



# GlideinWMS's use of cvmfsexec

Marco Mambelli, Namratha Urs - Fermilab

February 1, 2021

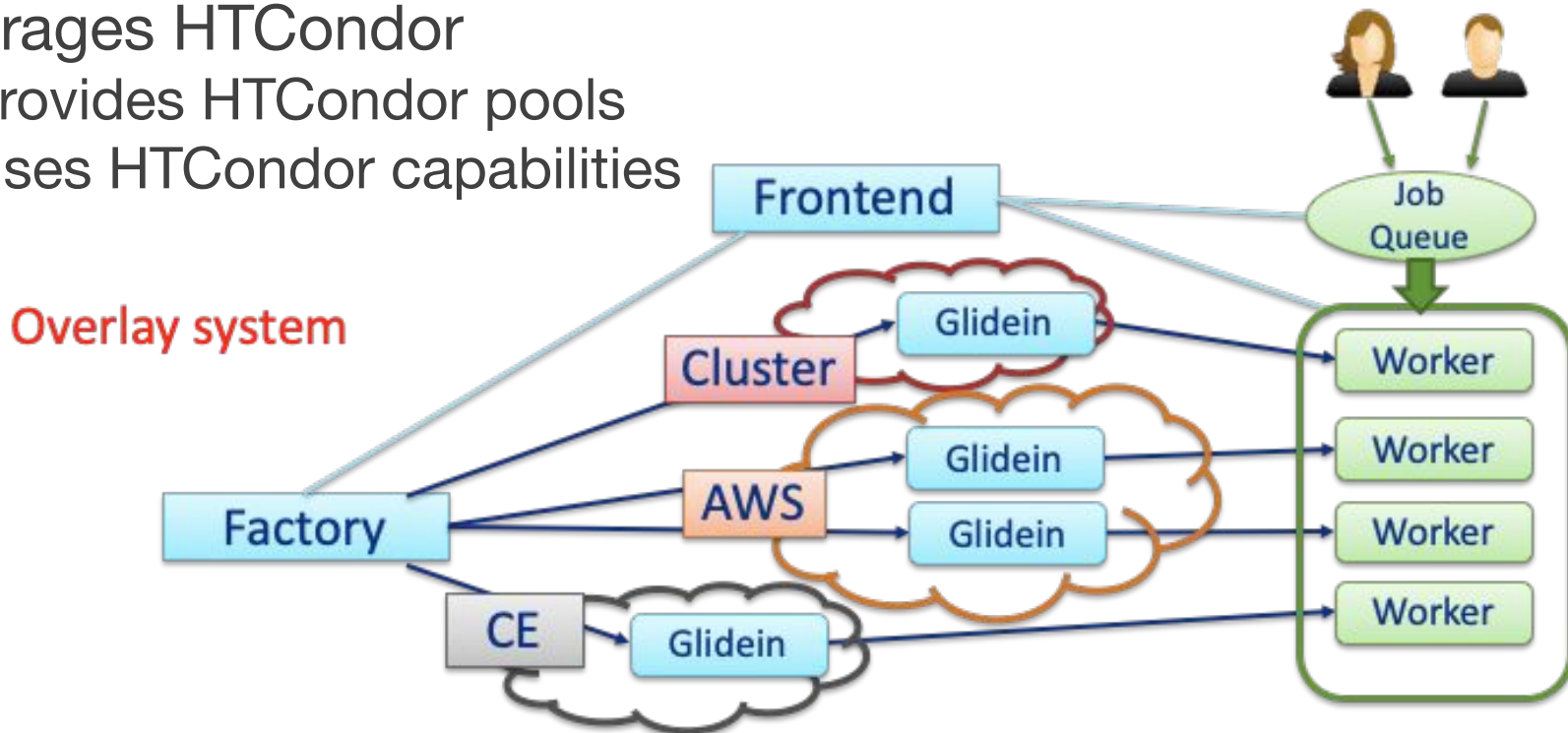
CernVM Users Workshop

NIKHEF (remote)

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

# GlideinWMS

- GlideinWMS is a pilot based resource provisioning tool for distributed High Throughput Computing
- Provides reliable and uniform virtual clusters
- Submits Glideins to unreliable heterogeneous resources
- Leverages HTCondor
  - Provides HTCondor pools
  - Uses HTCondor capabilities



