

glideinWMS Training @ UCSD

glideinWMS Frontend Installation

Part 2 – Frontend Installation

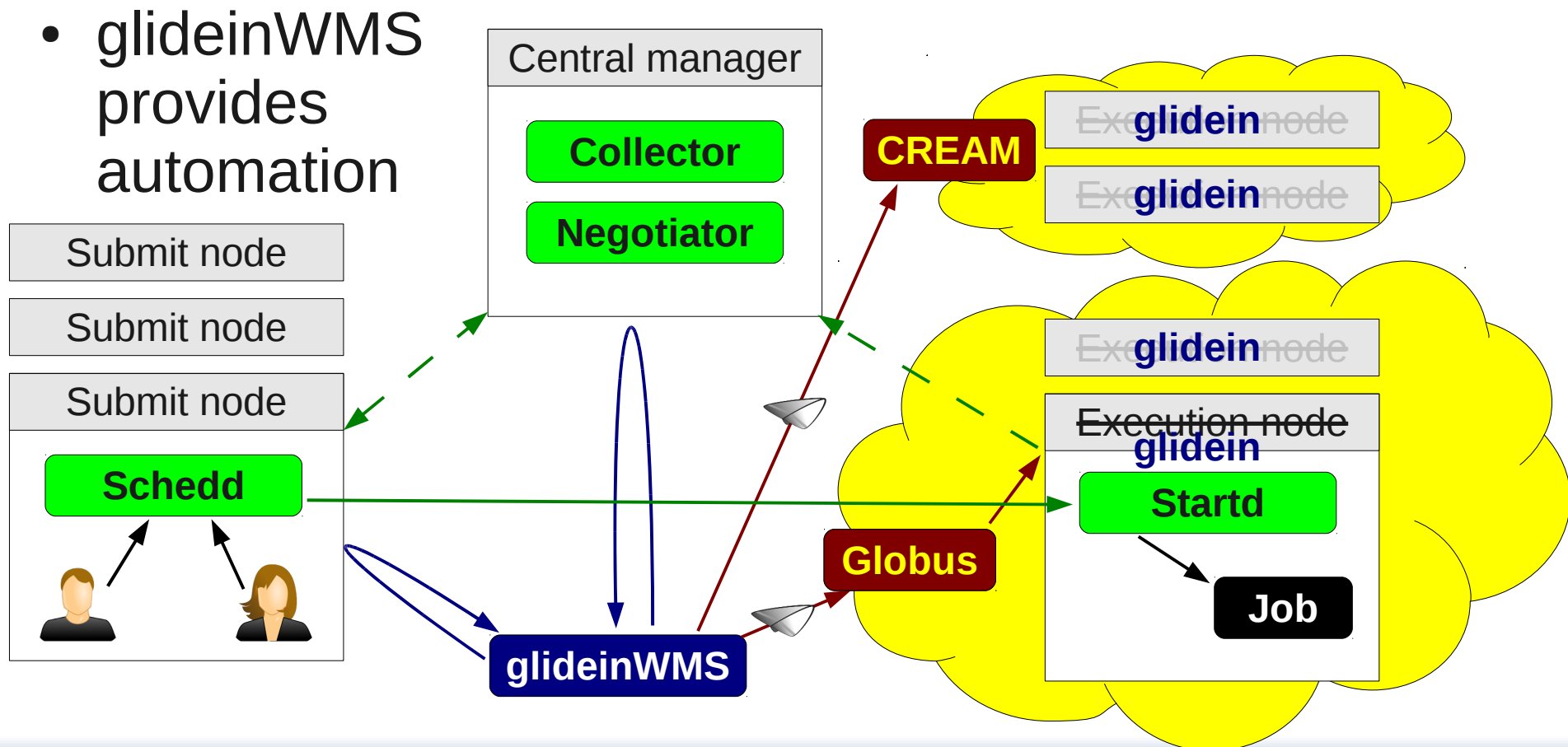
by Igor Sfiligoi (UCSD)

Overview

- Introduction
- Planning and System setup
- Prerequisites
- Actual Frontend Installation

