

glideinWMS Training @ UCSD

glideinWMS factory Internals

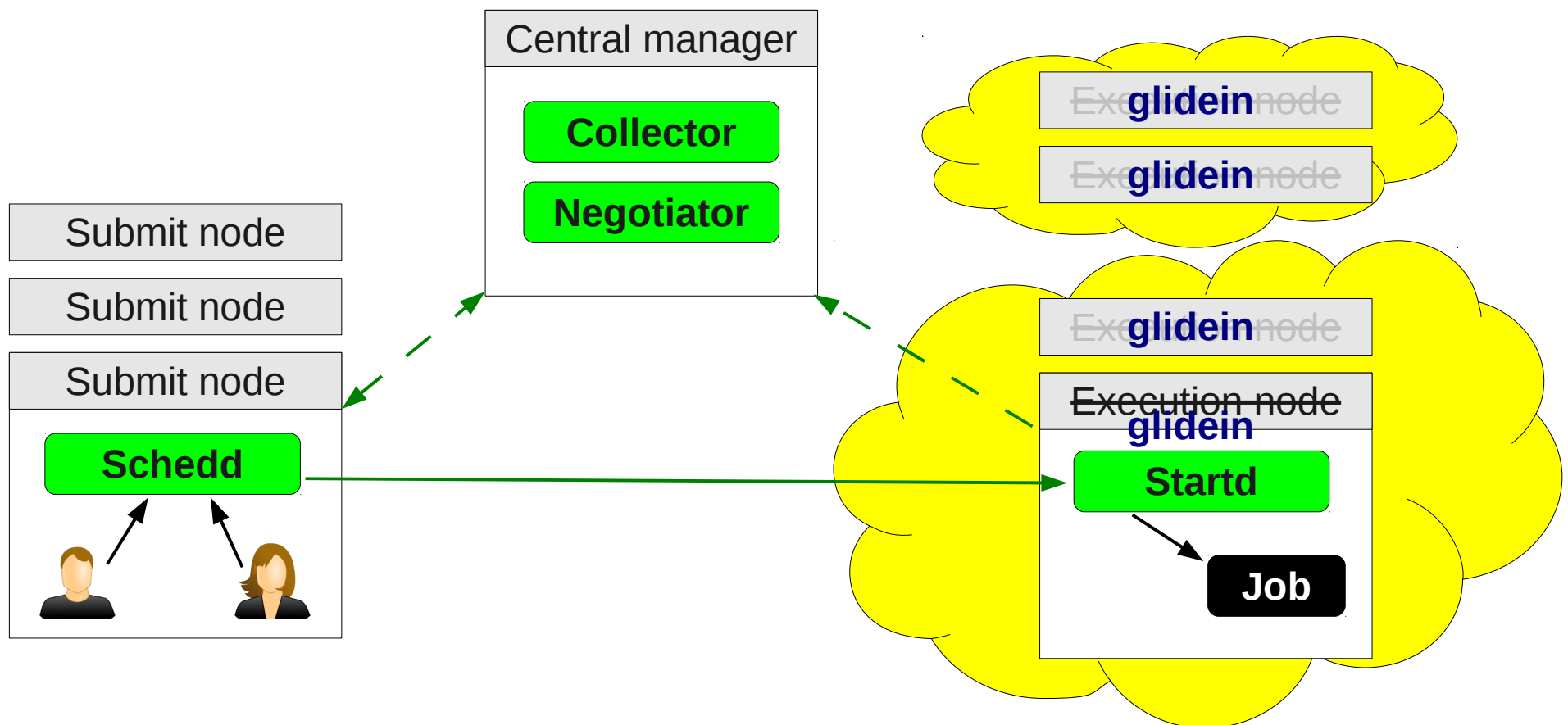
by Igor Sfiligoi (UCSD)

