Fermilab

The Compact Muon Solenoid

Joint EGEE/OSG VO Management at HPDC '08

GlideinWMS

The CMS pilot infrastructure

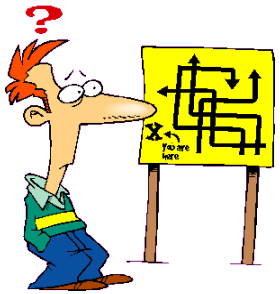
by Igor Sfiligoi (Fermilab)

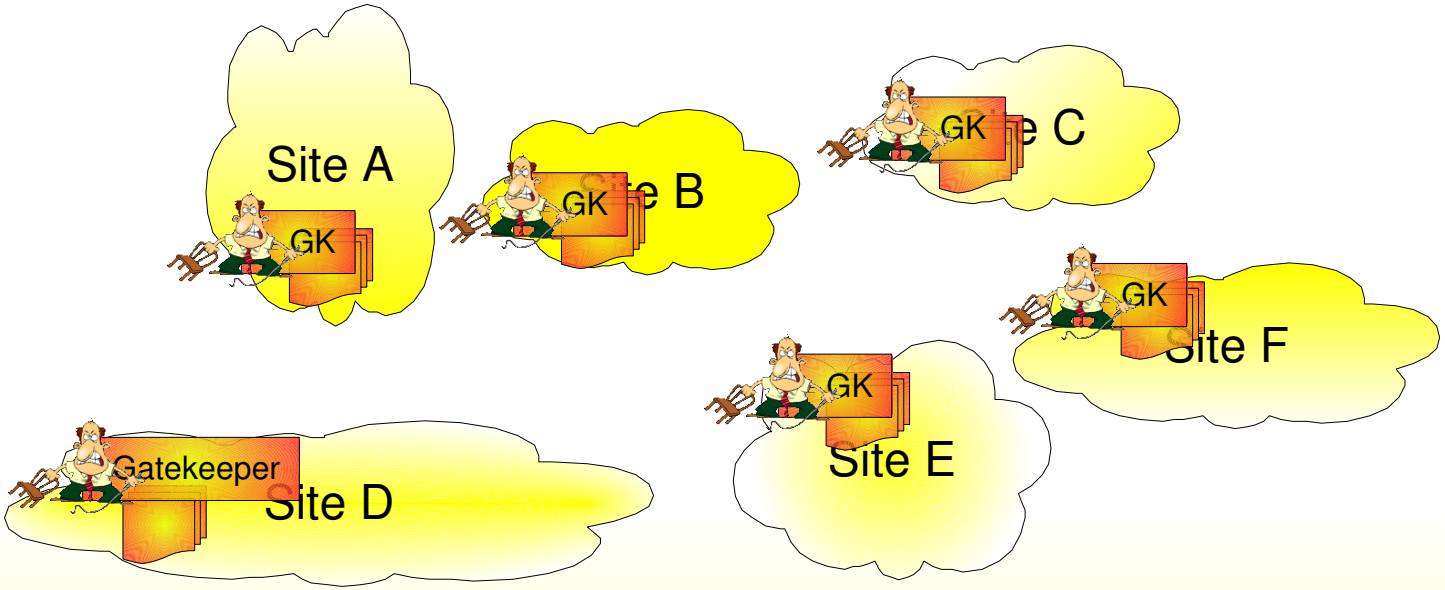
Outlook

- Grid computing overview
- The pilot paradigm
- Introducing Condor glideins
- glideinWMS description
- glideinWMS in real life
- Conclusions

Bare-bones Grid is complex

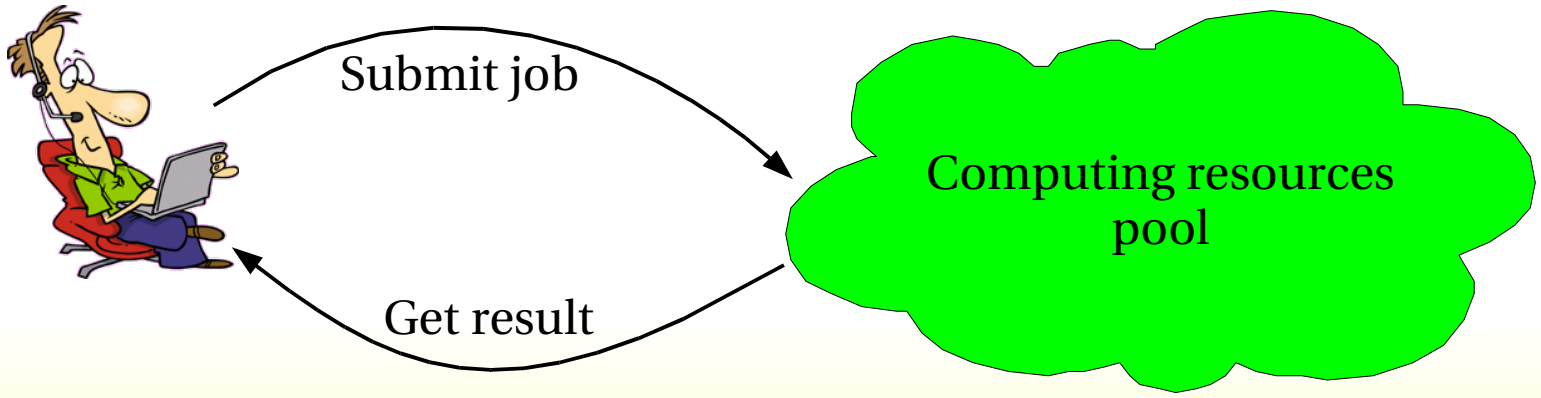
- Resources grouped in independent pools
  - Each with its own set of rules (often not published)
- Resources in different pools configured differently
  - Users expected to adapt (after they discover how they work)





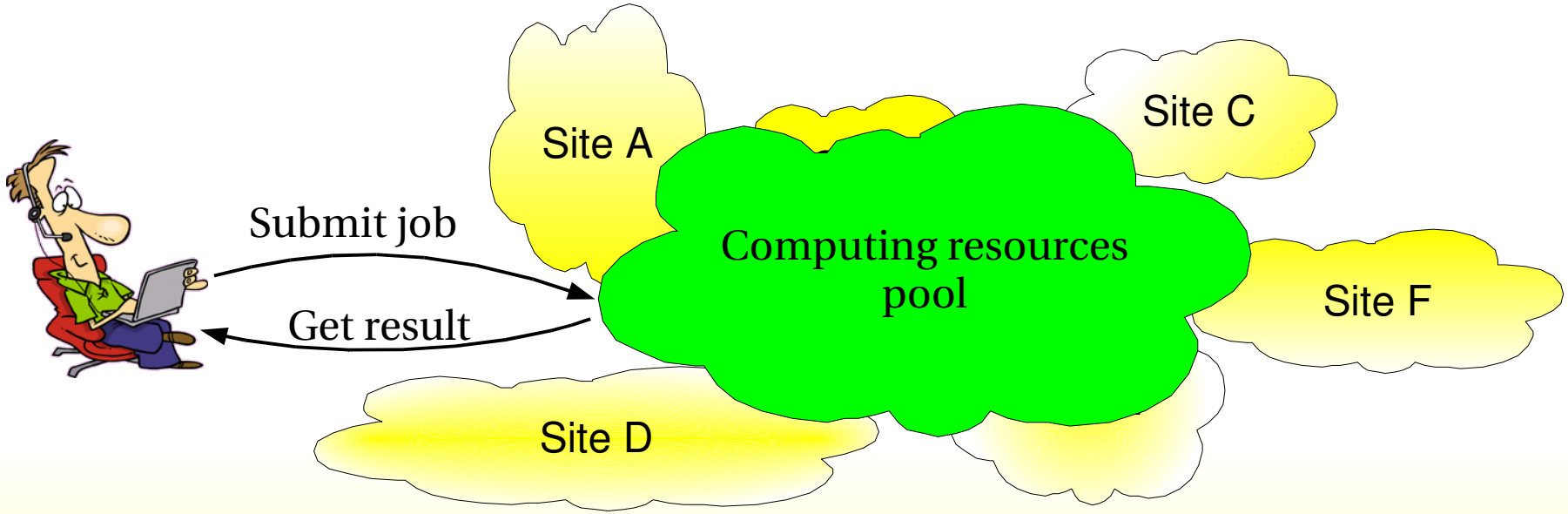
Scientific user needs

- Wants to spend most of his time thinking about the scientific problems
- Has large computing needs, but should be simple
  - Computing is just a tool
  - Time spent on computing problems is subtracting time available for scientific thinking!



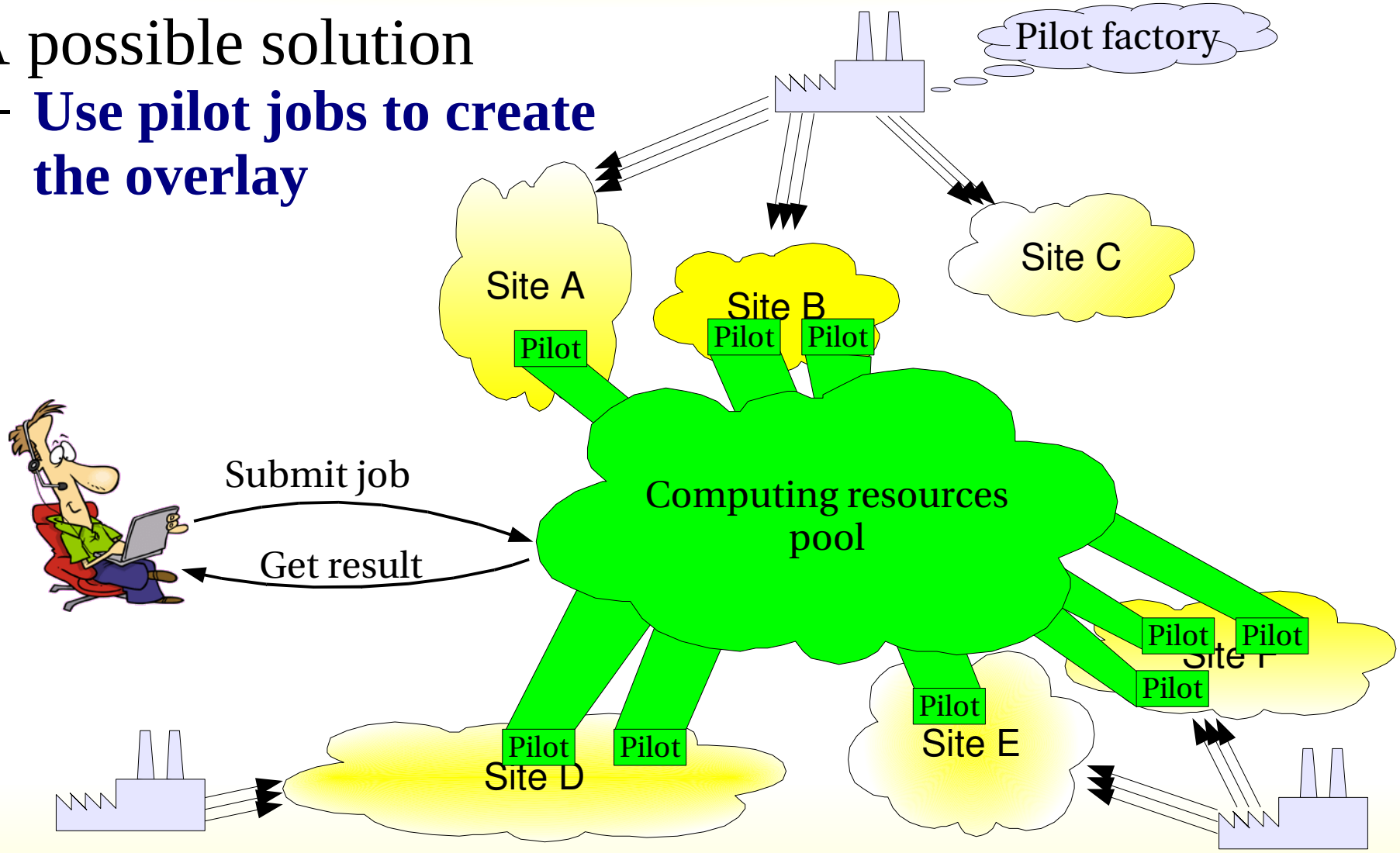
Can we get to the middle ground?

