

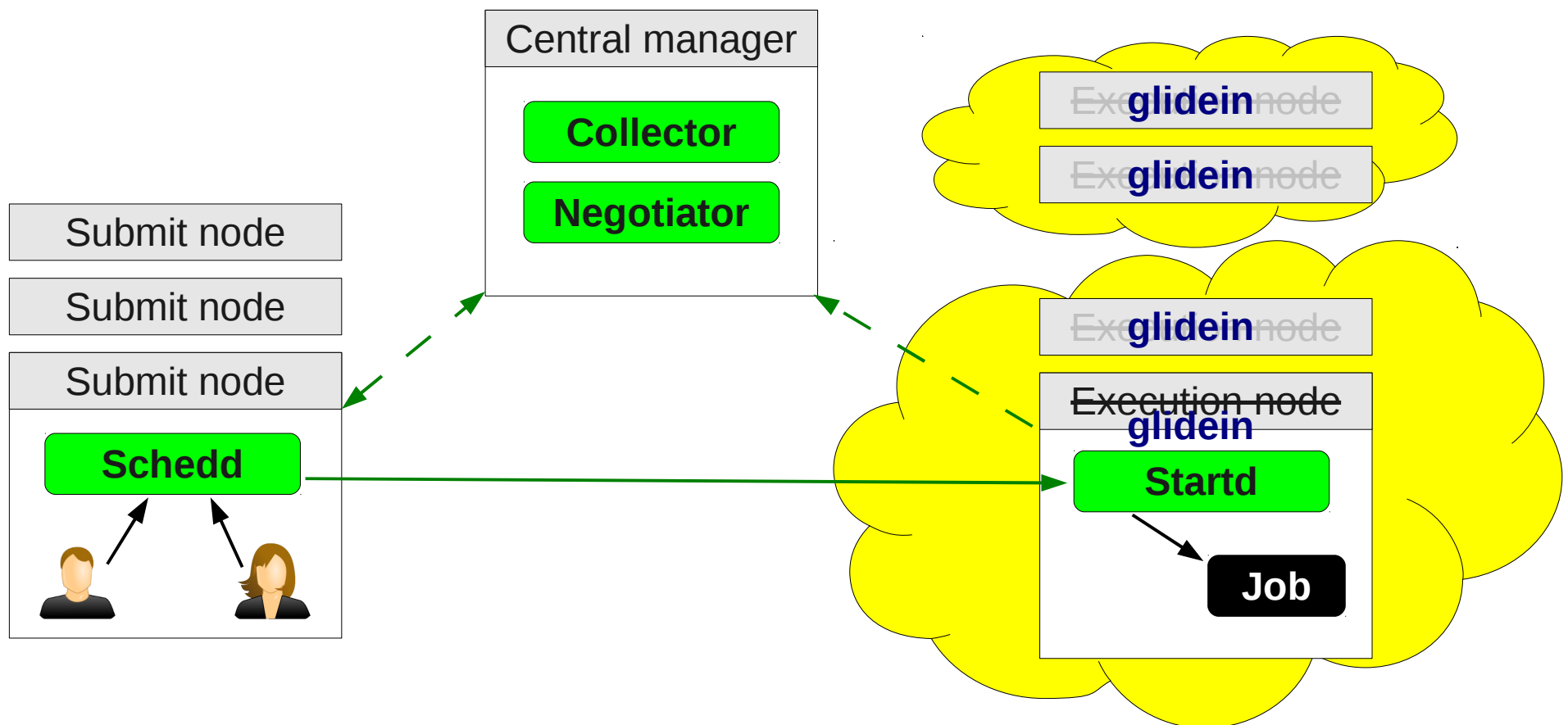
# glideinWMS Training @ UCSD

## **Glidein startup Internals**

by Igor Sfiligoi (UCSD)

