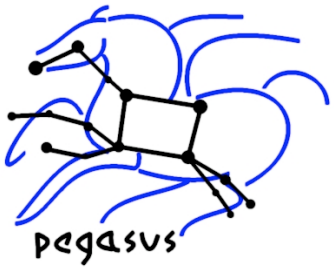
The Glidein Service

Gideon Juve
juve@usc.edu



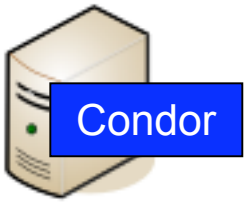
What are glideins?

- A technique for creating temporary, user-controlled Condor pools using resources from remote Grid sites
  1. Grid jobs (called “glideins”) are submitted to grid site using normal mechanisms (Globus, etc.)
  2. Glideins start Condor worker daemons on remote resources
  3. Glidein workers join user’s Condor pool and are used to run application jobs

How glideins work

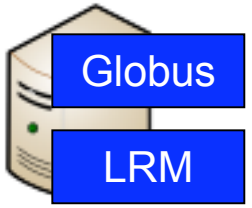
LOCAL SITE

User



Condor Central Manager

GRID SITE

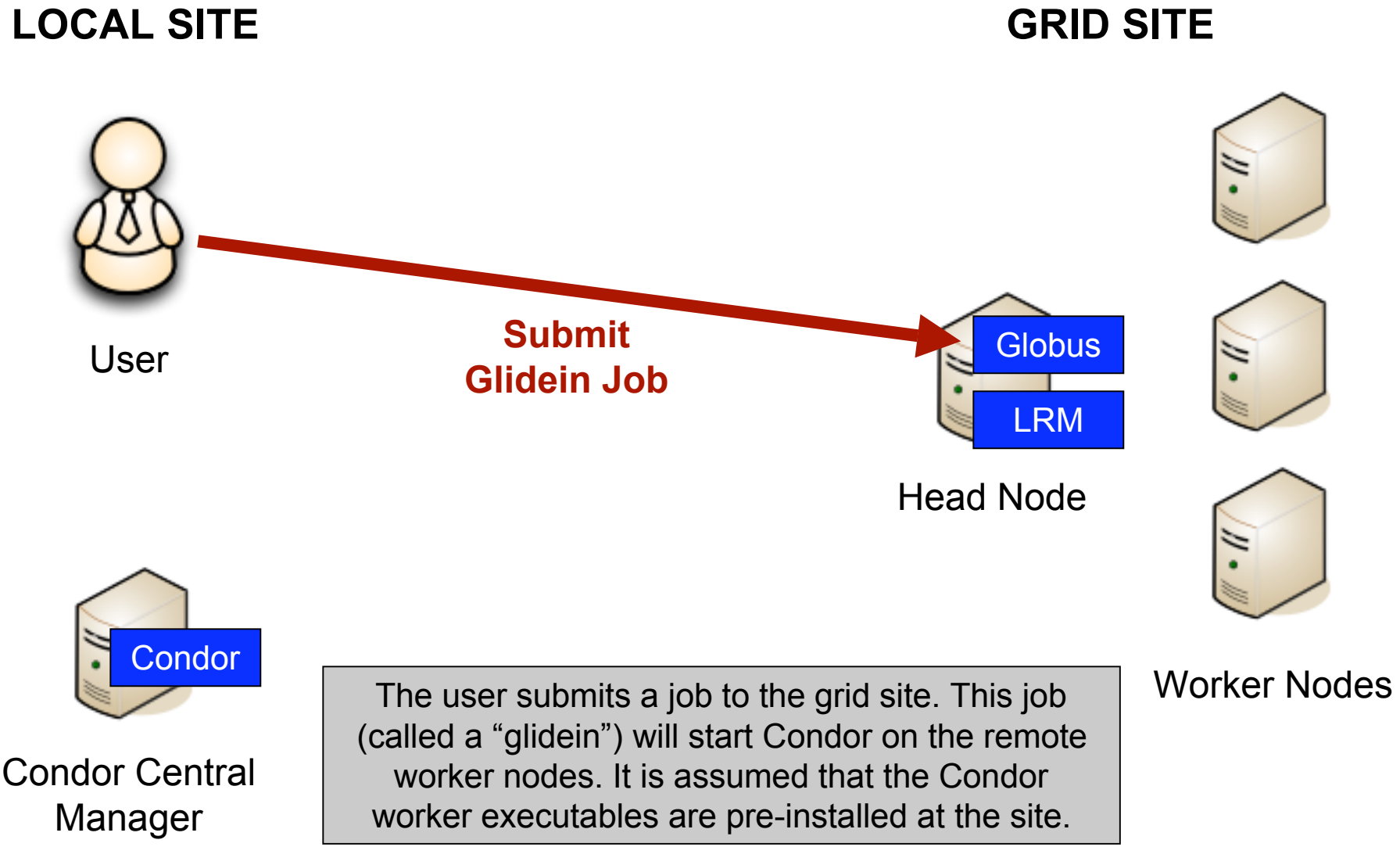


Head Node

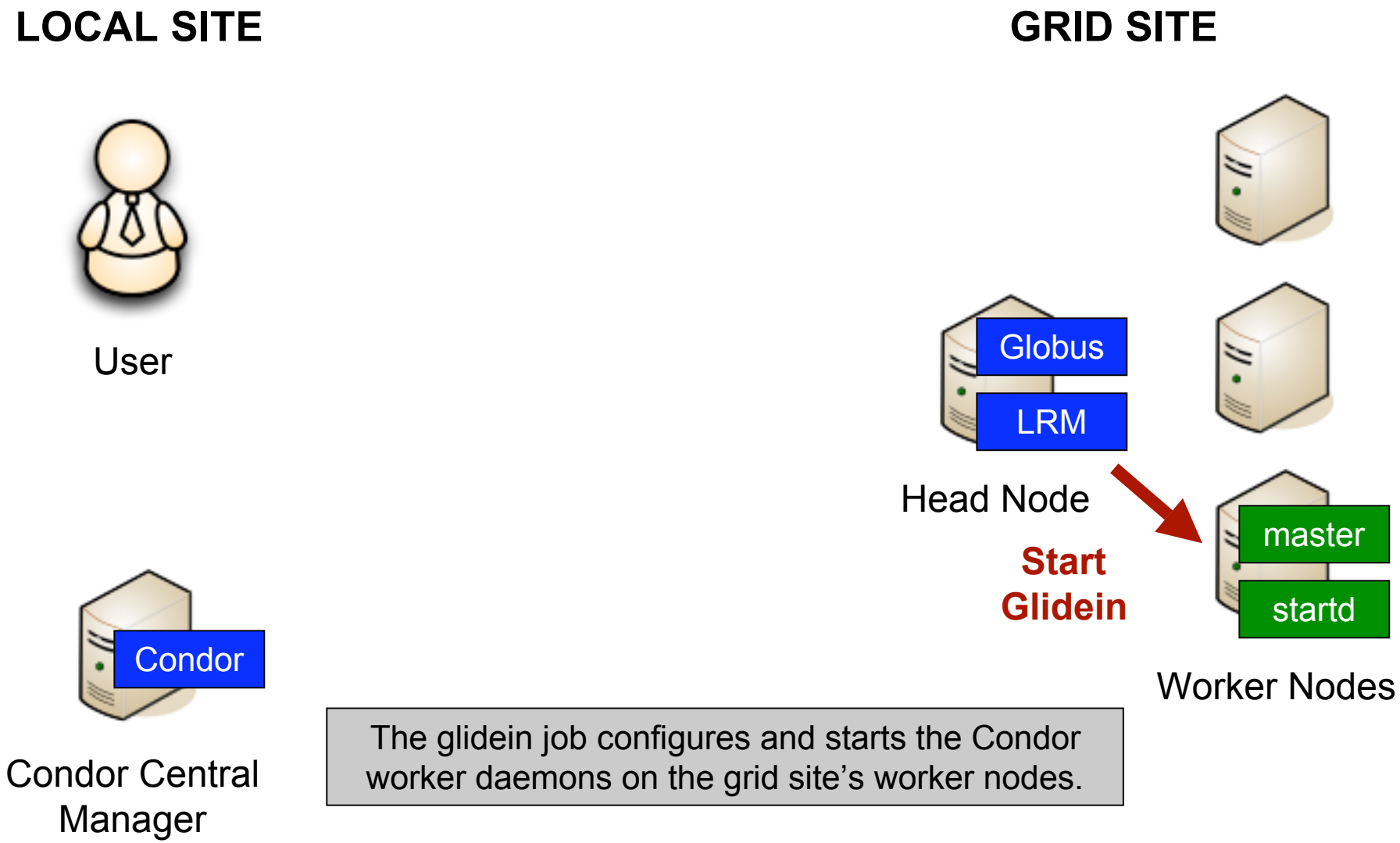
Worker Nodes

To use glideins, the user runs a Condor central manager on a local machine that they control. This Condor pool will manage glidein resources allocated from a remote grid site.