Overview

- Refresher
- Factory architecture
- Entry logic
- Security considerations

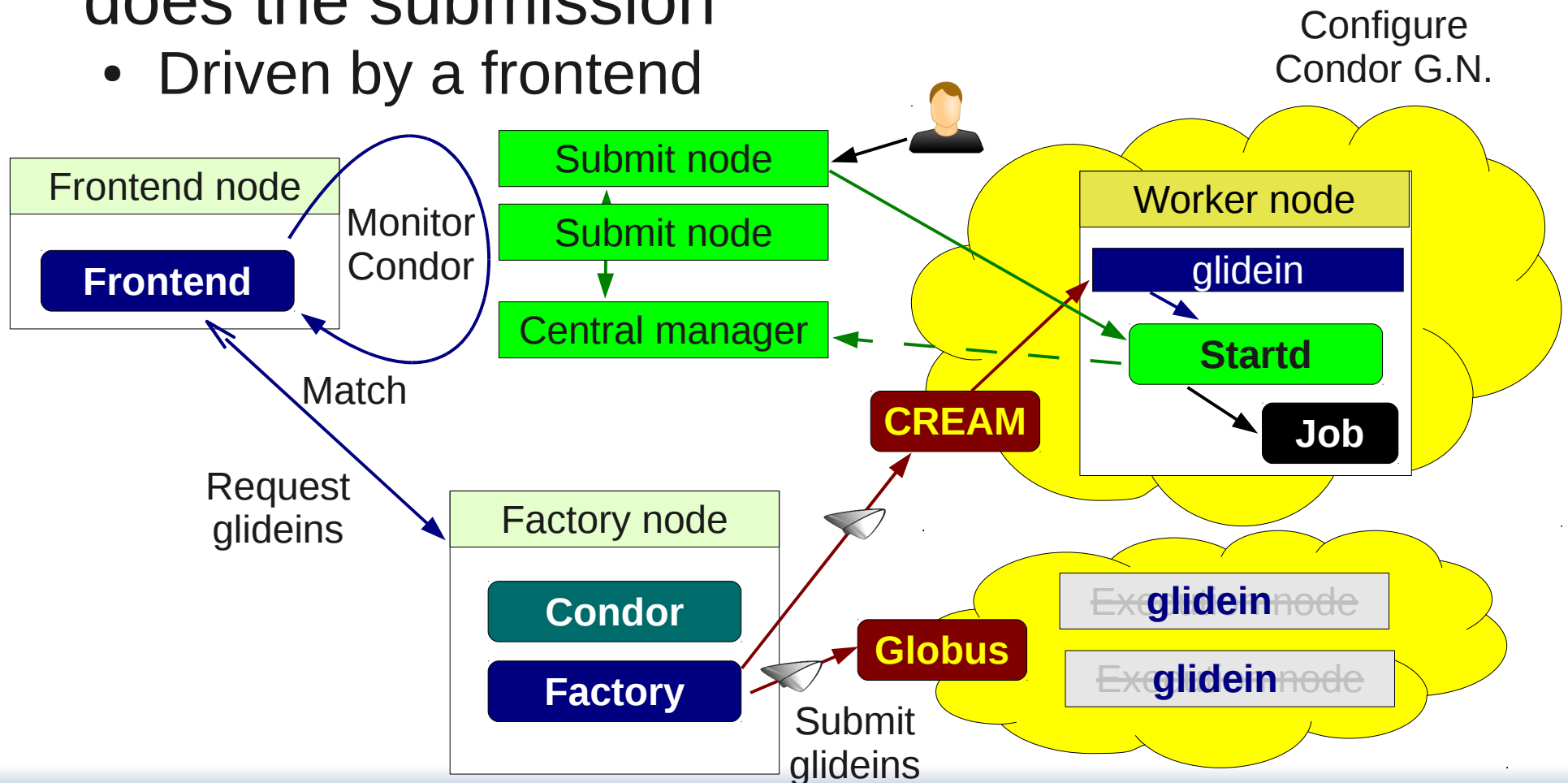
Refresher - What is a glidein?

