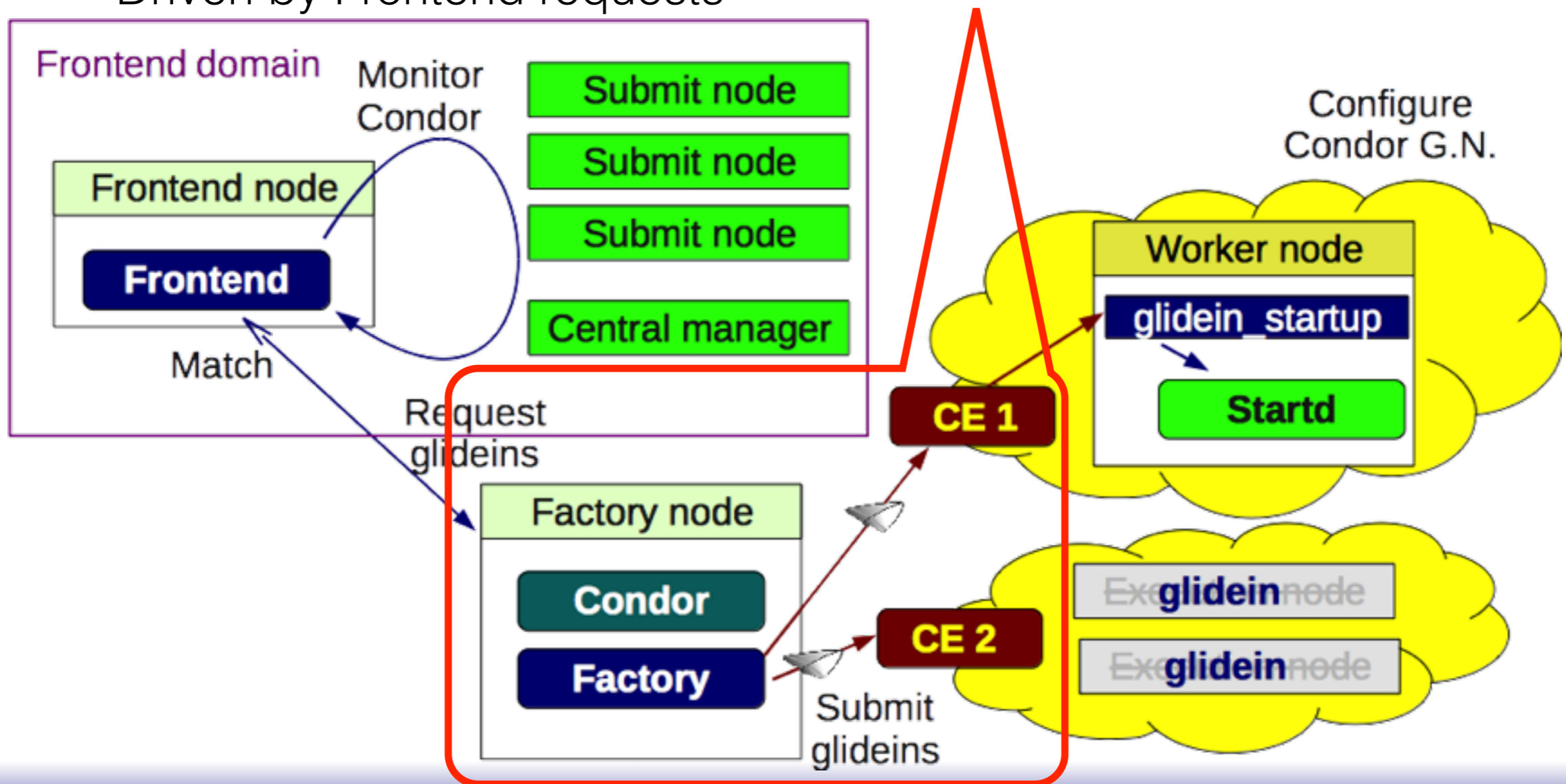


glideinWMS factory Internals

by Igor Sfiligoi, Jeff Dost (UCSD)

Refresher - glidein factory

- The factory knows about the grid and submits glideins
 - Driven by Frontend requests