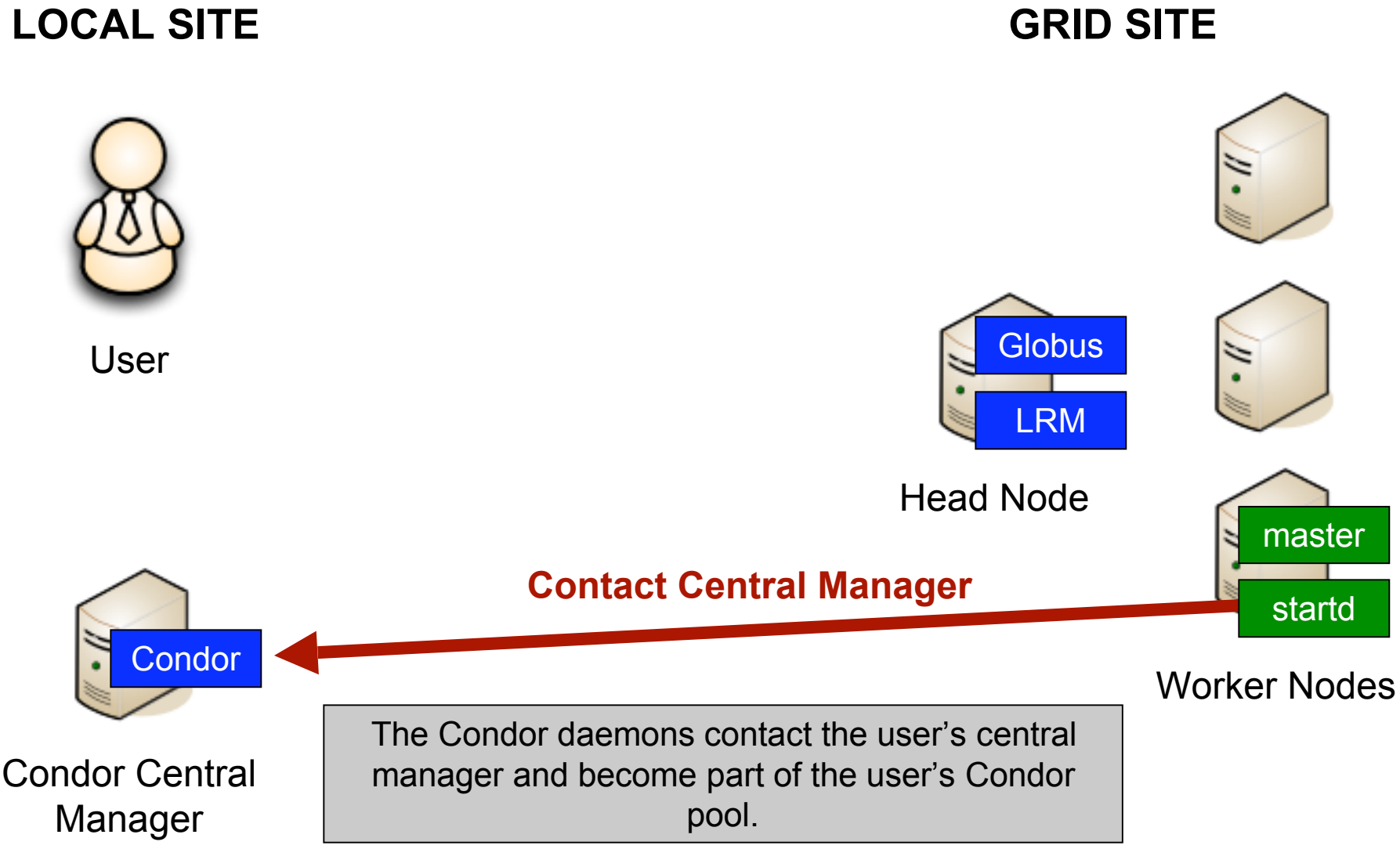
How glideins work



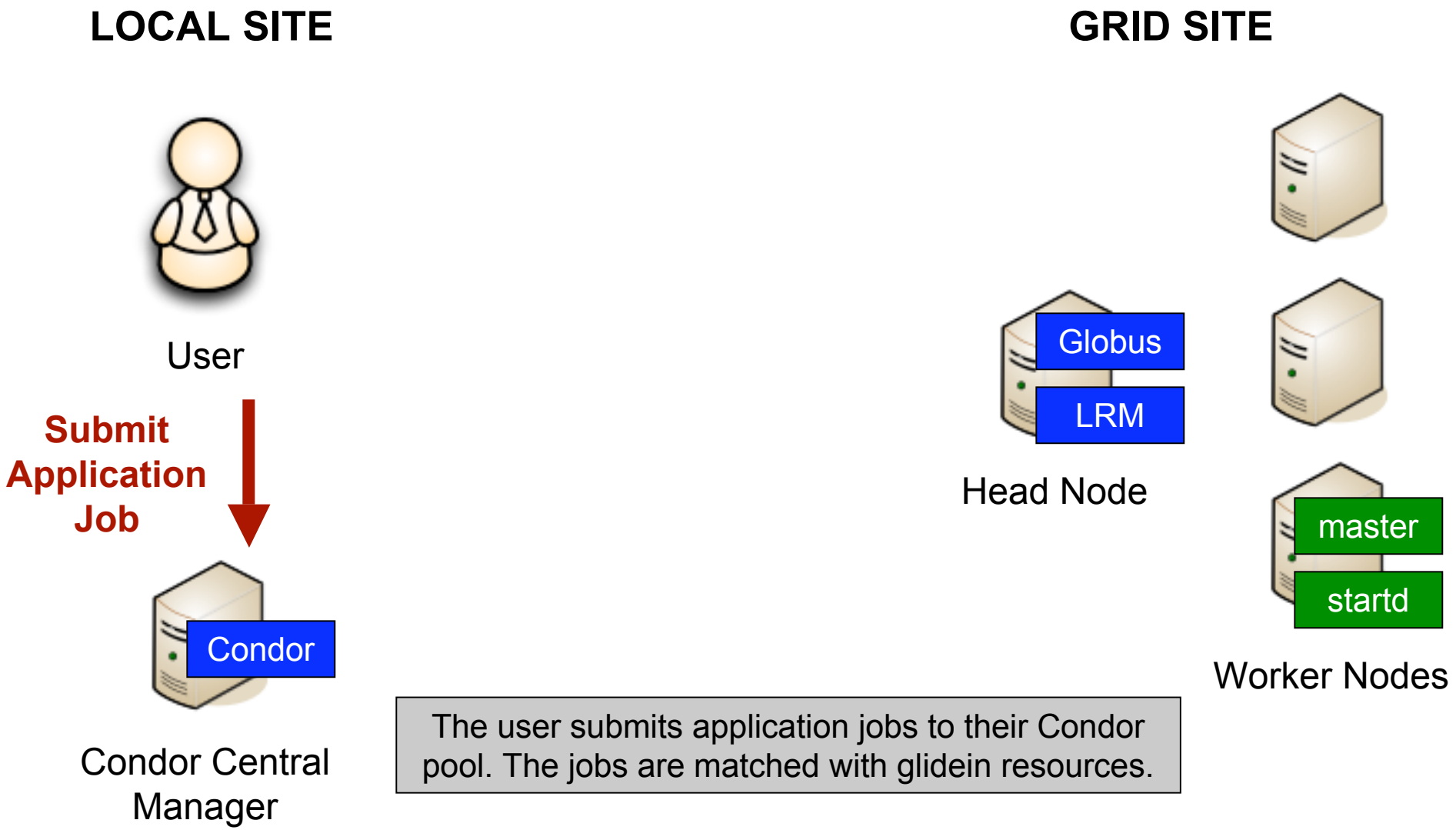
How glideins work



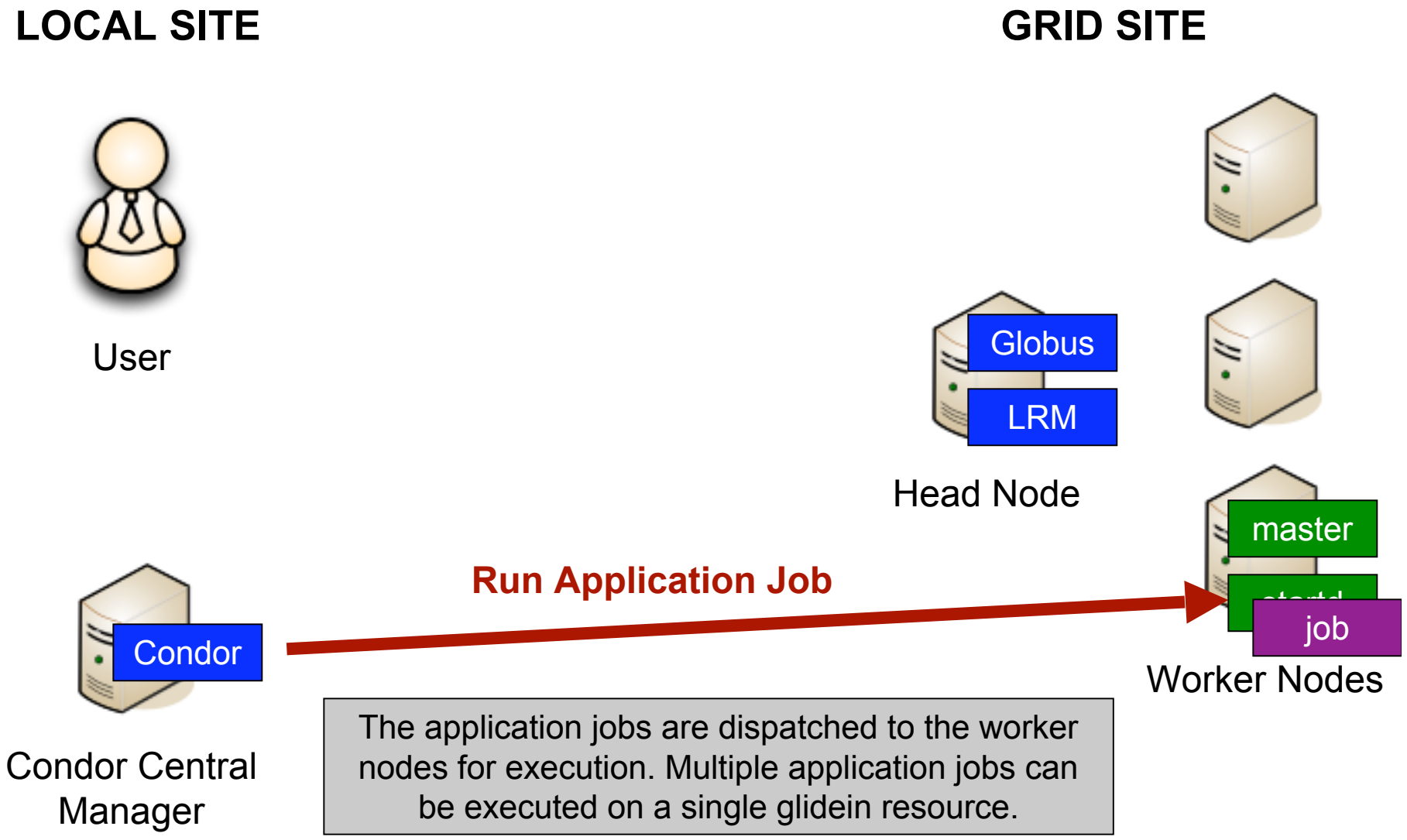
How glideins work



How glideins work



How glideins work



What are glideins good for?

- Running short jobs on the grid
  - Condor can dispatch jobs faster than Globus
- Bypassing site scheduling policies
  - Max submitted/running jobs
  - Priority for large jobs
- Avoiding competition for resources
  - Glideins reserve resources for multiple jobs
- Reducing load on head node/LRM
  - Fewer Globus jobmanagers polling for status

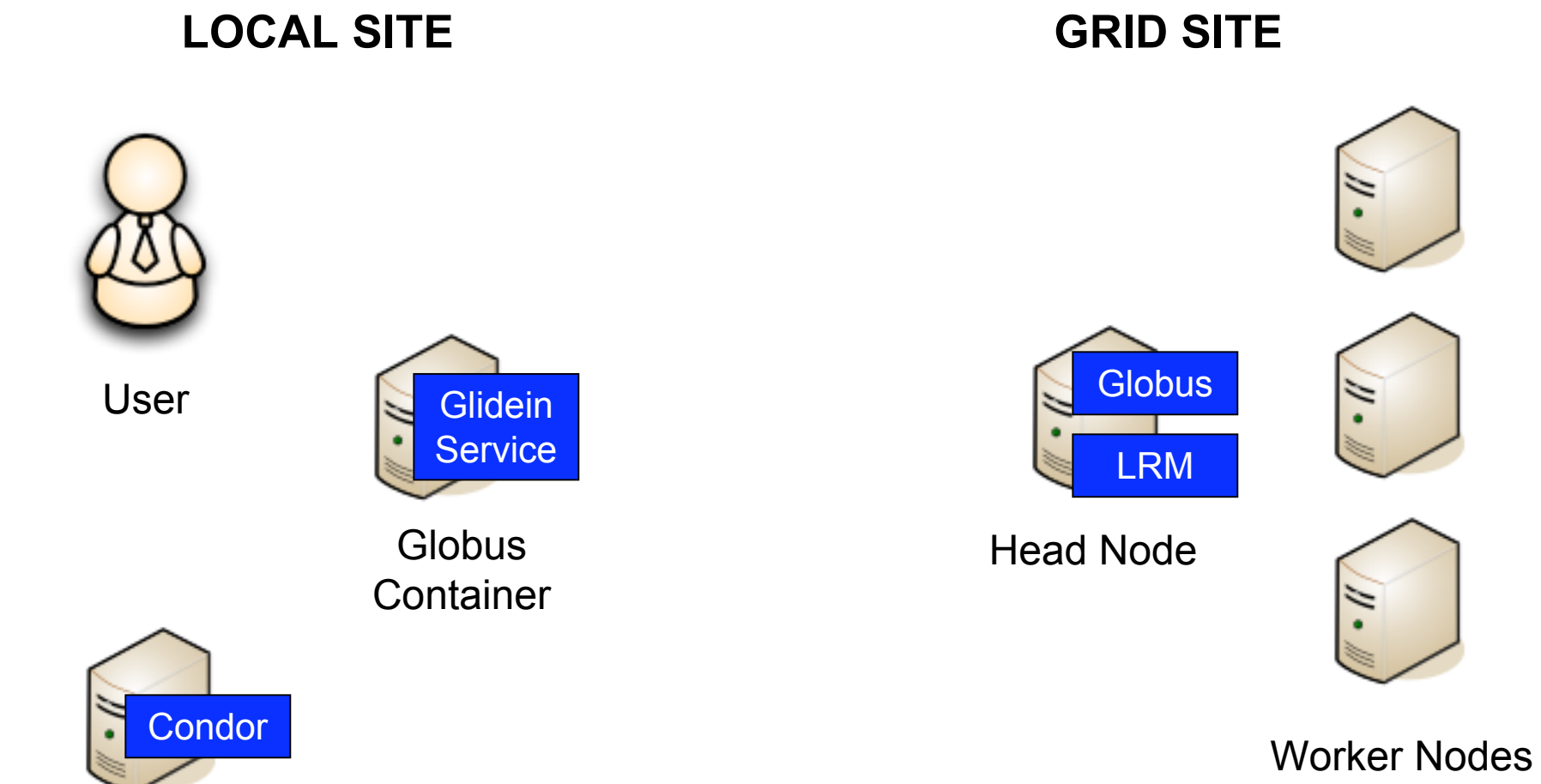
Other Approaches

- Advance Reservations
  - Ask the scheduler for exclusive access to resources
  - Not supported at many sites
  - Typically managed by site administrator (not users)
  - Users are charged a premium for resources
  - Unused reservations cannot be returned
- Task Clustering
  - Group multiple, independent jobs together
  - Can delay the release of some jobs
  - May reduce parallelism
- Not mutually exclusive
  - Can use a combination of techniques