- Maybe by creating an overlay over the Grid sites?
  - To hide the differences between sites
- Making the Grid look as a single, uniform pool?



The pilot paradigm

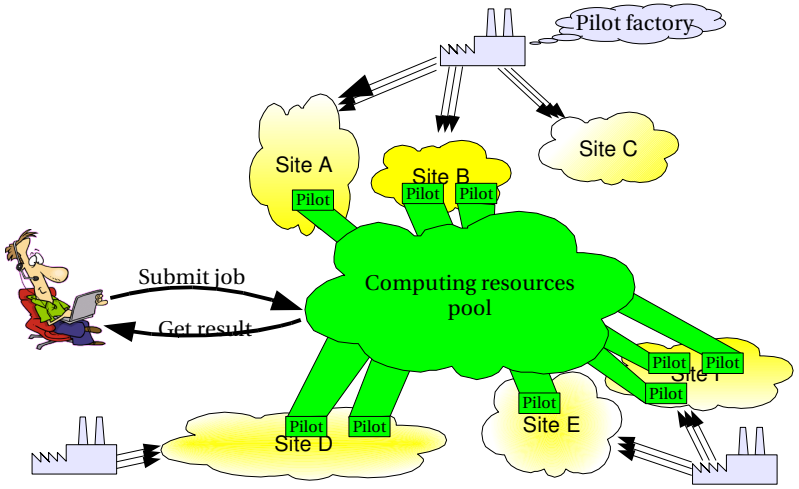
- A possible solution
  - **Use pilot jobs to create the overlay**



The pilot paradigm (continued)

User jobs never sent directly to Grid sites

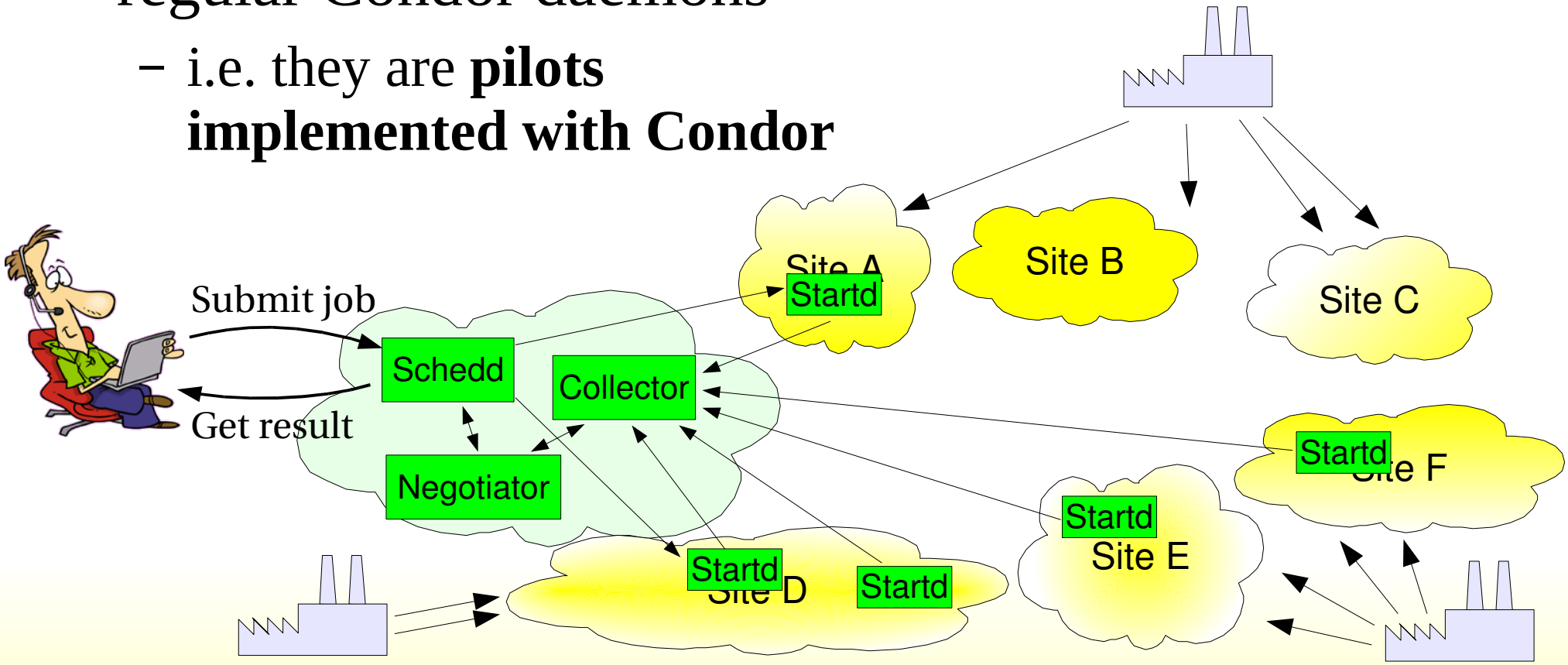
- Never send user jobs directly
  - Send pilot jobs to create the overlay, instead
- When a pilot lands on a Grid worker node
  - Validates Grid resource
  - Prepares the environment
  - Pulls a user job
- Hides Grid heterogeneity
  - Users see a fairly uniform computing pool



Condor glideins

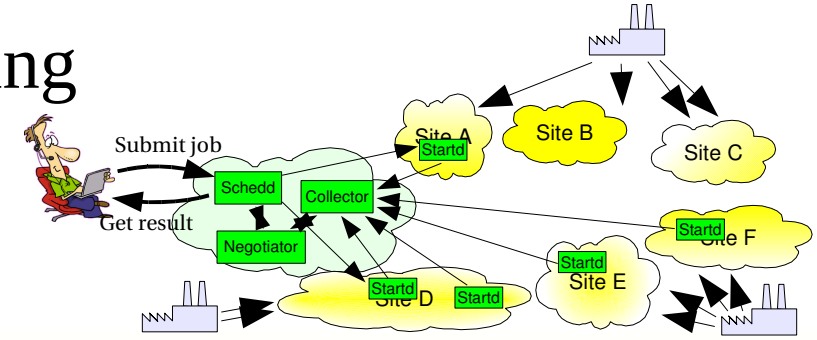
http://www.cs.wisc.edu/condor/

- Condor is based on a distributed architecture
- Condor glideins are Grid jobs that start regular Condor daemons
  - i.e. they are **pilots implemented with Condor**



Submitting glideins

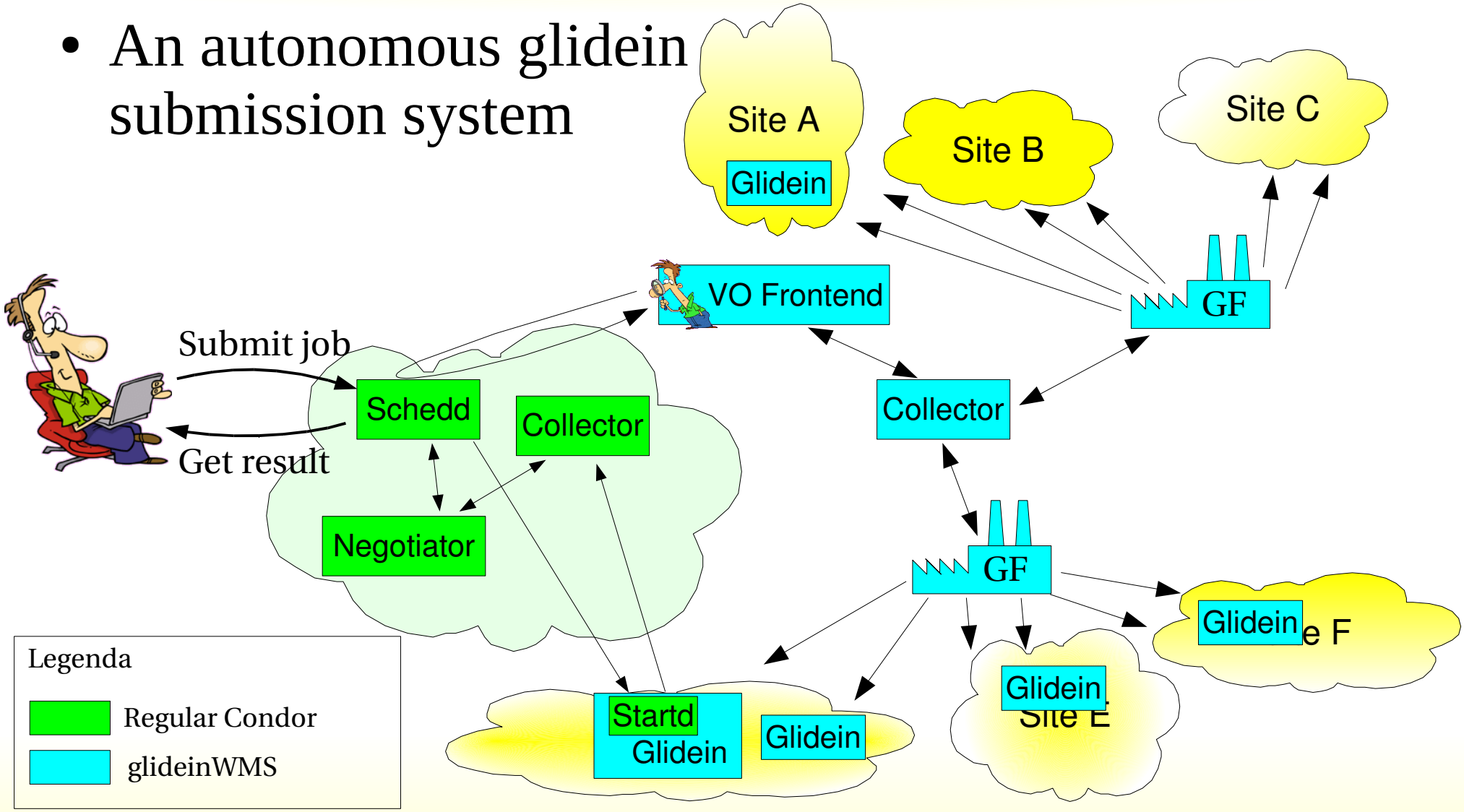
- Condor provides only a basic command line glidein submission tool
  - Good for trying out glideins
  - But not meant to be used as a glidein factory
- A few groups developed glidein factories
  - CDF has the CDF-specific GlideCAF
  - USCMS@FNAL is developing the **glideinWMS**



