Transitioning GlideinWMS, a multi domain distributed workload manager, from GSI proxies to tokens and other granular credentials

Marco Mambelli, Bruno Coimbra, Dennis Box

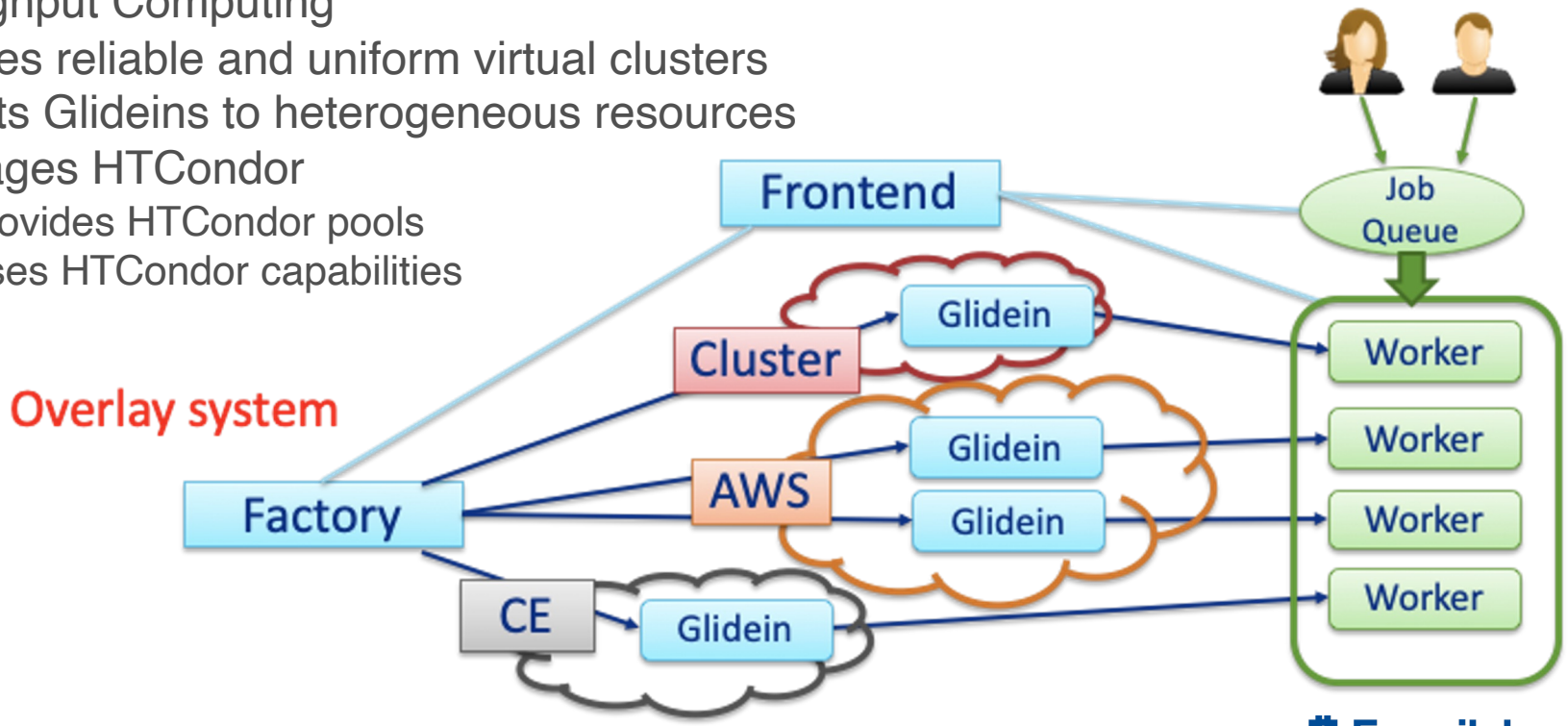
CHEP 2023

9 May 2023

This work was supported by the Fermi National Accelerator Laboratory, managed and operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy

GlideinWMS

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



Glidein: pilot job for node testing and customization

- Scouts for resources and validates the Worker node
  - Cores, memory, disk, GPU, OS, software installed, CVMFS, …
- Customizes the Worker node
  - Environment, GPU libraries, Starting containers (Apptainer, …)
- Provides monitoring and audit
- Runs one or more jobs in parallel or sequentially via HTCondor
- **Stores and uses Pilot credentials (e.g. VO Group)**
- **Safely receives and stores Job credentials**

