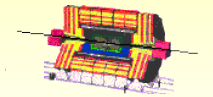
Fermilab



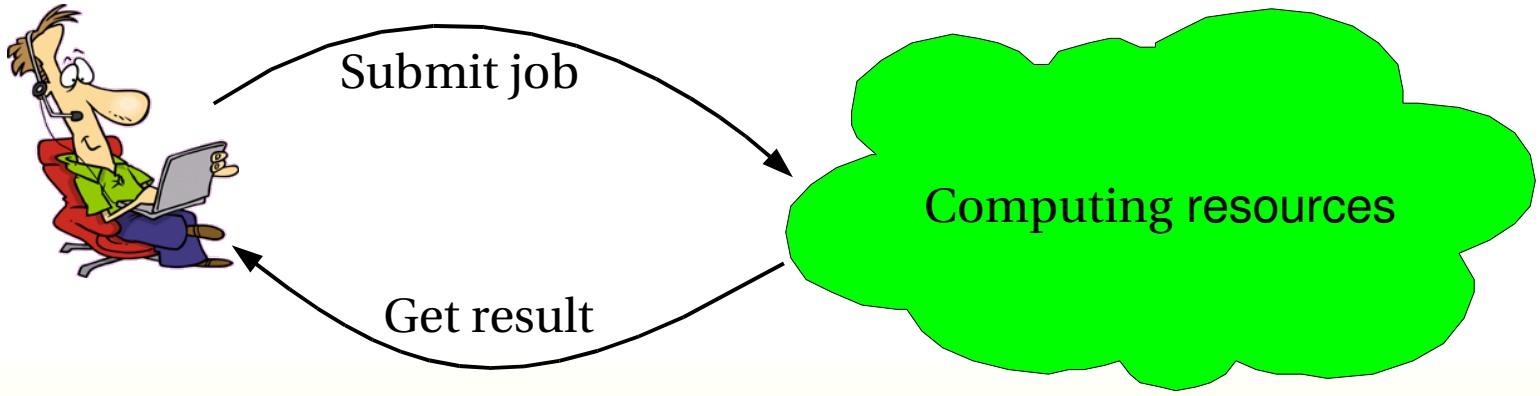
IEEE NSS 2007

Making Science in the Grid World
-
Using Glideins to Maximize Scientific Output

by Igor Sfiligoi (FNAL)