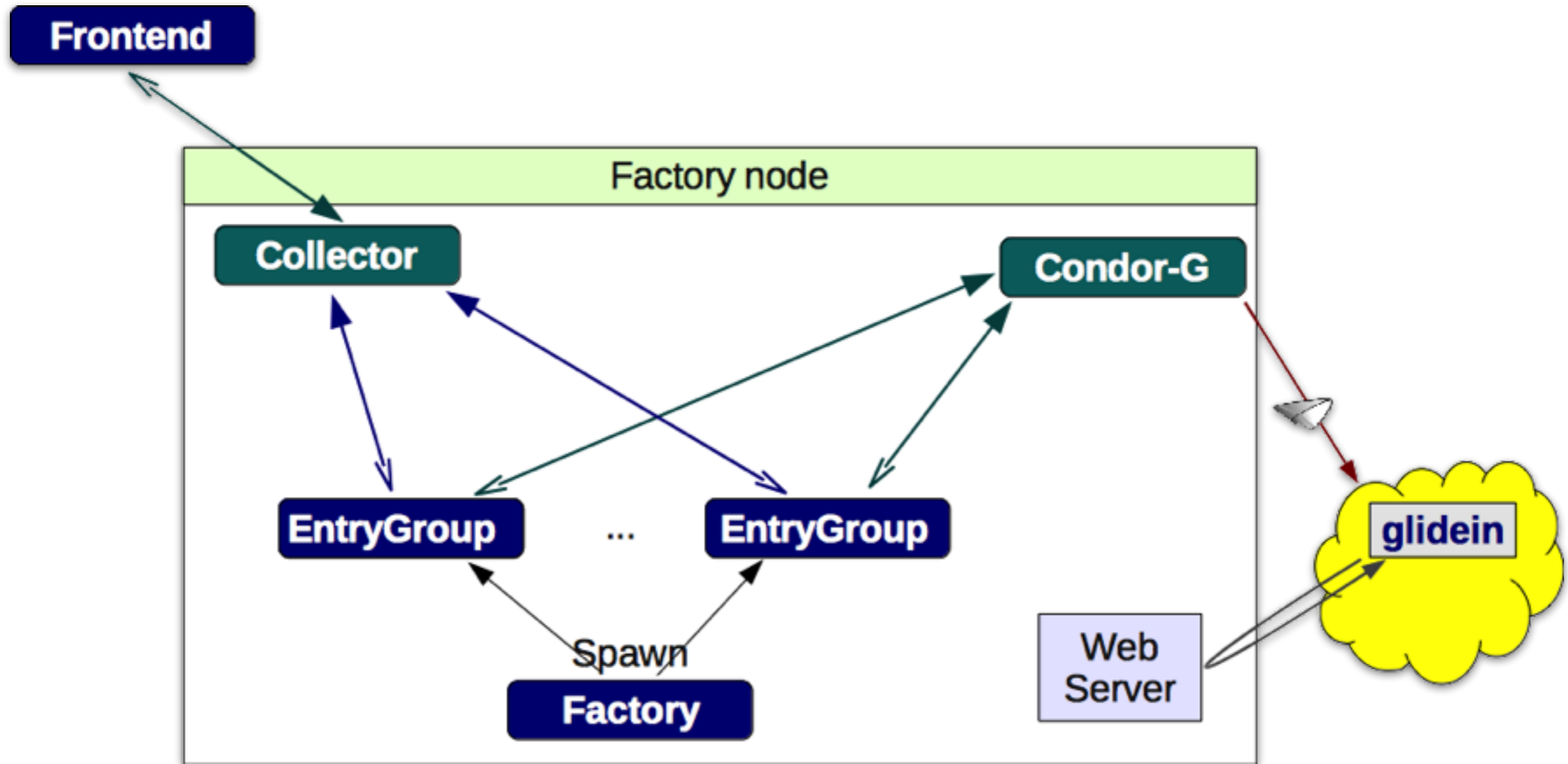
Overview

- Factory architecture
- Entry logic
- Security considerations

Factory architecture

- The factory is composed of:
 - The Condor collector – used for message passing
 - The glideinWMS factory proper
 - Condor-G – does the actual Grid submission
 - Web server – deliver code and data to glideins + monitoring
- The glideinWMS factory itself composed of:
 - EntryGroup processes – do the real work
 - Master factory – controls the EntryGroups and aggregates monitoring

