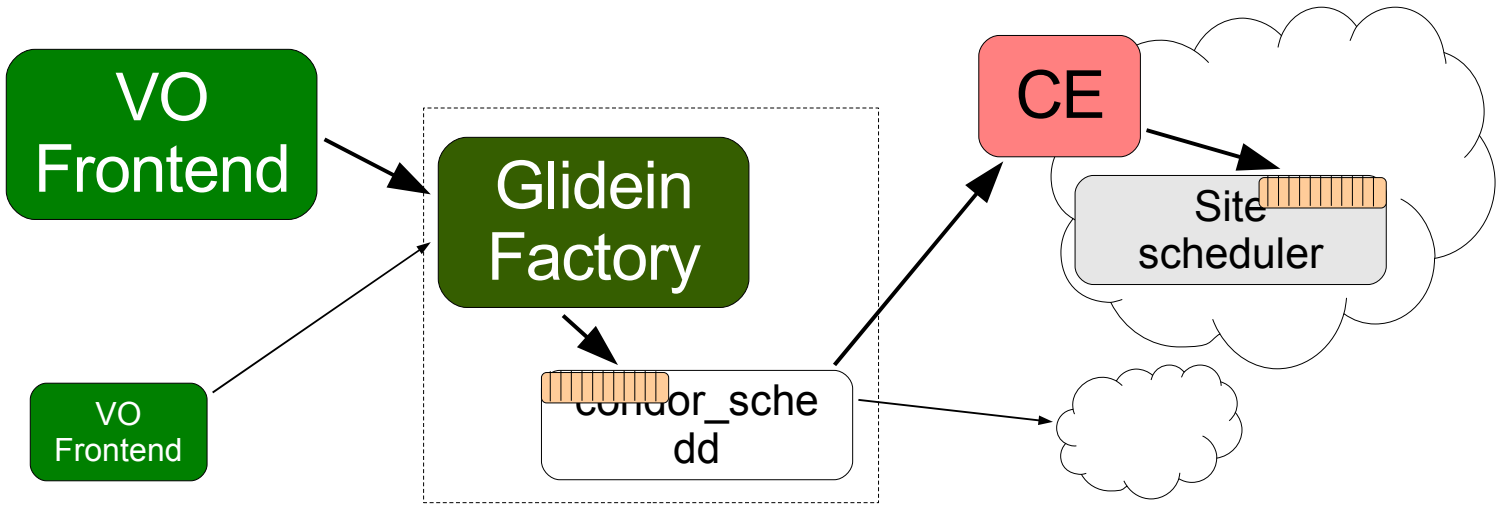
- The Glidein Factory is the interface to the Grid
  - Essentially, an additional abstraction layer
- Meant to be operated by an expert team on behalf of many user communities
  - Think of it as a service, not as a piece of SW
- The factory operators will deal with Grid details
  - Including debugging misbehaving glideins

Glidein Factory internals

Essentially a slave to VO Frontends

- Will submit on their behalf
- Using their certificate
- Main role is monitoring and debugging

Uses Condor-G under the hood



Glidein Factory internals

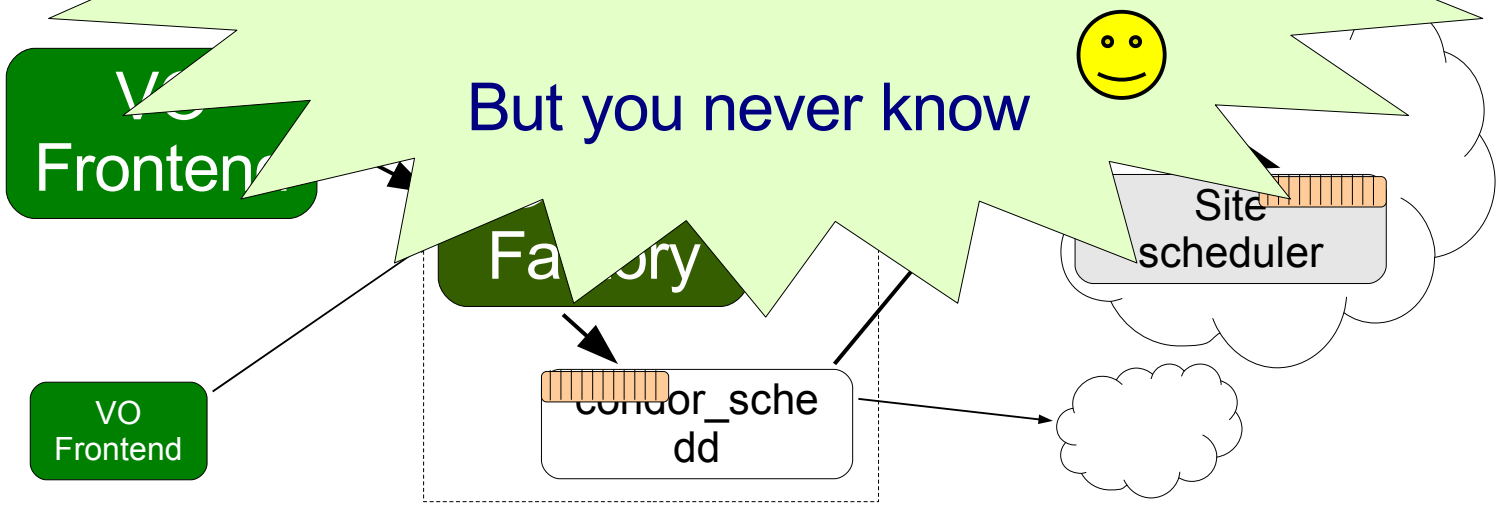
Essentially a slave to VO Frontends

- Will submit on t[illegible]ir h[illegible]alf
- Using t[illegible]
- Main[illegible]

Uses [illegible]

I don't expect you will ever need to operate a Glidein Factory.