Introducing the glideinWMS

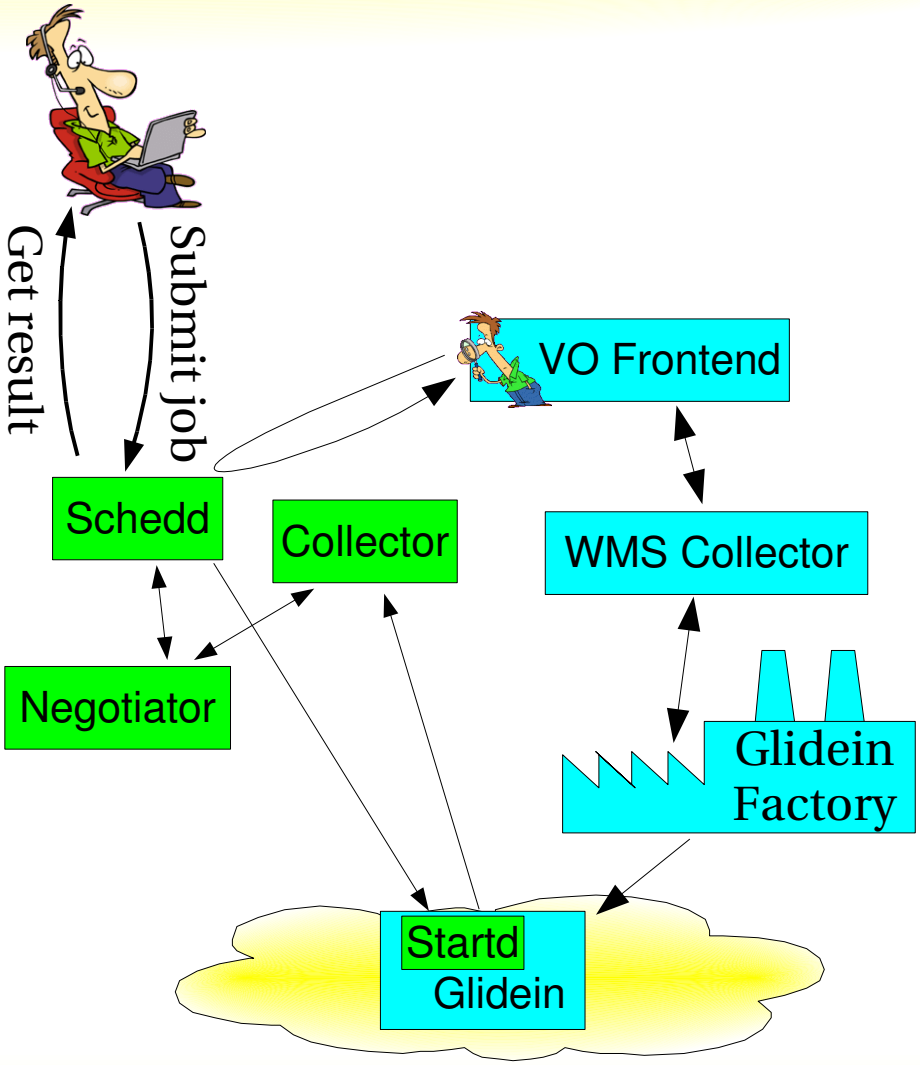
http://www.uscms.org/SoftwareComputing/Grid/WMS/glideinWMS/

- An autonomous glidein submission system



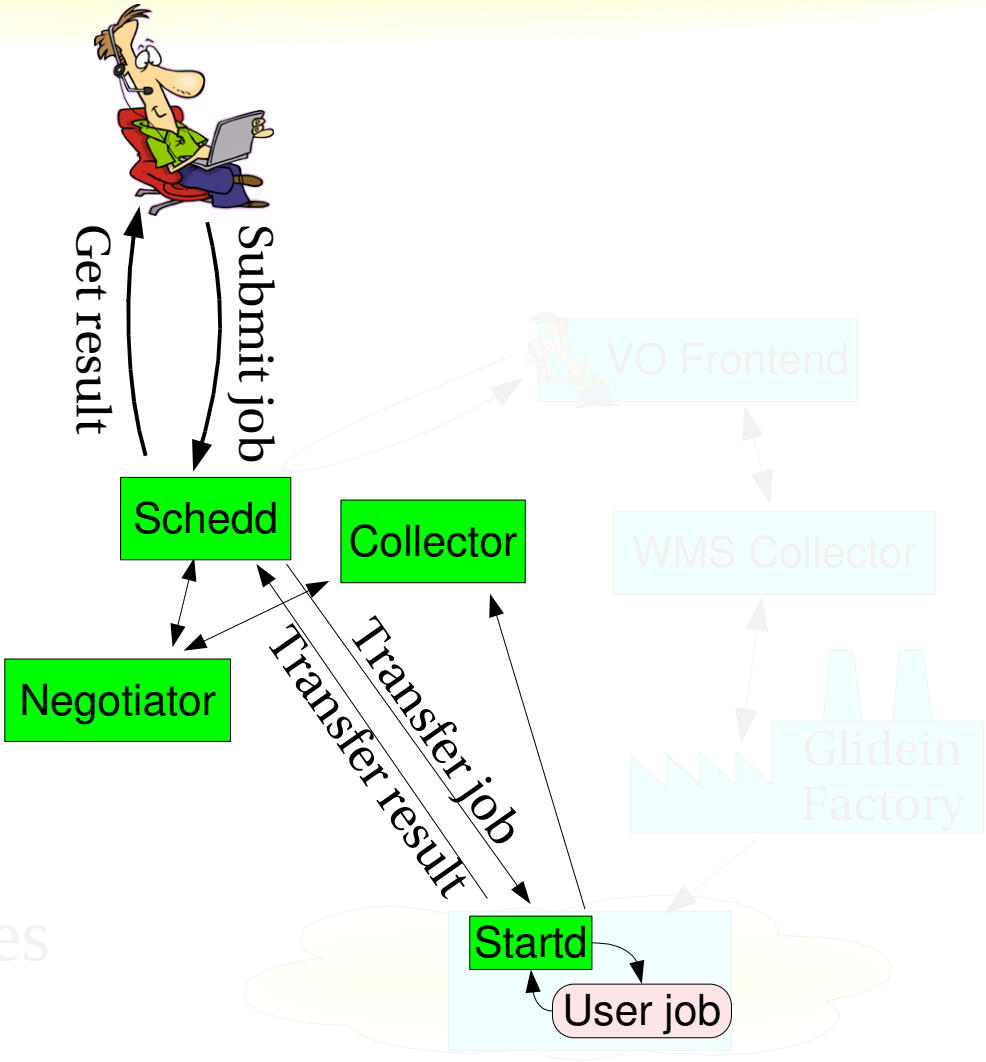
glideinWMS architecture

- glideinWMS composed of six logical pieces:
  - A Condor central manager (collector + negotiator)
  - One or more Condor submit machines
  - A glideinWMS collector
  - One or more VO frontends
  - One or more glidein factories
  - The glideins



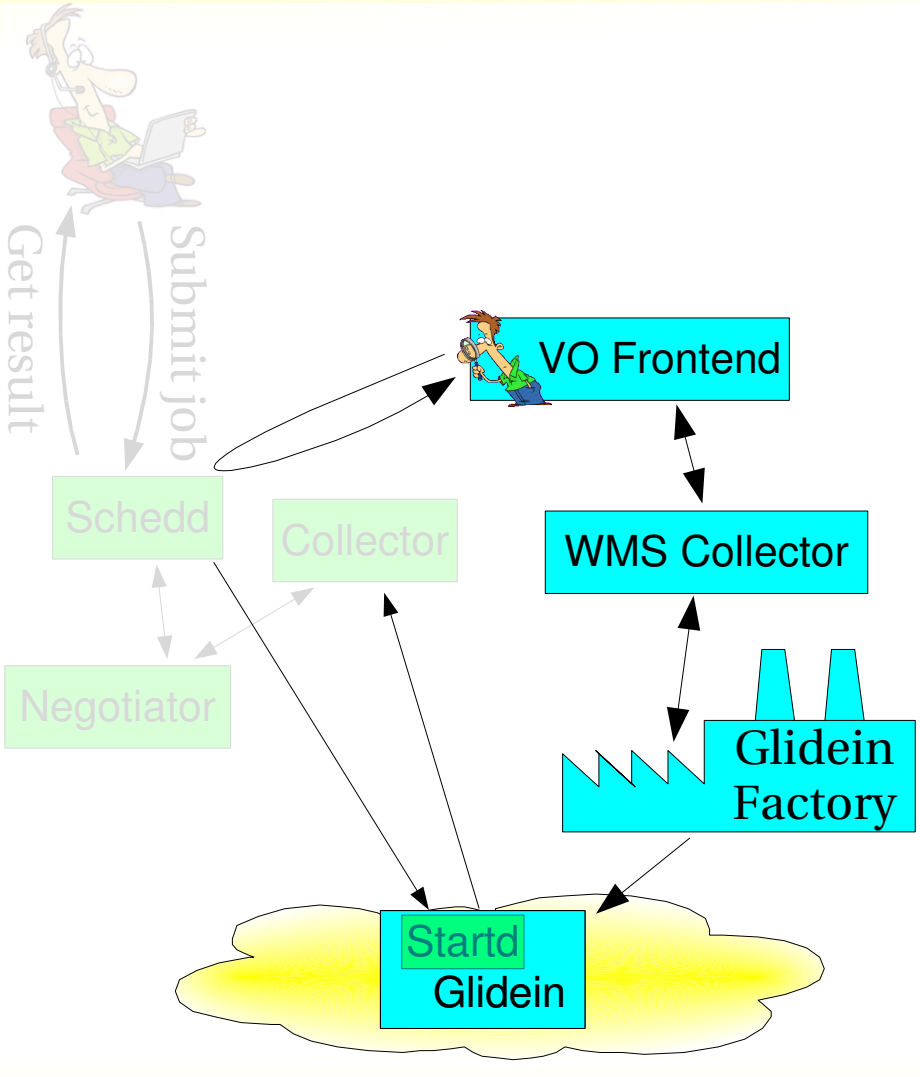
glideinWMS architecture (2)

