glideinWMS

Mats Rynge, rynge@isi.edu

USC / Information Sciences Institute

Igor Sfiligoi, isfiligoi@ucsd.edu
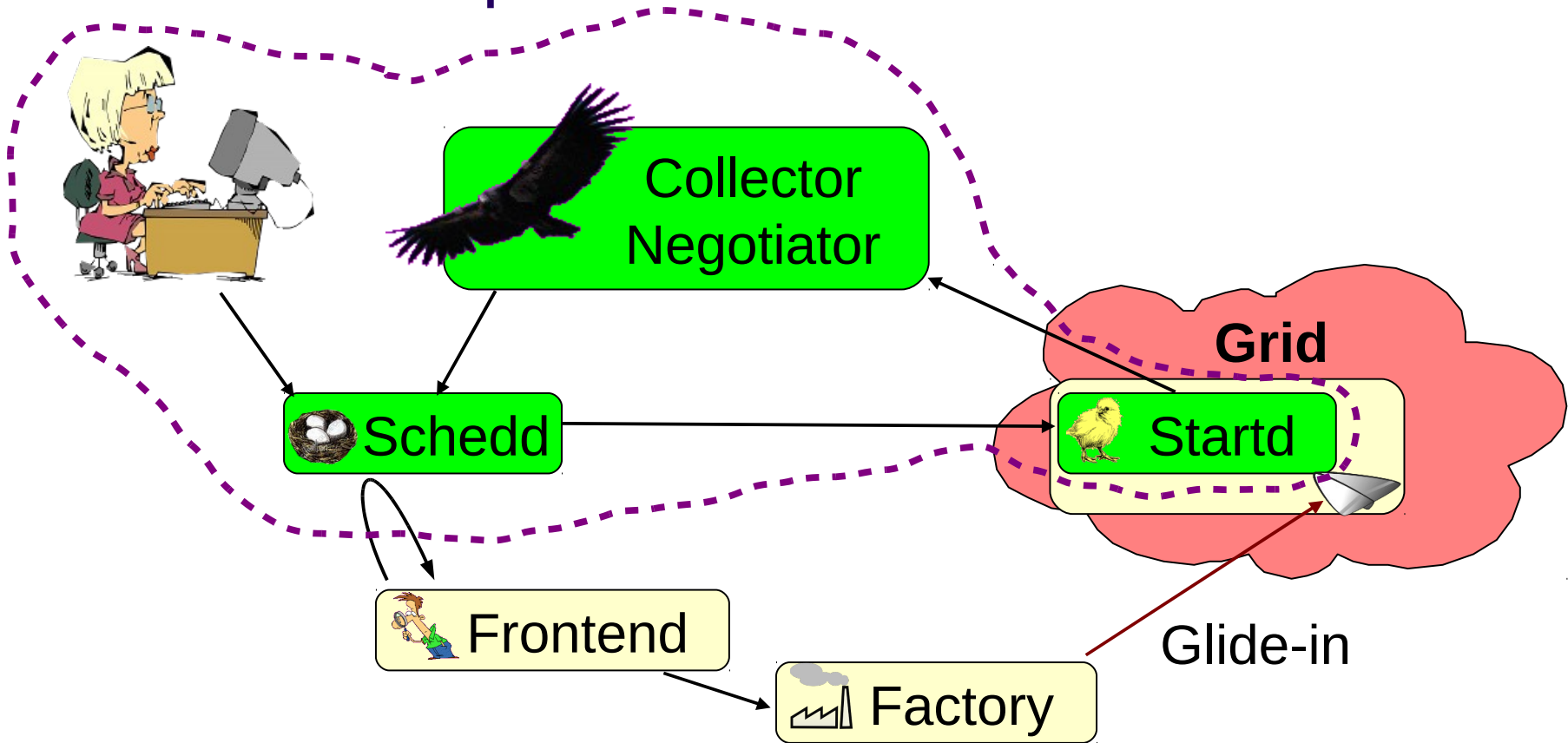
University of California San Diego