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



Refresher – Glidein factory

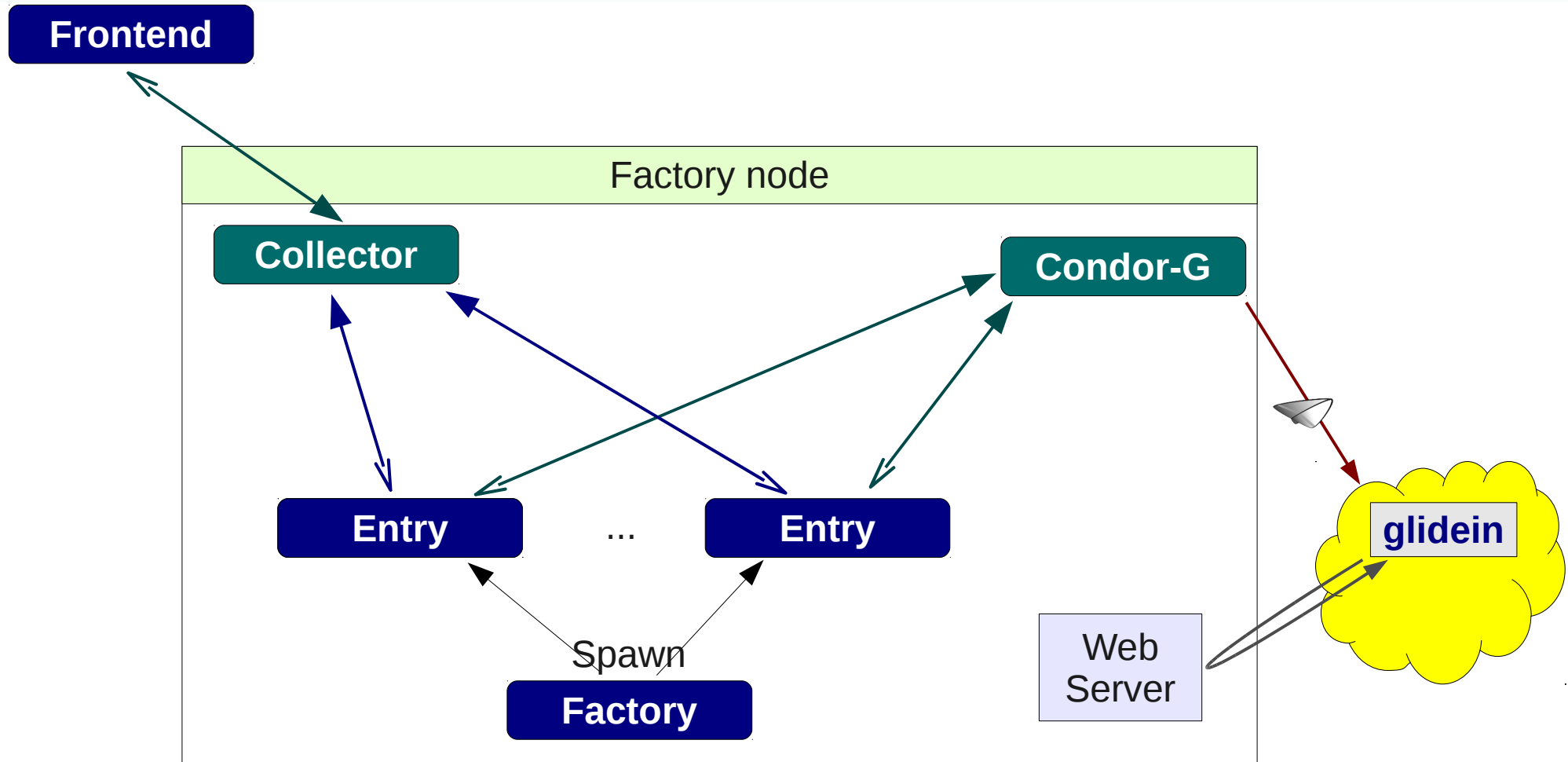
- The glidein factory knows about the sites and does the submission
 - Driven by a frontend



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:
 - Entry processes – do the real work
 - Master factory – controls the others and aggregates monitoring