(Glidein) Factory

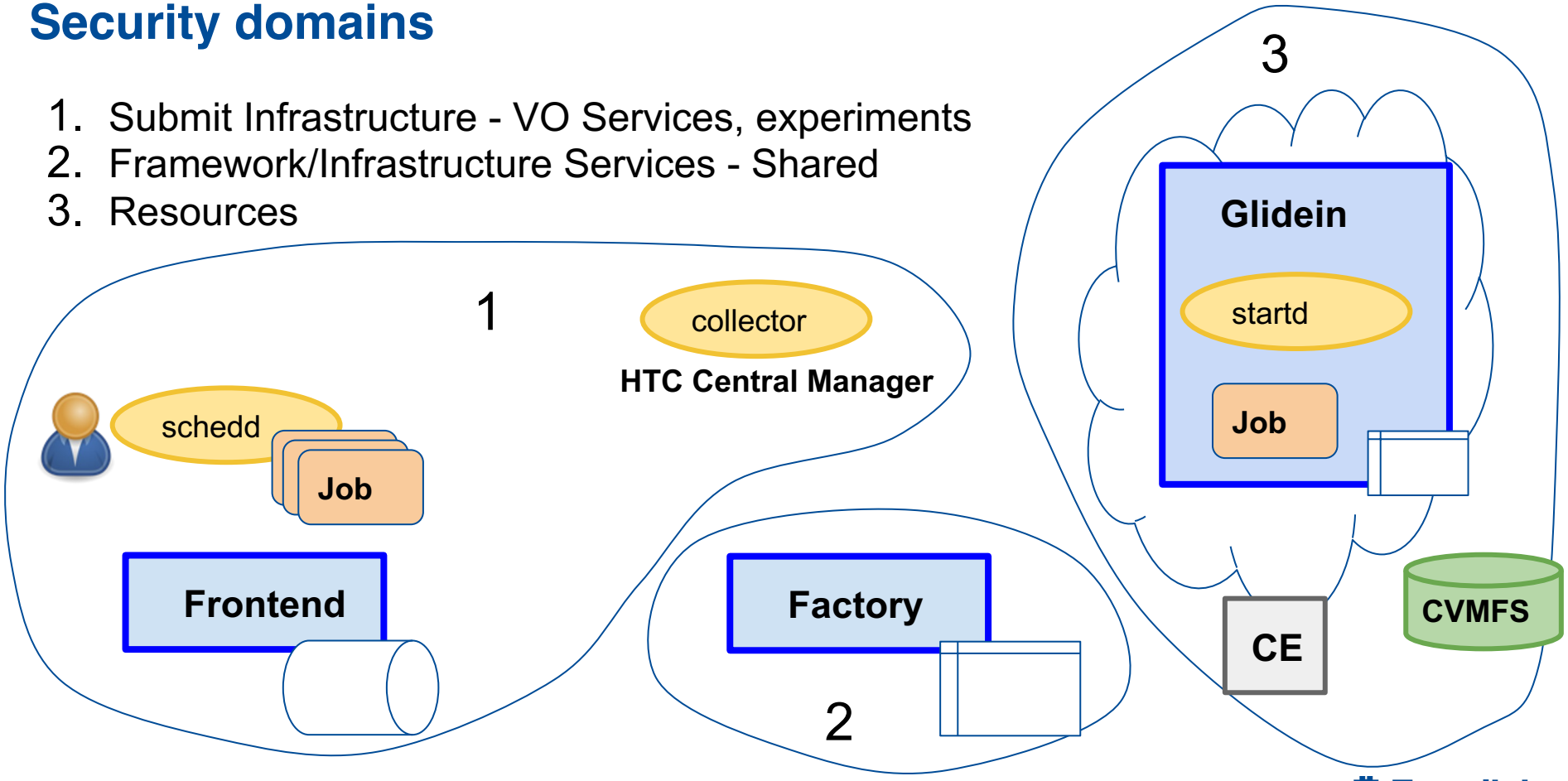
- Knows how to submit to sites
  - Sites are described in a configuration (curated, in VCS, or auto-generated)
    - **Authentication method**, **supported VOs,** expected resources, ..
  - Only trusted and tested sites are included in production
- Condor does the heavy lifting of submissions.
- **Keeps a cache of credentials used or forwarded to Glideins**

(VO) Frontend

- Pressure-based system controlling the Factory Glideins requests
  - Monitors job requests and available entries (sites)
  - Works keeping a certain number of Glideins running or idle at the sites
  - Limits Glideins requests to enforce policies, avoid spikes and overloads
- **Manages credentials and delegates them to the Factory and Glidein**
  - **Stores and owns auto-generated or VO-managed credentials**
  - **Has some long-term credentials, forwarding short term ones**
  - **Manual input, interacts with IAMs**

Security domains

1. Submit Infrastructure - VO Services, experiments
2. Framework/Infrastructure Services - Shared
3. Resources



Security and Credentials in GlideinWMS

Access resources, provide secure infrastructure, and user job needs

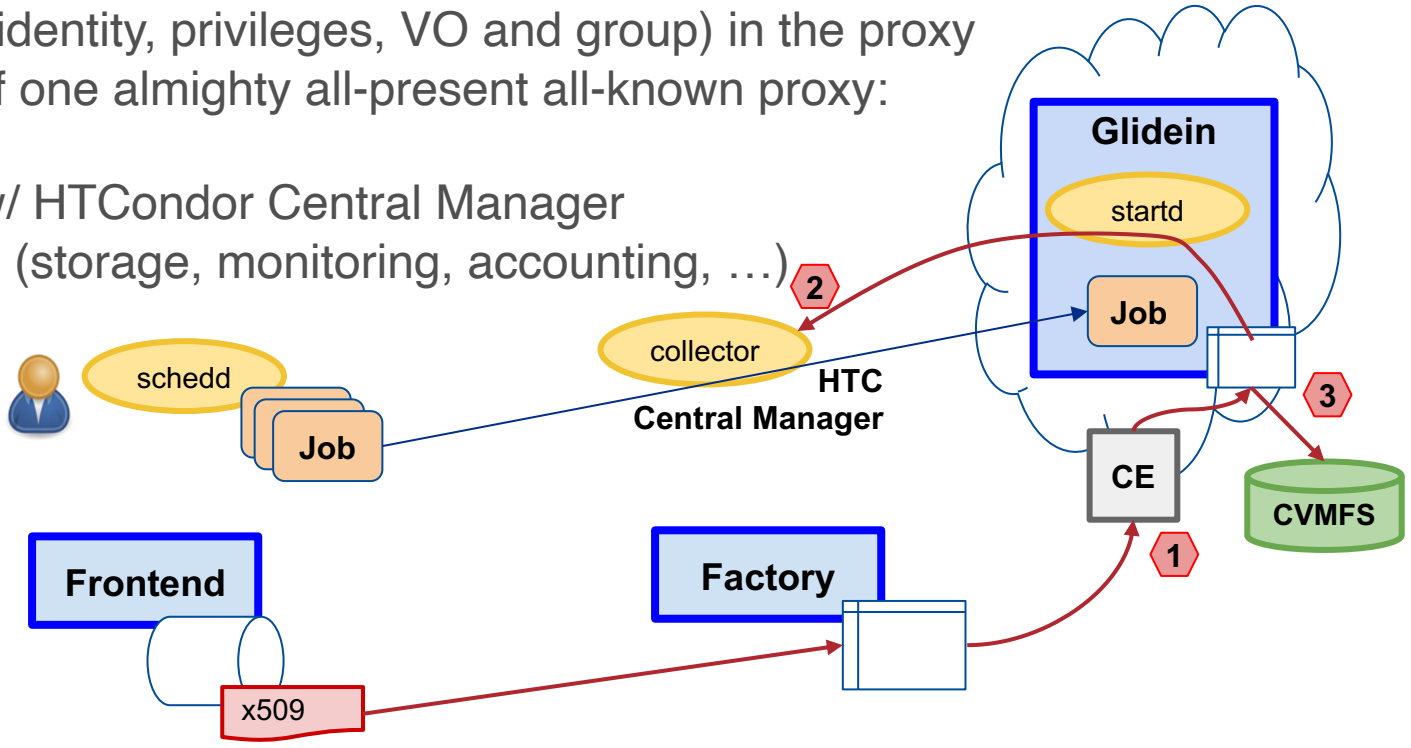
- IDTOKENS and HTCondor security (was GSI) Internally
- Credentials for different functions
  - Pilot submission
  - Pilot infrastructure operation (virtual cluster security, monitoring, …)
  - Pilot VO services
  - User job operation
- Credentials of different types, both identity or capability based
  - Tokens, GSI, SSH keys, …
- Agnostic about the credential type
  - Provider and service must be compatible
- Hierarchic groups of credentials from users