But you never know

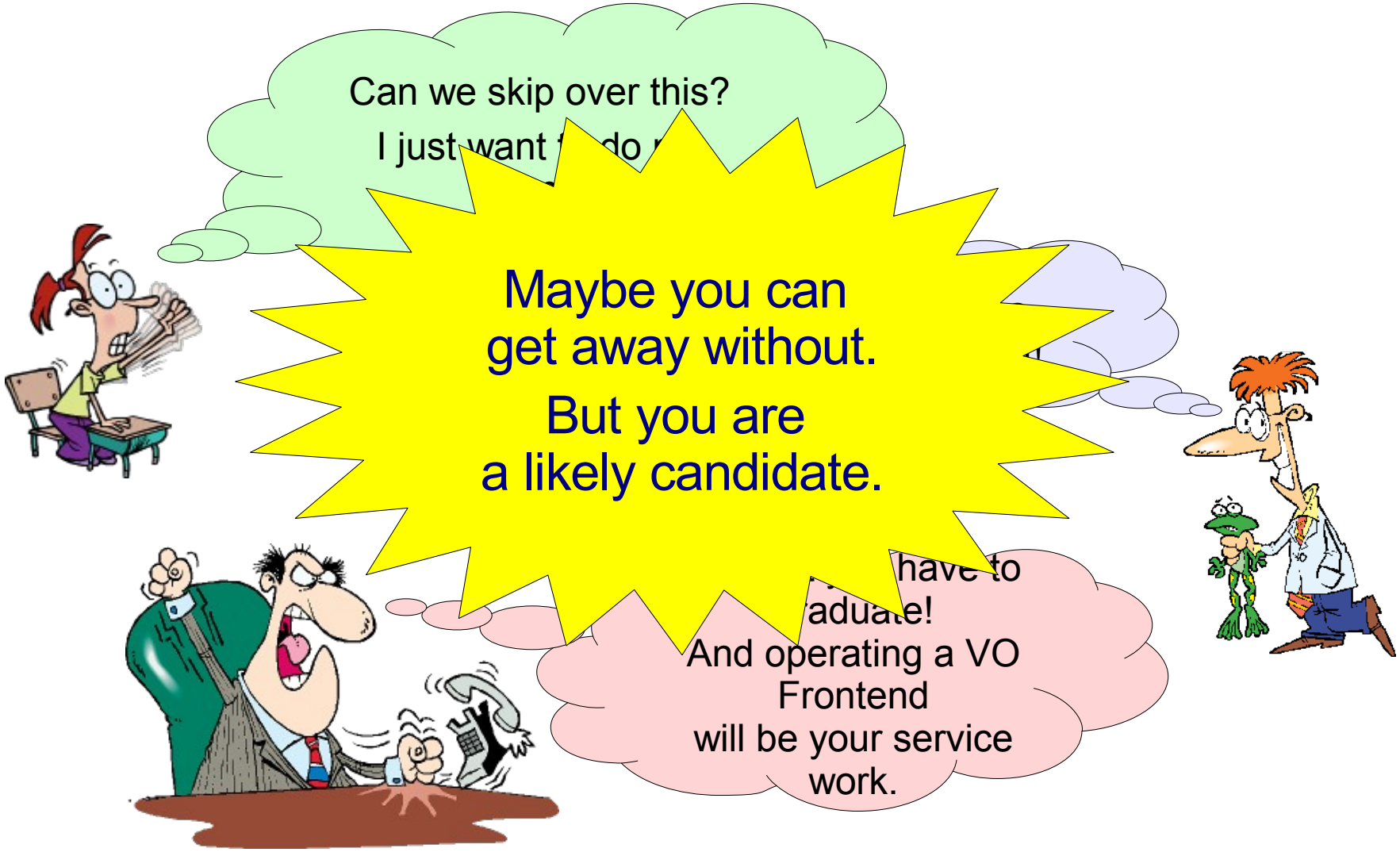


Open Science Grid

VO Frontend

- The “brain” of a glideinWMS system
  - Decides when and where to send glideins
  - Will talk to one or more gfactories
- Each user community needs one
  - i.e each VO == Virtual Organization
  - Alongside the Condor daemons
- Not much Grid knowledge needed here
  - Apart from when things go really wrong!

VO Frontend Matchmaking



VO Frontend Matchmaking

- The VO Frontend config defines the matchmaking policy
  - For both levels of matchmaking
- Unfortunately, the two levels expressed in two different languages
  - Python expression – Frontend logic
  - ClassAd expression – Startd requirements

Example config

```
<match
   match_expr='glidein["attrs"]["GLIDEIN_Site"] in job["DESIRED_Sites"].split(",")'
   start_expr='stringListMember(GLIDEIN_Site,DESIRED_Sites,",")' />
```

Monitoring and debugging

- The system mostly run itself
  - But sometimes things do go wrong
- Two major sources of monitoring
  - Condor itself (condor_q, condor_status)
  - The VO Frontend Web page and logs
- Similar for debugging

Open Science Grid

Questions?

Questions? Comments?

Feel free to ask me questions later:

Rob Quick <rquick@iu.edu>

Open Science Grid

Copyright notice

This presentation contains images copyrighted by ToonClipart.com

These images have been licensed to Rob Quick for use in his presentations

Any other use of them is prohibited