glideinWMS

- At a glance
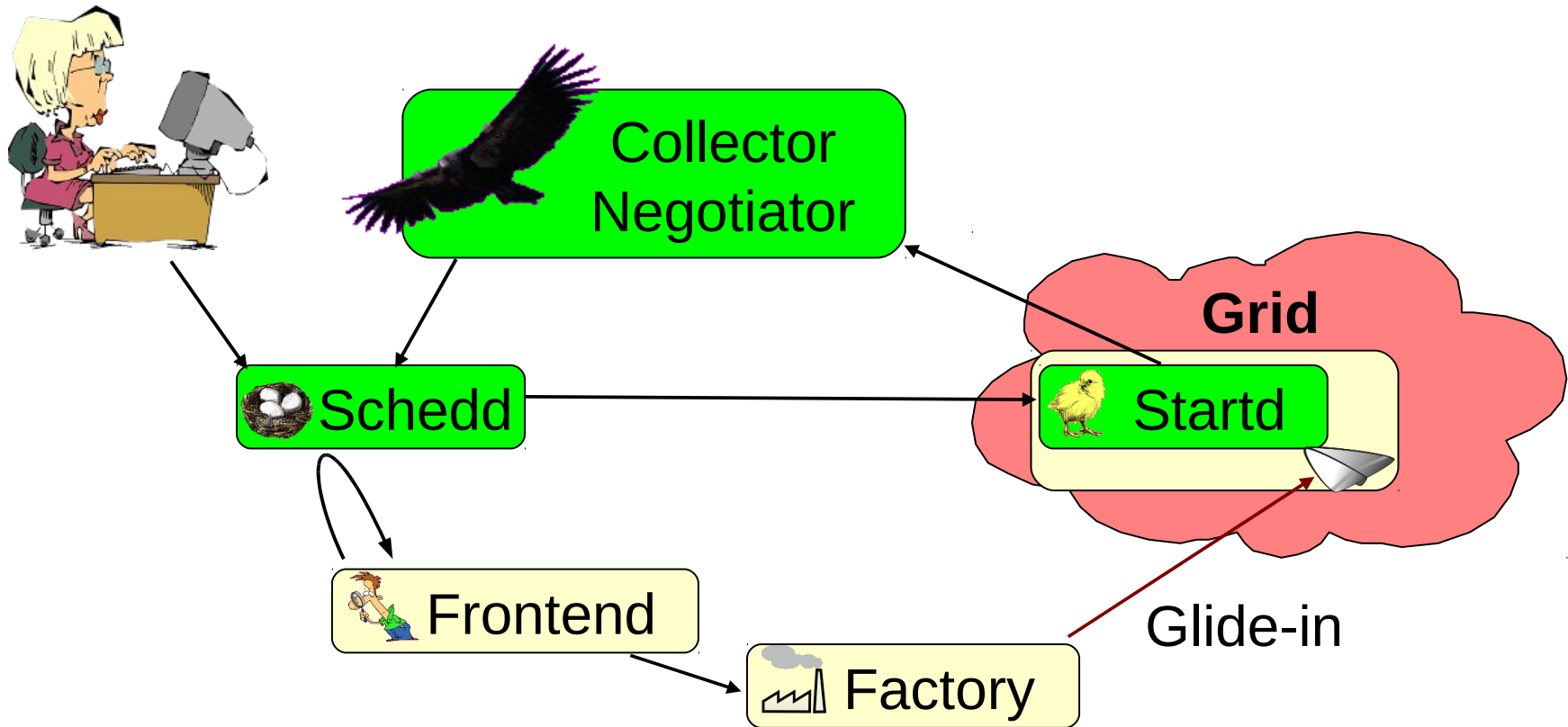
- The user really only sees the regular Condor pool

Just a regular-looking pool

- ## Submit vanilla jobs

```
sfiligoi@vdt-itb$cat j.submit
Universe       = vanilla
Executable     = mytest.sh
Output         = j.out.$(Cluster).$(Process)
Error          = j.err.$(Cluster).$(Process)
Log            = j.log
queue

sfiligoi@vdt-itb$condor_submit j.condor
```