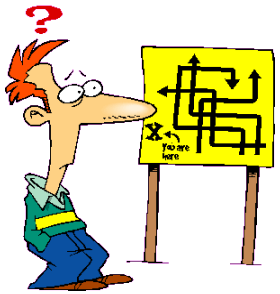
Portrait of a scientist

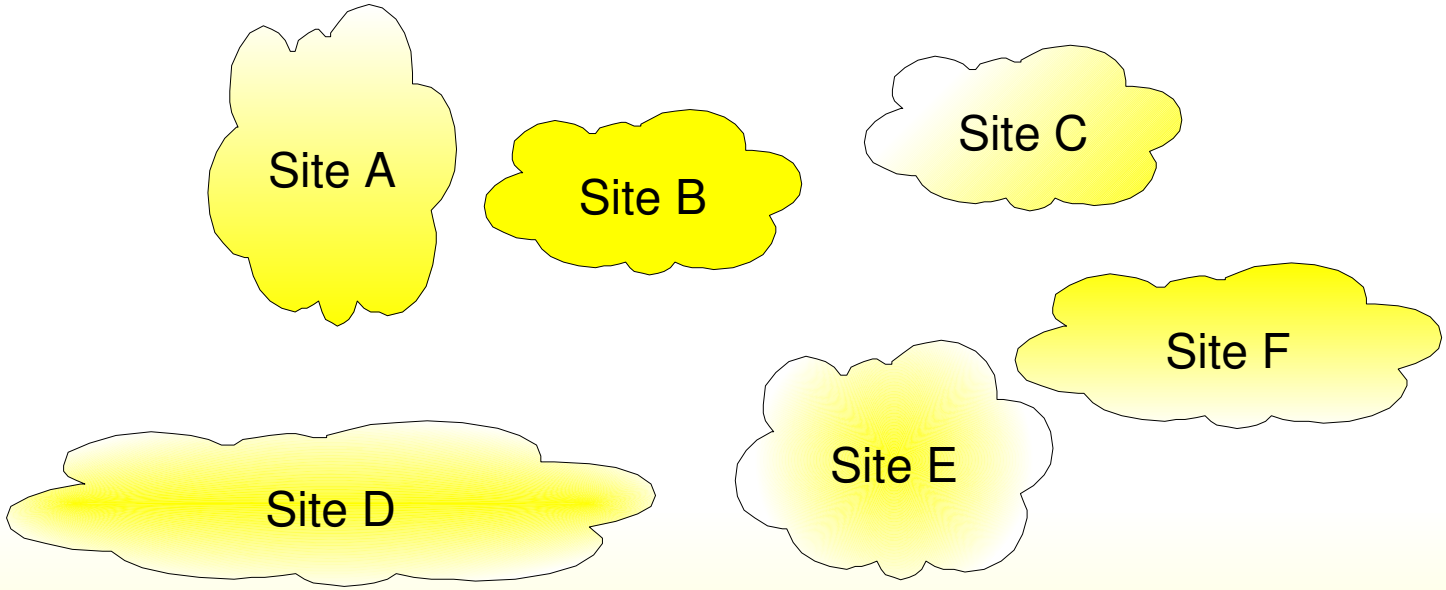
- Needs many computing cycles to analyze his data
  - More than can get from a personal desktop
- Wants to spend most of his time thinking about the scientific problems
  - Computing is just a tool



Portrait of the Grid

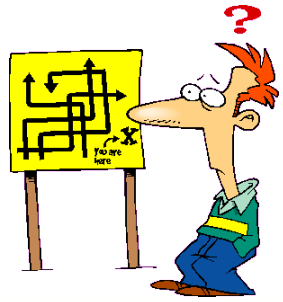
- Resources grouped in independent pools
  - Each with its own set of rules
- Resources in different pools configured differently
  - Users expected to adapt





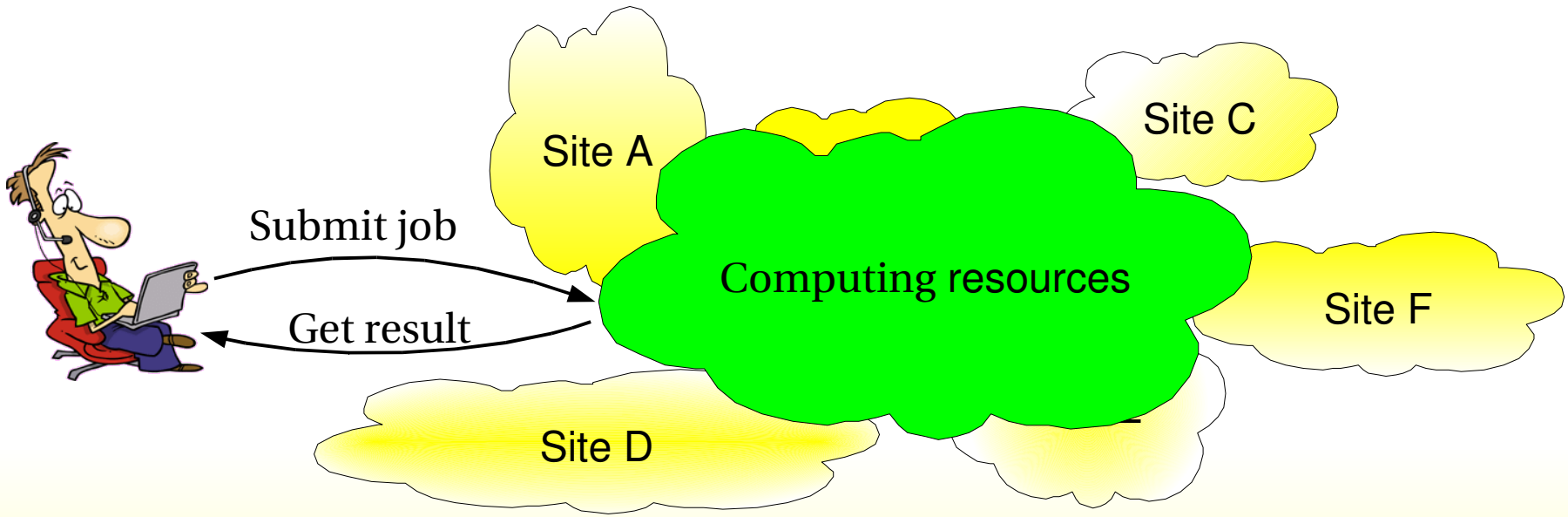
We have a problem!