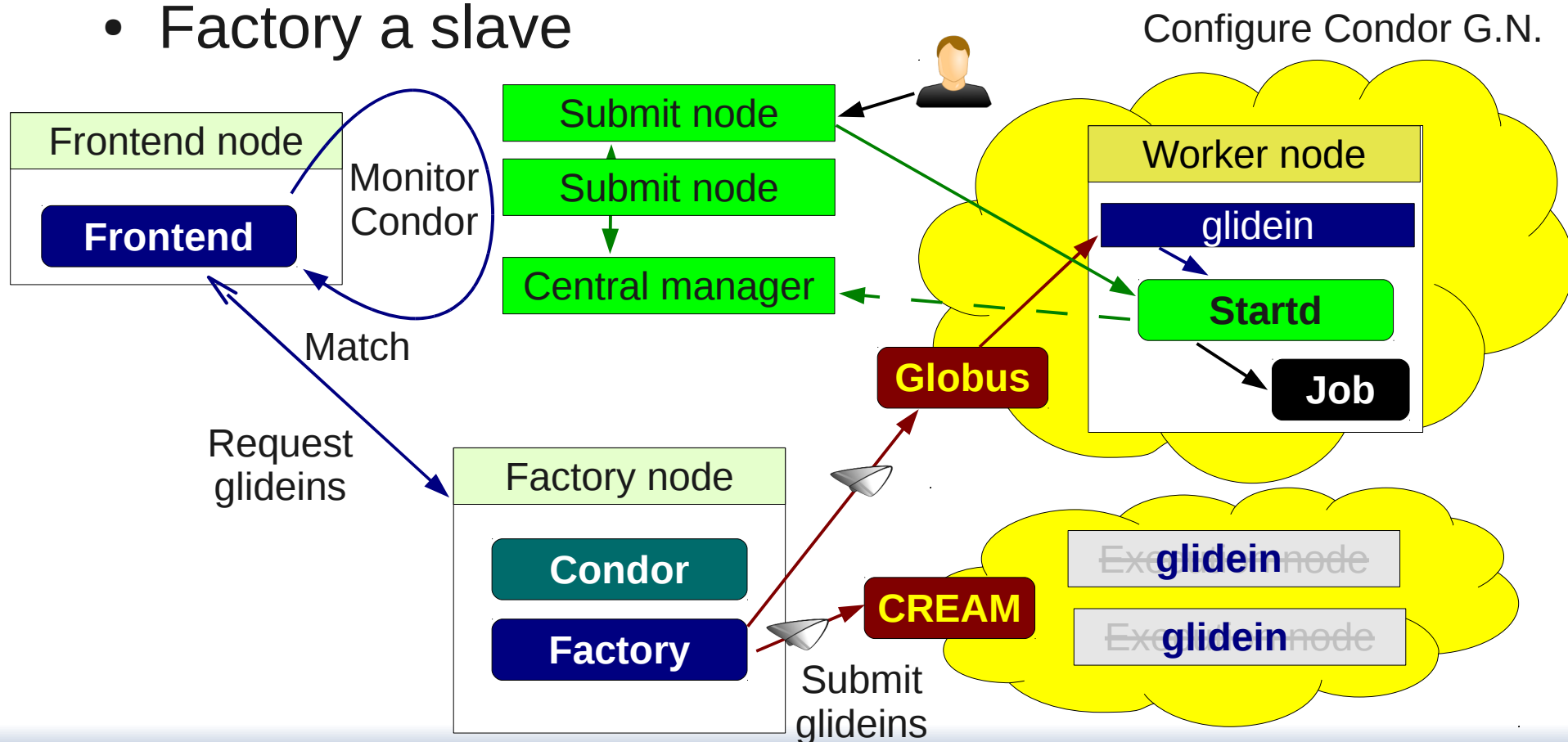
Refresher - Glideins

- A glidein is just a properly configured Condor execution node submitted as a Grid job