# Frontend

- Monitors jobs to see how many Glideins are needed
- Compares what entries (sites) are available
- Requests Glideins from the Factory
- Requests Factory to kill Glideins if there are too many
- Pressure-based system
  - Works keeping a certain number of Glideins running or idle at the sites
  - Gradual Glideins requests to avoid spikes and overloads
- Manages credentials and delegates them to the Factory

# Factory

- A Glidein Factory knows how to submit to sites
  - Sites are described in a local configuration
  - Only trusted and tested sites are included
- Each site entry in the configuration contains
  - Contact info (hostname, resource type, queue name)
  - Site configuration (startup dir, OS type, ...)
  - VOs authorized/supported
  - Other attributes (Site name, core count, max memory, ...)
  - Glideins can also auto-detect resources
- Configuration can be auto-generated (e.g. from CRIC), admin curated, stored in VCS (e.g. GitHub)
- Condor does the heavy lifting of submissions.

# Glidein: node testing and customization

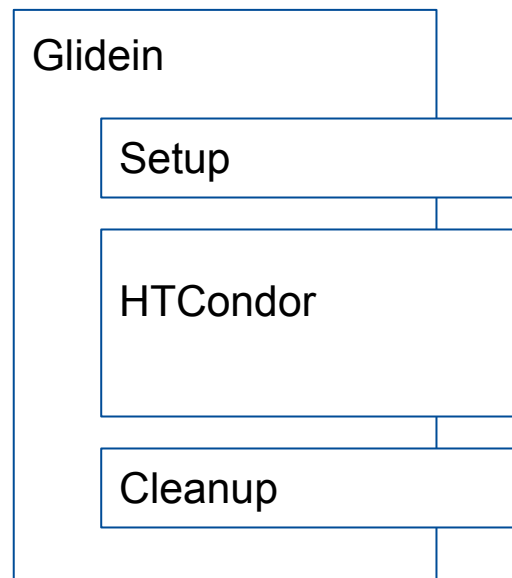
- Mostly shell scripts
- Scouts for resources and validates the Worker node
  - Cores, memory, disk, GPU, ...
  - OS, software installed
  - CVMFS
  - VO specific tests
- Customizes the Worker node
  - Environment, GPU libraries, ...
  - Starting containers (Singularity, ...)
  - Setup of CVMFS
  - VO specific setup
- Provides a reliable and customized execute node to HTCondor
- Reports back to the Factory

# GlideinWMS and CernVM-FS

- Used by the Glideins
- Hosts the Singularity binary provided by OSG
- Hosts most Singularity images shared by OSG
  - Fermilab worker nodes replicas
  - OSG images (default for GlideinWMS)
- Hosts the software of several VOs