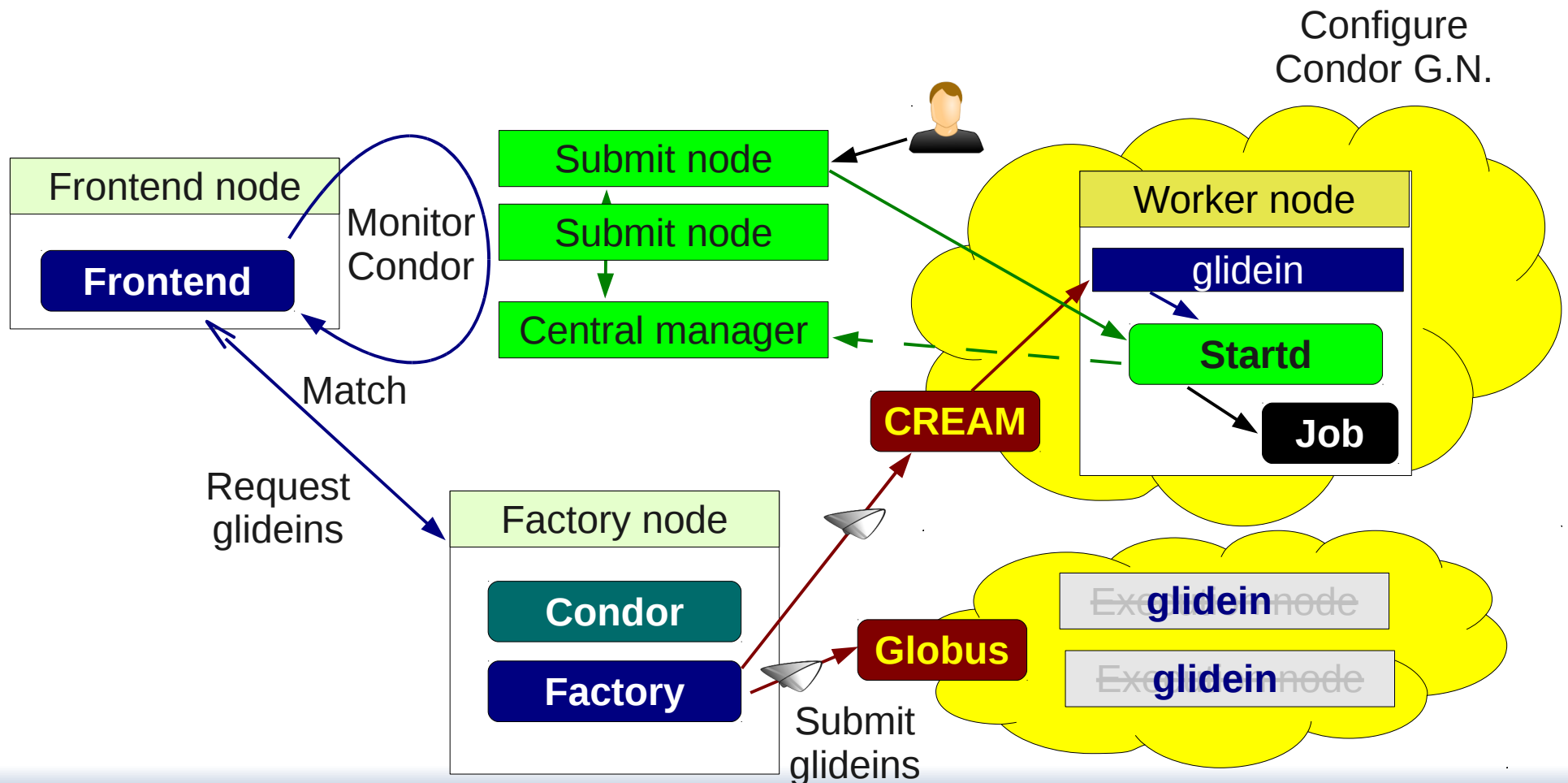
# Refresher - What is a glidein?

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



# Refresher – Glidein startup

- glidein\_startup configures and starts Condor



# glidein\_startup tasks

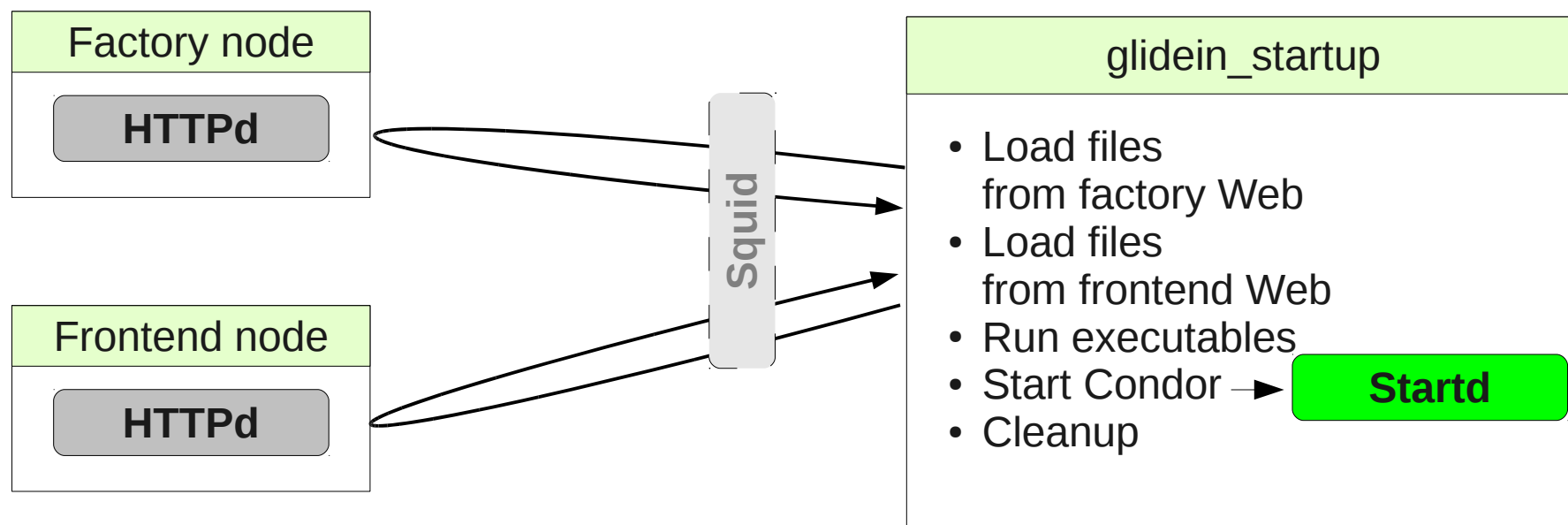
- Download scripts, parameters and Condor bins
- Validate node (environment)
- Configure Condor
- Start Condor daemon(s)
- Collect post-mortem monitoring info
- Cleanup