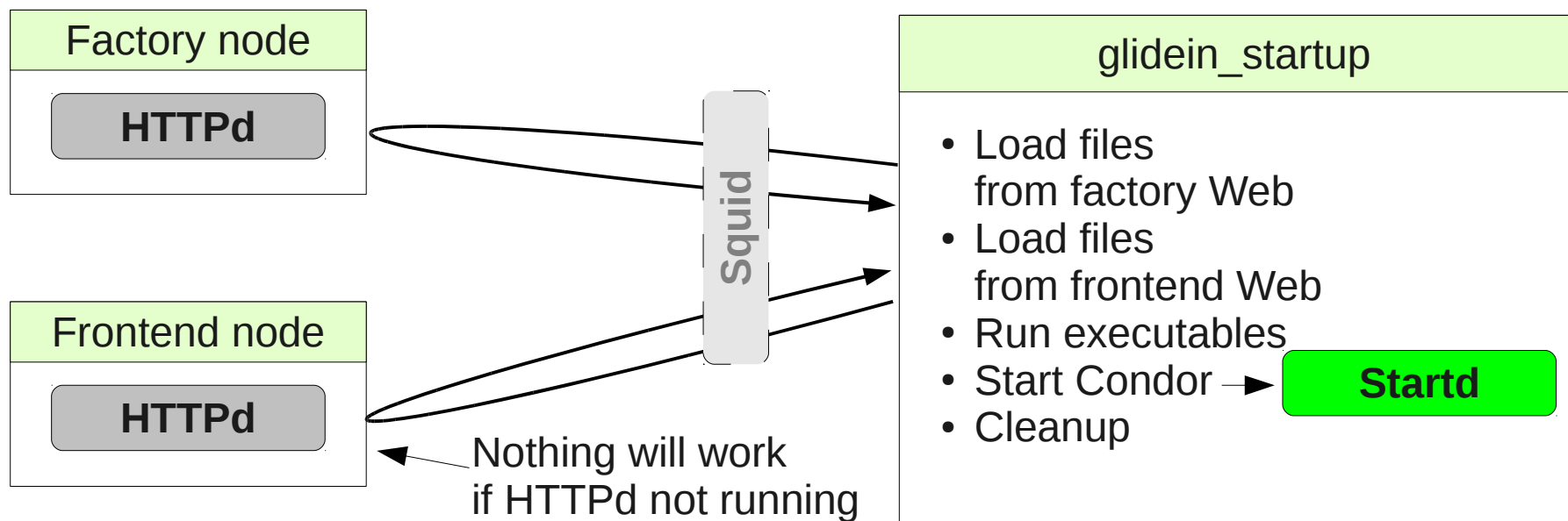
Refresher – VO Frontend

- The frontend monitors the user Condor pool, does the matchmaking and requests glideins
 - Factory a slave



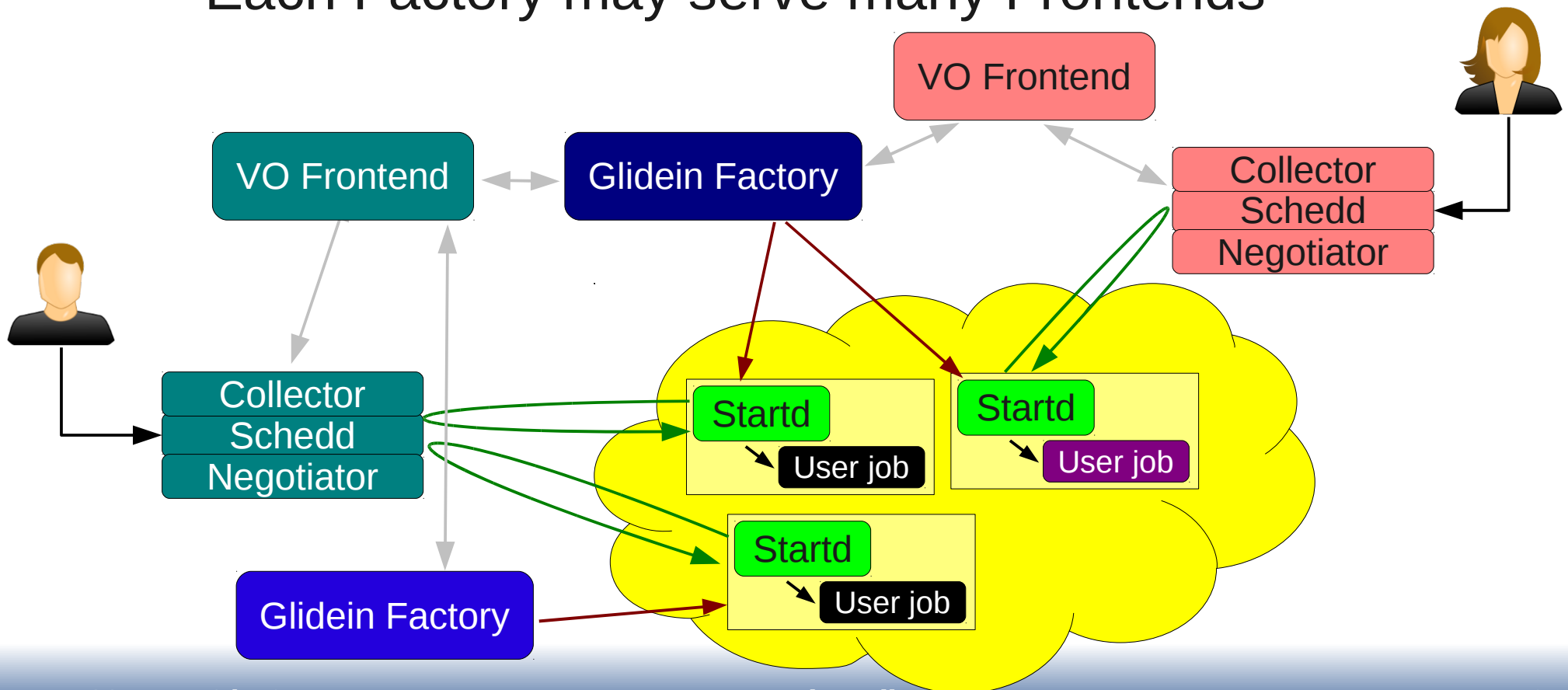
Refresher - glidein_startup

- Glidein startup script mostly an empty shell
- Real code and config files downloaded
 - From both Factory and Frontend Web servers