- ## Monitor resources

```
sfiligoi@vdt-itb$condor_status -total -state
```

		Total	Owner	Unclaimed
Claimed	Preempting	Matched	Backfill	
	Busy	928	0		0
928		0	0	0	
	Idle	32	0		32
0		0	0	0	
	Retiring	42	0		0
42		0	0	0	
	Total	1002	0		32
970		0	0	0	

```
sfiligoi@vdt-itb$
```
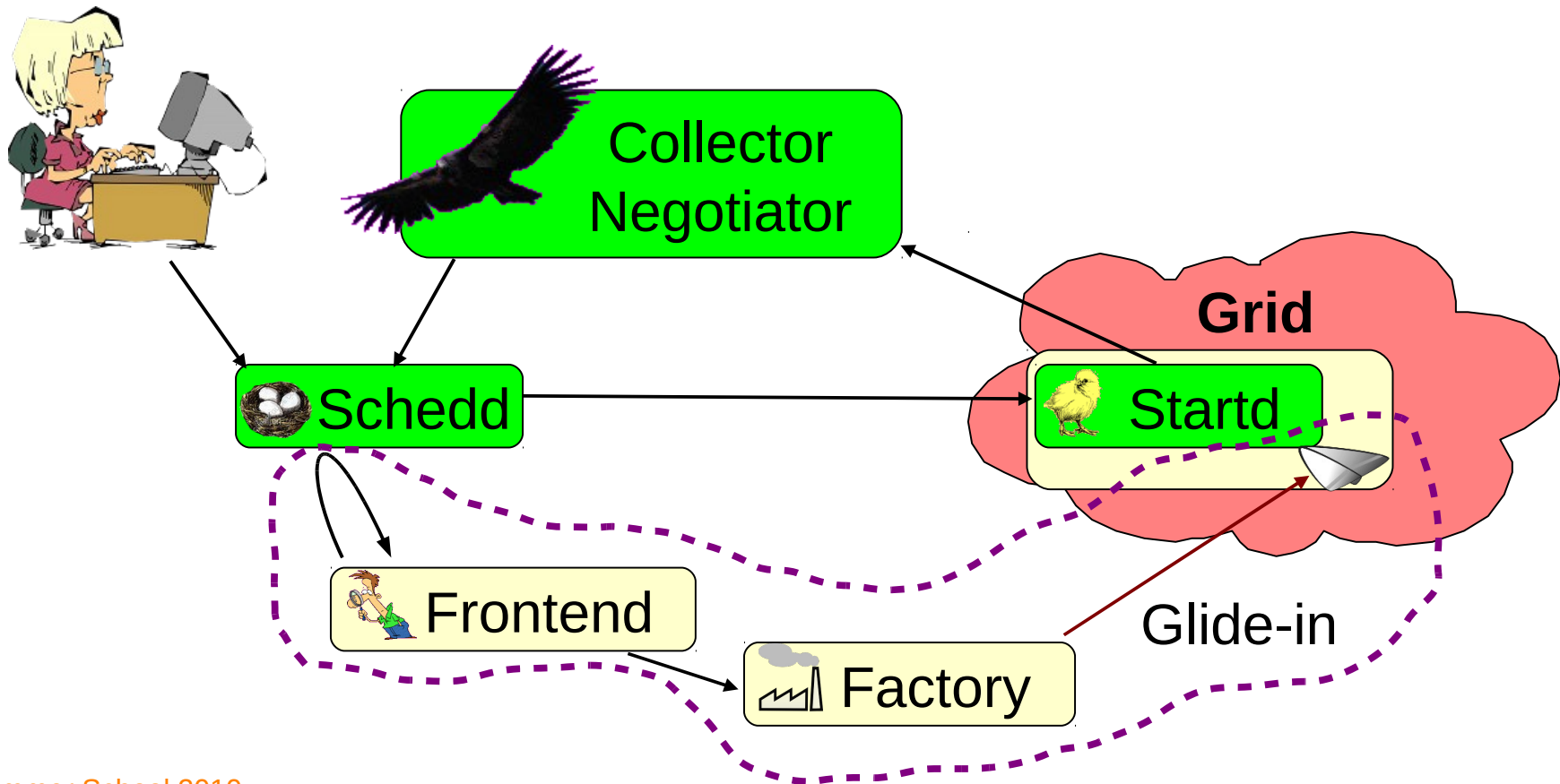
Just a dynamic one