- glideinWMS composed of six logical pieces:
  - A Condor central manager (collector + negotiator)
  - One or more Condor submit machines
  - A glideinWMS collector
  - One or more VO frontends
  - One or more glidein factories
  - The glideins (startd)



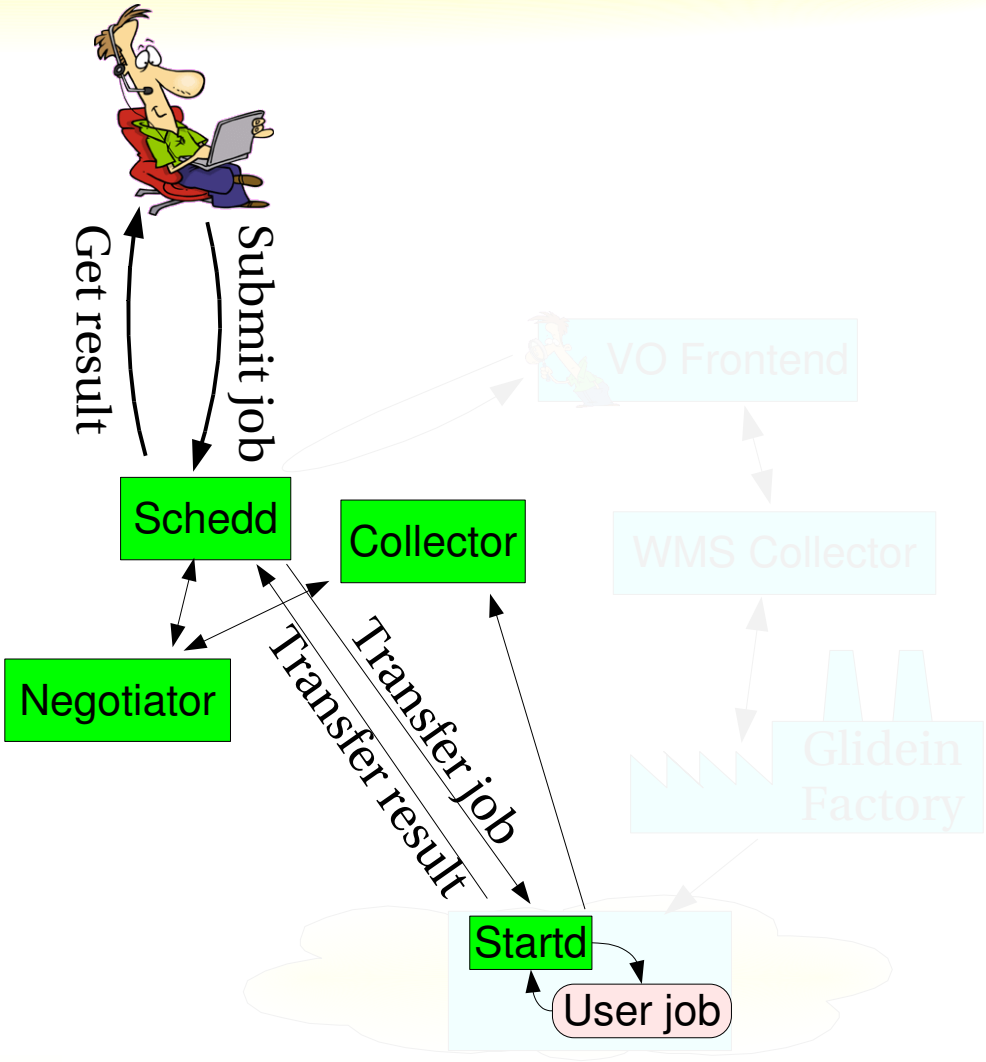
glideinWMS architecture (3)

- glideinWMS composed of six logical pieces:
  - A Condor central manager (collector + negotiator)
  - One or more Condor submit machines
  - A glideinWMS collector
  - One or more VO frontends
  - One or more glidein factories
  - The glideins



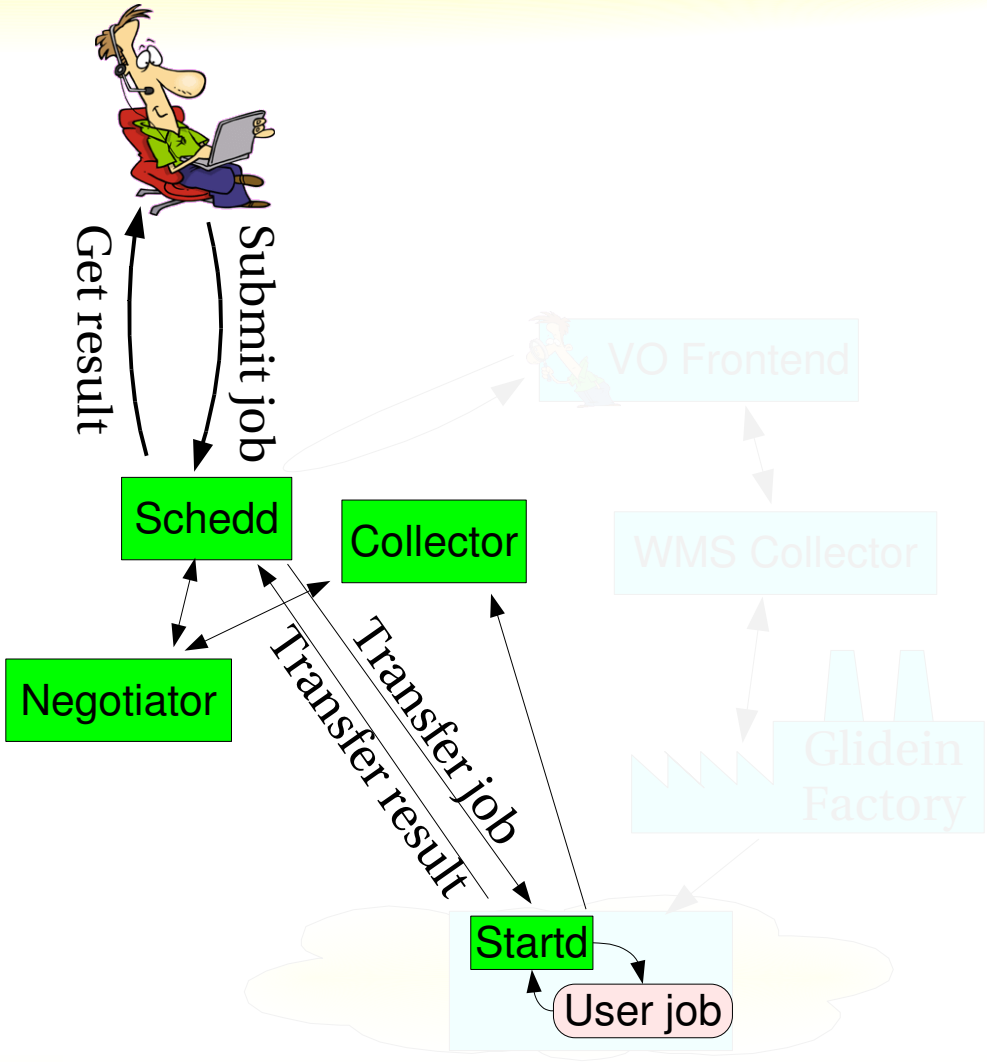
Condor handles user jobs

- A glidein Condor pool is still a Condor pool
  - Just a very dynamic one
- All Condor features available
  - ClassAds
  - Job requirements
  - Fair share
- Users really don't know about the glideinWMS



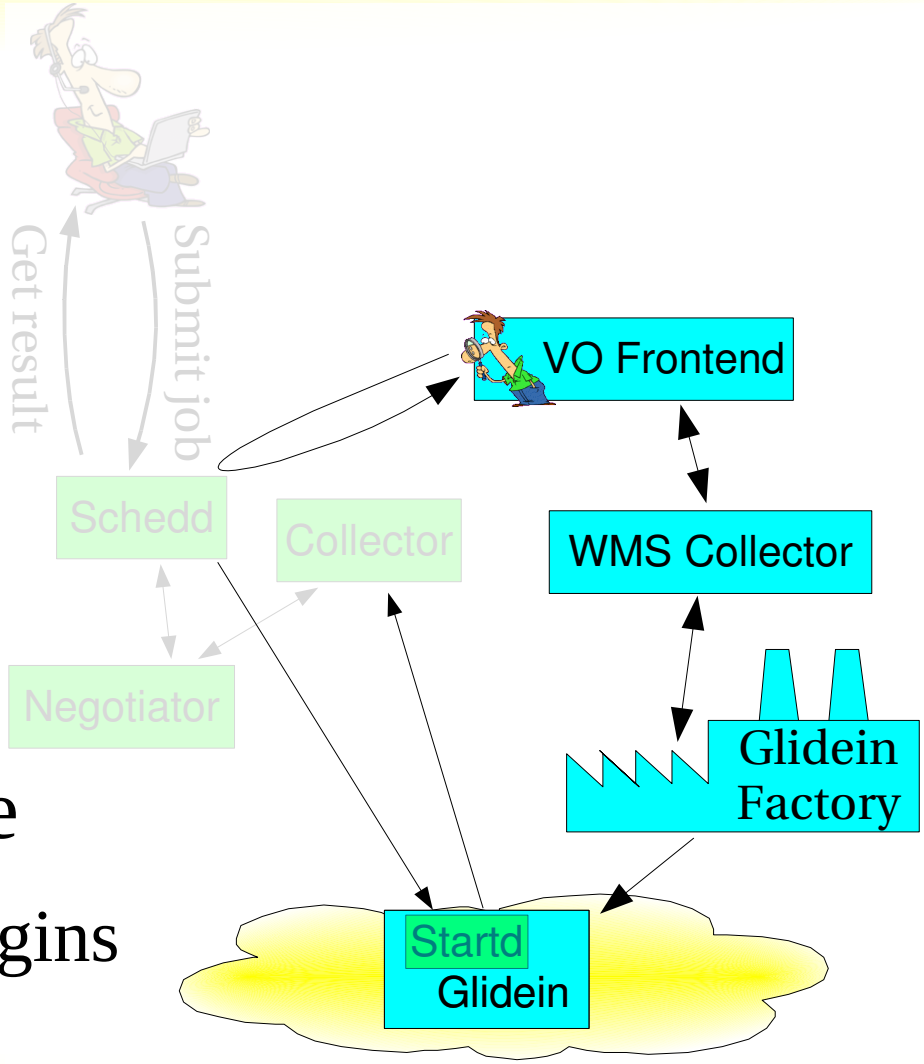
Condor handles user jobs (2)