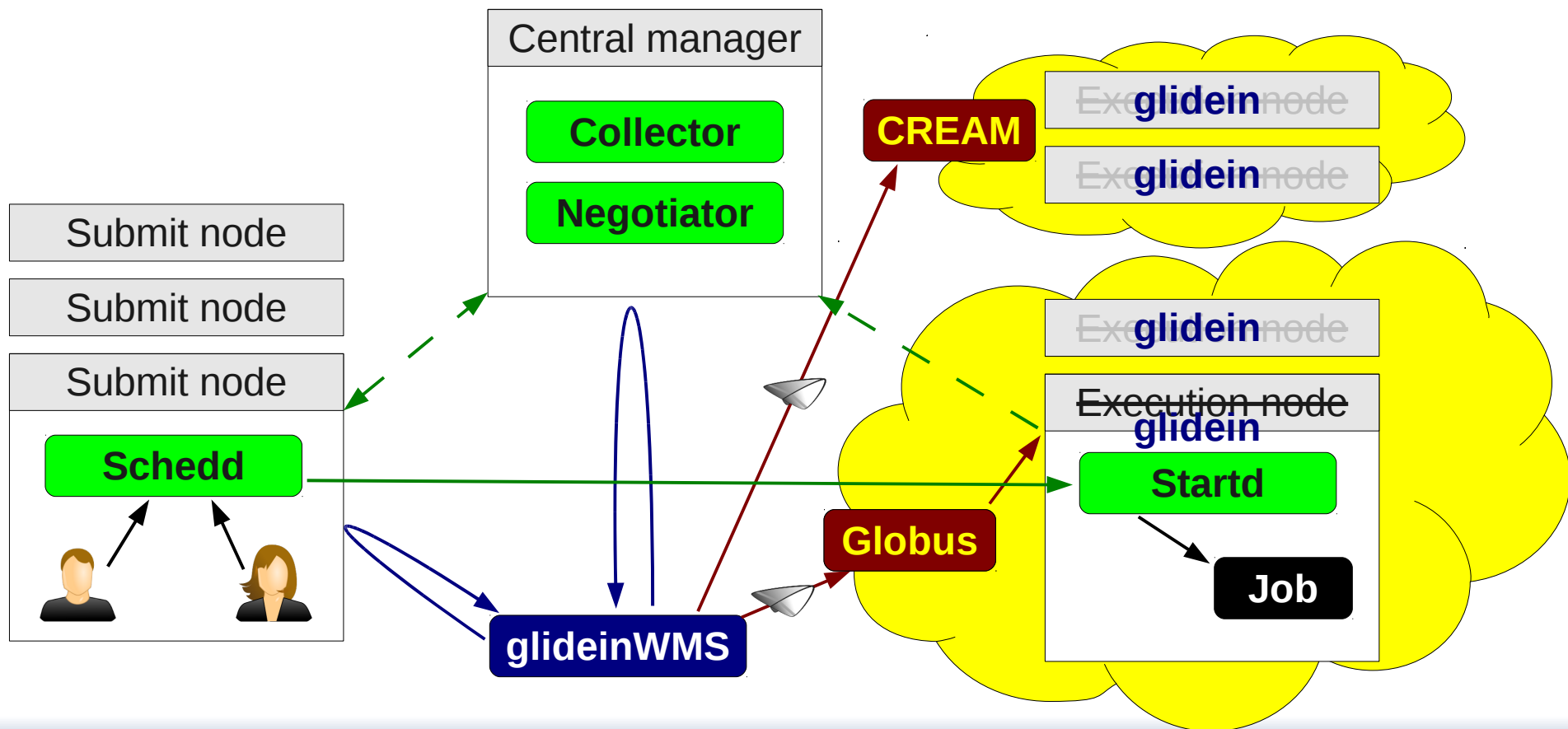
Refresher - Cardinality

- N-to-M relationship
 - Each Frontend can talk to many Factories
 - Each Factory may serve many Frontends



Refresher - Matchmaking

- The glideinWMS triggers glidein submission
 - The “regular” negotiator matches jobs to glideins



Refresher – Security Handles

- **Mutual authentication with Factory**
- Frontend provides (and Factory whitelists)
 - Service Proxy to talk to Factory Collector
 - Frontend Security name
 - Pilot Proxy
 - Proxy Security Class

Can have many such pairs

One set for whole Frontend (all Groups)
- Frontend whitelists (obtained from Factory admins)
 - Factory Collector DN
 - Own mapping @Factory
 - Factory mapping @Factory