- Resources come and go
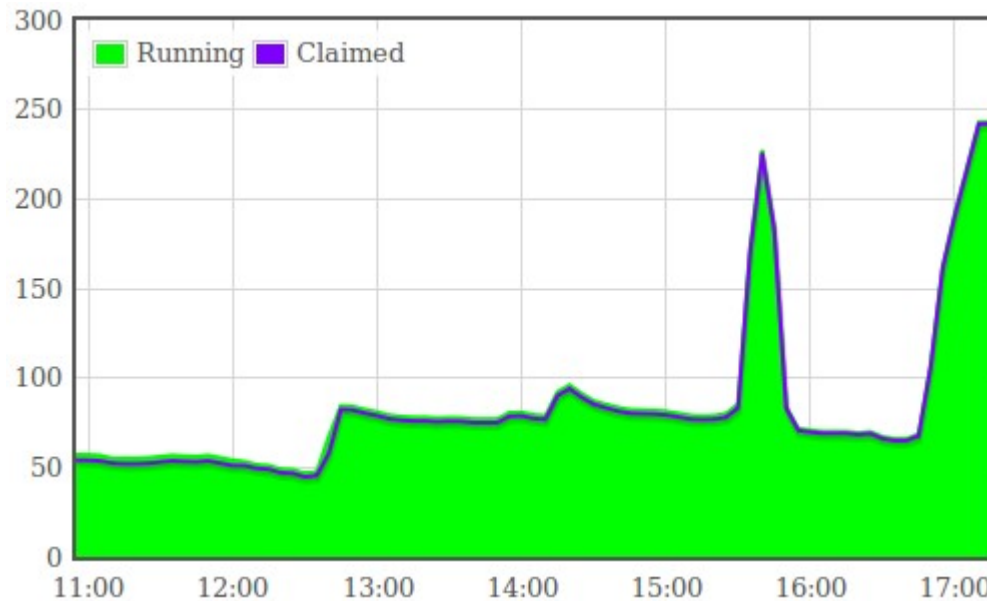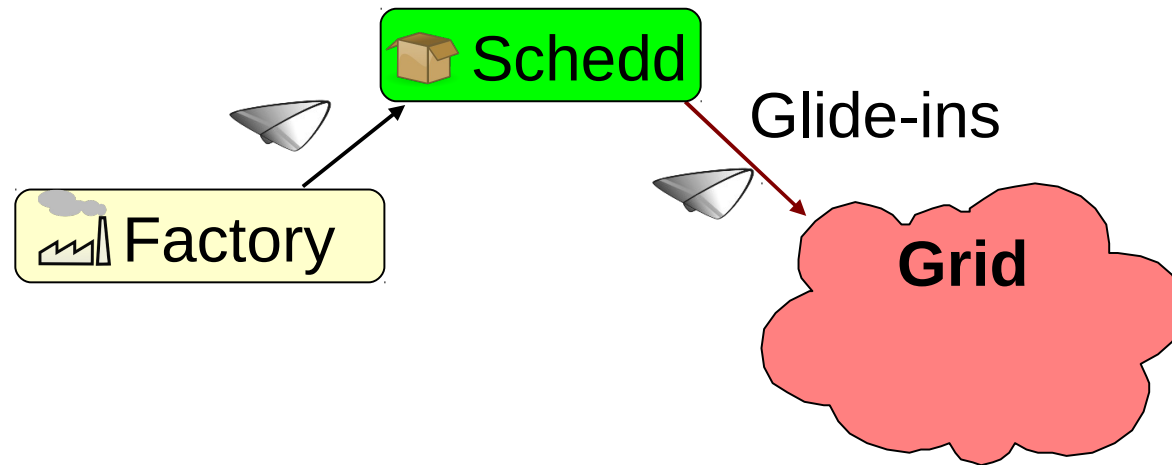

Courtesy of the UCSD CMS frontend

glideinWMS backend

- glideinWMS processes make sure users have resources when needed

Glidein Factory

- The glidein factory actually submits the Grid jobs – the glideins
- Uses Condor-G for the task

Frontend

- The frontend decides how many glideins to submit
- The decision based on how many jobs need more resources and where