- Internal VO policy management completely in VO hands
  - Sites only need to handle inter-VO policies
- Condor provides
  - Accounting groups (with quotas)
  - Fair share between users withing a group



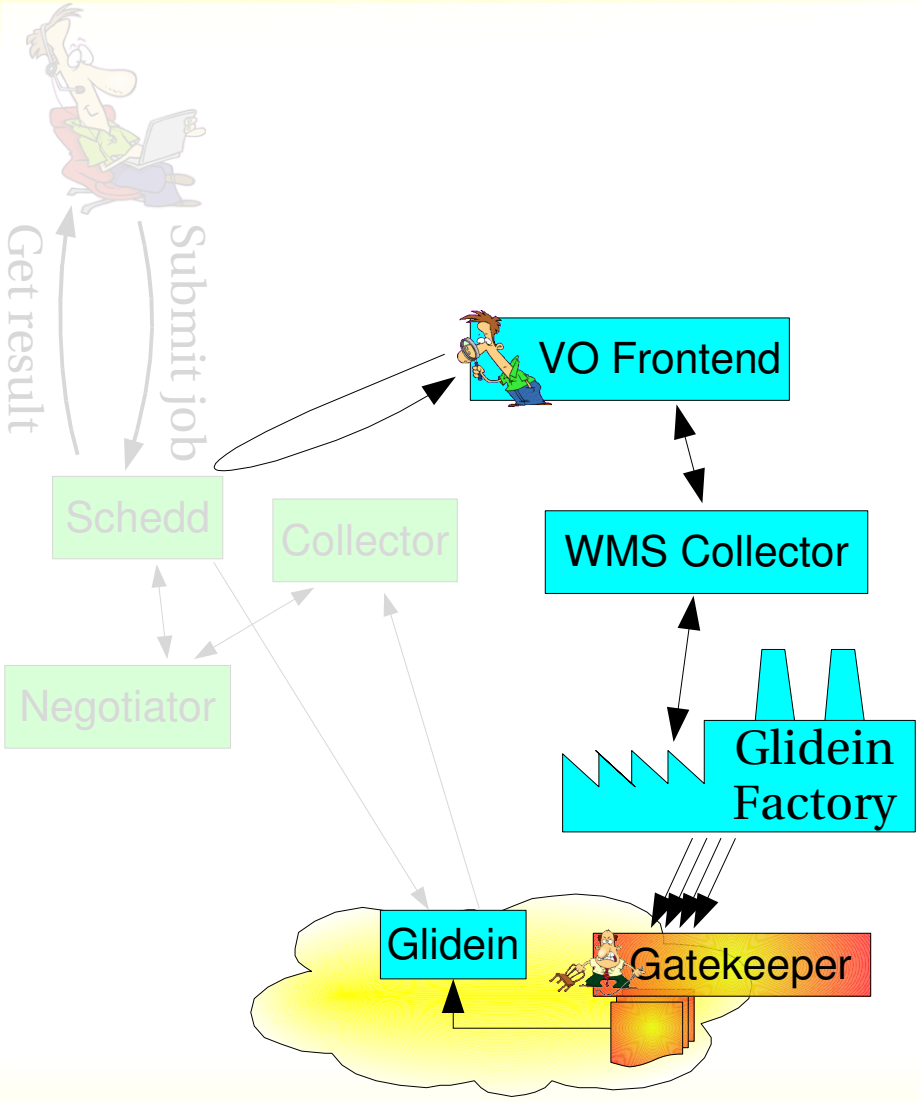
Glidein submission

- glideinWMS processes are responsible **only** for startd startup
  - A glidein just configures and starts it
  - Once started, startd has full control
- Glideins highly customizable
  - Glidein factory allows for plugins



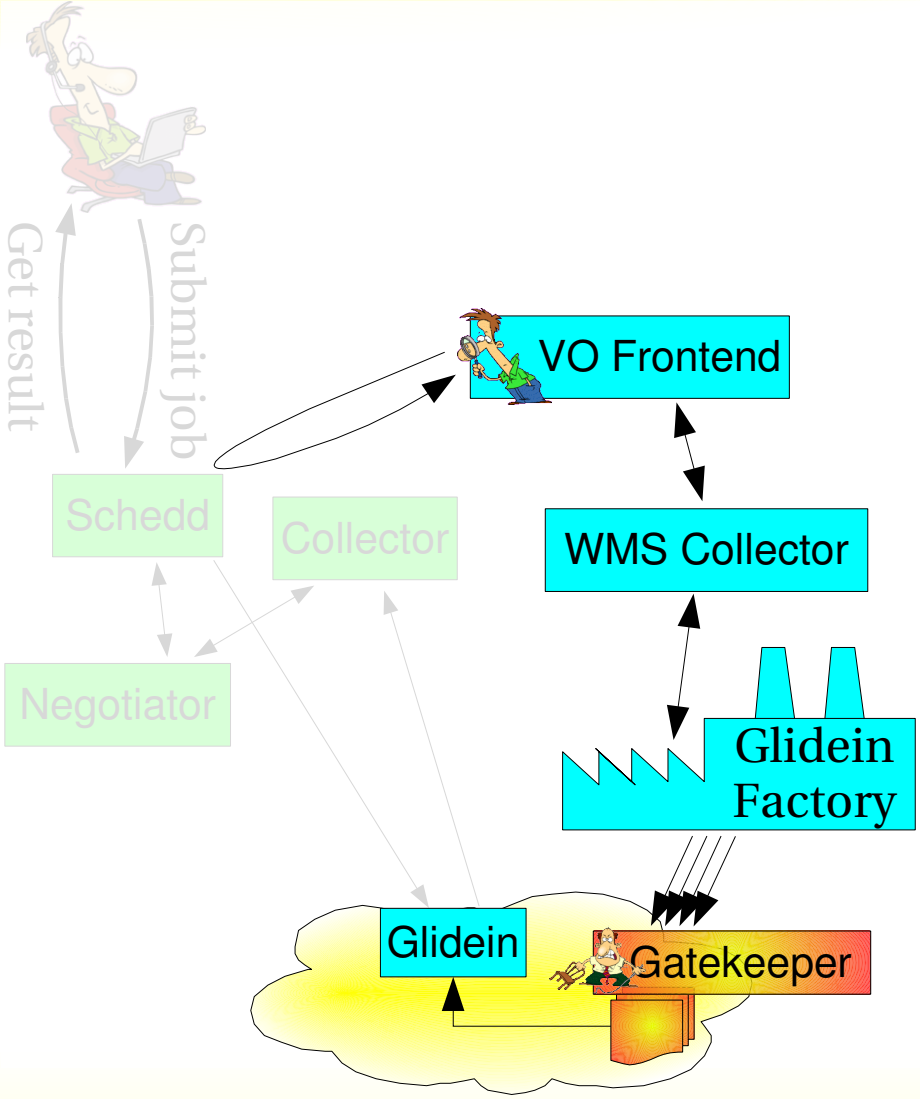
Glidein submission (2)

- Condor-G used for submission to Grid sites
  - Current implementation
- Glidein submission is a collaborative work
  - VO frontend decides how many glideins to submit
  - Glidein factory actually does the submission
  - WMS collector is used for message passing



Glidein submission (3)

- Based on the principle of constant pressure
  - As long as there are enough waiting jobs in the queue, a fixed number of glideins are kept at each suitable Grid site
- Works nicely for systems with lots of waiting jobs
  - Will waste resources on seldom used systems

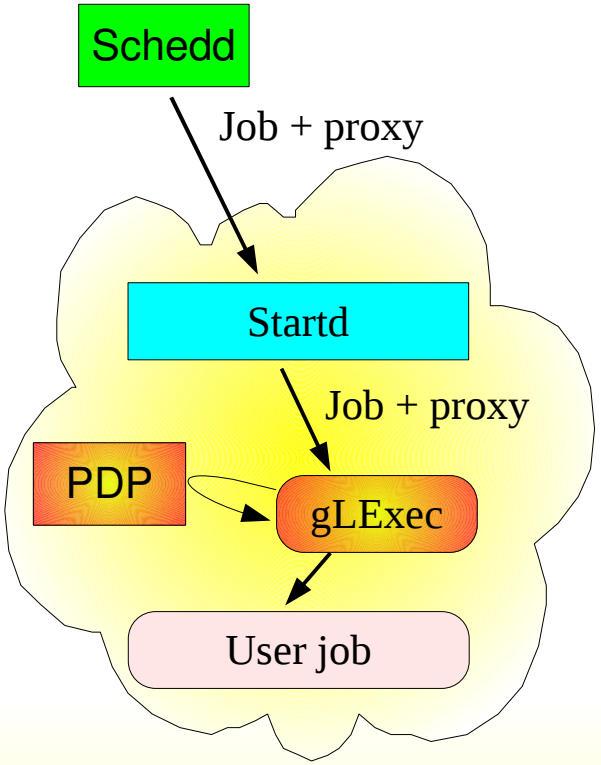


Security considerations

- GlideinWMS **requires** security over the wire
  - WAN network connections cannot be blindly trusted!
- All network traffic features integrity checks
  - Prevents man-in-the-middle attacks
- GSI authentication (X509 certificates/proxies) used for all interactions with Condor daemons over the network
  - Based on access lists (like gridmap file, but with regex)
  - Attributes not used right now
    - But will probably need it