Factory arch - picture



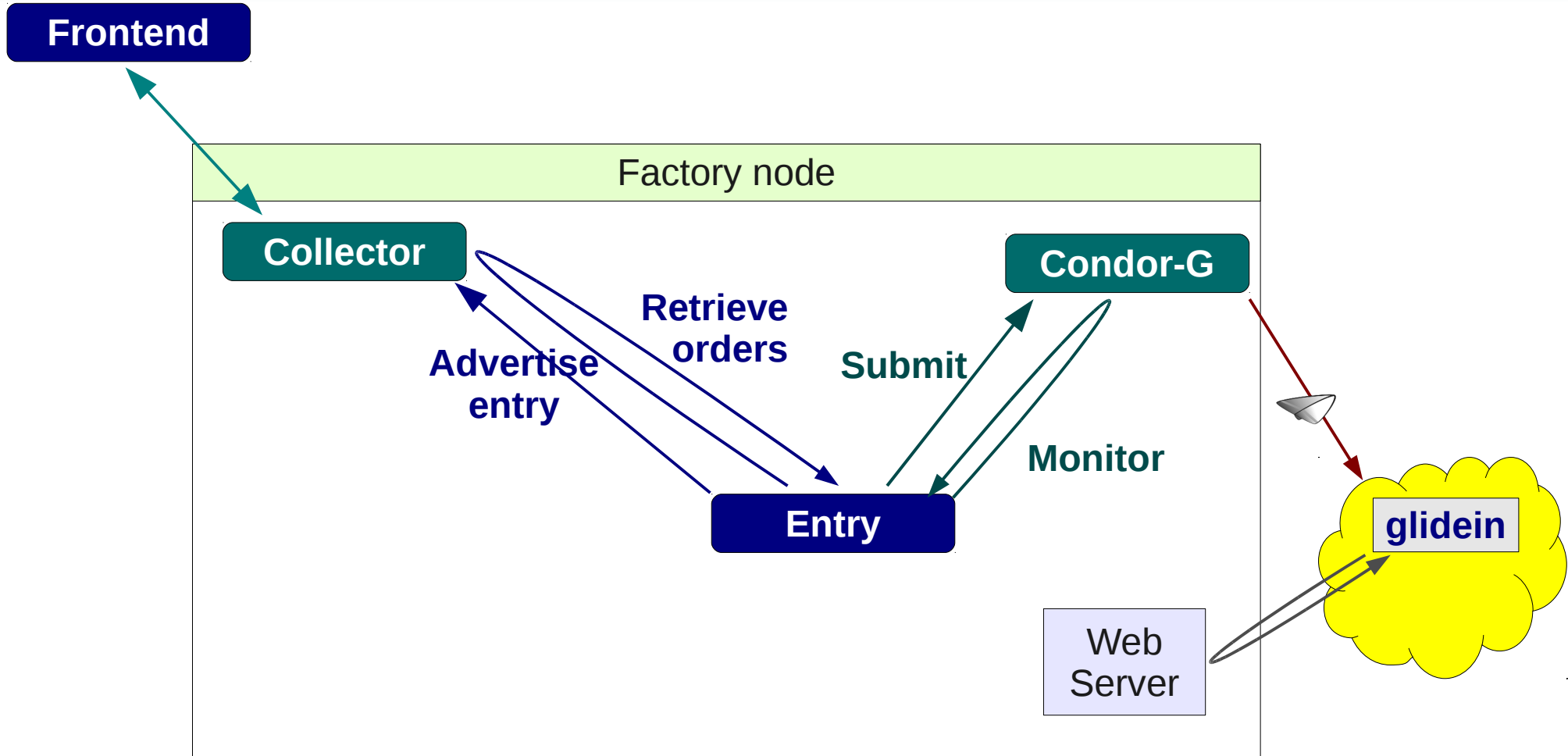
Factory processes

- Real work performed by Entry process
 - `glideFactoryEntry.py`
 - One process x Entry
- They are controlled by master Factory
 - `glideFactory.py`
 - Starts the other processes
 - Aggregates monitoring

Entry logic

- Essentially a slave
 - Will do what a frontend tells it to
- Uses the Collector for communication
 - Advertise own existence and attributes
 - Polls the collector for commands
 - Everything ClassAd based
 - All security implemented in the Collector
- Glideins submitted via Condor-G
 - Then just monitors them

Factory Entry - picture



Entry loop

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

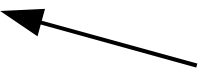
Entry Attributes - Essential

- The Entry Classad has many attributes
 - A few attributes are essential
 - Needed to submit glideins
- Essential attributes:
 - Condor-G: gridtype, gatekeeper, rsl
 - Resource Type: wax_walltime, glexec, os_type
- Required info attributes:
 - Name
 - Site name – related, but does not need to be unique