Traditional X509 authentication - Pilot proxy

Credentials at Frontend, proxy cached at Factory and Glidein
All the information (identity, privileges, VO and group) in the proxy
Multiple functions of one almighty all-present all-known proxy:

1. Pilot submission
2. Authentication w/ HTCondor Central Manager
3. Services access (storage, monitoring, accounting, …)

…

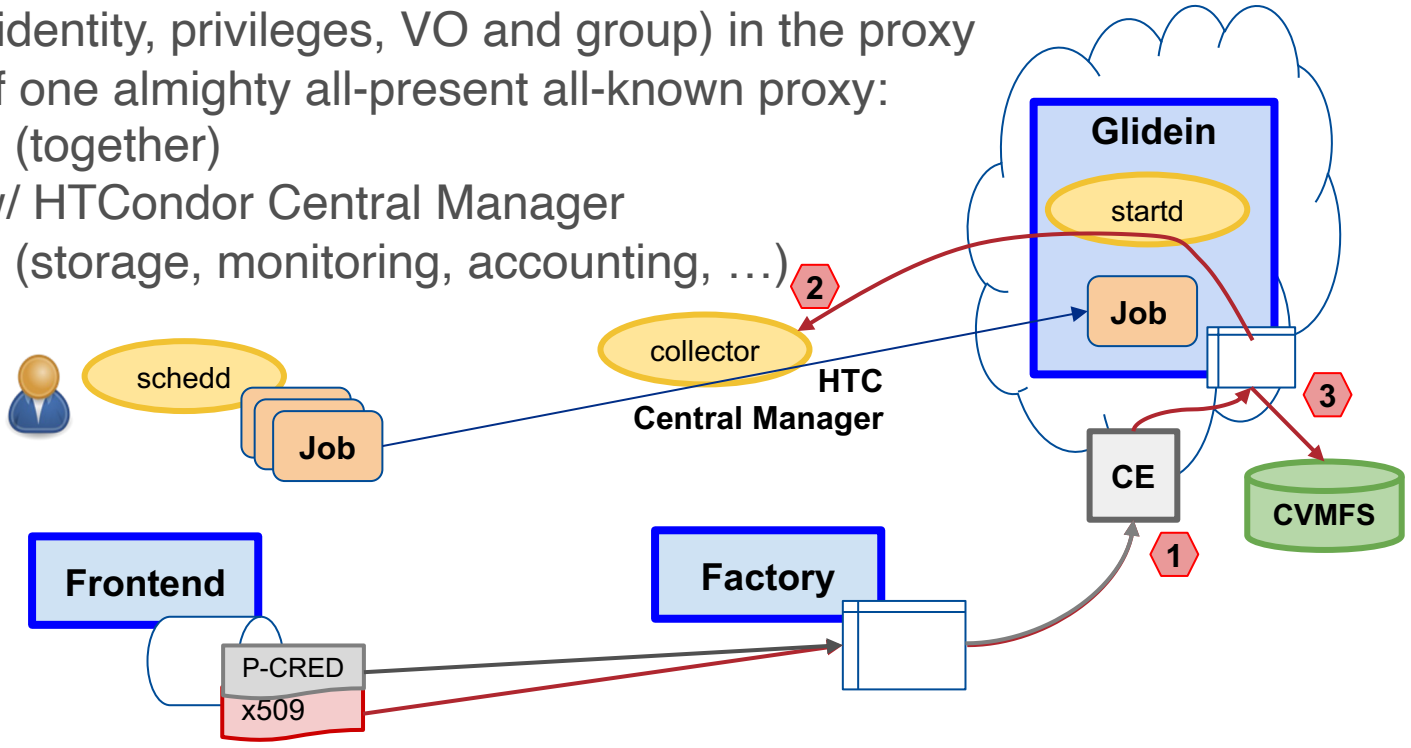


Traditional X509 authentication - Pilot proxy

Credentials at Frontend, proxy cached at Factory and Glidein
All the information (identity, privileges, VO and group) in the proxy
Multiple functions of one almighty all-present all-known proxy:

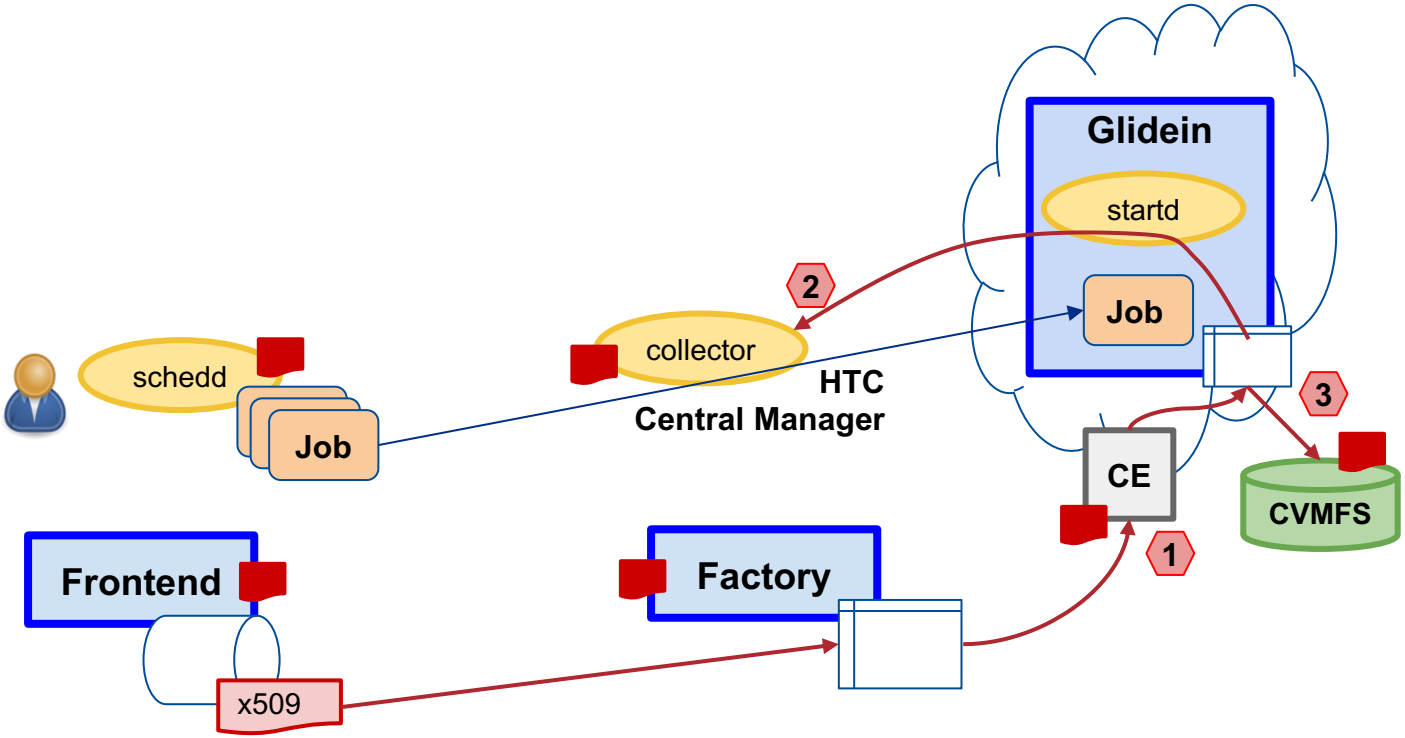
1. Pilot submission (together)
2. Authentication w/ HTCondor Central Manager
3. Services access (storage, monitoring, accounting, …)

…



Traditional X.509 authentication - Host certificates

X.509 host certificates used to identify the servers



Fermilab

From X.509 proxy to JWT

Benefits [#139]

- Industry Standards
- Ease of Use
- Flexible Authentication
- Finer Grained Access Control (in time and space)
- Security Benefits

From X.509 proxy to JWT

Benefits [#139]

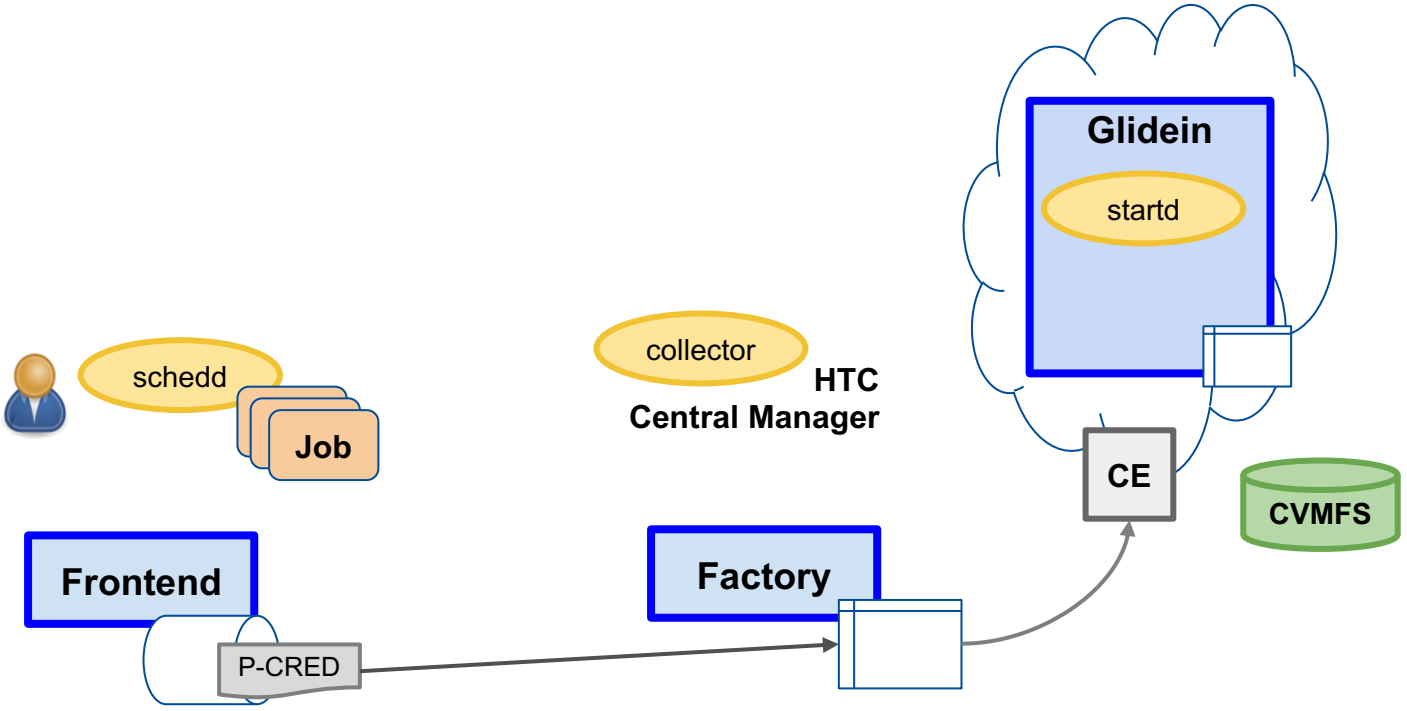
- Industry Standards
- Ease of Use
- Flexible Authentication
- Finer Grained Access Control (in time and space)
- Security Benefits

Caveats

- Replace existing mechanisms
- Rethink credentials: number and scope
- Number and differentiation cause complexity
- Dynamic generation to accommodate granularity
- New refresh mechanisms
- Transition and hybrid systems