One set per factory collector

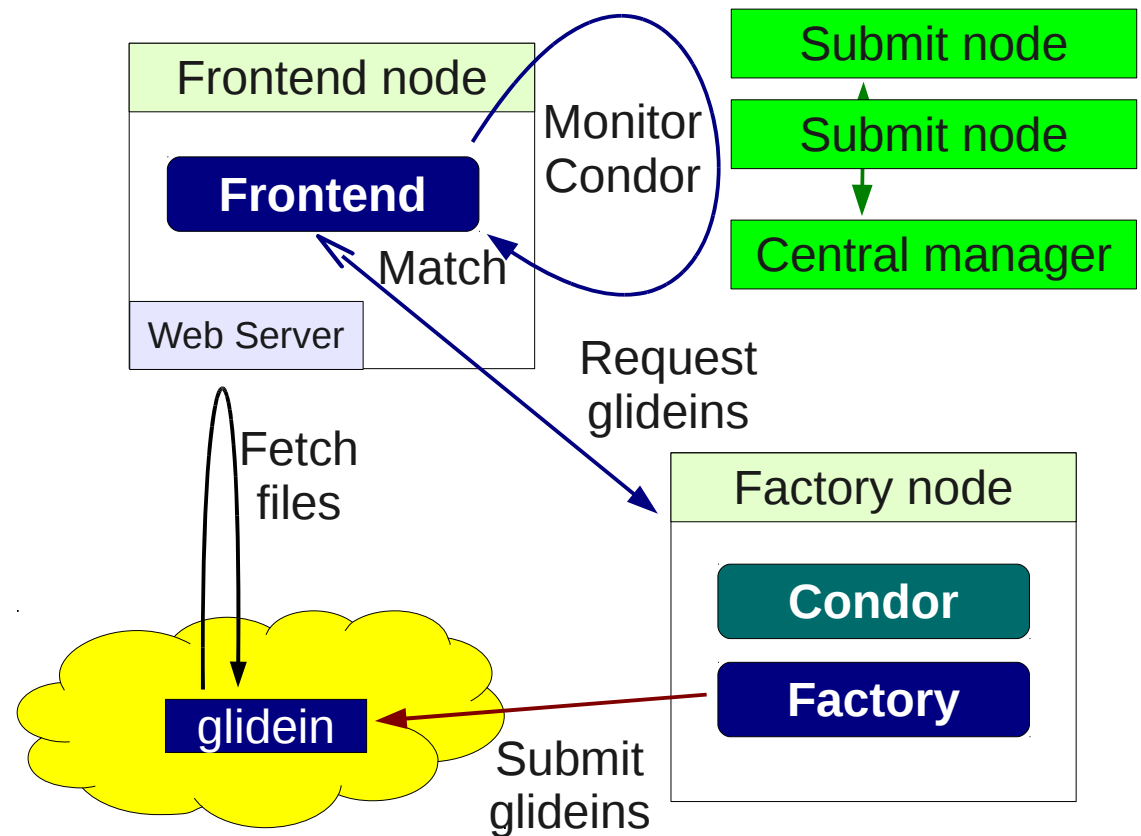
A word about monitoring

- The frontend keeps historical stats for monitoring purposes
- Storage based on Round Robin Databases
rrdtool - <http://oss.oetiker.ch/rrdtool/>
- Web interface to RRDs also provided
 - No server side code
 - Fully client side rendering in Javascript
JavascriptRRD - <http://javascriptrrd.sourceforge.net/>

Planning and System setup

It is a big world

- Many pieces, Frontend processes just a part
 - Condor
 - Web Server
 - Factory
 - The Grid
- Must play nicely with all of them



Condor

You should have Condor installed by now!

See yesterday's talk(s)

Planning


- Frontend can be installed on its own node
 - Or shared with other services
- **Avoid nodes where many users log into**
 - **For security reasons**
 - Will hold the pilot cert(s)
- Condor Central Manager a good companion
 - If it has enough Hardware for both

Experience: CMS has been switching between dedicated node and co-locating with CM as the pool grew and HW got replaced

Hardware requirements

- The CPU and Memory resources scale with the Condor use, both
 - Jobs in the queue
 - Glideins registered with the VO Collector
- Each Frontend group spawns 5 processes
 - So useful to have **multiple CPUs**
- Minimal IO use

Actual memory usage depends on attributes used for matchmaking
O(1k)-O(100k)



Components requiring root

- The Frontend processes **do not** need root privileges
 - And can be installed fully as non-root
- However, a few components are easier to install as root
 - The (OSG) Grid Client & CAs
 - The Web server
 - rrdtool python libraries
- Frontend RPM also available