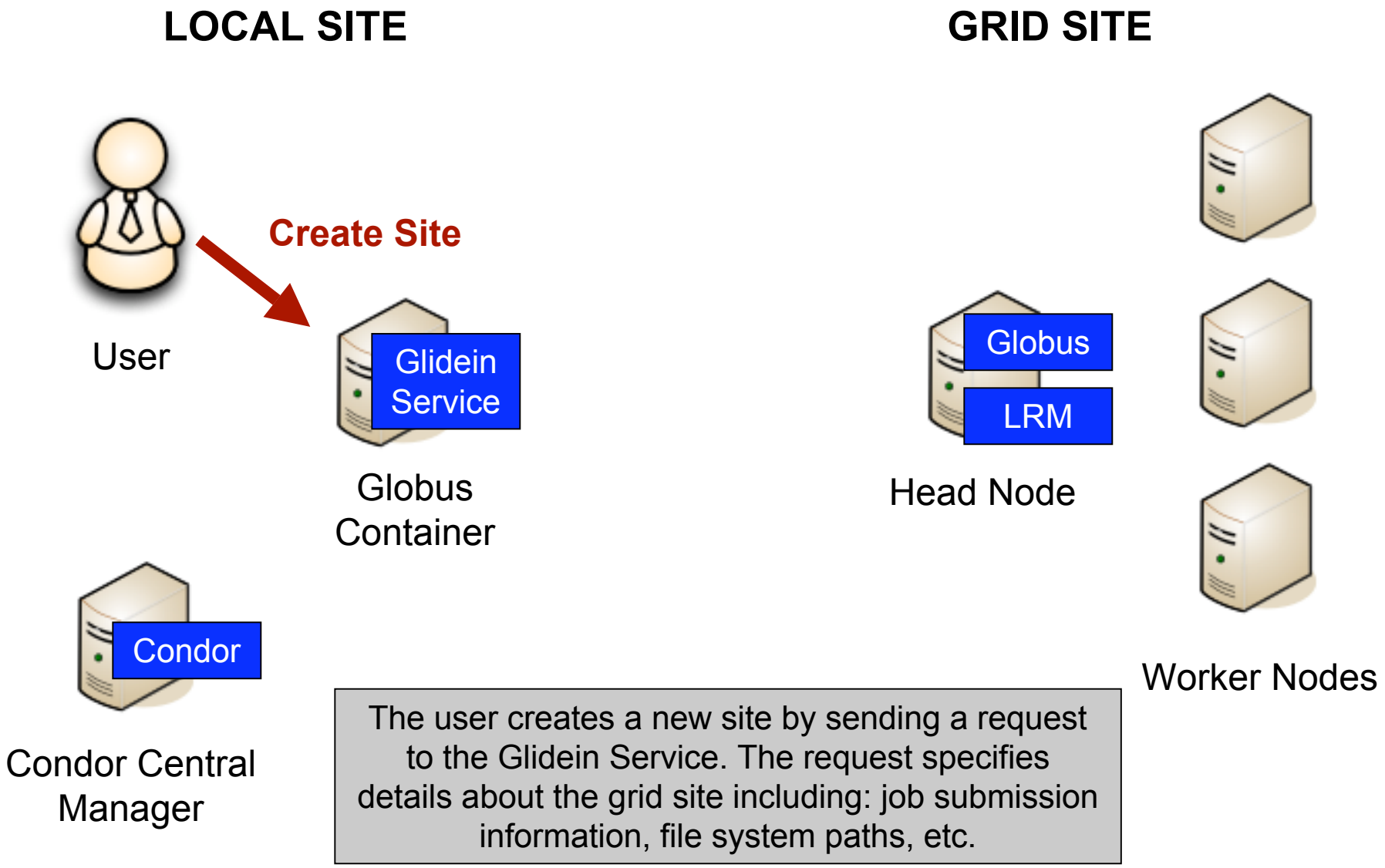
Glidein Service

- GT4 grid service for running glideins
  - Automates the installation and configuration of Condor on grid site
  - Simplifies the complex setup and configuration required to run glideins
- Separate setup and provisioning steps
  1. Create “sites” for remote installation and setup of Condor executables
  2. Create “glideins” for resource allocation