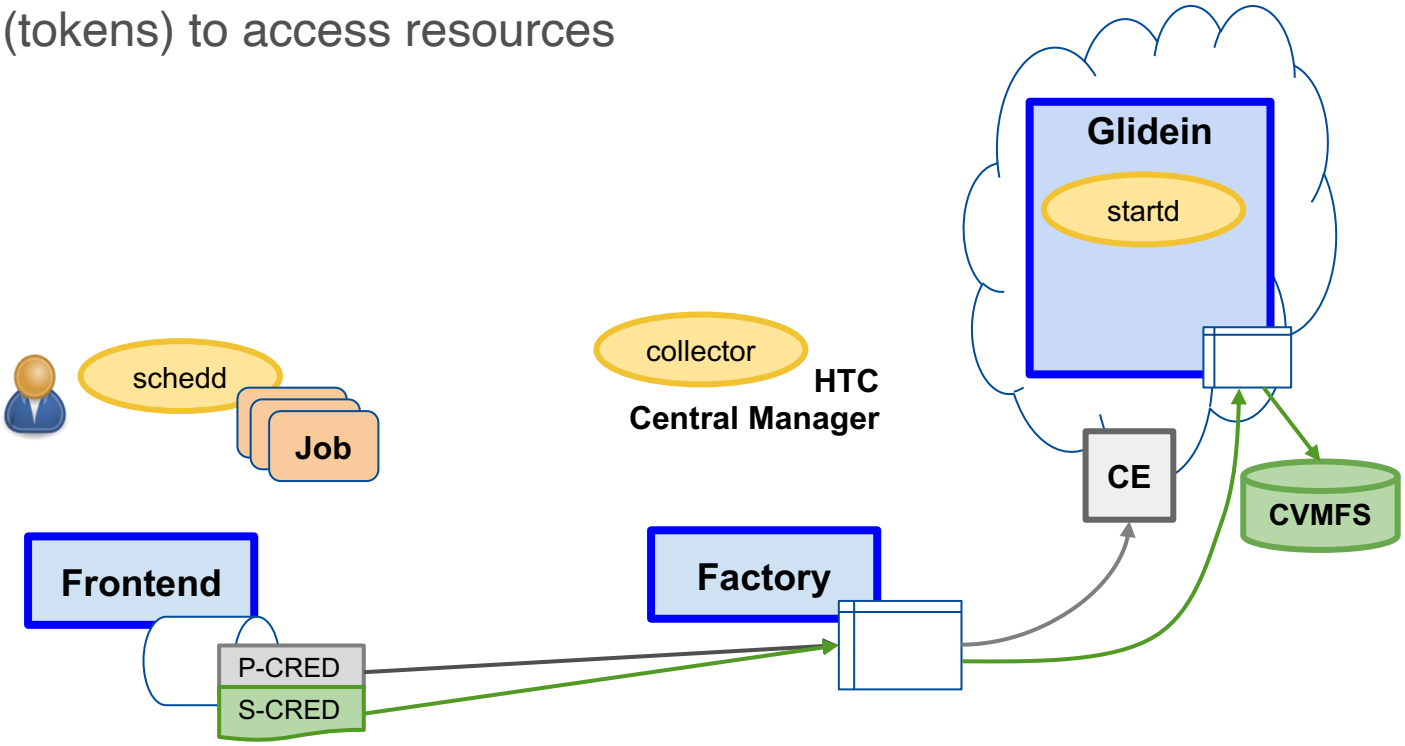
Token authentications

1. Pilot submission credential (SSH key, SciToken/ WLCG token)



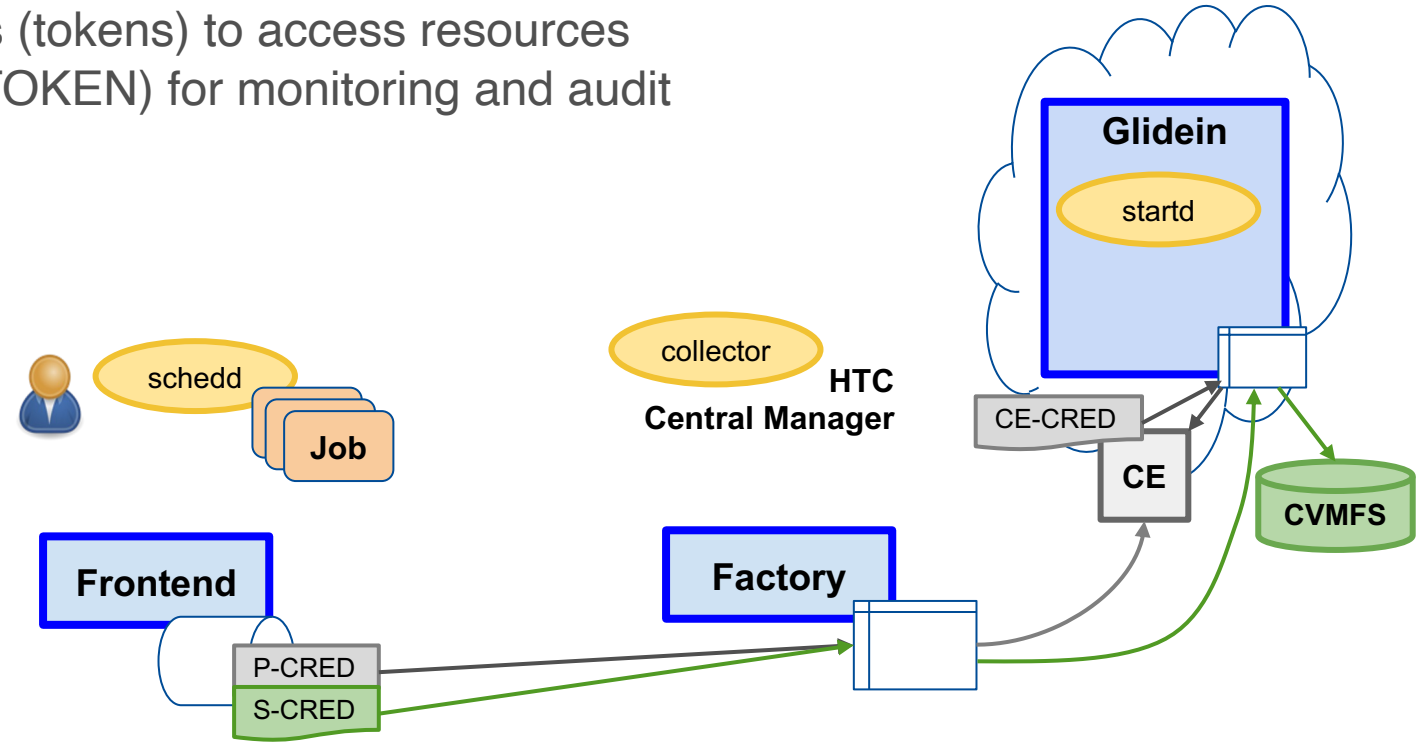
Token authentications

1. Pilot submission credential (SSH key, SciToken/ WLCG token)