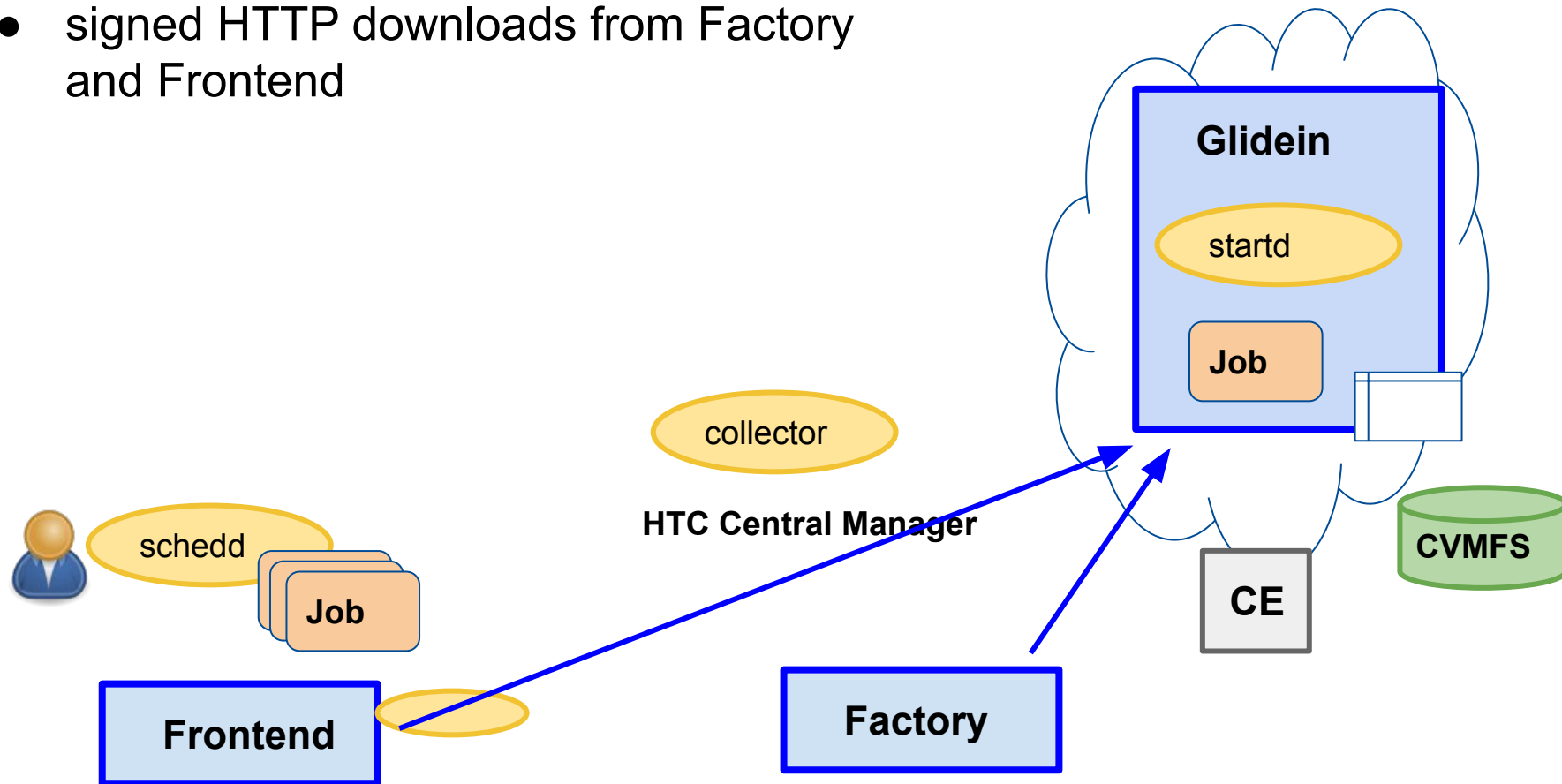
## Glidein structure (current\*)

- Spawns multiple Glideins if needed (MPI, multi-glidein, ...)
- Initial checks
- Download of scripts and setup
  - includes CVMFS mounting and Singularity testing