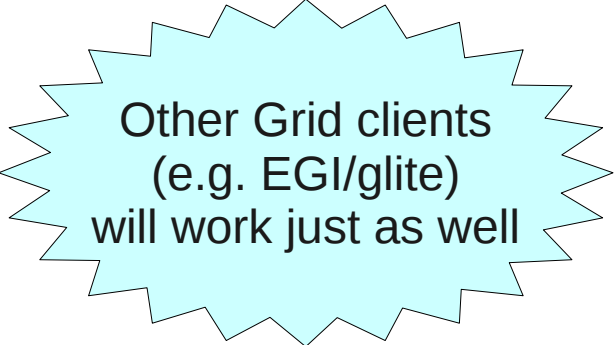

Will thus use this method in this talk

Not my favorite, but will shortly describe it toward the end

OSG Grid Client

- Requires RHEL5-compatible Linux
 - RHEL6 support promised for early 2012
- Procedure in a nutshell
 - Add EPEL and OSG RPM repositories to sys conf.
 - `yum install osg-ca-certs`
 - `yum install osg-client`
 - Enable CRL fetching crontab

Pacman based
VDT could be
installed as non-root
but now deprecated



Other Grid clients
(e.g. EGI/glite)
will work just as well

<https://twiki.grid.iu.edu/bin/view/Documentation/Release3/InstallOSGClient>

Web server

- Any Web server will do
 - Serving just static files
- The glideinWMS provided installer will clean up the config
 - Will use system yum install
 - But get rid of any eventual old httpd beforehand
- Create a directory owned by the frontend user
 - `/var/www/html/vofrontend`

```
# install the web server
yum remove -y httpd
cd glideinWMS/install/
# Q&A installer in batch mode
./glideinWMS_install << EOF
8
C
80 ← Port to use
1
EOF
```

RRDTool

- The Frontend is a python script
 - Thus needs the RRDTool python libraries
- Easiest way is getting them from EPEL
 - Which you already have if using the OSG Client

```
% yum install rrdtool-python
```


Prerequisites

Before you begin

- Before you start the actual installation, you need a few things:

- Decide what UID will you be using
- A host/service certificate for the Frontend
- One or more pilot certificates
 - And they must be registered with your VOMS
- Decide on the Security Name for your Frontend
- Find a Factory to support you

Assuming
“frontend”
in this talk



Hint: The OSG glidein factory



Frontend certificate

- OSG provides a script to talk to DOEGrids

https://twiki.grid.iu.edu/bin/view/Documentation/Release3/GetHostServiceCertificates#Request_a_Service_Certificate

- Unfortunately, must be run as root
- Procedure in a nutshell
 - Install OSG client
 - `yum install osg-cert-scripts`
 - `cert-request ...` ← Ask for the “VOfrontend” service
 - Wait for email
 - `cert-retrieve ...`
 - `cp` into `~frontend/.globus/`
 - `chown` to `frontend:frontend`

If you have other ways to obtain a service cert, feel free to use them

Pilot certificate(s)

- Just another type of service certificate
 - Use service name **vopilotxx**
- Follow instructions on previous slide
- Then register it with your VOMS

Try to make it
Globally Unique

Each VO uses a slightly different
procedure, so I am not going to
describe this step

If you decide to
use multiple pilot certificates
with different privileges you may want
to
pick multiple **security class** names
for added security
(default is "frontend")

The Security Name

- The Frontend **Security Name** is a **Globally Unique name** your Frontend will use to securely communicate with the Factories
 - A good pick is “**VOsomething**”
- This may or may not be the same as the **Frontend Name**
 - Which also must be Globally Unique
 - and is used as a ClassAd Name
 - The Frontend name often contains a version string; Security name does/must not

Contact the factory admins

- You will need a Factory to submit glideins
 - While the Factory is a slave to the Frontend
 - It chooses its masters!
 - To use the OSG glidein factory, send an email to osg-gfactory-support@physics.ucsd.edu
 - You will need to provide
 - Frontend certificate DN
 - Security name
- And you will get back the factory information you need during the installation

<http://hepuser.ucsd.edu/twiki2/bin/view/UCSDTier2/OSGgfactory>

Frontend Installation

Frontend Installation

- Now you should have all the pieces to start the installation

- In a nutshell

I assume you already have the glideinWMS “binaries”

<http://tinyurl.com/glideinWMS>

- Automate proxy creation
- Get access to a Condor client
- Get JavascriptRRD tarball and extract it

<http://sourceforge.net/projects/javascriptrrd/>

Get the latest one with flot and tooltip


- Create Frontend config file
- Instantiate the Frontend
- Start the Frontend

All as a regular user (e.g. “frontend”)

Proxies vs certs

- In the previous slides, you were told to obtain a service and pilot certificates
- But **the Frontend needs proxies!**
 - You must convert certs into proxies
- Proxies are short lived
 - Should be hours for the service proxy
 - A few days for the pilot proxy
- **You must keep them valid at all times**
 - With a cron script (or equivalent)

Must be longer than the glidein lifetime (and no less than 12h)