2. VO credentials (tokens) to access resources



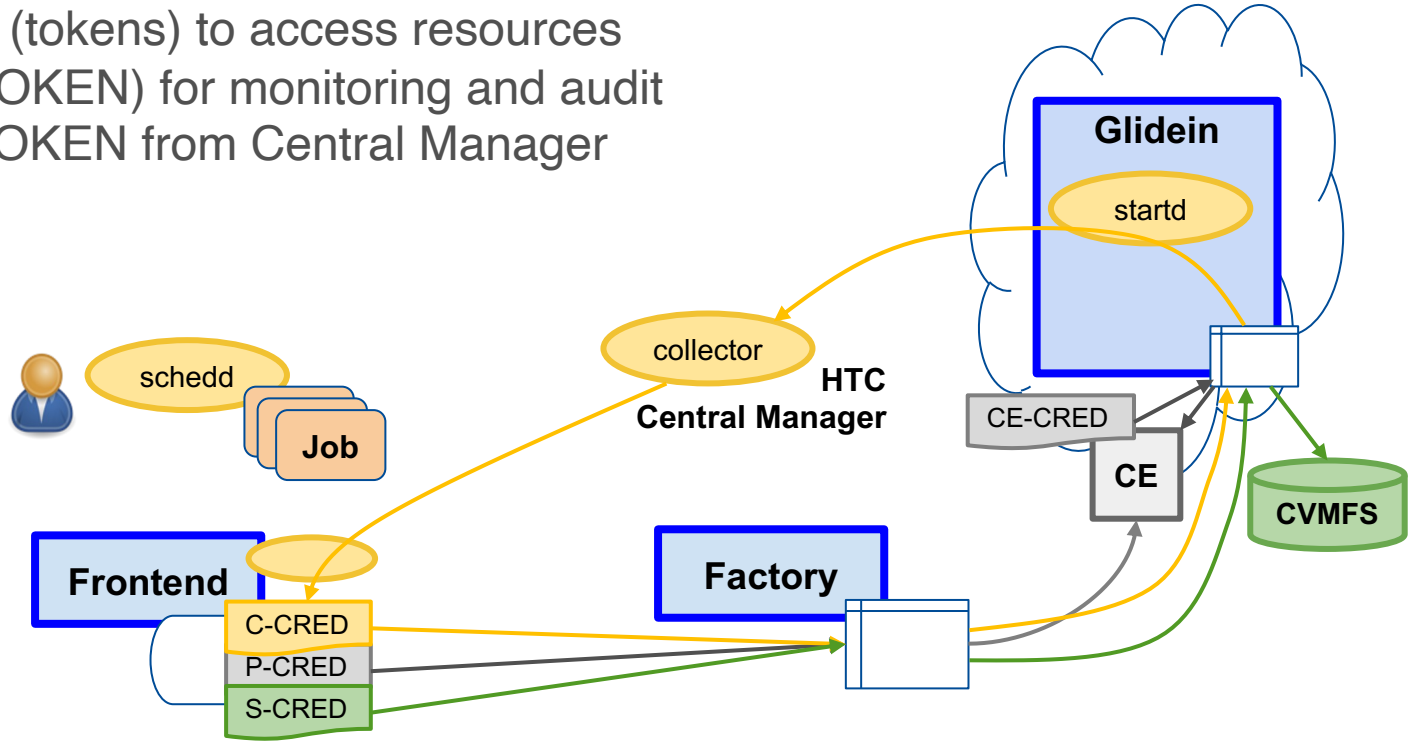
Token authentications

1. Pilot submission credential (SSH key, SciToken/ WLCG token)
2. VO credentials (tokens) to access resources
3. CE Token (IDTOKEN) for monitoring and audit