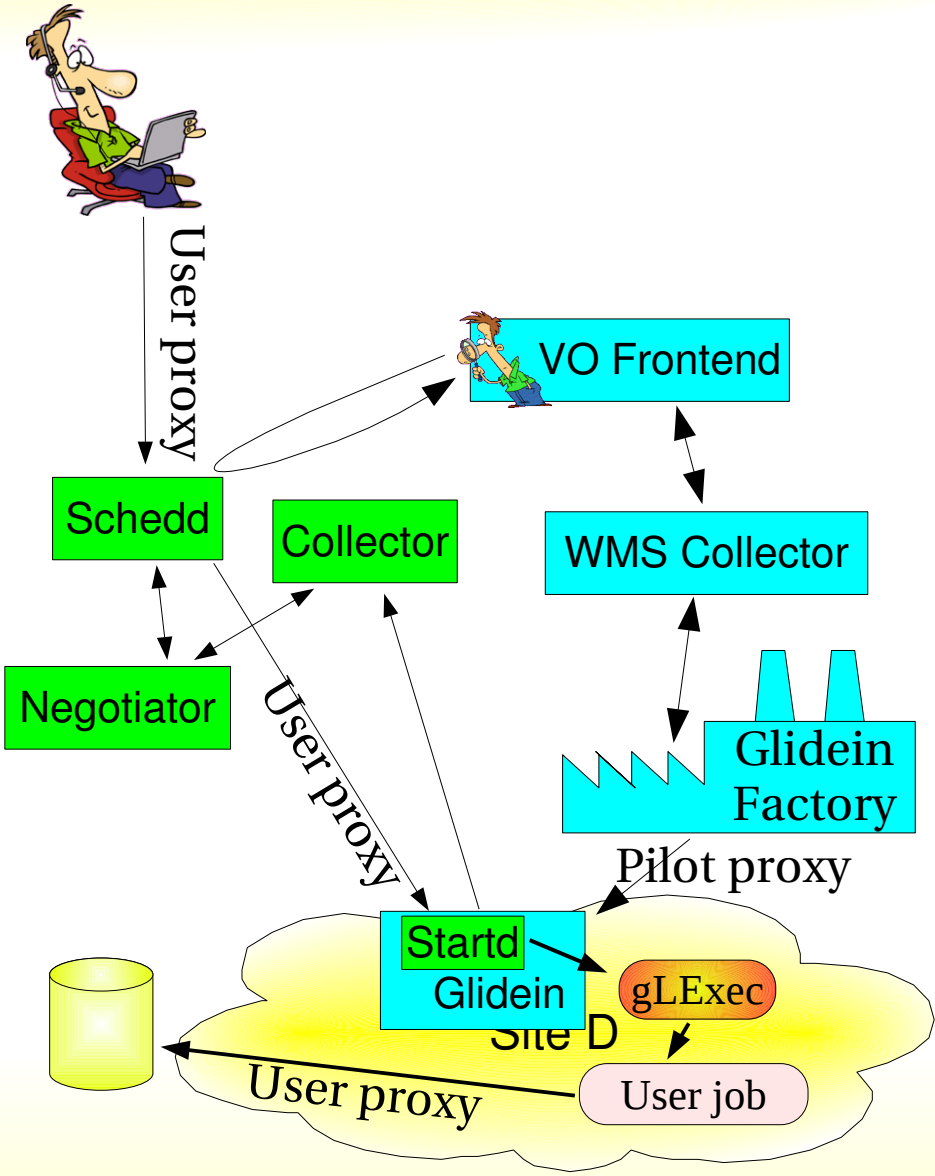
Security considerations (2)

- Startd not running as a privileged user
  - Cannot change UID by itself when starting user job
  - Malicious user job could hijack the startd if running under the same UID
- Condor interfaced to gLExec
  - gLExec allows to change UID given user proxy
  - Startd protected from the user job
- gLExec interfaces with local security infrastructure
  - Enforces site policies
  - When pilots use it



Proxy handling

- A single identity used to submit all the glideins
  - Should have a pilot role
- Condor ships user proxy to worker node
  - User can use it access other resources
- Proxy lifetime management and renewal not handled by glideinWMS
  - Users expected to do it



User job monitoring

- Good monitoring a must for most users
- Condor provides a plethora of monitoring tools
  - Most useful are condor_q and condor_status
  - Third parties provide additional Condor monitoring tools
- glideinWMS provides tools for pseudo-interactive monitoring
  - ls, cat, top on the worker nodes
- The glidein factory also maintains a basic Web based graphical view
  - plus machine readable XML and rrd data

glideinWMS monitoring

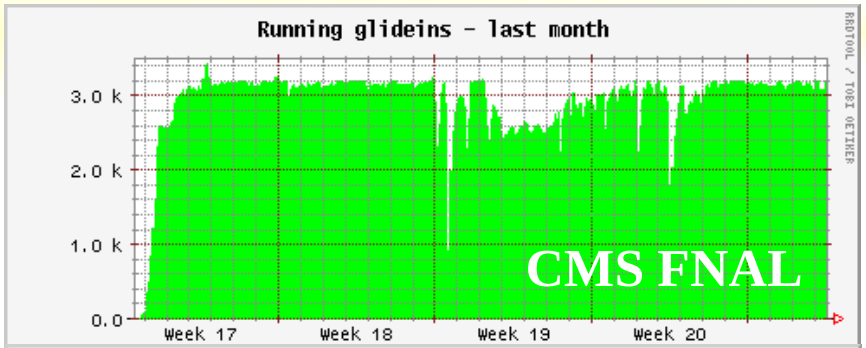
- Good monitoring a must for most administrators, too
- Condor-G provides some tools
  - Mostly condor_q
- The glidein factory maintains a rich Web based graphical view
  - plus machine readable XML and rrd data
- Glideins return comprehensive logs
  - Useful for low level debugging
  - But require some expertise to browse though

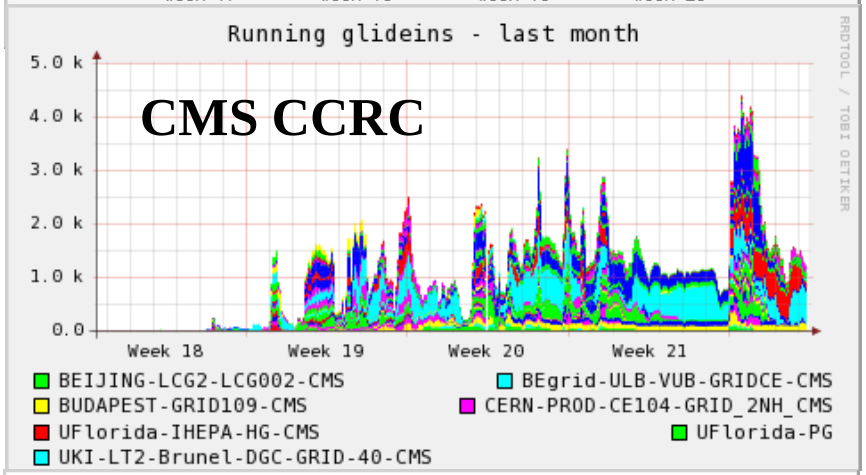
Status of glideinWMS

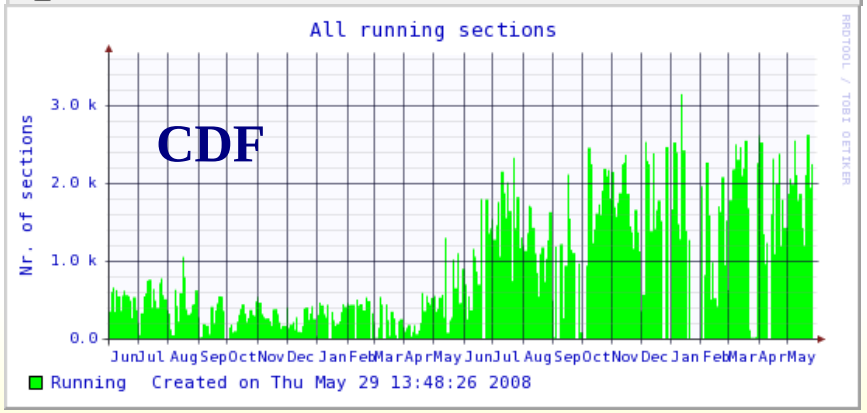
- Version 1.2.1 released May 30th
- Should be usable out of the box for most users
  - CMS is using it since v1.1
- Still in active development phase
  - More monitoring
  - More automated error checking
  - More automated error recovery
  - Better integration with other systems
- Condor also an evolving product

Glidein deployments in HEP

- CMS using glideins for production jobs at FNAL
  - Recently across all seven T1s
- CMS used them for analysis jobs in CCRC08
  - Across 40 T2s
- CDF and MINOS using them for user analysis





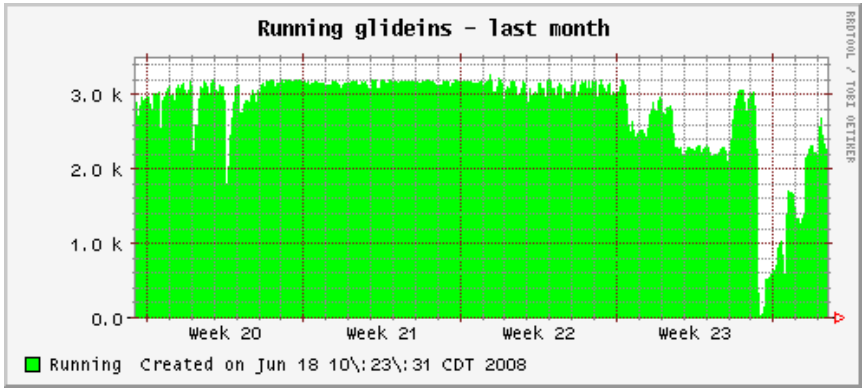


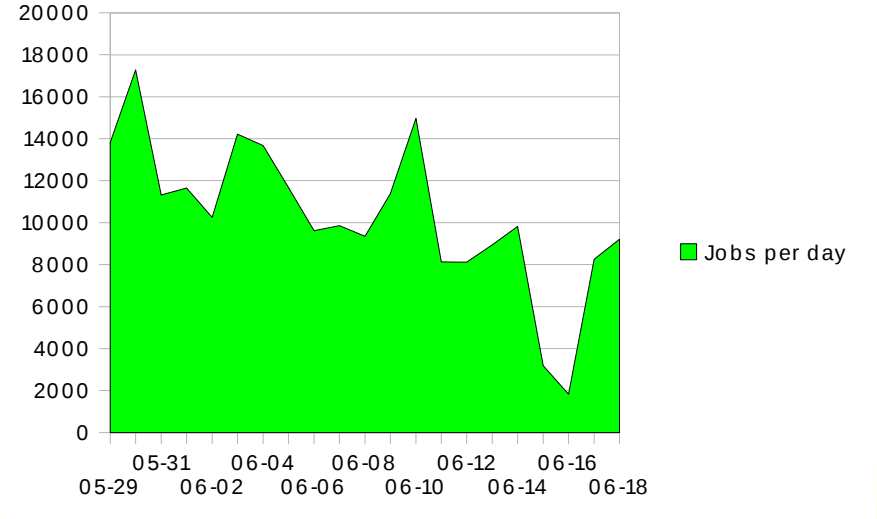
glideinWMS in numbers

- Deployed systems
  - CMS@FNAL stable 3k glideins for the past 6 months
  - CMS@CCRC up to 4k glideins over 40 sites globally
  - CDF average 2k glideins with 100s of users for past 2 years (by using the GlideCAF)
- glideinWMS Tested on a dedicated test pool, scaled without major problems to
  - 10k glideins at any time
  - 100k user jobs queued