- Scientists are forced to spend a significant amount of time thinking about computing
  - And every time a new site is added, the process starts again
- Time spent on computing problems is subtracting time available for scientific thinking!



Let's make the Grid uniform

- ... by creating an overlay over the Grid sites
  - Hiding differences between sites
  - Making the Grid look as a single, uniform pool

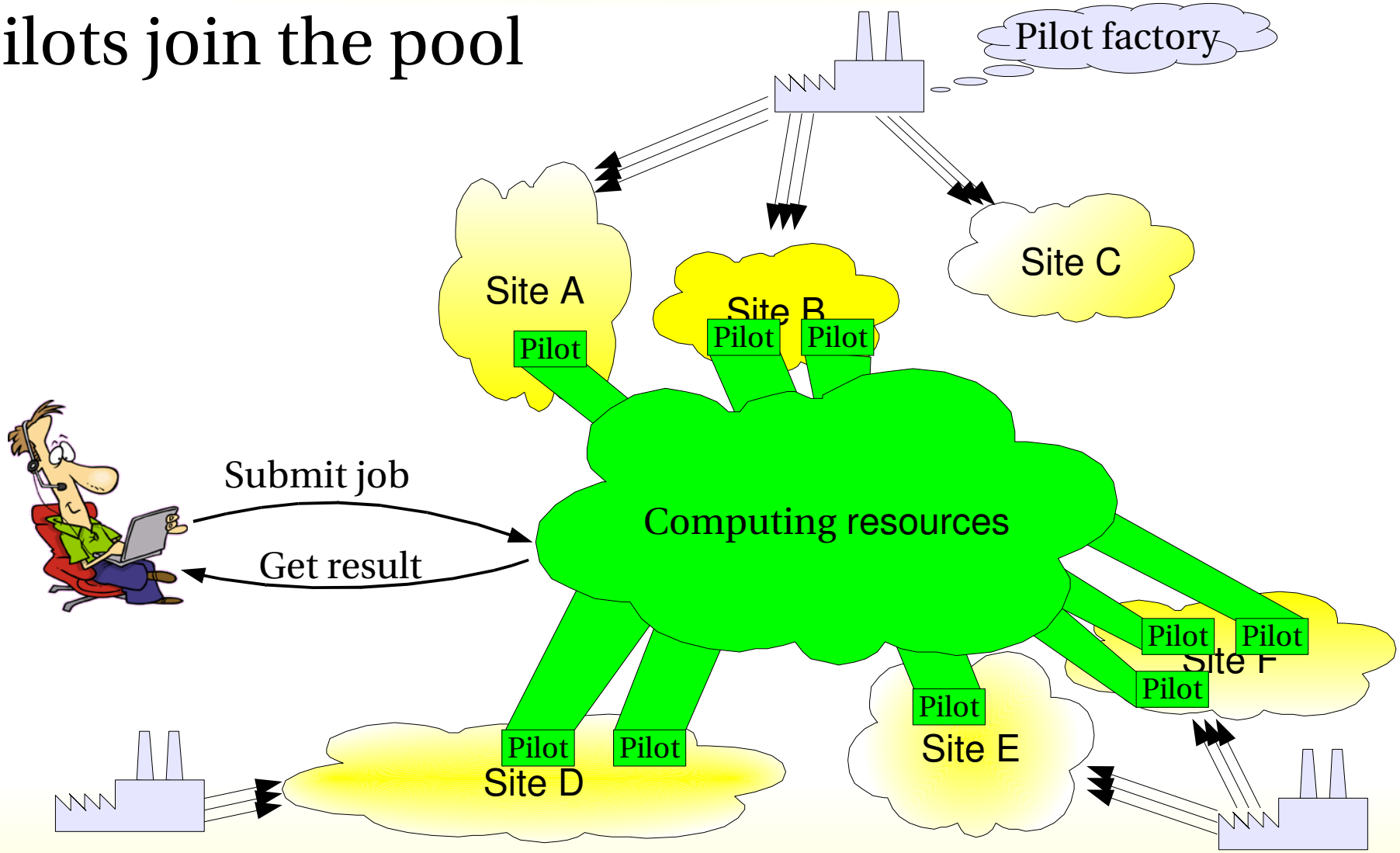


The pilot paradigm

- Never send user jobs to the Grid sites
  - Send pilots instead
- When pilots start
  - Validate Grid resource
  - Prepare the environment
  - Pull user jobs
- Pilot admins tailor the pilots to Grid sites
  - Users see a uniform pool