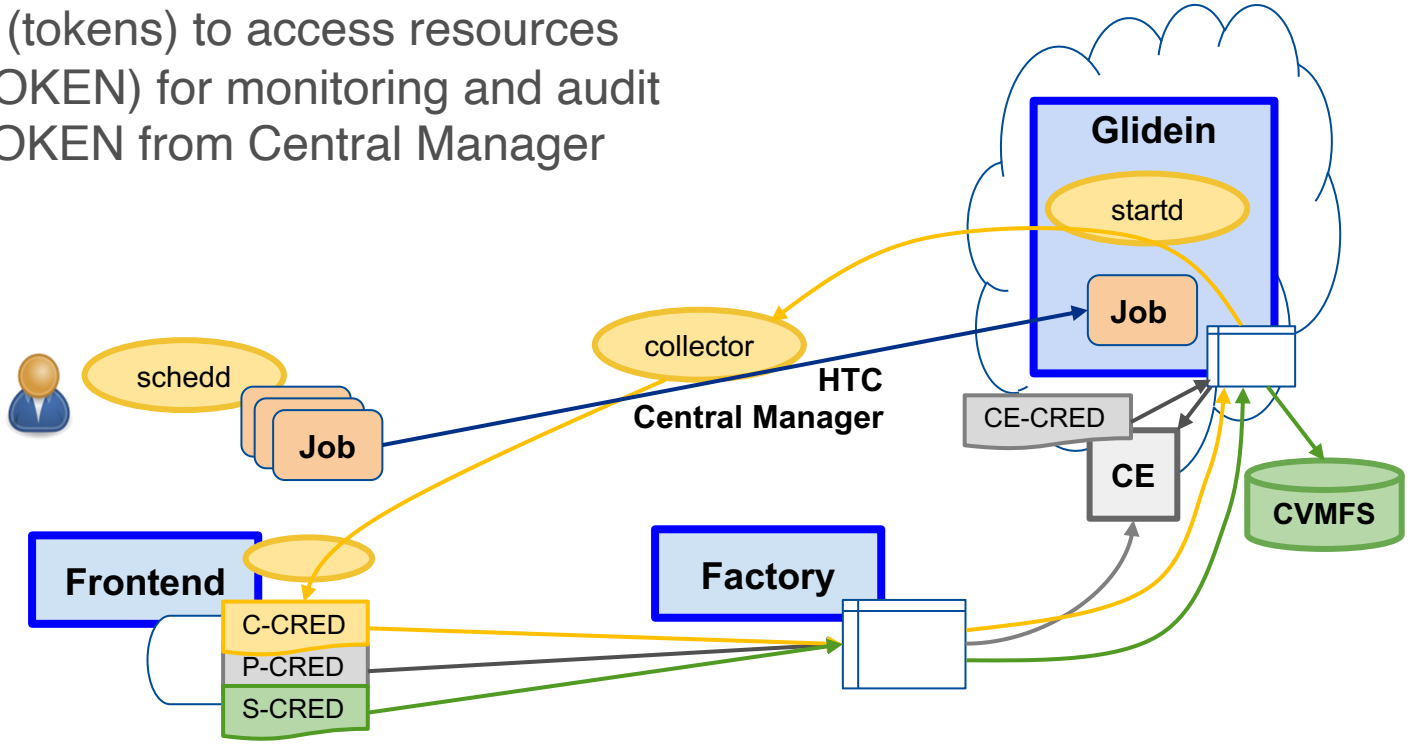
Token authentications

1. Pilot submission credential (SSH key, SciToken/ WLCG token)
2. VO credentials (tokens) to access resources
3. CE Token (IDTOKEN) for monitoring and audit
4. HTCondor IDTOKEN from Central Manager



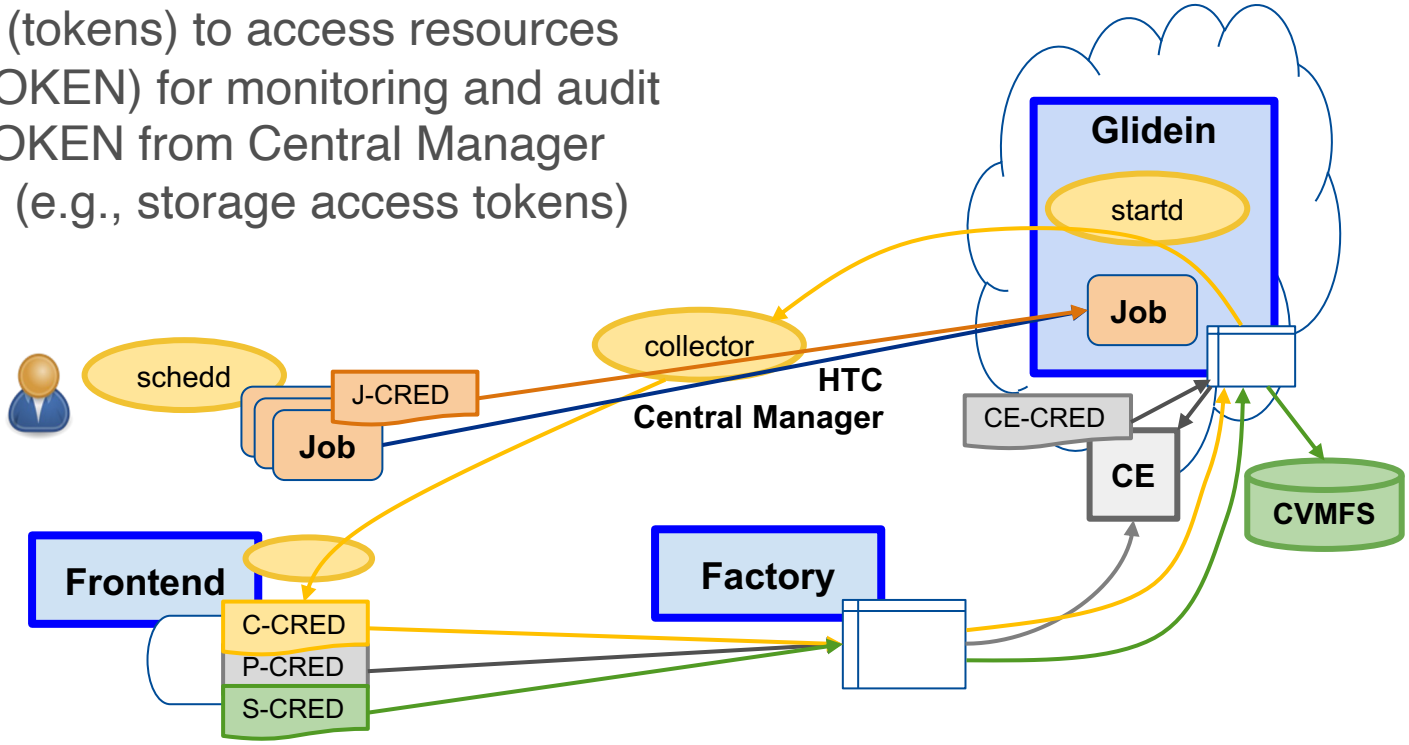
Token authentications

1. Pilot submission credential (SSH key, SciToken/ WLCG token)
2. VO credentials (tokens) to access resources
3. CE Token (IDTOKEN) for monitoring and audit
4. HTCondor IDTOKEN from Central Manager