Factory arch - picture



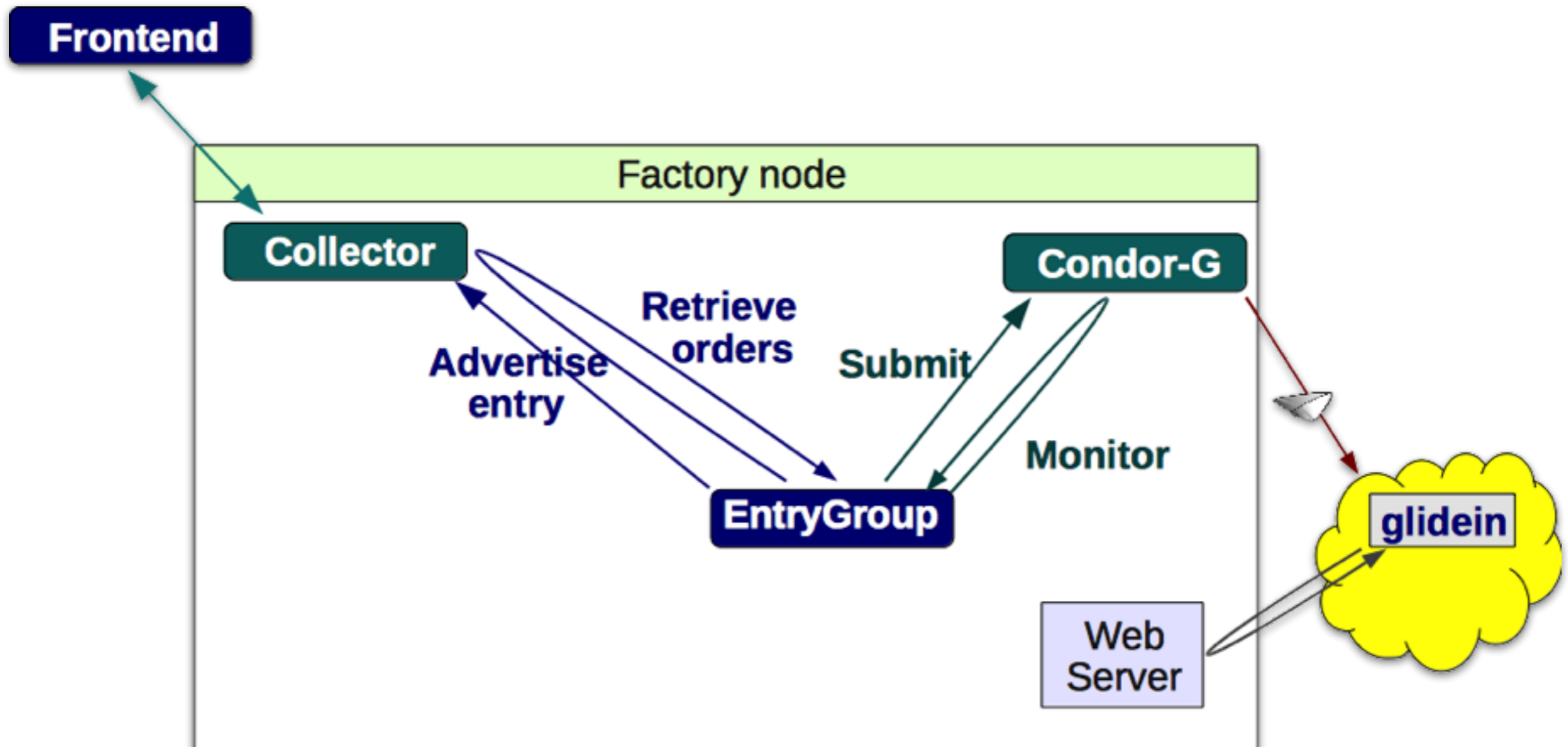
Factory processes

- Real work performed by EntryGroup processes
 - **n** glideFactoryEntryGroup.py processes are forked
 - **n** is configurable as **entry_parallel_workers** in config
 - each EntryGroup handles **N / n** entries where **N** is total number of entries in the factory
- They are controlled by master Factory
- glideFactory.py
 - Starts the other processes
 - Aggregates monitoring

EntryGroup logic

- Essentially a slave
 - Will do what a frontend tells it to
- Uses the Collector for communication
 - Advertise existence of **N** / **n** entries and their attributes
 - Polls the collector for commands
 - Everything is ClassAd based
 - All security is implemented in the Collector
- Glideins are submitted via Condor-G
 - Then EntryGroups just monitor them

EntryGroup - picture



EntryGroup loop

- EntryGroup in continuous loop
 - Advertise → Read → Submit → Monitor → Advertise...
- The monitoring information is stored
 - In internal log files,
 - Web accessible location, and
 - Monitoring ClassAds

Entry Attributes - Essential

- The Entry Classad has many attributes
 - A few attributes are essential and are required to submit glideins
- Essential attributes:
 - Condor-G: gridtype, gatekeeper, submit attrs (rsl)
 - Resource Type: max_walltime, glexec, os_type
- Required informational attributes:
 - Name — must be unique
 - Site name – does not need to be unique, but used to group multiple CEs as part of same site for matchmaking