- Start HTCondor startd and join the pool
  - Start Singularity for each job individually (via wrapper)
- Cleanup
  - included modular scripts



# Glidein downloads

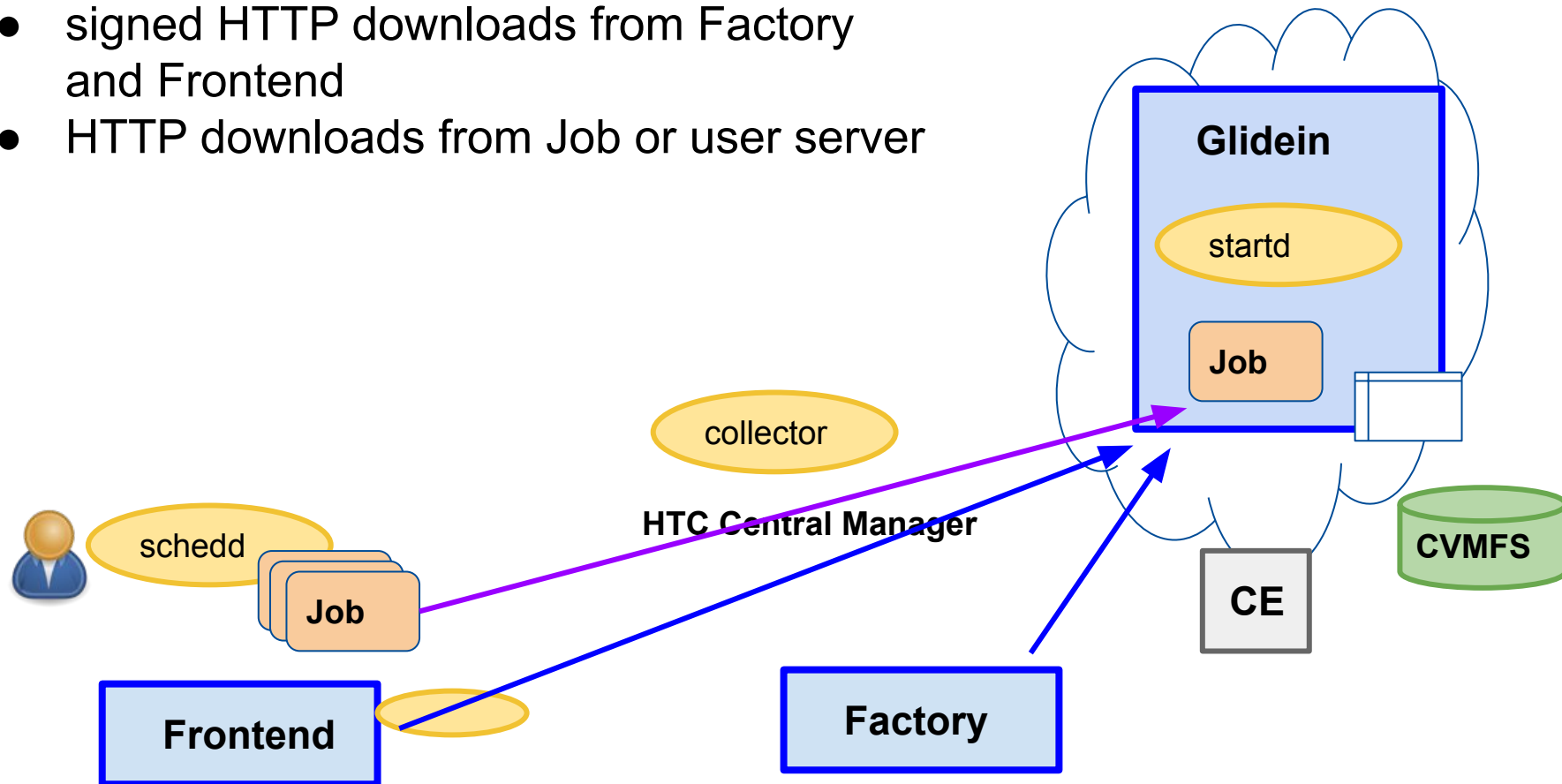
- signed HTTP downloads from Factory and Frontend