Schedd

condor_q

50 running
75 idle

Frontend

Factory

Please submit 25 more
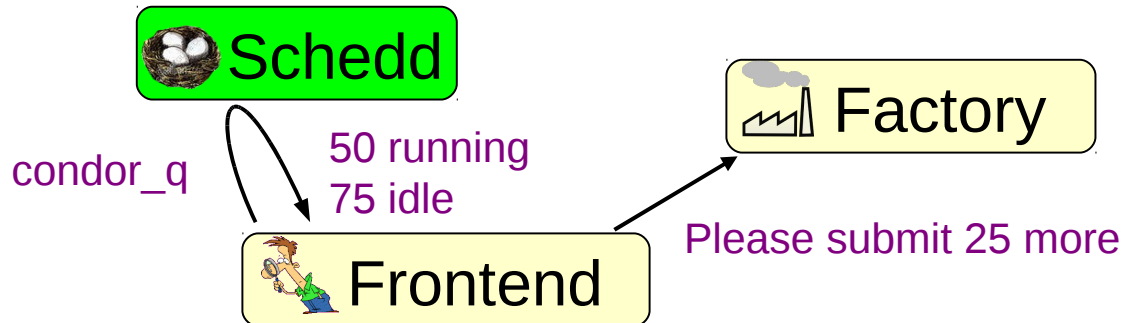
The glideins

- Glideins are regular Grid jobs
  - Nothing special about this
- Will configure a Condor instance
  - Get the binaries
  - Discover/validate WN environment
  - Create condor config
- Finally start a Condor startd
  - Condor takes care of itself after that

- Glideins are not specific to a user
  - The startd will pick the highest priority job that matches
  - Just like in a regular pool
- A glidein can run many jobs
  - Possibly from many users
  - Just like in a regular pool
- Uses its own credential for Grid submission
  - Usually a service proxy

Security considerations

- Traffic to and from a Grid site flows over untrusted Internet
- All WAN connections must have security measures in place
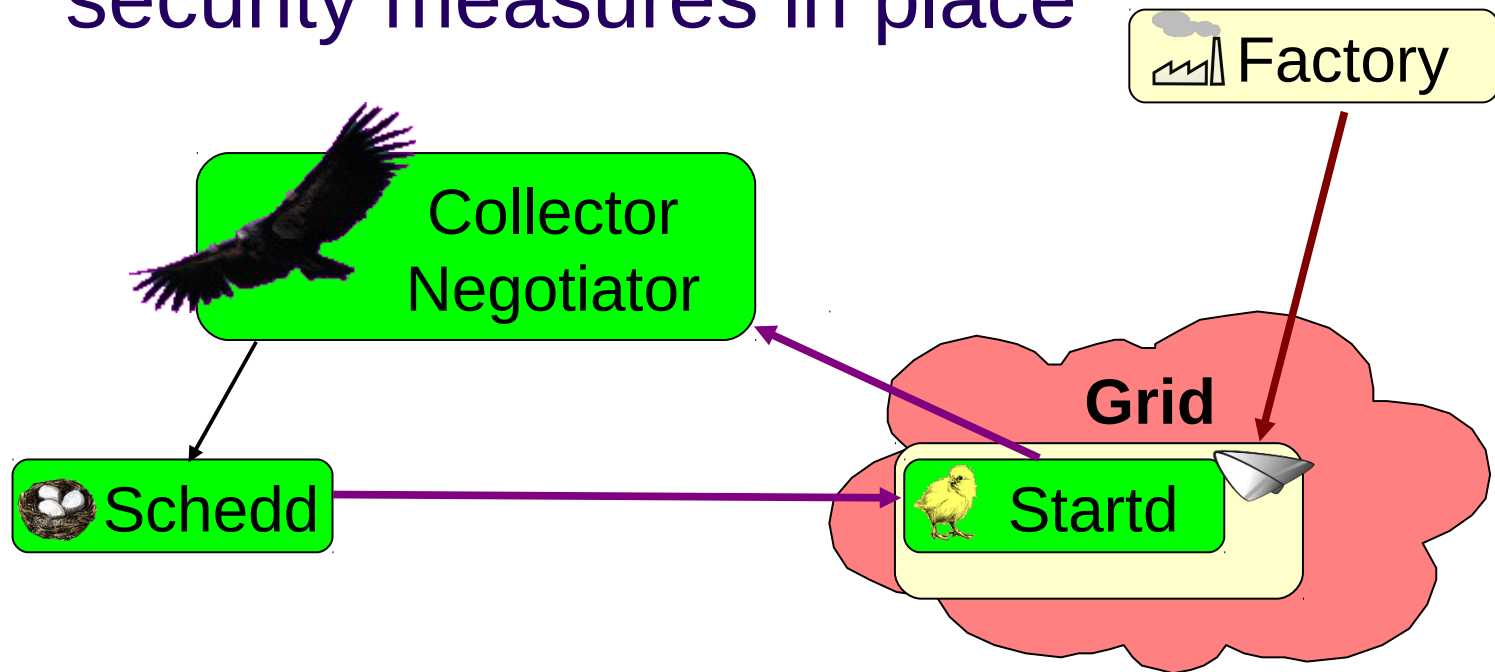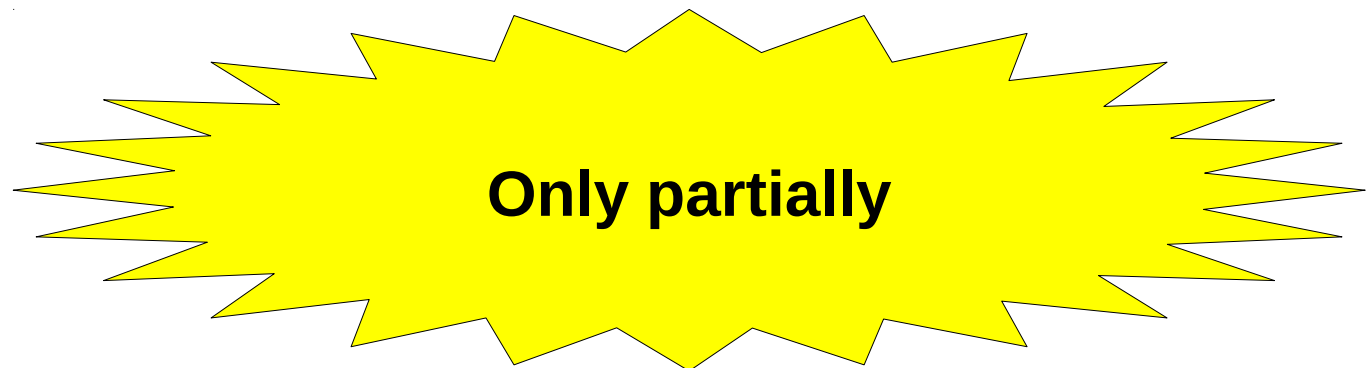
Security considerations (2)