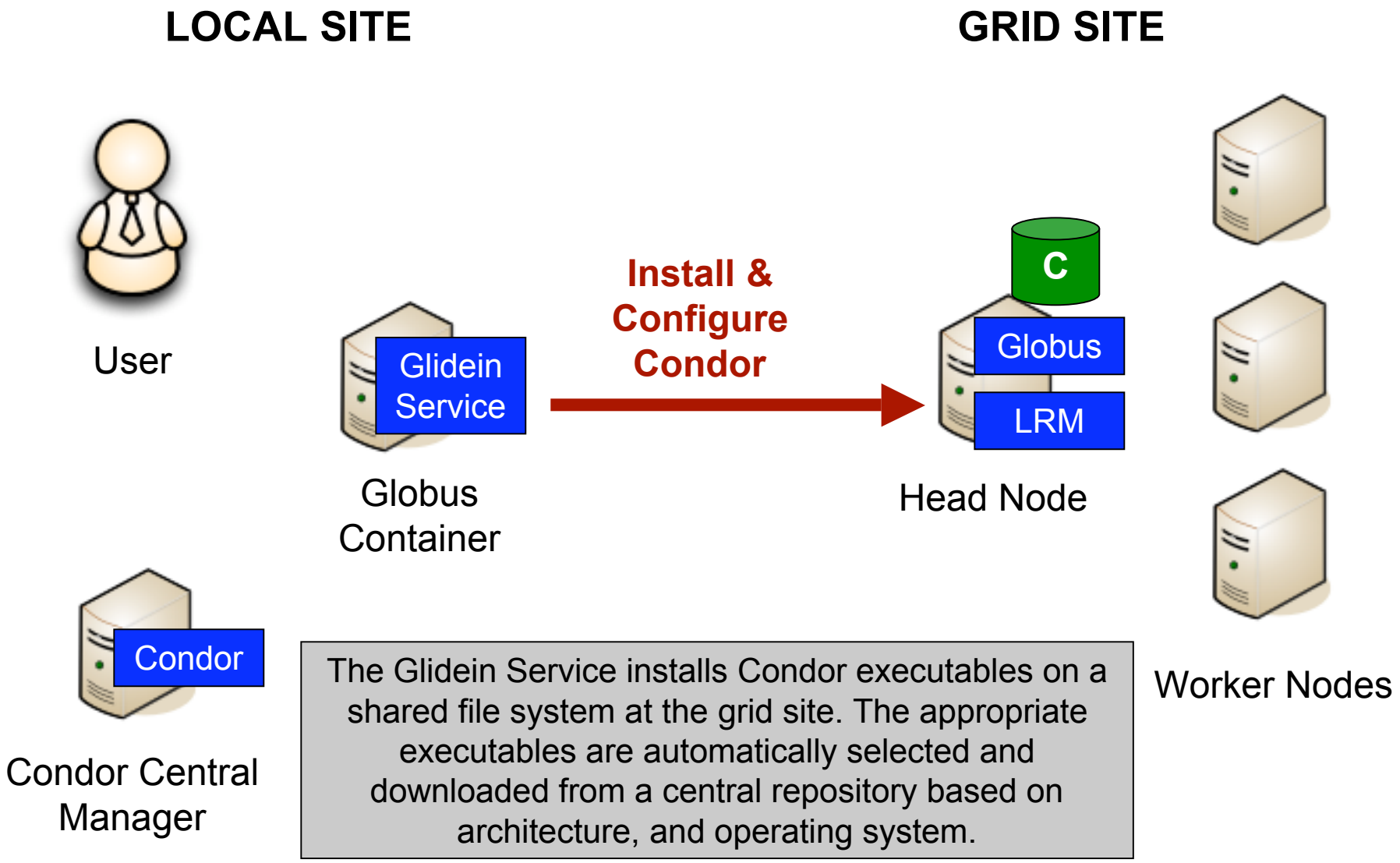
How the Glidein Service works



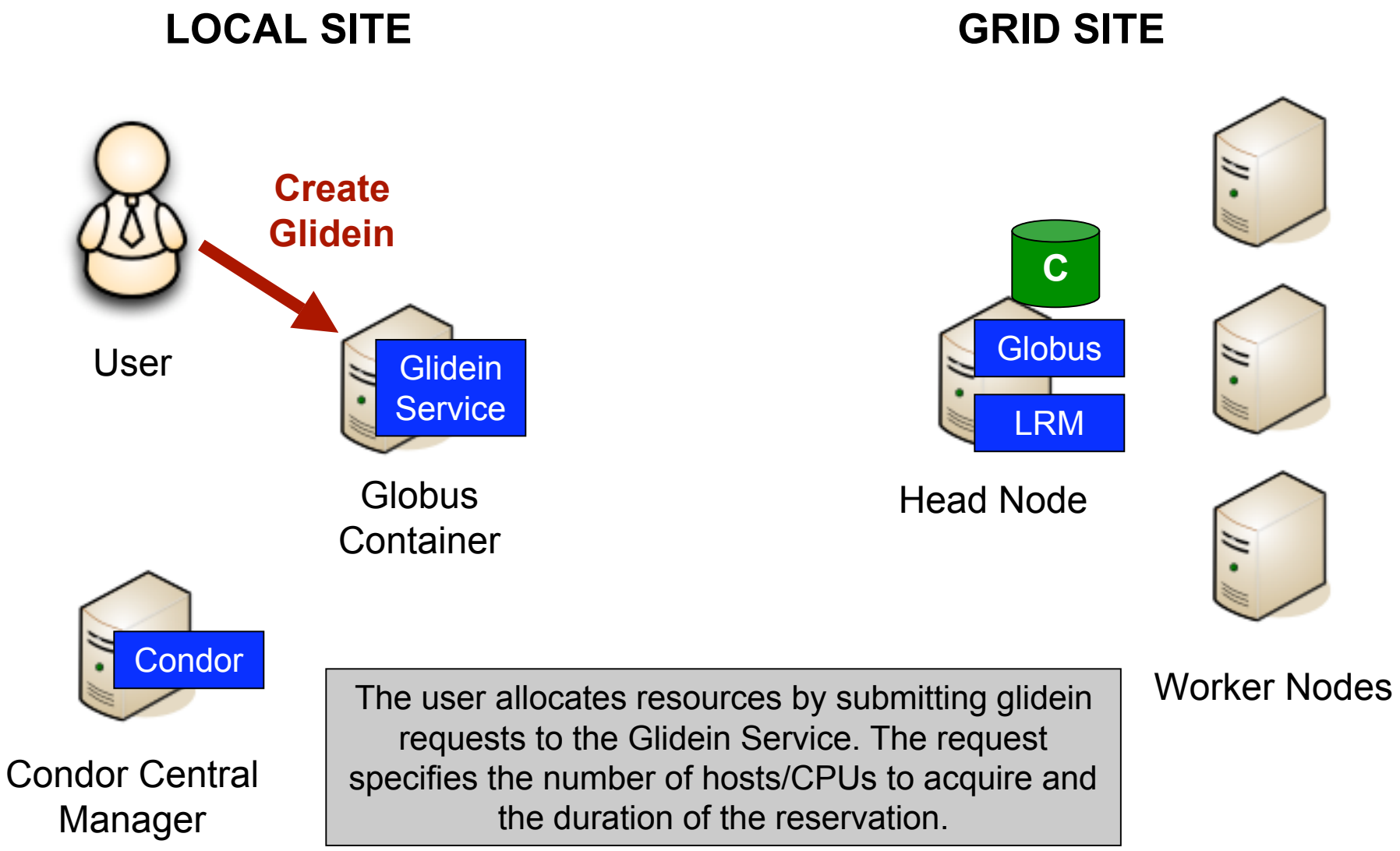
How the Glidein Service works



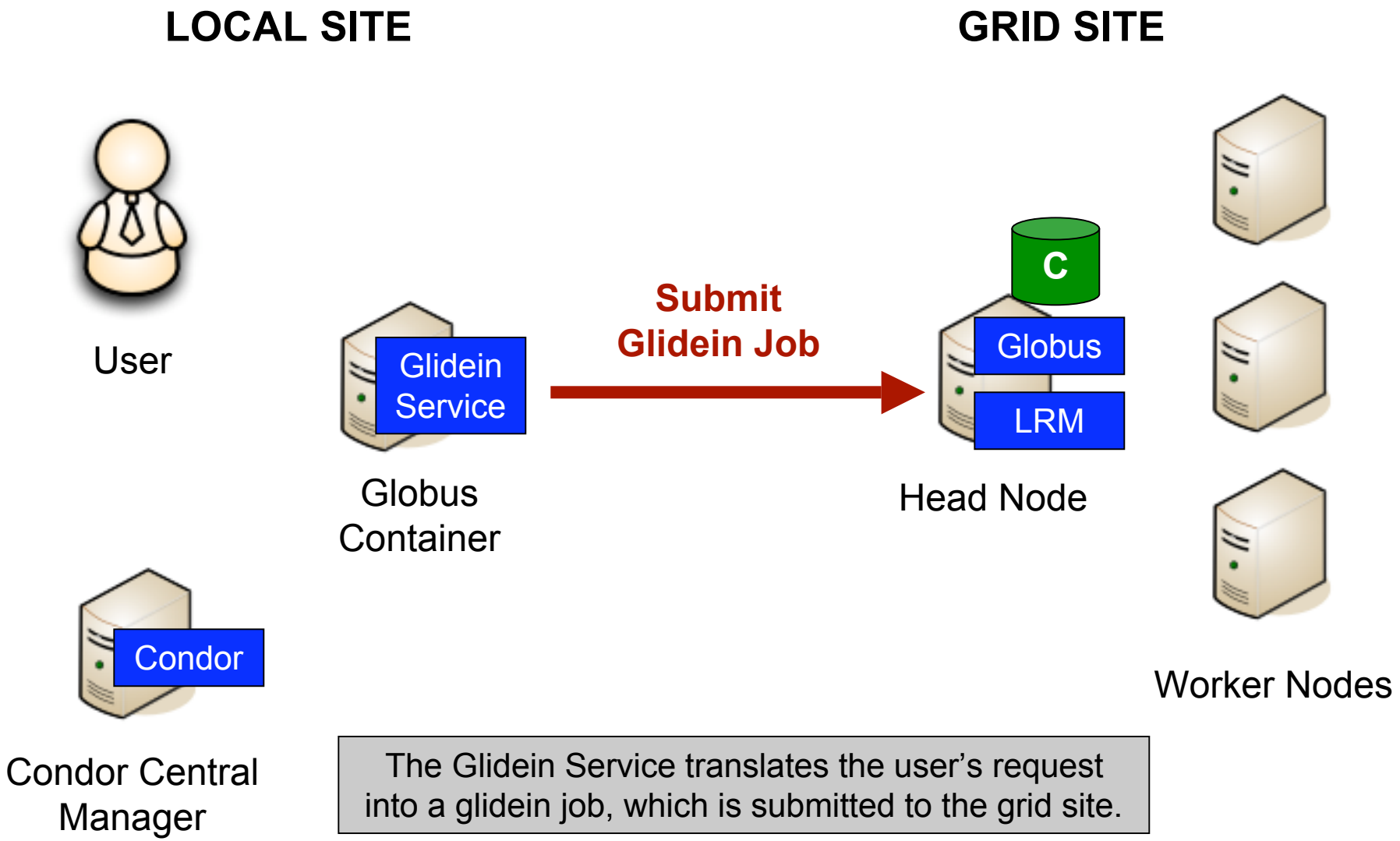
How the Glidein Service works



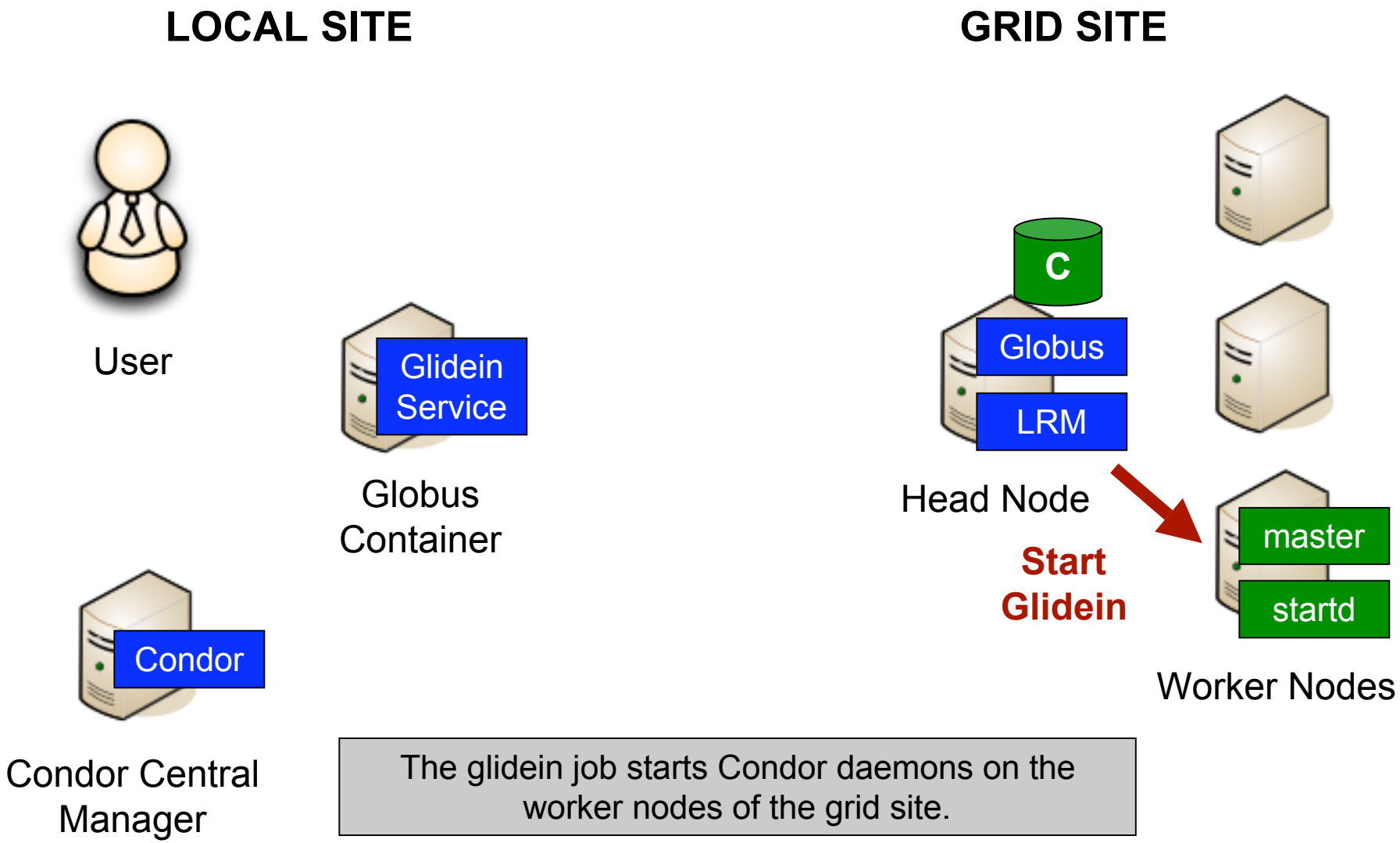
How the Glidein Service works



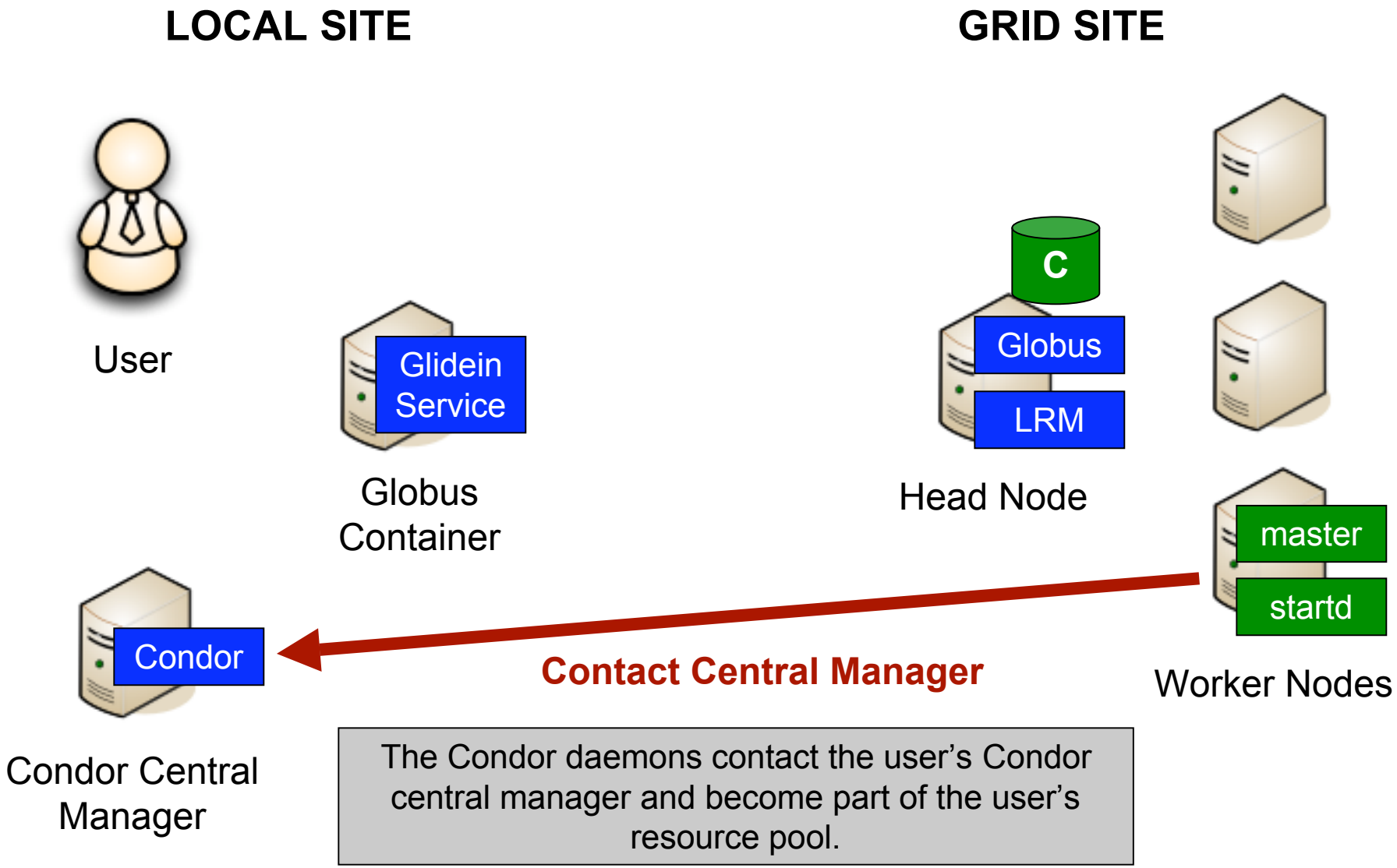
How the Glidein Service works



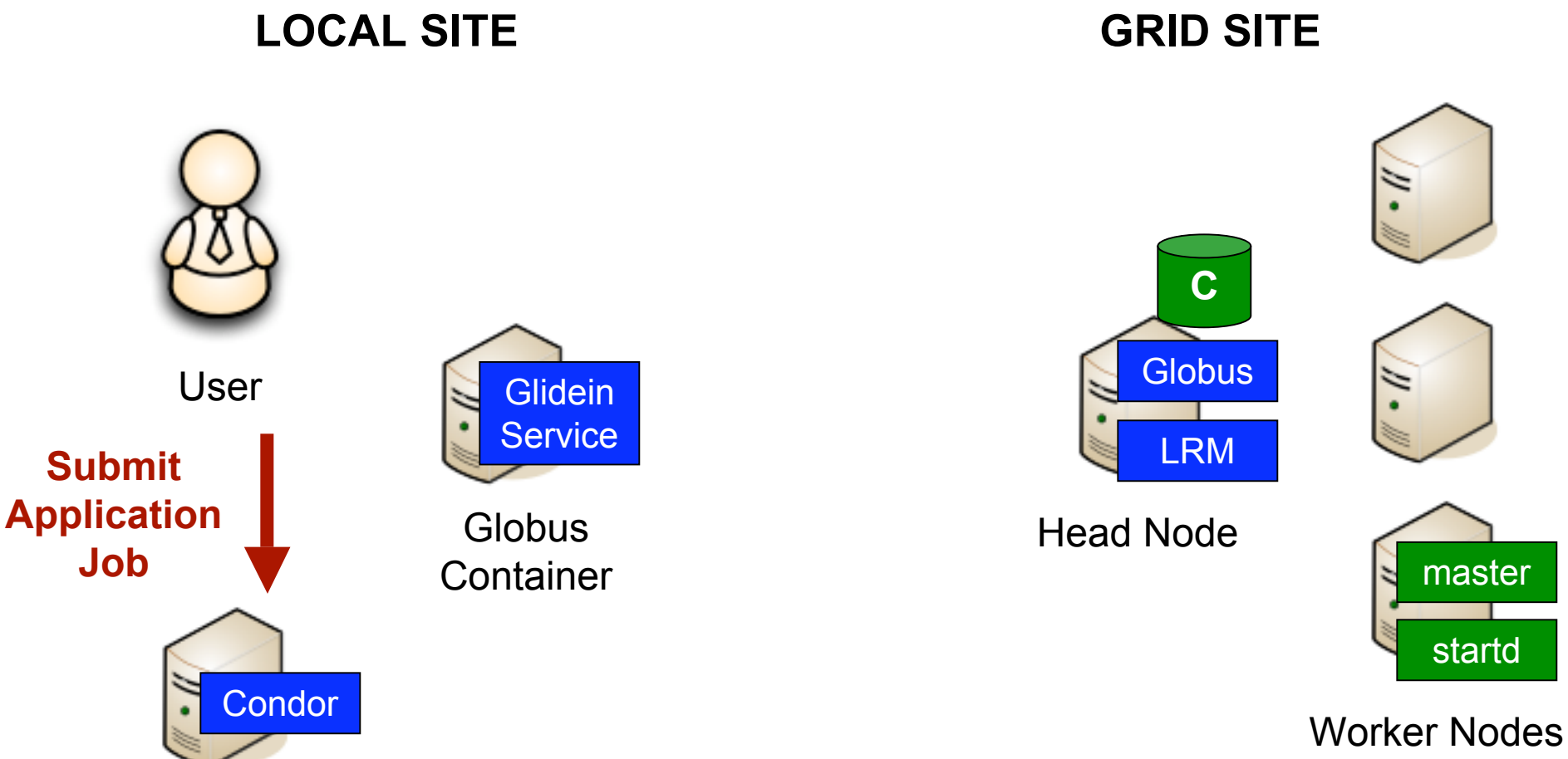
How the Glidein Service works



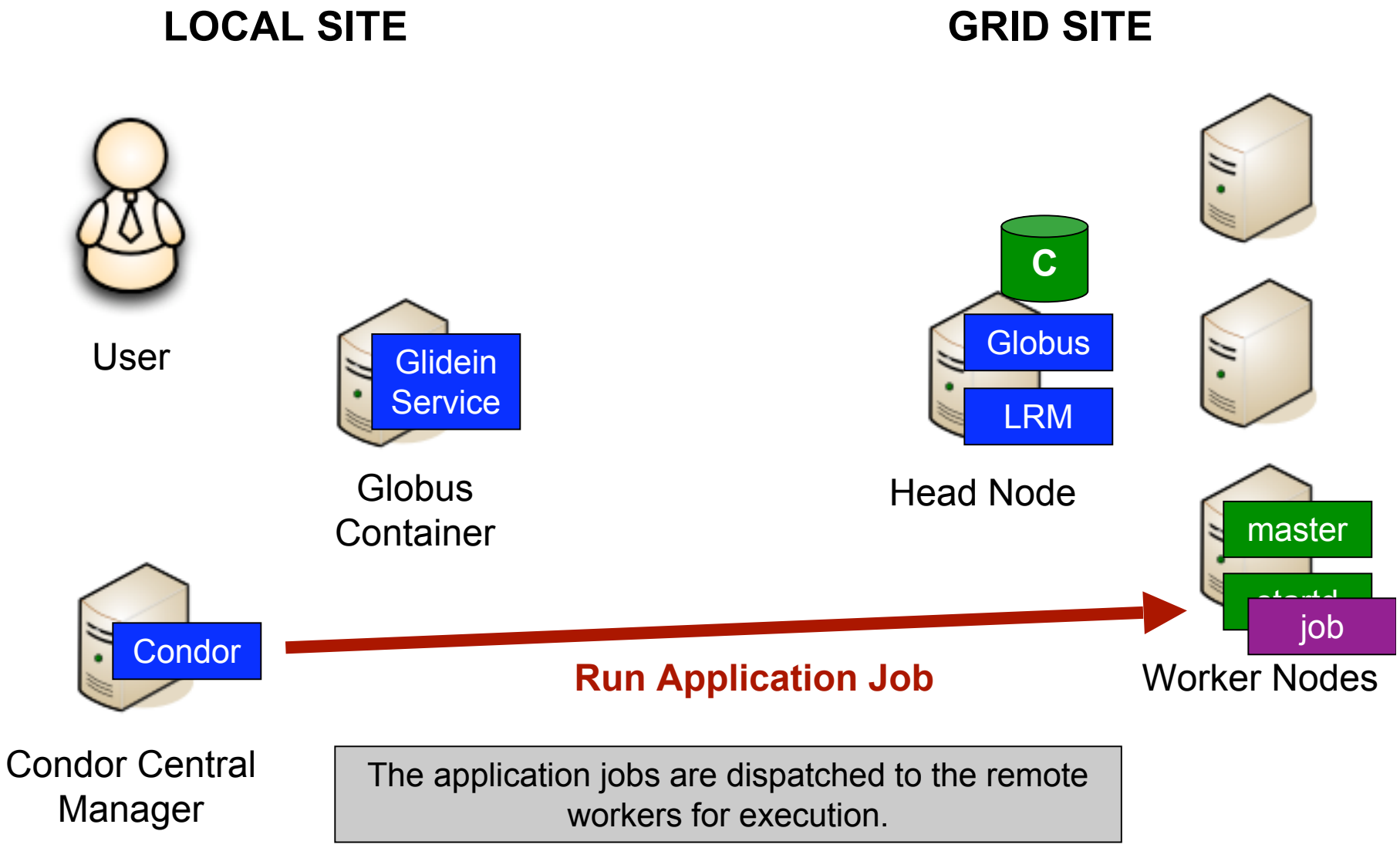
How the Glidein Service works



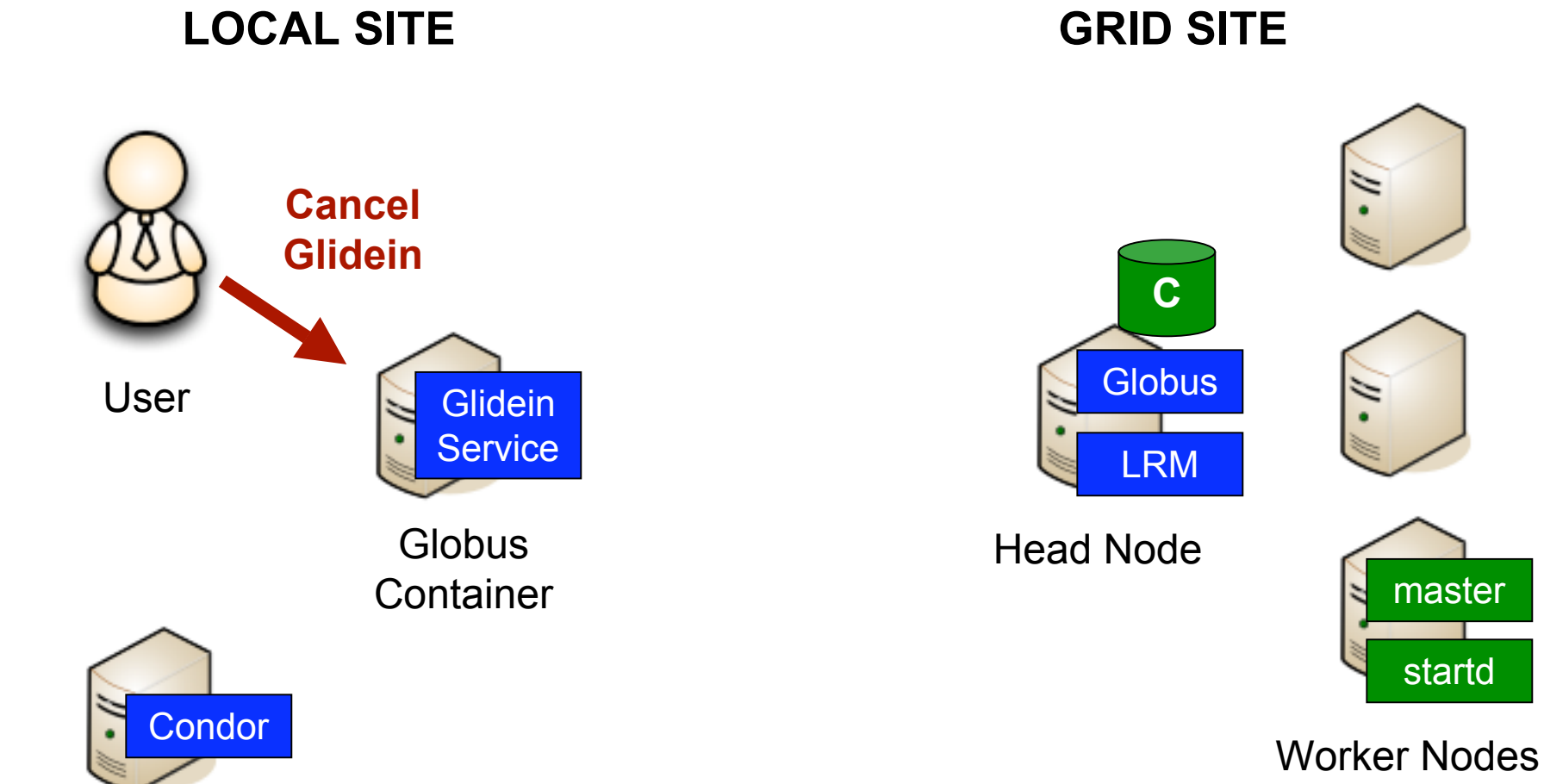
How the Glidein Service works