Example cron script

- Really just voms-proxy-init

```
#!/bin/bash
home=/home/frontend/.globus
umask 0077

idstr=service
export X509_USER_PROXY=${home}/x509_${idstr}.proxy
voms-proxy-init -cert ${home}/${idstr}cert.pem -key ${home}/${idstr}key.pem \
                -hours 12 -valid 12:0

idstr=vopilot01
export X509_USER_PROXY=${home}/x509_${idstr}.proxy
voms-proxy-init -cert ${home}/${idstr}cert.pem -key ${home}/${idstr}key.pem \
                -voms VO:/VO/role=pilot -hours 72 -valid 72:0
```

- Then add it to crontab

```
# crontab -l
31 3,9,15,21 * * * /home/frontend/.globus/refresh_proxy.sh >/dev/null 2>/dev/null
```

The Condor client

- The Frontend uses the Condor client cmdline tools to talk to both the rest of the VO Condor daemons and to the Factory(s)
 - Just the binaries
- Unless you host the Frontend on a submit node **(which is not recommended)** you will need to **install a private version** of Condor binaries
 - Properly configuring it will make it usable for debugging, too

Installing the Condor client

- As usual, using the provided installer the easiest way
- Manual install would require:
 - Unpack the tarball
 - Point it to the CM
 - Make sure no daemons are ever started (it is a client!)

Installing with the Q&A Installer

```
~/glideinWMS/install$ ./glideinWMS_install
...
Please select: 6
[6] Condor for VO Frontend
...
Where do you have the Condor tarball? /home/frontend/Downloads/condor-7.6.4-x86_rhap_5-stripped.tar.gz
Where do you want to install it?: [/home/condor/glidecondor] /home/frontend/glidecondor/glidecondor
If something goes wrong with Condor, who should get email about it?: me@myemail
Do you want to split the config files between condor_config and condor_config.local?: (y/n) [y] y
...
Do you want to get it from VDT?: (y/n) y
Do you have already a VDT installation?: (y/n) y
Where is the VDT installed?: /etc/osg/wn-client
Will you be using a proxy or a cert? (proxy/cert) proxy
Where is your proxy located?: /home/frontend/globus/x509_service.proxy
My DN = 'DN1'
...
DN: DNxxx
nickname: [condor001] uidxxx
Is this a trusted Condor daemon?: (y/n) y
...
DN:
What node is the collector running (i.e. CONDOR_HOST)?: collectornode.mydomain
```

Creating the frontend config file

- This is the main step of the installation!
- Here I explain only the basics
 - Actual frontend config tuning in the next talk
- As usual, using the provided installer makes life easier
 - And is the **recommended way** to get the basic template
 - You will likely want to tune the created config, though

Using the Q&A Installer - 1/3

```
$ ./glideinWMS_install
```

```
...
```

```
Please select: 7
```

```
[7] VO Frontend
```

```
Do you have already a javascriptRRD installation?: (y/n) y
```

```
Where is javascriptRRD installed?: /home/frontend/javascriptrrd-0.6.1
```

```
Where will you host your config files?: [/home/frontend/frontstage] /home/frontend/frontstage
```

```
Where will the web data be hosted?: [/var/www/html/vofrontend] /var/www/html/vofrontend
```

```
What Web URL will you use?: [http://my.node/vofrontend/] http://my.node/vofrontend/
```

```
Where will you host your log files?: [/home/frontend/frontlogs] /home/frontend/frontlogs
```

```
Give a name to this VO Frontend?: [myVO-cabinet-10-10-5] VOfe
```

```
Give a name to this VO Frontend instance?: [v1_0] v1_0
```

← This is your
security name

```
What node is the WMS collector (i.e. the gfactory) running?: factory.node
```

```
What is the classad identity of the glidein factory?: [gfactory@factory.node] gfactory@factory.node
```

```
What is the WMS collector DN (i.e. subject)?: DN1
```

```
Where is your proxy located?: /home/frontend/globus/x509_service.proxy
```

```
voms-proxy-info is needed to extract the VO information from the proxy
```

```
Do you want to get it from VDT?: (y/n) y
```

```
Do you have already a VDT installation?: (y/n) y
```

```
Where is the VDT installed?: /etc/osg/wn-client
```

```
For security reasons, we need to know what will the WMS collector map us to.
```

```
What is the mapped name?: vo1@factory.node
```

← You get
this data
from the
factory
admins

Using the Q&A Installer - 2/3

Using pool collector collector.node

What is the pool collector DN (i.e. subject?): **DN2**

List and secondary pool collectory the glideins should use instead of the main collector.
Leave an empty collector name when finished.

Collector name: **collector.node:9620-9819**

Collector DN (i.e. subject): **DN3**

Collector name:

Tree of collectors

The following schedds have been found:

[1] schedd1.node

...

Do you want to monitor all of them?: (y/n) **y**

What is the DN (i.e. subject) for schedd schedd1.node?: **DNxxx** } x N

...