# Work area

- Before writing any files, a work area must be chosen
  - CWD often not the right choice
- Where to cd into provided as an argument
- A unique tmp dir created there
  - So several glideins running on the same node don't step on each other toes

# Downloading files

- Files downloaded via HTTP
  - From both the factory and the frontend Web servers
  - Can use local Web proxy (e.g. Squid)
  - Mechanism tamper proof and cache coherent



# URLs

- All URLs passed to `glidein_startup` as arguments
  - Factory Web server
  - Frontend Web server
  - Squid, if any

# Security measures

- Using SHA1 hashes
  - Each file has a SHA1 hash associated with it
- The hashes delivered in a single file
  - List of (fname,hash) pairs
- The hash of the hash list is delivered as a parameter to `glidein_startup`
  - This is guaranteed to be secure with GRAM



# Cache coherence

- Files never change, once uploaded to the Web area
- If the source file changes, a file with a different name is created on the Web area
  - Essentially `fname.date`
- There is a logical to physical name map file
  - `file_list.id.lst`
  - The id of this list is passed as an argument to `glidein_startup`

# Node validation

- Run scripts / plugins provided by both factory and frontend
  - The list of files that are scripts vs “regular” files is separate, so obvious for glidein\_startup
- If a script returns with `exit_code != 0`, glidein\_startup stops execution
  - Condor never started → no user jobs ever pulled
  - Will sleep for 20 mins → blackhole protection

# Condor configuration

- Condor config values coming from files downloaded by glidein\_startup
  - Static from both factory and frontend config files
- Scripts can add, alter or delete any attribute
  - Based on dynamic information
  - Either discovered on the WN or by combining info from various sources
- More details at [http://tinyurl.com/glideinWMS/doc.prd/factory/custom\\_scripts.html](http://tinyurl.com/glideinWMS/doc.prd/factory/custom_scripts.html)

# Glidein lifetime

- Glideins are temporary resources
  - Must go away after some time
- We want them to go away by their own will
  - So we can monitor progress and clean up
- Condor daemons configured to die by themselves
  - Just need to tell them when
  - Mechanism changed between 2.5.3 and 2.5.4, but policies still the same

# Glidein lifetime policies

- Two termination triggers
  - Reach End-of-life – glidein\_max\_walltime
  - Unused for too long - glidein\_max\_idle
- Running jobs may be killed if EOL reached
  - Resulting in badput
- Reaching max\_idle another waste
  - Especially when 0 jobs ran

# Monitoring

- After Condor terminates, the logs are mined for job information
  - Number jobs ran
  - Job exit codes, exist by signal?
  - Wasted time
- Condor logs also compressed and copied into stderr
  - Only way to get them to the factory
  - GRAM does not guarantee files returning

# Cleanup

- glidein\_startup will remove all files before terminating
  - Unless killed by the OS, of course
- The tmp dir is removed as well
- Only stdout and stderr survive the glidein

# Factory provided values

- Factory provides defaults
  - From glideinWMS/creation/web\_base/condor\_var\*
- Factory also provides
  - Max\_walltime and max\_idle
  - Glxec location
  - Condor binaries (including os arch and version)
  - High level network setup (CCB, TCP, etc.)
- Factory DOES NOT provide
  - Collector location



# Pointers

- The official project Web page is <http://tinyurl.com/glideinWMS>
- glideinWMS development team is reachable at [glideinwms-support@fnal.gov](mailto:glideinwms-support@fnal.gov)
- OSG glidein factory at UCSD  
<http://hepuser.ucsd.edu/twiki2/bin/view/UCSDTier2/OSGgfactory>  
[http://glidein-1.t2.ucsd.edu:8319/glidefactory/monitor/glidein\\_Production\\_v4\\_1/factoryStatus.html](http://glidein-1.t2.ucsd.edu:8319/glidefactory/monitor/glidein_Production_v4_1/factoryStatus.html)

# 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