CMS @ FNAL experience

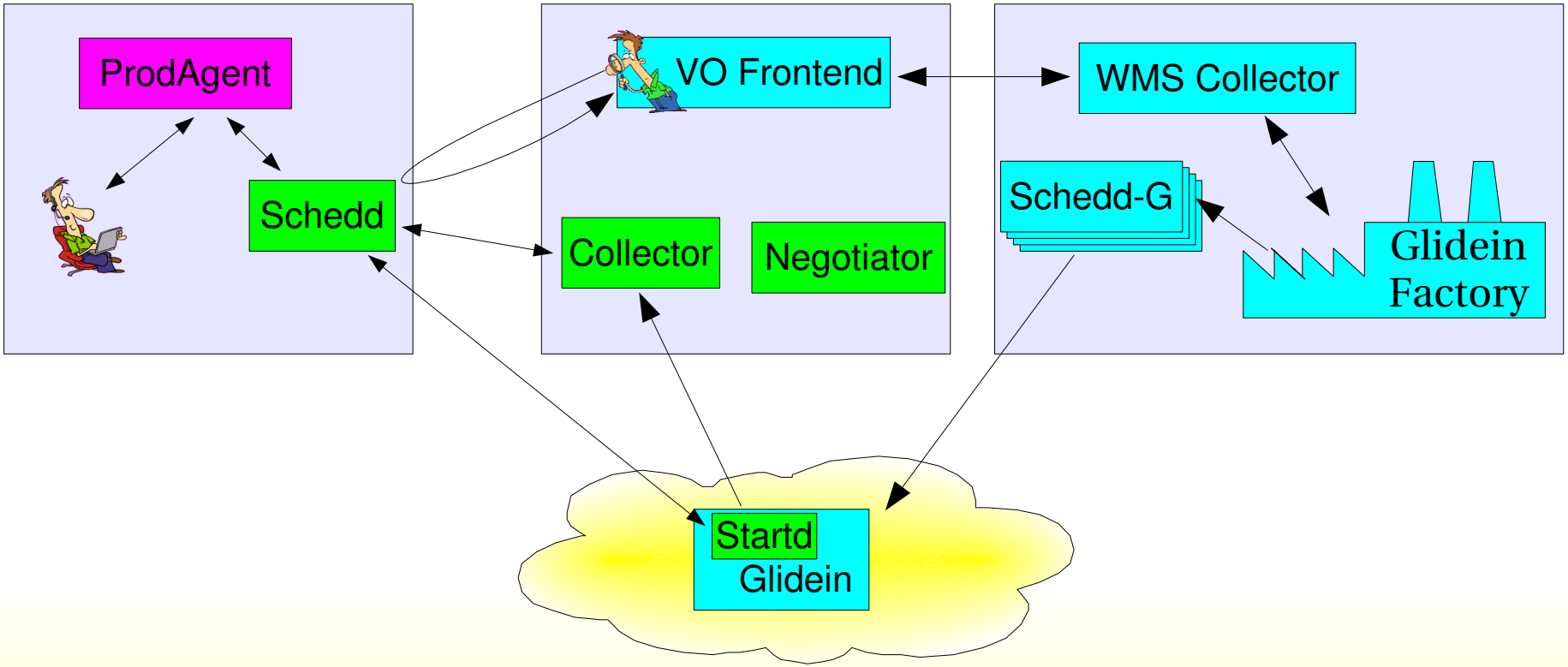
- Using ProdAgent to submit jobs to local schedd
- Gliding into a single site
  - over LAN
  - Using 3 CEs
- Saturating the FNAL T1
  - ~3200 slots
- Quick job turnaround
  - >10k jobs per day on average
  - >150k jpd during CSA07
- Few failures
  - Mostly storage related





CMS @ FNAL setup

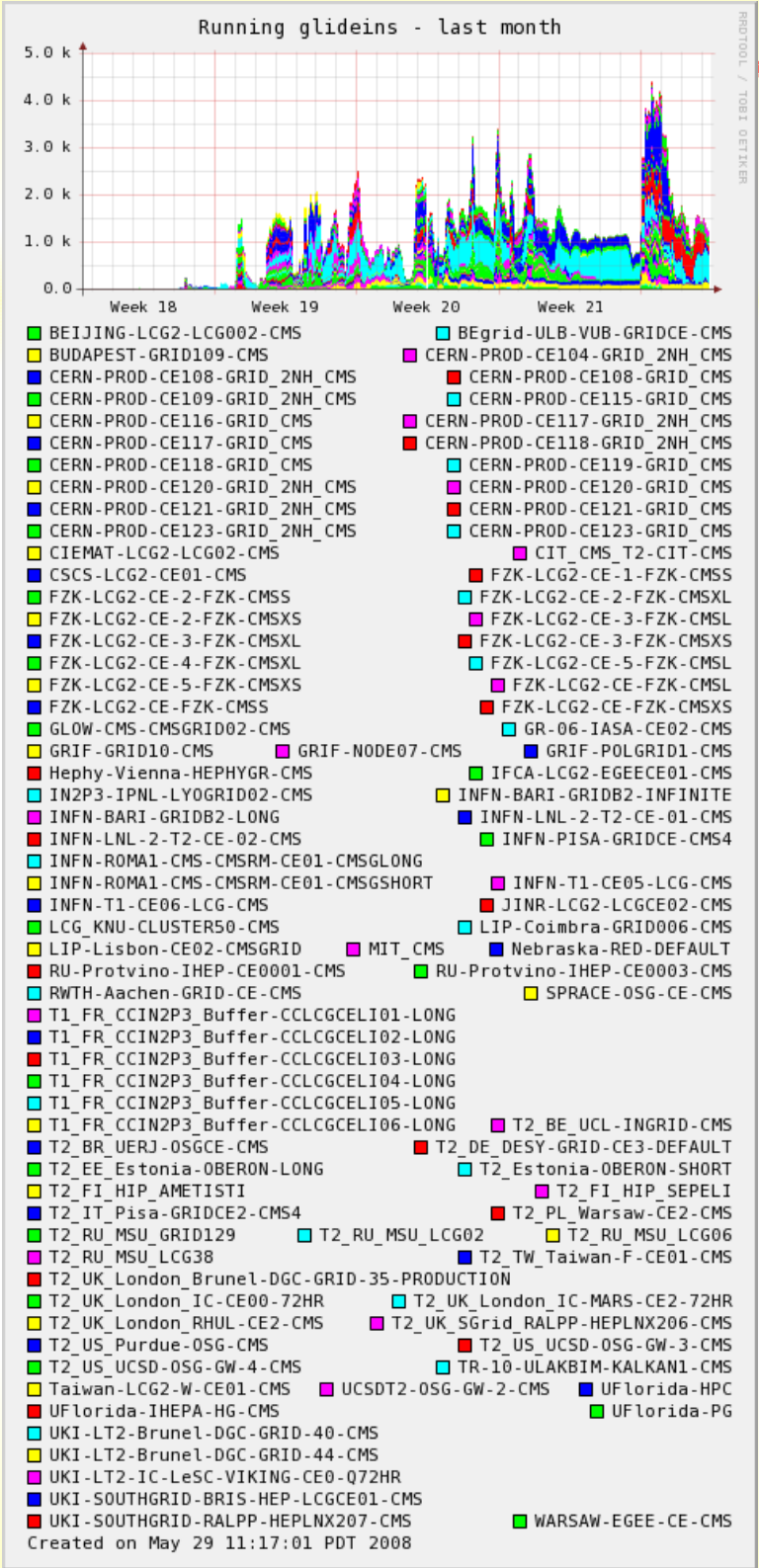
- 3 nodes used (+Grid worker nodes)
- No GCB (LAN)
- No gLExec (only production team, single logical user)



Fermilab

CMS CCRC08 experience

- Using CRAB to submit to the local schedd(s)
- Submitting to 40 T2s
  - All over the world
  - OSG, EGEE and Nordugrid (a first for CMS)
- Ran 300k jobs over 4 weeks
  - Mix of CPU intensive and IO intensive jobs

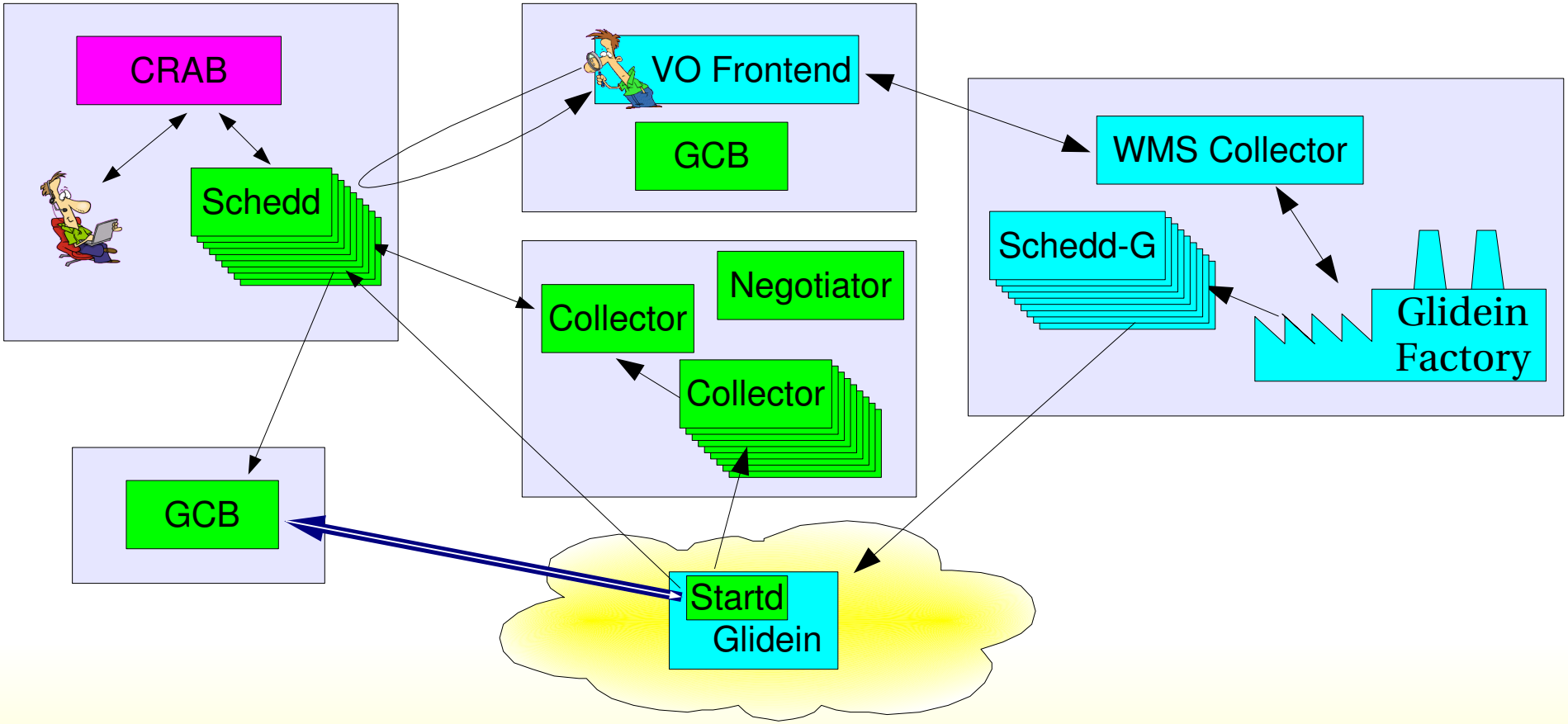


CMS CCRC08 experience (2)