Using the Q&A Installer - 3/3


```
What kind of jobs do you want to monitor?: [JobUniverse==5] JobUniverse==5
Give a name to the main group: [main] main
What expression do you want to use to match glideins to jobs?
Match string: [True] glidein["attrs"]["GLIDEIN_Site"] in job["DESIRED_Sites"].split(",")
What job attributes are you using in the match expression?
Job attributes: [DESIRED_Sites] DESIRED_Sites
What glidein/factory attributes are you using in the match expression?
Factory attributes: [GLIDEIN_Site] GLIDEIN_Site

Do you want to use the frontend proxy to submit glideins: (y/n) [y] n
An empty entry means you are done.
proxy fname: /home/frontend/globus/x509_vopilot1.proxy
proxy fname:

How do you want to categorize the use of glexec if available at sites?: [OPTIONAL] OPTIONAL
Do you want to expose the Grid env. to the user jobs?: (y/n) y

Do you want to create the VO Frontend instance (as opposed to just the config file)?: (y/n) [n] n
Configuration file is located at /home/frontend/frontstage/instance_v1_0.cfg/frontend.xml
```

Will use glexec
wherever available



Here is your config file



Instantiating the Frontend

- Once you are happy with the config, run **create_frontend**
- It will create the Frontend instance in **~/frontstage/frontend_vOfe-v1_0**
 - You will work in there from now on

By default, the security name is used to create the frontend name

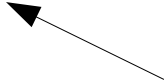
```
~/glideinWMS/creation$ ./create_frontend ~/frontstage/instance_v1_0.cfg/frontend.xml
Created frontend 'VOfe-v1_0'
Active entries are:
  main
Work files can be found in /home/frontend/frontstage/frontend_VOfe-v1_0
Log files can be found in /home/frontend/frontlogs/frontend_VOfe-v1_0
Support files are in /var/www/html/vofrontend/stage/frontend_VOfe-v1_0
Monitoring files are in /var/www/html/vofrontend/monitor/frontend_VOfe-v1_0
```

Starting the Frontend

- The Frontend instance has a init.d-like script
`./frontend_startup start|stop`
- This script also must be used to make **any config changes** to the Frontend
 - **Never change the frontend.xml in the Frontend dir**
 - Work on a copy
(possibly in `../instance_v1_0.cfg/`)
and then run
`./frontend_startup reconfig new_config`

Installing the Frontend from the OSG RPM repository

The Frontend RPM

- OSG RPM repository provides a frontend RPM
 - Advantages:
 - It is an RPM, so tighter system integration
 - Partially pre-configured to talk to the OSG factory
 - Disadvantages:
 - It is an RPM, thus installed and maintained as root
 - No support for versioning
 - Requires more manual configuration
- 
- Processes still running as frontend

<https://twiki.grid.iu.edu/bin/view/Documentation/Release3/InstallGlideinWMSFrontend>

Installation of the RPM

- Similar to OSG Grid Client
 - Indeed good practice to install the OSG Client first
- **Requires a RPM-based Condor installation**
 - But will provide basic Condor config files in `/etc/condor/config.d/`
(not going into details in this talk, see Twiki)
- The actual RPM install a simple **`yum install glideinwms-vofrontend`**
 - You still need to configure it, though

<https://twiki.grid.iu.edu/bin/view/Documentation/Release3/InstallGlideinWMSFrontend>

Configuration

- The RPM comes with a basic template in **`/etc/gwms-frontend/frontend.xml`**
- **All security sections must be modified:**
 - To talk to the rest of the Condor pool (CM and submit nodes)
 - For the Factory interaction (the info you got from the Factory admins)
 - Regarding the service and pilot proxies

<https://twiki.grid.iu.edu/bin/view/Documentation/Release3/InstallGlideinWMSFrontend>

Installation and startup

- The RPM provides a init.d script
/etc/init.d/gwms-frontend
- You will need to **run the reconfig once** to finish the installation

```
# /etc/init.d/gwms-frontend reconfig
/var/lib/gwms-frontend/vofrontend/frontend.xml
Warning: Cannot find /var/lib/gwms-frontend/vofrontend/frontend.xml
If this is the first reconfig, you can ignore this message.
{ -force_name name -writeback yes|no -update_scripts yes|no -xml xml | -help }

Reconfiguring the frontend                [OK]
```

- After that, it is ready to be started
/etc/init.d/gwms-frontend start

The End

Pointers

- The official glideinWMS project Web page is <http://tinyurl.com/glideinWMS>
- glideinWMS development team is reachable at glideinwms-support@fnal.gov
- The OSG glidein factory is reachable at osg-gfactory-support@physics.ucsd.edu

Acknowledgments

- The glideinWMS is a CMS-led project developed mostly at FNAL, with contributions from UCSD and ISI
- The glideinWMS factory operations at UCSD is sponsored by OSG
- The funding comes from NSF, DOE and the UC system