- The glidein will have its own proxy when landing
  - Will use it to authenticate to the rest of the system
- The other processes have a certificate as well
  - Mutual authentication
- Using whitelist authorization
- Also enabling integrity checks in Condor
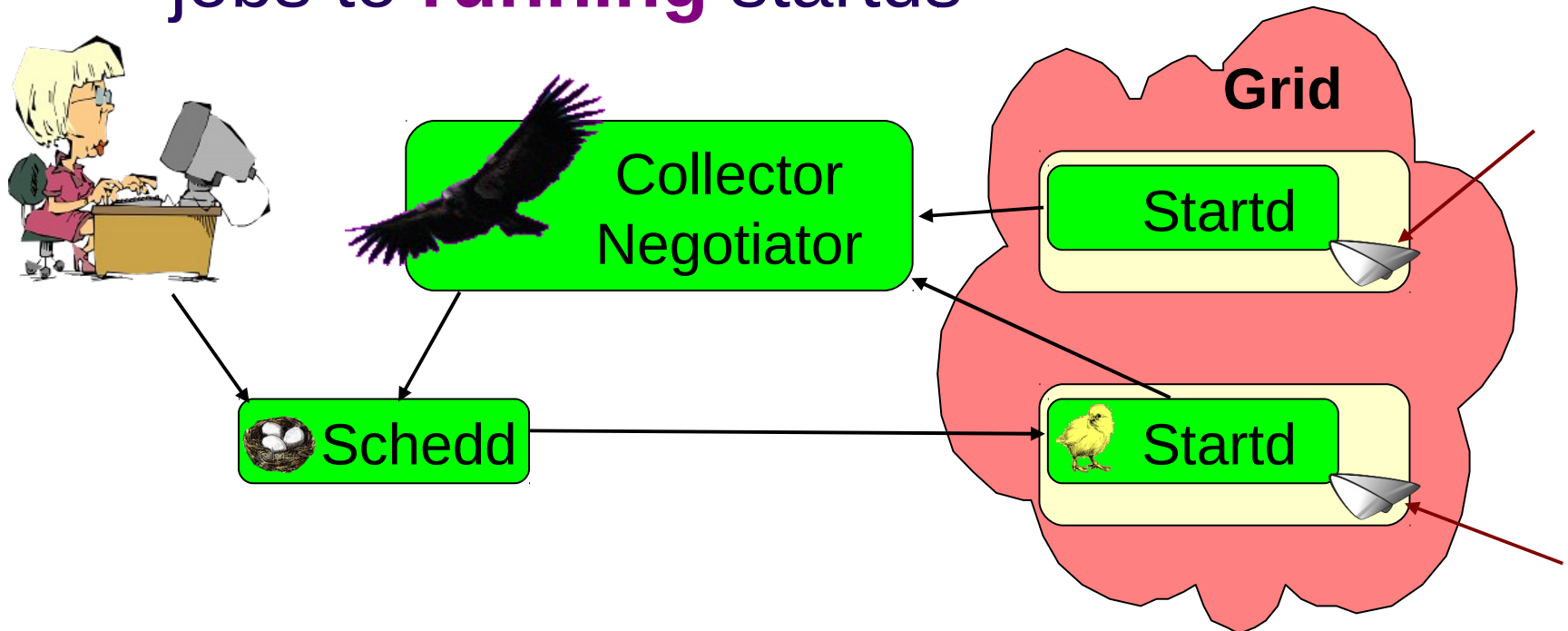
Anything glideinWMS specific?