- Latencies have bitten us
  - Condor uses blocking connections for security handshake
    - Condor working on fix
  - For CCRC solved by using multiple condor daemons
    - Hierarchy of collectors
    - Multiple schedds
- Still very successful
  - CMS pleased with the experience

CMS CCRC08 setup

- 5 nodes used (+Grid worker nodes)
- No gLExec (Only one CCRC user)



CMS glidein plans

- Production over all T1s using glideinWMS should start soon (from FNAL)
  - Prototype in place
  - Need to sort out operational issues
- But should be pretty straightforward
- Still only one logical user

CMS glidein plans (2)

- UCSD offered to host an analysis service
  - Serving physicists
  - Using the CRABServer
  - Using gLExec
  - Expected to be setup over the summer
- Proxy handling may become an issue
- Using a single pilot proxy may not be the best strategy
  - Problems when competing with PUSH-based WMSes (like gLite WMS and plain Condor-G)

Conclusions

- Bare-bones Grid difficult to use
  - Glideins can hide the Grid complexity and make it look as a uniform computing pool
- CMS has used glideinWMS for the past 6 months
  - Great success at FNAL
  - Good results in tests over T1s and T2s
- Glideins allow for flexible VO policy managing
  - CDF had excellent experience over past few years
  - CMS will go after this soon

Backup Slides

glideinWMS contact info

GlideinWMS home page:

http://www.uscms.org/SoftwareComputing/Grid/WMS/glideinWMS/

Condor home page:

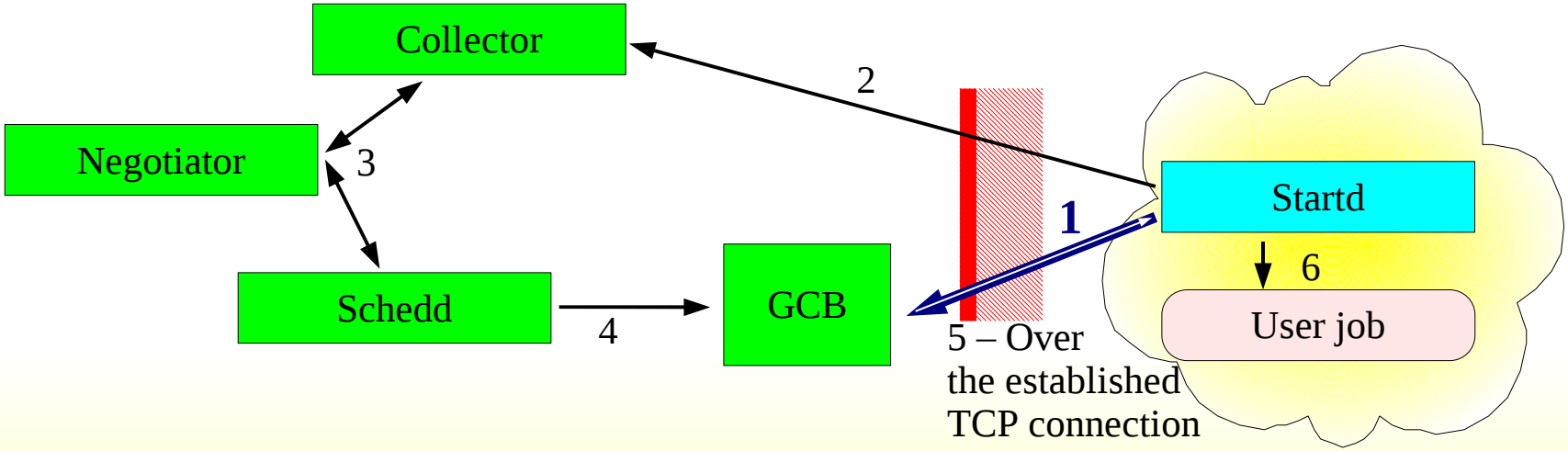
http://www.cs.wisc.edu/condor/

email: sfiligoi@fnal.gov

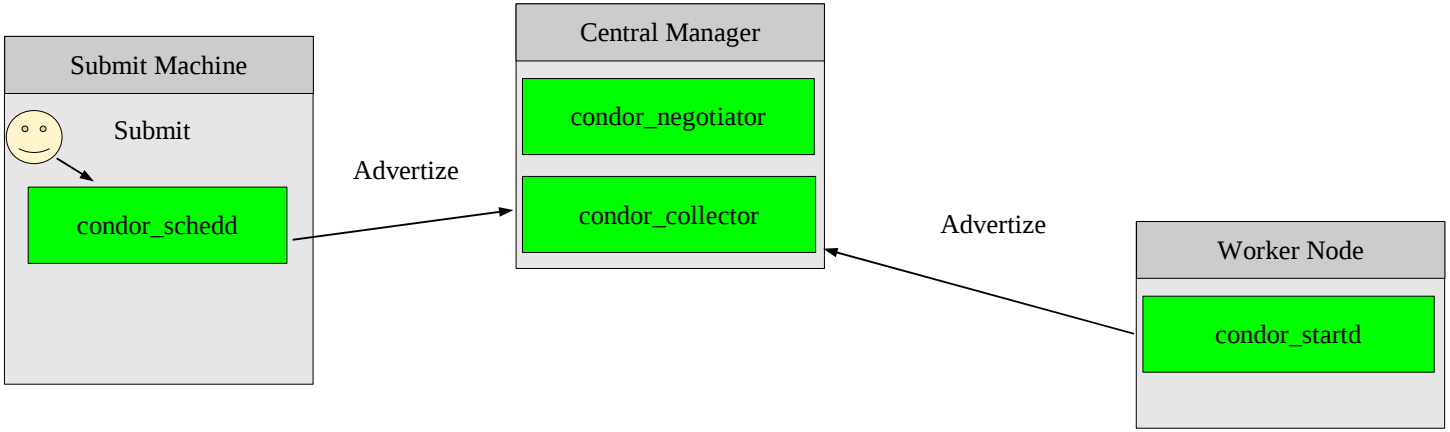
Working over the firewalls

http://www.cs.wisc.edu/condor/gcb/

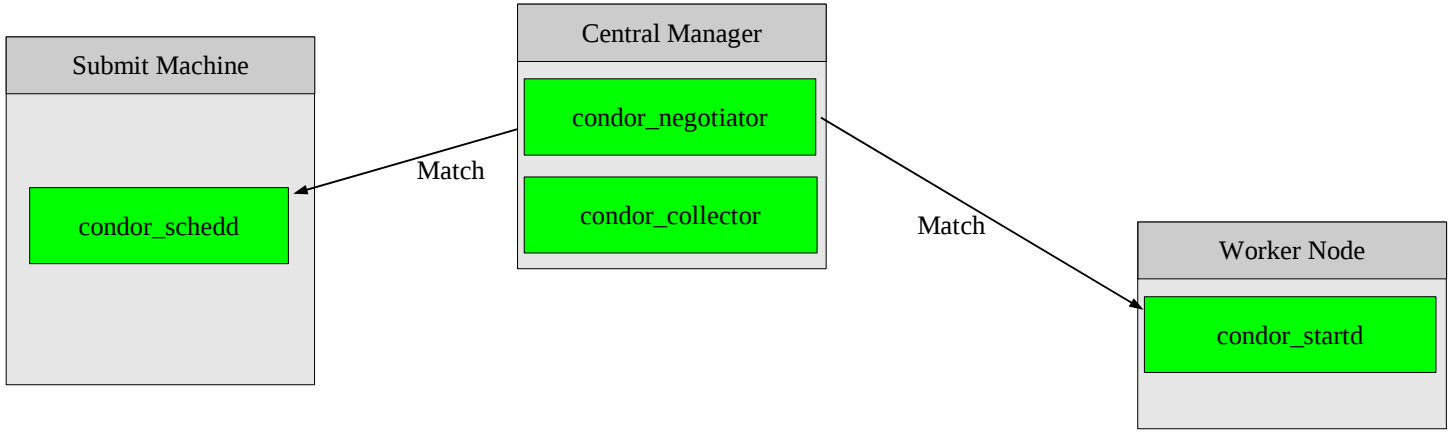
- Condor uses two-way communication
  - But incoming connection often blocked by Grid sites
- Can use Condor GCB (Generic Connection Broker) to make all communications one-way
  - By opening a long lived TCP connection
  - Outgoing connectivity always needed



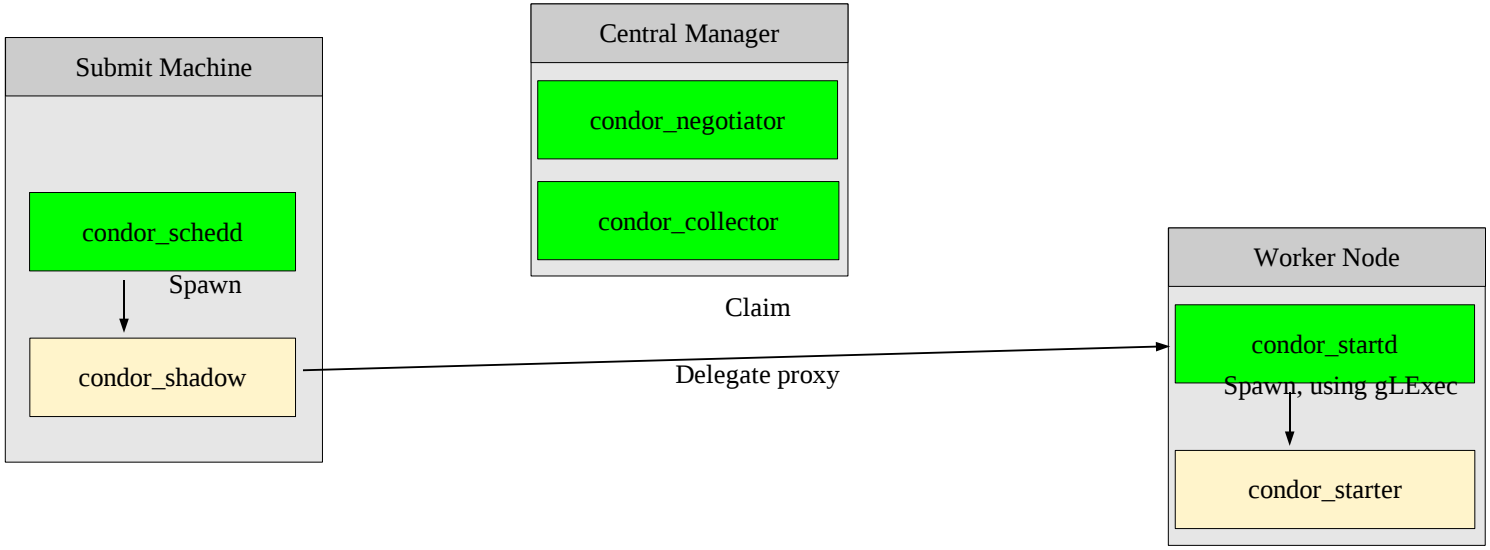
Condor Internals



Condor internals



Condor internals



Condor Internals