How the Glidein Service works

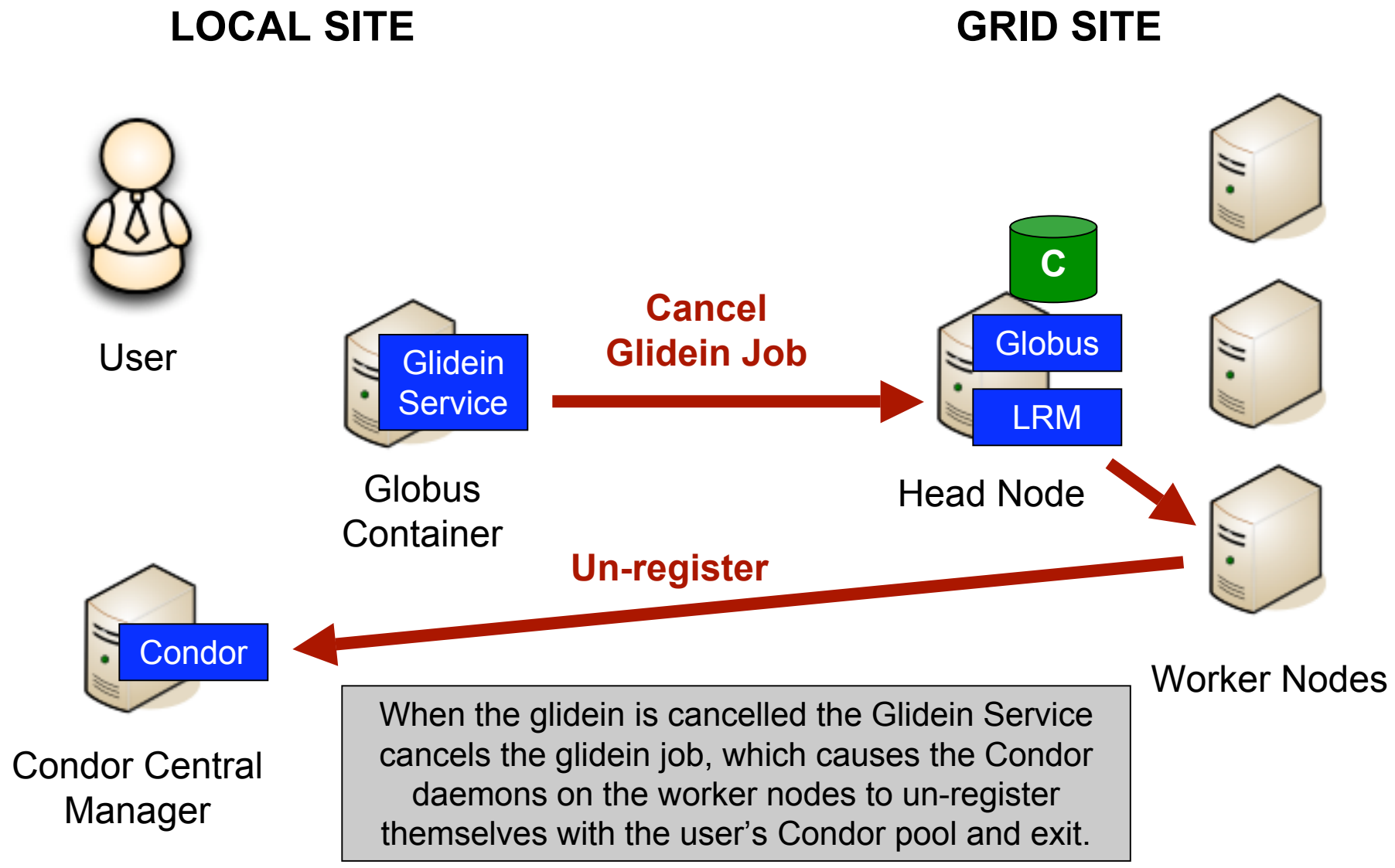


How the Glidein Service works

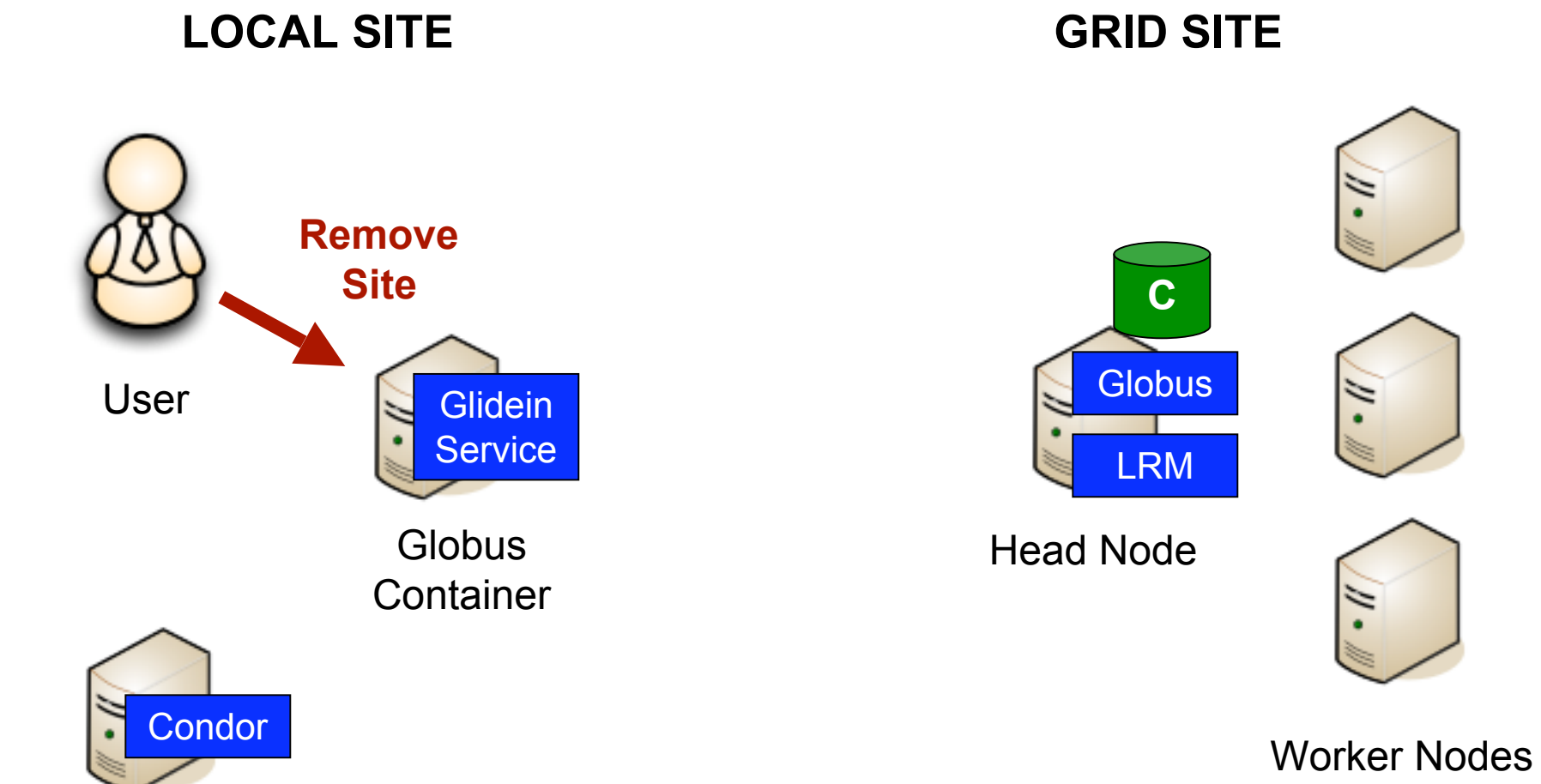


When the user is finished with their application they cancel their glidein request (or let it expire automatically).

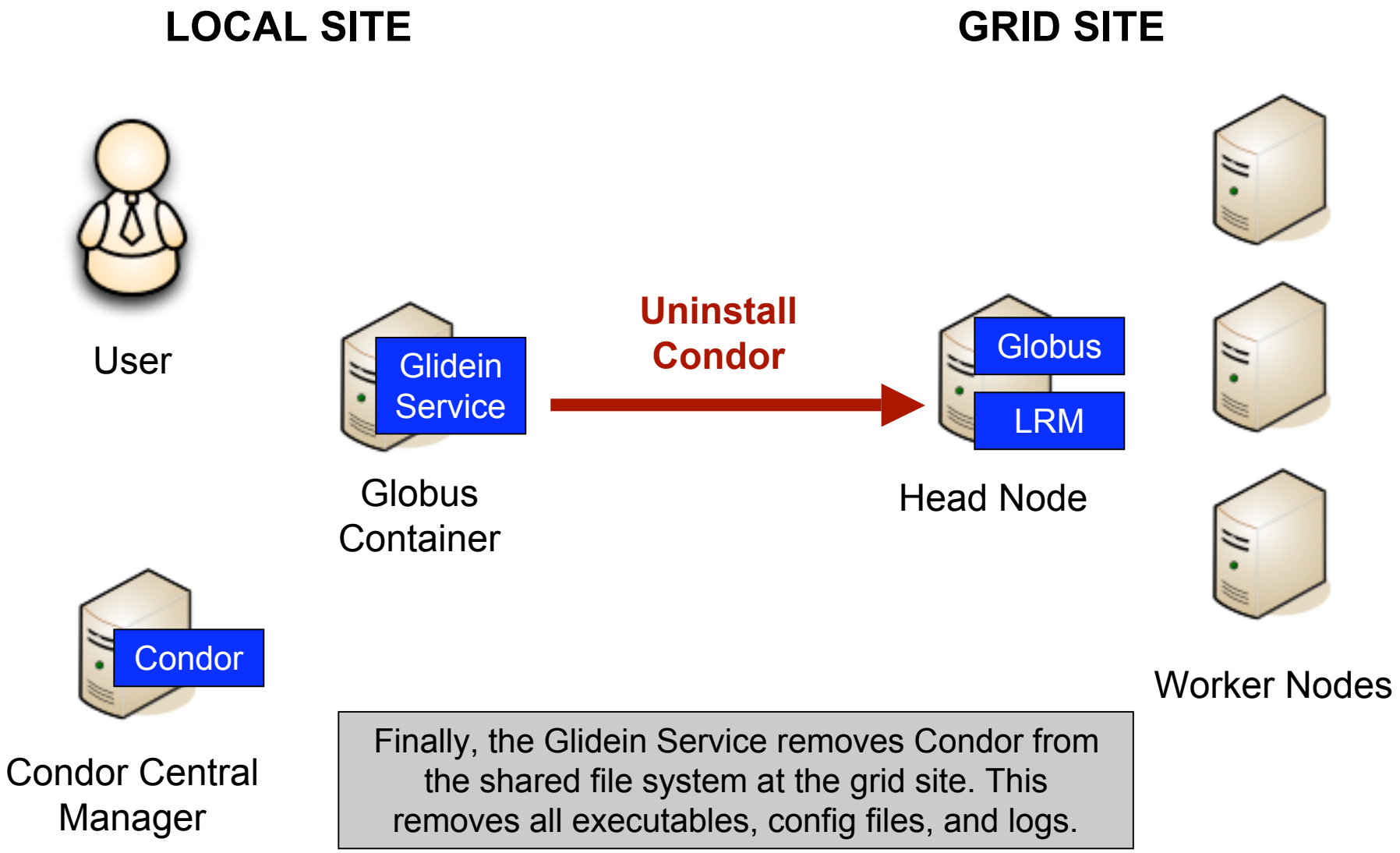
How the Glidein Service works



How the Glidein Service works



How the Glidein Service works



Features

- Auto-configuration
  - Detect architecture, OS, glibc -> Condor package
  - Determine public IP
  - Generate Condor config file
- Multiple interfaces
  - Command-line
  - SOAP
  - Java API