Token authentications

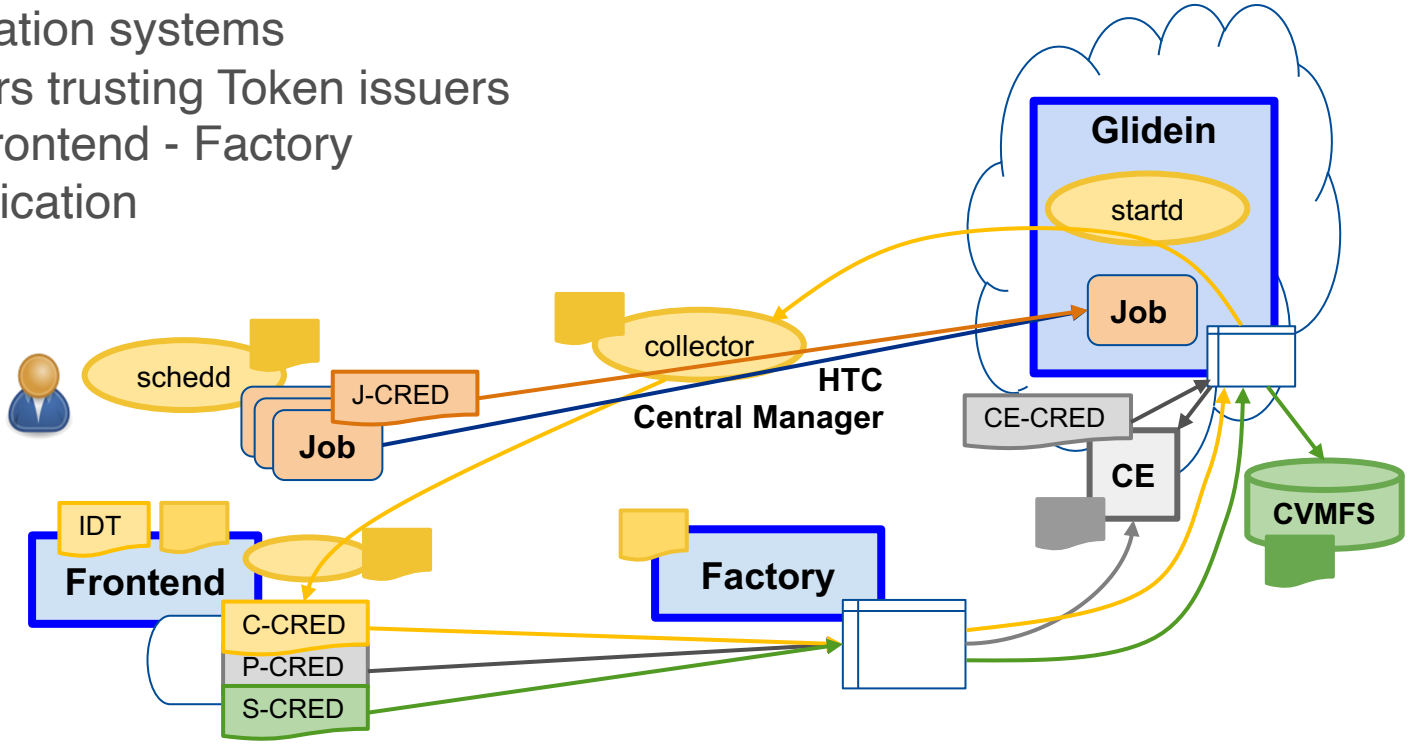
1. Pilot submission credential (SSH key, SciToken/ WLCG token)
2. VO credentials (tokens) to access resources
3. CE Token (IDTOKEN) for monitoring and audit
4. HTCondor IDTOKEN from Central Manager
5. Job credentials (e.g., storage access tokens)



Token authentications - Hosts

Host certificates used to bootstrap TLS (DN not important)
Different authentication systems

- Service providers trusting Token issuers
- IDTOKEN for Frontend - Factory
- HTCSS authentication



Token Support Milestones

- 2019/05 - Use of tokens (security without x509 certificates) in roadmap
- 2019/09 – Use of token authentication becomes high priority, dedicated developer
  Collaboration w/ HTCondor and OSG: Use token-auth to authenticate Glideins, support sites with sci-token, tokens to authenticate Factories w/ Frontends
- 2019/10/22 – **Proof of concept** (GlideinWMS using tokens and no proxy)
- 2020/03/31 - v3.7 Use of HTCondor token-auth for Glideins authentication
- 2020/11/03 – v3.7.1 SciTokens authentication with sites, IDTOKENS authentication between Factory and Frontend (Using tokens but with GSI)
- 2021/03/25 – v3.7.3 GlideinWMS configured without GSI but with SciToken and IDTOKENS successfully run jobs
- 2021/09/02 – v3.7.5 Fix IDTOKEN generation, SciToken credential per-entry, **CMS and OSG production**
- 2022 – v3.9.x to 3.10.1 - More automation, configurability, better handling of transition, credential generator plugins, token support for Grid/Batch universe, GCE, and AWS, support HTCondor-CE collector
- 2023/07 - 3.11.2 Expected refactored credentials handling

Surprises

Early adopters

- HTCondor pre-release, 8.9, 9.0, 9.x, 10.0, 10.x
  - TOKEN auth, IDTOKEN
  - Changes of defaults and security model
  - Undocumented features (e.g. Job router generating IDTOKENS)
- Multiple feature requests changes
  - per-site credentials
  - hybrid support

GSI and Proxies engrained from years of use

- in the GlideinWMS code
  - Name of credential-related classes/functions Proxy…
  - Verifications
- in the systems
  - CE monitoring
  - Audit Log
  - Accounting
  - Renewal of tokens not supported

Transition challenges

- Factory configuration (auth_method)

Credentials refactoring

Reasons

- Be more flexible to make the adoption of new credential types easier in the future
- Need to handle multiple credential types in entries and groups
- The ability of specifying fallback credentials would ease Factory and Frontend operations
- Our code is starting to have too many conditions to treat special authentication scenarios

Benefits

- Support for lists of credential sets
- More orthogonal and consistent credential model
- Improve the way “fair split” handles entries with multiple credentials of different types
- Ease adoption of new credential types in the future
- Allows code reuse by other GWMS Factory clients such as the HEPCloud DE

Conclusions

- GlideinWMS token and hybrid support in production since end of 2021
- Early adoption has its complications
- Token migration is not a drop-in replacement
  - Time granularity
  - Space granularity
  - Multiple credentials
  - Dynamic/Late credentials generation/request
- Configurable duration and granularity will help in transition
- Code refactoring is paying off

Acknowledgements

This work was done under the GlideinWMS project

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