Entry attributes - Optional

- But Entry can publish any attribute
- Recommended ones:
 - Supported VOs
 - Well defined resource name: CMSSite, BDIIResourceName, etc.
- Plus any other info the Factory admin thinks the Frontend may be interested in

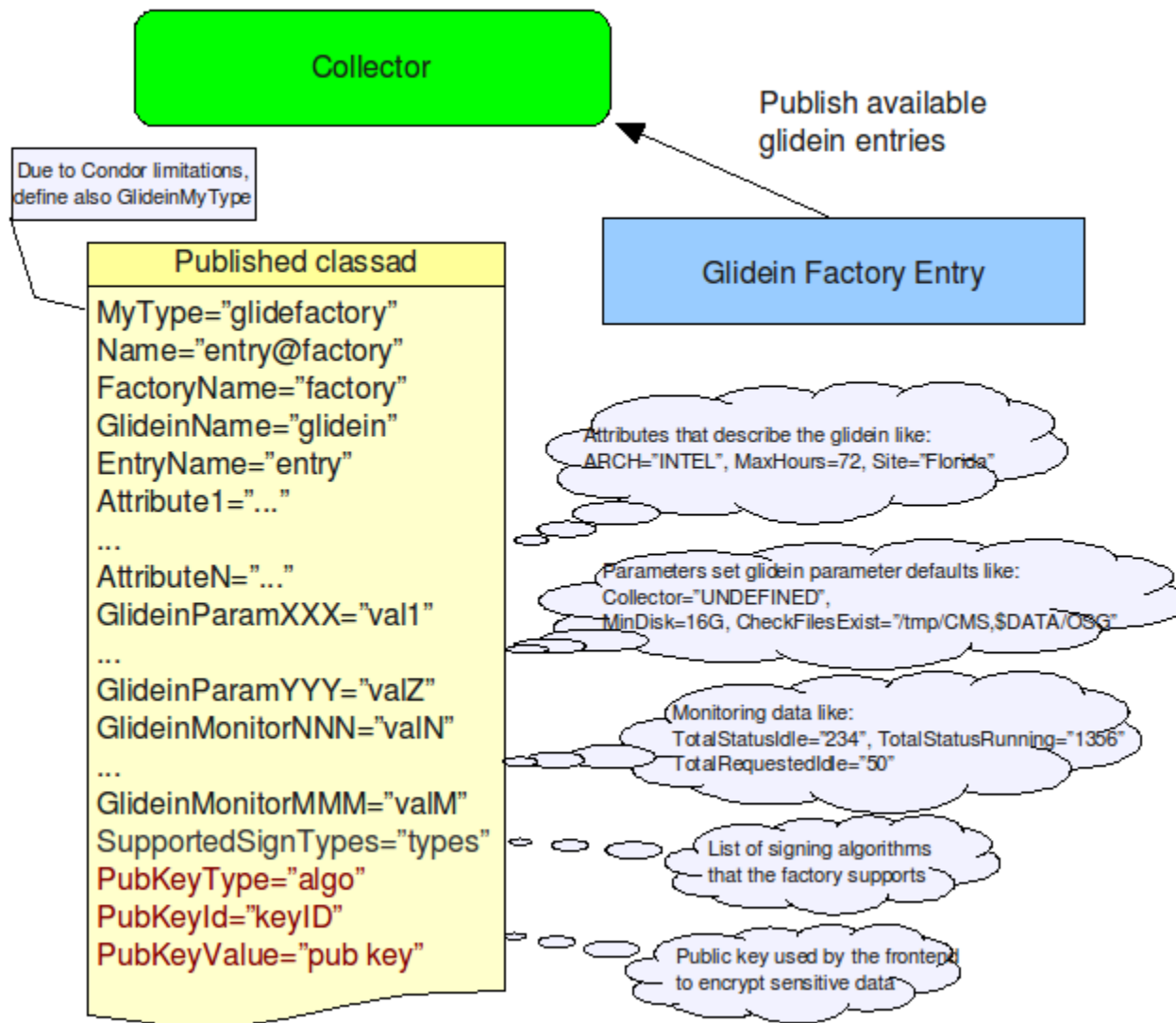
Attributes from Frontend

- Entry attributes are really just for the Frontend
 - But the Frontend provides back attributes for Entry to use
 - Mostly used to influence the glidein behavior
 - So can be delivered in other ways (e.g. Web)
 - Two attributes influence factory:
 - Pilot proxy
 - RSL modifiers 
- New in 2.5.3, used only for Corral