- Automatic resubmission of glideins
  - Indefinitely, N times, until date/time
- Notifications
- History tracking

Challenges

- Firewalls / Private IPs
  - Hinder communication between glideins and user's pool
  - Potential solutions
    - GCB
    - VPN?
    - Run central manager on head node (restrictions)
- Shared File System
  - Currently required for Glidein Service to stage Condor executables

Command Line Example

```
$ glidein create-site --site-name mercury --condor-version 7.0.0 \
          --environment GLOBUS_LOCATION=/usr/local/globus-4.0.1-r3 \
          --install-path /home/ac/juve/glidein \
          --local-path /gpfs_scratch1/juve/glidein \
          --staging-service 'gt2 grid-hg.ncsa.teragrid.org/jobmanager-fork' \
          --glidein-service 'gt2 grid-hg.ncsa.teragrid.org/jobmanager-pbs'
```

```
$ glidein list-site
ID      NAME            CREATED         LAST UPDATE     STATE     MESSAGE
3       mercury         08-27 09:16     08-27 10:17     READY     Installed
```

```
$ glidein create-glidein --site 3 --count 1 --host-count 1 --num-cpus 1 \
          --wall-time 30 --condor-host juve.isi.edu
```

```
$ glidein list-glidein
ID   SITE          CONDOR HOST     SLOTS  WTIME  CREATED       LAST UPDATE   STATE     MESSAGE
1    mercury (3)   juve.isi.edu     1     30     08-27 09:23   08-27 10:24   RUNNING   Running
```

```
$ condor_status

Name                OpSys      Arch    State     Activity LoadAv Mem   ActvtyTime

tg-c408.ncsa.terag LINUX      IA64  Unclaimed Idle     0.100  1024  0+00:00:05

                     Total Owner Claimed Unclaimed Matched Preempting Backfill

            IA64/LINUX     1     0       0         1       0          0        0

                Total      1     0       0         1       0          0        0
```