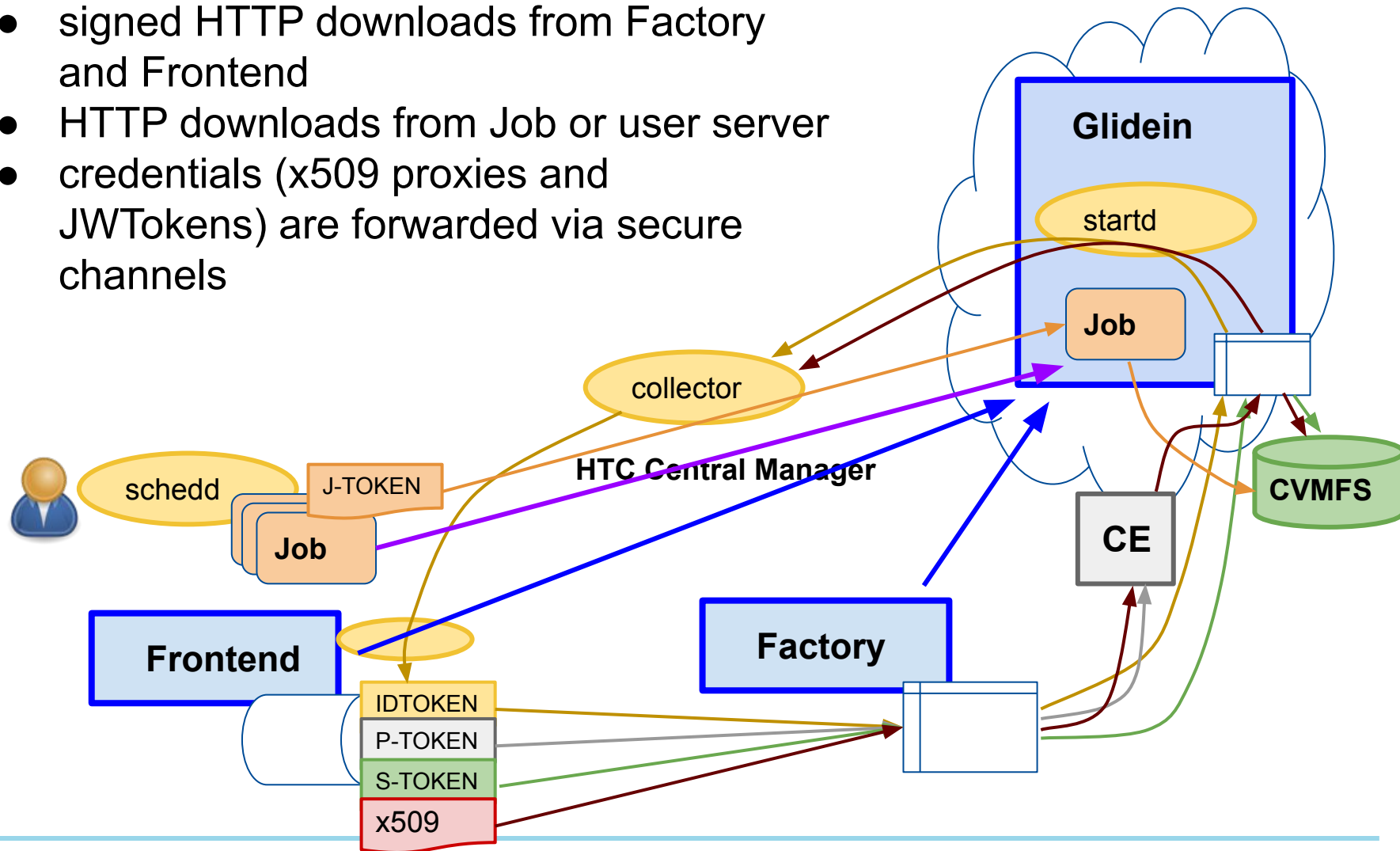
# Glidein downloads

- signed HTTP downloads from Factory and Frontend
- HTTP downloads from Job or user server