Pilots – An overview

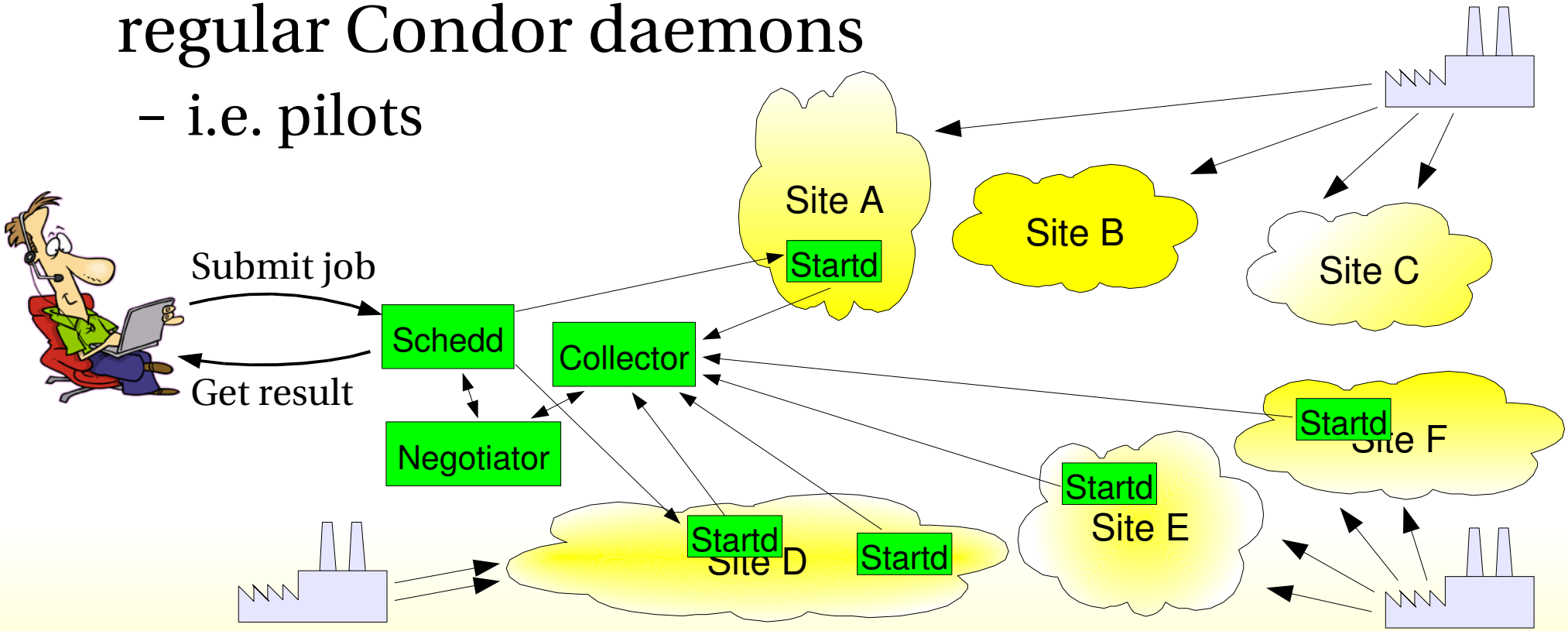
- Pilots join the pool



Condor glideins

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

- Condor distributed architecture ideally suited for this task
- Condor glideins are Grid jobs that start regular Condor daemons
  - i.e. pilots