Entry Attributes - Optional

- Entry can publish any arbitrary attribute
- Recommended ones:
 - Supported VOs
 - Well defined resource name: CMSSite, Infosys ResourceName, etc.
- Plus any other info the Factory admin thinks the Frontend may be interested in
 - In particular anything to facilitate matchmaking

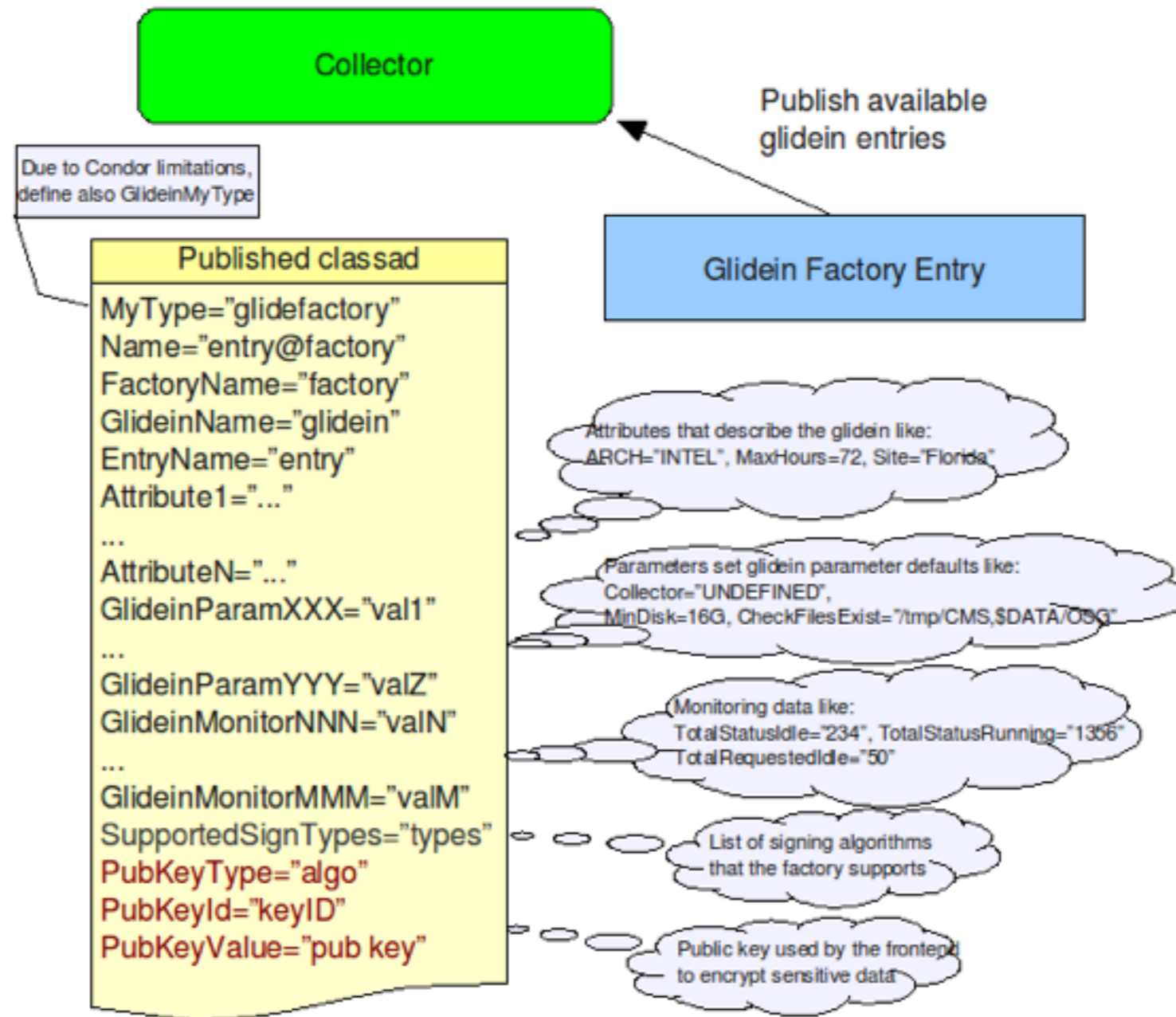
Attributes from Frontend

- Factory Entry attributes can be considered defaults for the Frontend
- Frontend can override or append attributes to Entry
 - Mostly used to influence the glidein behavior
 - Can be delivered in other ways (e.g. Web)
- Pilot proxy is the only **required** attribute sent by the frontend

Pilot proxy

- The Factory submits glideins on Frontend's behalf
 - Uses the Frontend provided proxy
- The proxy is delivered as an attribute
 - It is **Encrypted** before being sent to the factory
- The Factory provides a public key as an attribute in its own ClassAd
 - Frontend encrypts it using factory pub key
 - the factory decrypts it with its own private key