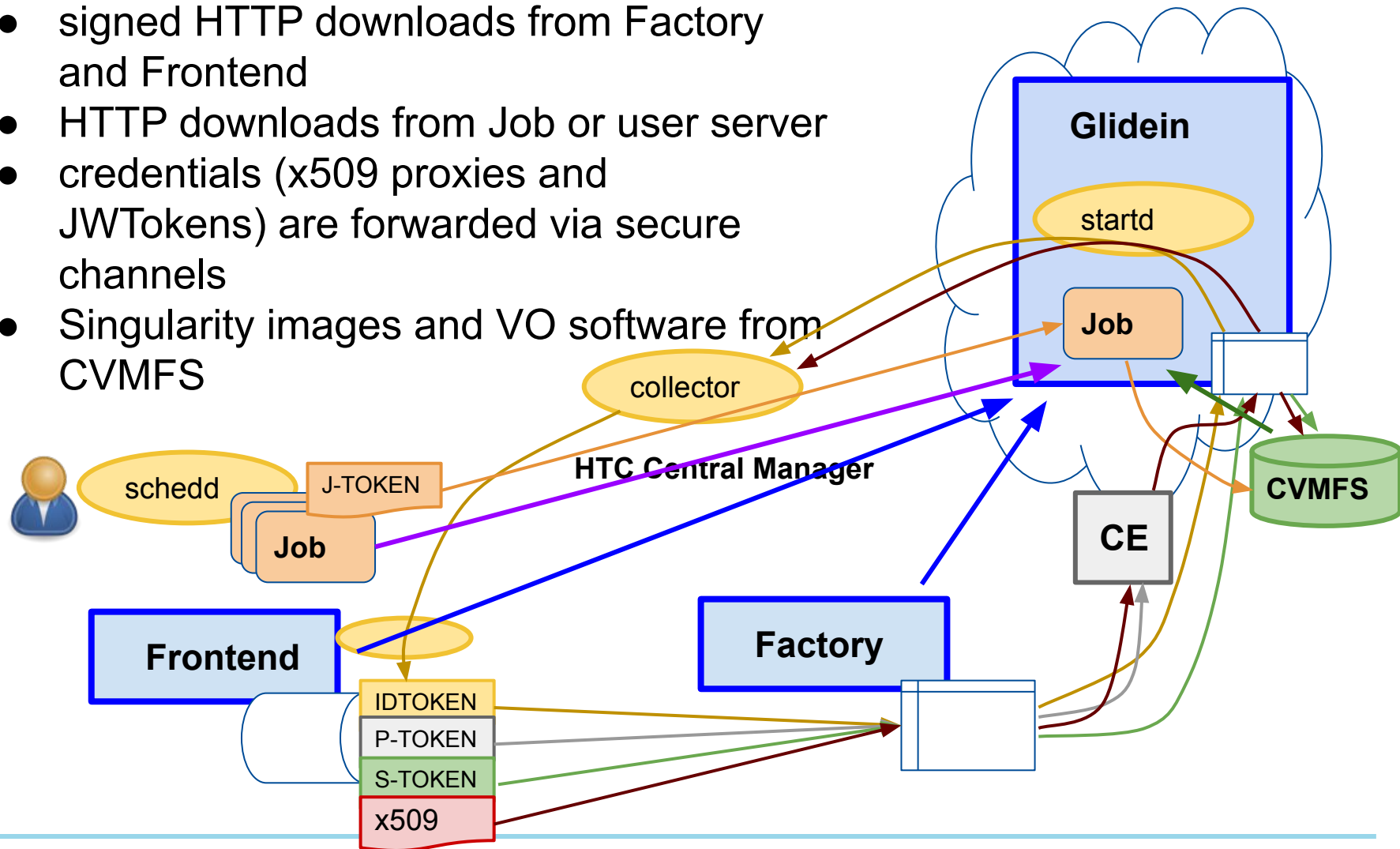
# Glidein downloads

- signed HTTP downloads from Factory and Frontend
- HTTP downloads from Job or user server
- credentials (x509 proxies and JWTokens) are forwarded via secure channels



# Glidein downloads

- signed HTTP downloads from Factory and Frontend
- HTTP downloads from Job or user server
- credentials (x509 proxies and JWTokens) are forwarded via secure channels
- Singularity images and VO software from CVMFS



## cvmfsexec use (current\*)

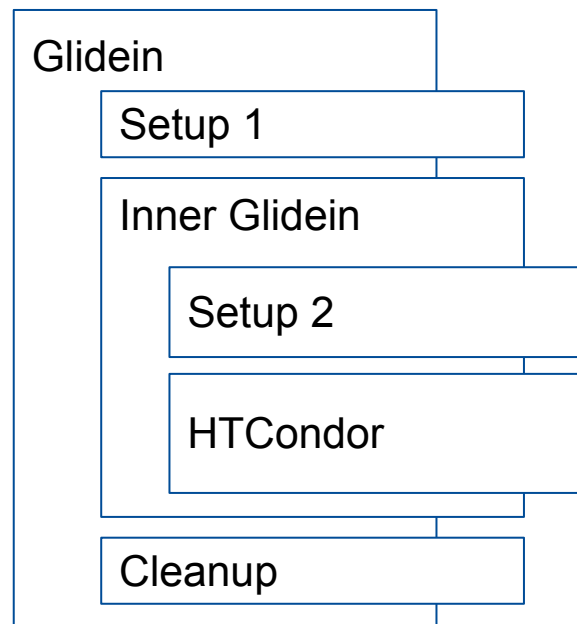
- During setup
  - check OS
  - check support for unprivileged user namespaces
  - check FUSE (packages installed, user in fuse group)
  - download cvmfsexec package (both OSG and EGI distributions)
  - if desired use cvmfsmount (with the configuration selected)
    - mounted on /cvmfs if possible
- During Singularity startup (job wrapper)
  - bind mount the mount directory to /cvmfs
- During cleanup
  - use cvmfsumount if there are mounted file systems

## cvmfsexec use (planned)

- During initial setup
  - check OS
  - check support for unprivileged user namespaces
  - check FUSE (packages installed, user in fuse group)
  - download cvmfsexec distribution if needed (only the configuration desired)
  - decide best option between cvmfsexec and cvmfsmount (given the OS and configuration)
    - mounted on /cvmfs if possible
- During re-invocation (if needed)
  - use cvmfsexec
- During Singularity startup (job wrapper)
  - bind mount the mount directory to /cvmfs
- During cleanup
  - use cvmfsumount if needed

# Glidein structure (planned)

- Spawns multiple glideins if needed (MPI, multi-glidein, ...)
- Initial checks
- Download of scripts and setup
  - includes CVMFS mounting and Singularity testing
- **Exec the inner part of the Glidein**
- Complete setup
- Start HTCondor startd and join the pool
  - Start Singularity for each job individually (via wrapper)
- Cleanup
  - included modular scripts



# Acknowledgements and Credits

This work was done under the GlideinWMS project

Thank you to Namratha Urs for most of the development

Thank you to Dave Dykstra for all the support

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

## References

<https://github.com/glideinWMS/glideinwms>

# Summary

- Glideins are pilot jobs that allow setup and cleanup for each experiment job
- Currently (next production version) support `cvmfsmount/umount`
  - Thanks to Singularity CVMFS always available in `/cvmfs` for the experiments
- To improve reliability will support all `cvmfsexec` modes