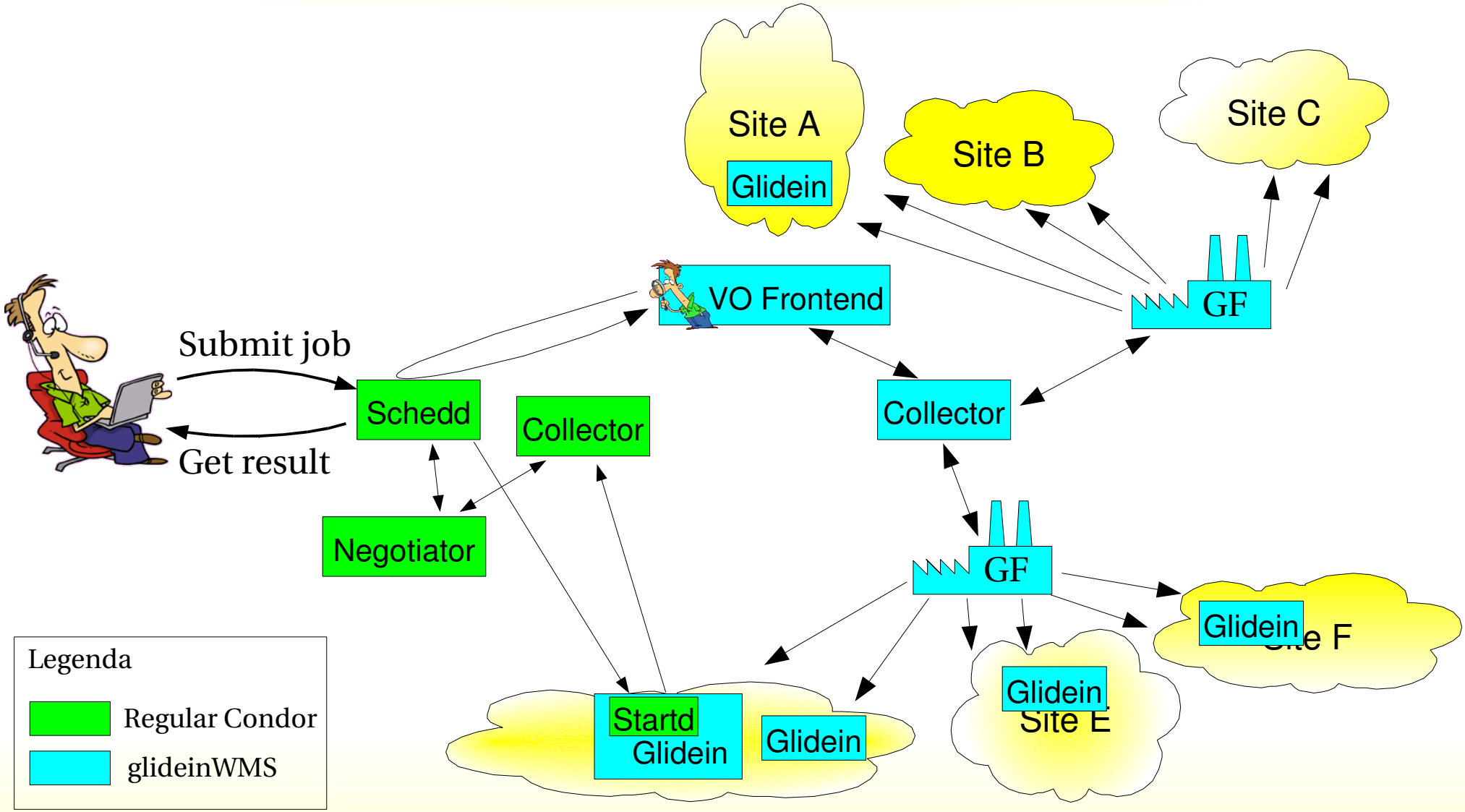


Submitting glideins

- glideinWMS is a system for automatic glidein submission
  - glidein submission triggered by user jobs waiting in the queue
- Composed by two types of services
  - VO frontends – monitor user queues and regulate glidein submission rates
  - Glidein factories – handle glidein configuration and submit glideins
- Uses condor collector as a glue

glideinWMS – An overview

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



GlideinWMS factory config

- Condor-G used for job submission
  - allows submission to several Grids
- Uses static configuration
  - Pilot admins can tailor each site as needed
  - Base configuration easy to generate using Grid information systems (like ReSS and BDII)

GlideinWMS pilot content

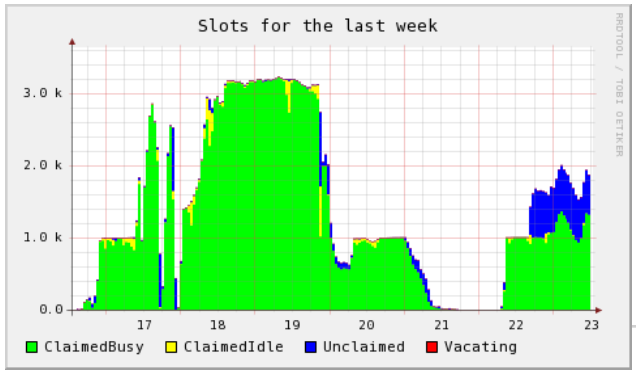
- A simple shell script
  - Downloads other scripts and binaries using HTTP
    - All network transfers have integrity checks
- These additional executables
  - Validate the node
  - Prepare the environment, install user software
  - Configure Condor daemons (policies, security, proxies, etc.)
- Finally, condor_startd is launched