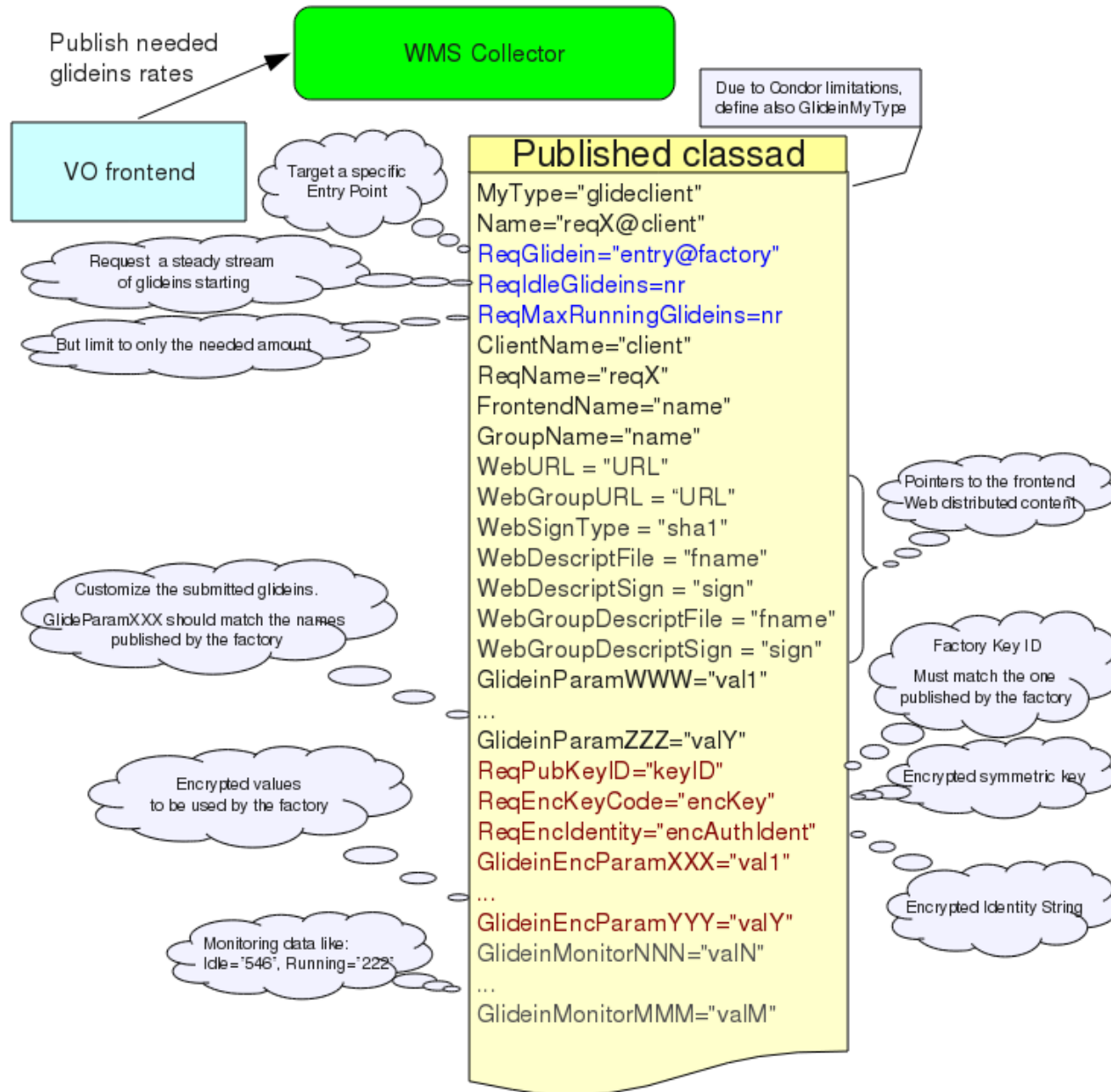
Pilot proxy

- The Entry submits glideins in Frontend name
 - Uses the Frontend provided proxy
- The proxy is delivered as an attribute
 - **Encrypted**
- The Entry provides a public key as an attribute in its own ClassAd
 - Frontend encrypts it using that key
 - Entry decrypts it with own private key