Entry ClassAd

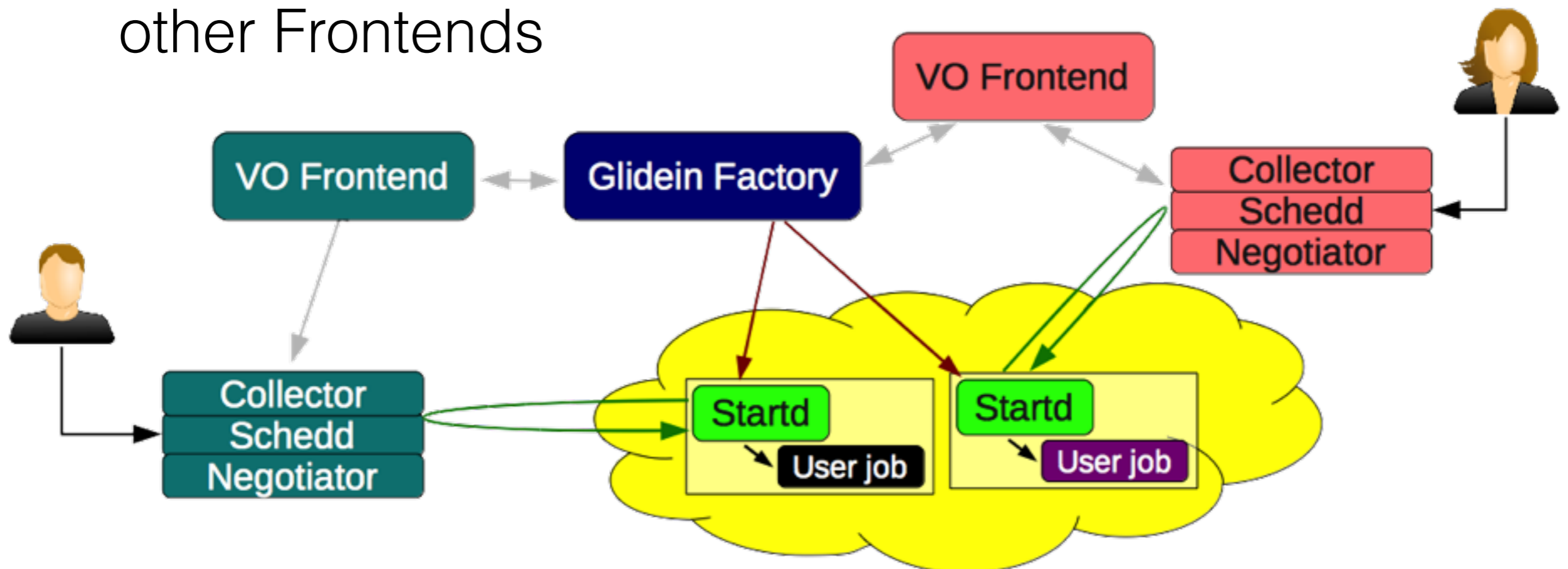


Frontend ClassAd



Security considerations

- An Entry can serve multiple Frontends
 - So it will have multiple proxies
- Frontends trust the factory, but not necessarily other Frontends



User insulation

- Factory must use different UIDs for different Frontends
 - Both for file access, and
 - Condor-G (processes run in the name of that user)
- Factory has a FrontendName → UID map
- Must prevent spoofing
 - i.e. bad Frontend pretending to be a different one
 - Use Collector Auth to get TrustedIdentity, then whitelist FrontendName ↔ TrustedIdentity

Monitoring

- Monitoring is available in several locations
 - Condor monitoring
 - GlideinWMS monitoring
- In particular Condor logs:
 - Collector logs provides authentication info
 - Gridmanager logs provide info about the interaction with resource providers (e.g. Grid sites)

Entry monitoring

- Activity log
- RRD files with statistics (running, held, etc.)
- XML files with the current snapshot
- Frontend Monitoring ClassAd sent to factory collector with current snapshot of FE usage
- Master factory process aggregates RRD and XML files, and writes factory-wide monitoring totals into its own area
- All of the above are available to view in factory monitoring webpages

Pointers

- glideinWMS development team is reachable at glideinwms-support@fnal.gov
- The official project Web page is <http://glideinwms.fnal.gov>
- OSG glidein factory at UCSD
<http://http://www.t2.ucsd.edu/twiki2/bin/view/UCSDTier2/OSGgfactory>
<http://gfactory-1.t2.ucsd.edu/factory/monitor/>

Acknowledgments

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