  - does most of the work

Monitoring

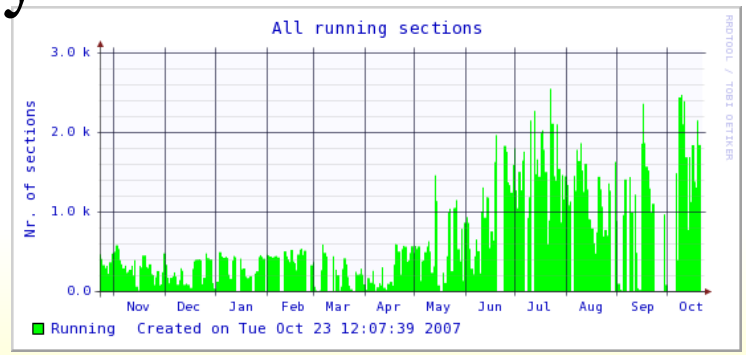
- All the standard monitoring of a Condor local pool
  - condor_q and condor_status
- glideinWMS provides tools for pseudo-interactive monitoring
  - ls, cat, top on the worker nodes
- The glidein factory also maintains a Web based graphical view

Glidein deployments in HEP

- CMS using glideins for production jobs



- ATLAS also using them for production jobs
- CDF using them for user analysis



Conclusions

- The average scientist should not (yet) be exposed directly to the Grid
  - Computing related overhead too high
- Glideins can hide the Grid complexity and make it look as a local computing pool
- Several HEP collaborations are happily using glideins in the real life
  - Other communities could benefit as well
  - glideinWMS is an easy path there

Backup Slides

Security considerations

- GSI security used between endpoints
  - Only trusted glideins can join the pool
  - Prevent man-in-the-middle attacks
- Interfaced to gLExec
  - Adhering to Grid rules when multi-user glideins are used
- Condor GCB used to bridge firewalls

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