Entry ClassAd

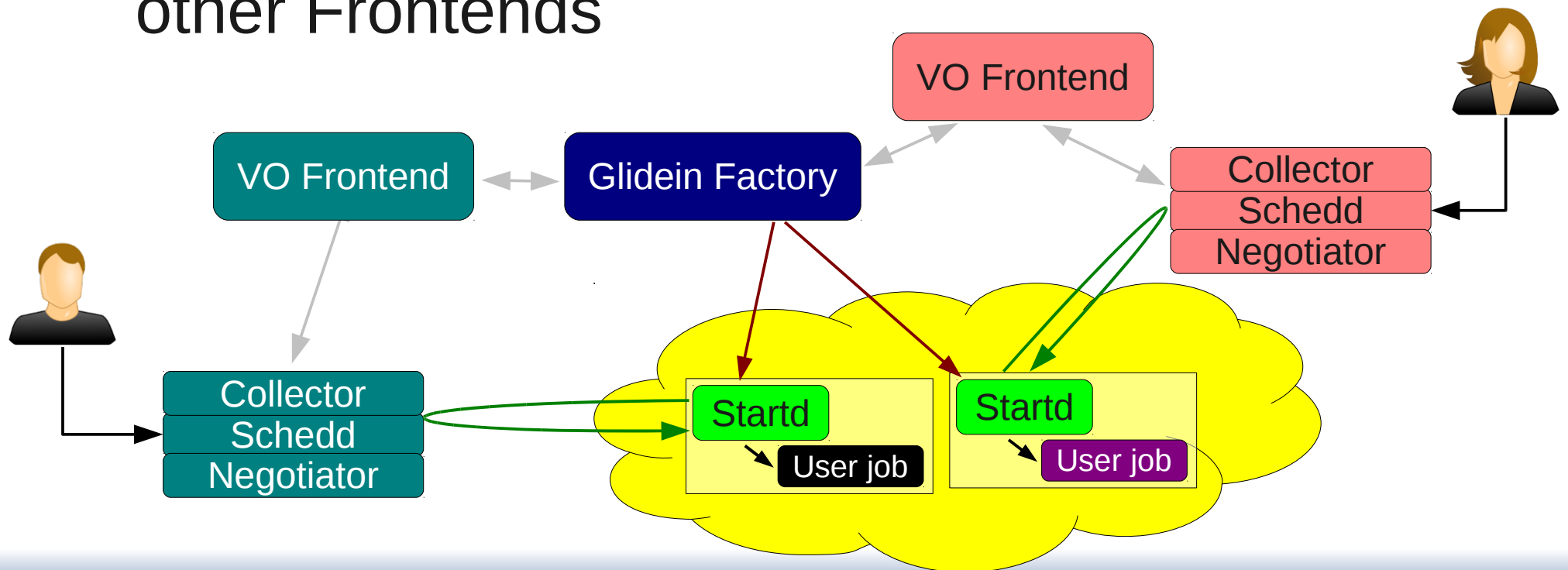


Frontend ClassAd



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

- Entry must use different UIDs for different Frontends
 - Both for file access, and
 - Condor-G (processes run in the name of that user)
- Entry 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 available in several locations
 - Condor monitoring
 - GlideinWMS monitoring
- More details on Condor monitoring tomorrow, but in summary
 - Collector provides Auth logs
 - Gridmanager provides 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 current snapshot
 - Monitoring ClassAd with current snapshot
-
- Master factory aggregates RRD and XML files, and writes the in its own area

Pointers

- The official project Web page is <http://tinyurl.com/glideinWMS>
- glideinWMS development team is reachable at 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

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