- From the user's point of view,
  isn't this just a regular Condor pool?

- Can't I just set the
  "**requirements**"
  and be done?
 (i.e. the negotiator will take care of it)

Only partially

- The **"requirements"** only used by the negotiator

- The negotiator only matches jobs to **running** startds

Grid

Collector
Negotiator

Startd

Schedd

Startd

Where will the glideins run?

- There may be many Grid sites
  - OSG g.factory @UCSD serves ~160
- Frontend will match
  user job attributes
  against
  site attributes
 **(provided by the factory)**
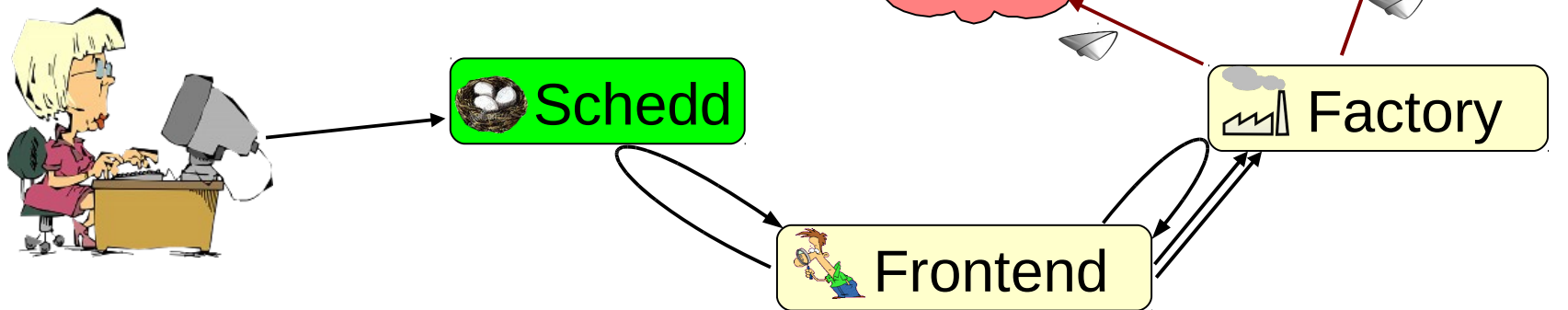
Frontend matchmaking

- The standard "requirements" attribute cannot be used by the frontend
  - The glidein factory attributes are likely different than the glidein-startd attributes
- The factory has only generic information about the site
  - Glideins can augment it with information discovered on the worker nodes

Two level matchmaking

- Based on different information
- Can potentially be a problem
  - Glideins that start and don't get matched
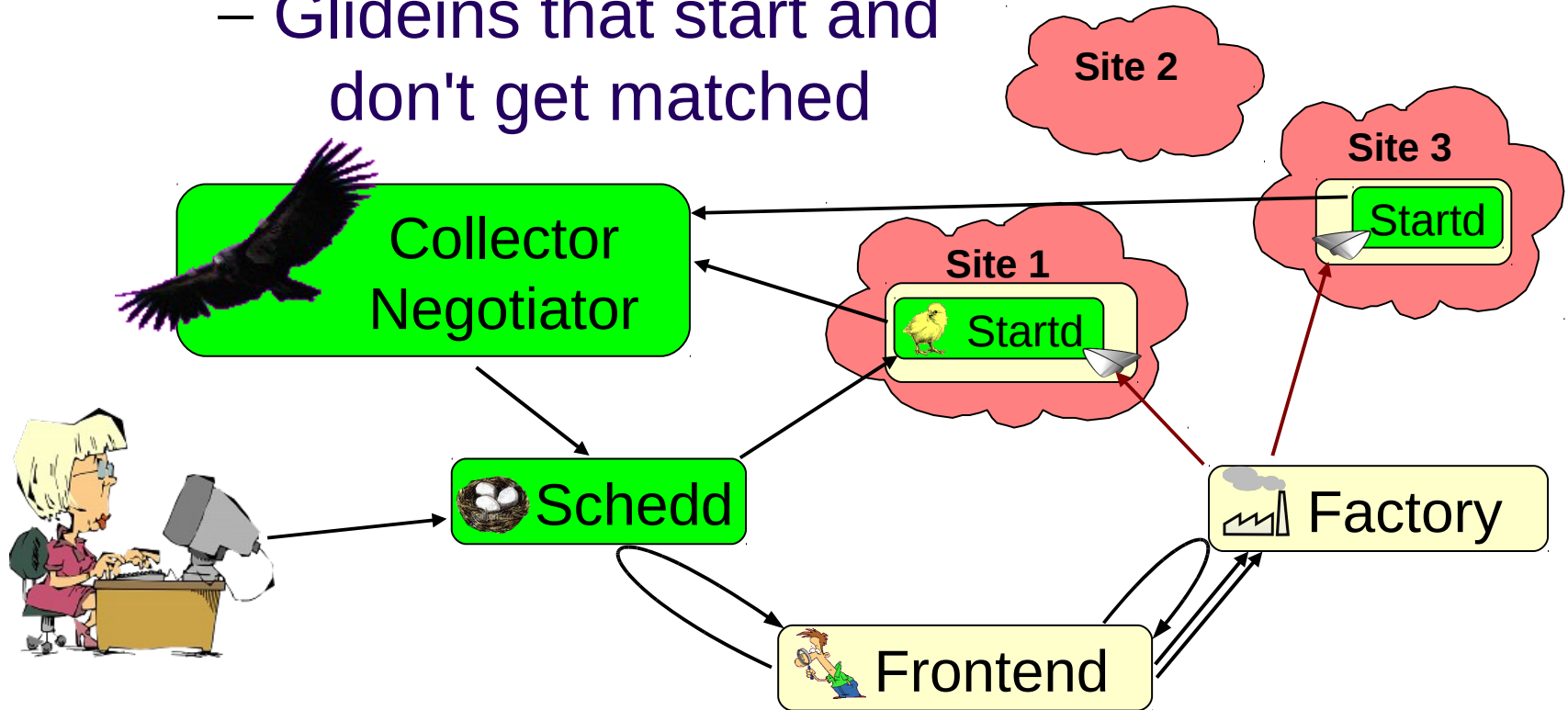
An example

- Site selection criteria:
  - Site=!="Nebraska" && HasBLAST

- Startd requirements:
  - BLAST_version>2 && Disk>100000

Summary

- glideinWMS looks almost like a regular Condor pool to the users
  - Most of the Grid details taken care of by the glideinWMS admins

- Users may need to add additional attributes for Grid site selection

Exercises

- Blast
- Montage

NASA IPAC Montage

mProject

mDiff

mFitplane

mConcatFit

mBgModel

mBackground