Fermilab's Transition to Token Authentication

Presented by Nick Smith
on behalf of the transition team

CHEP 2024

22 October 2024

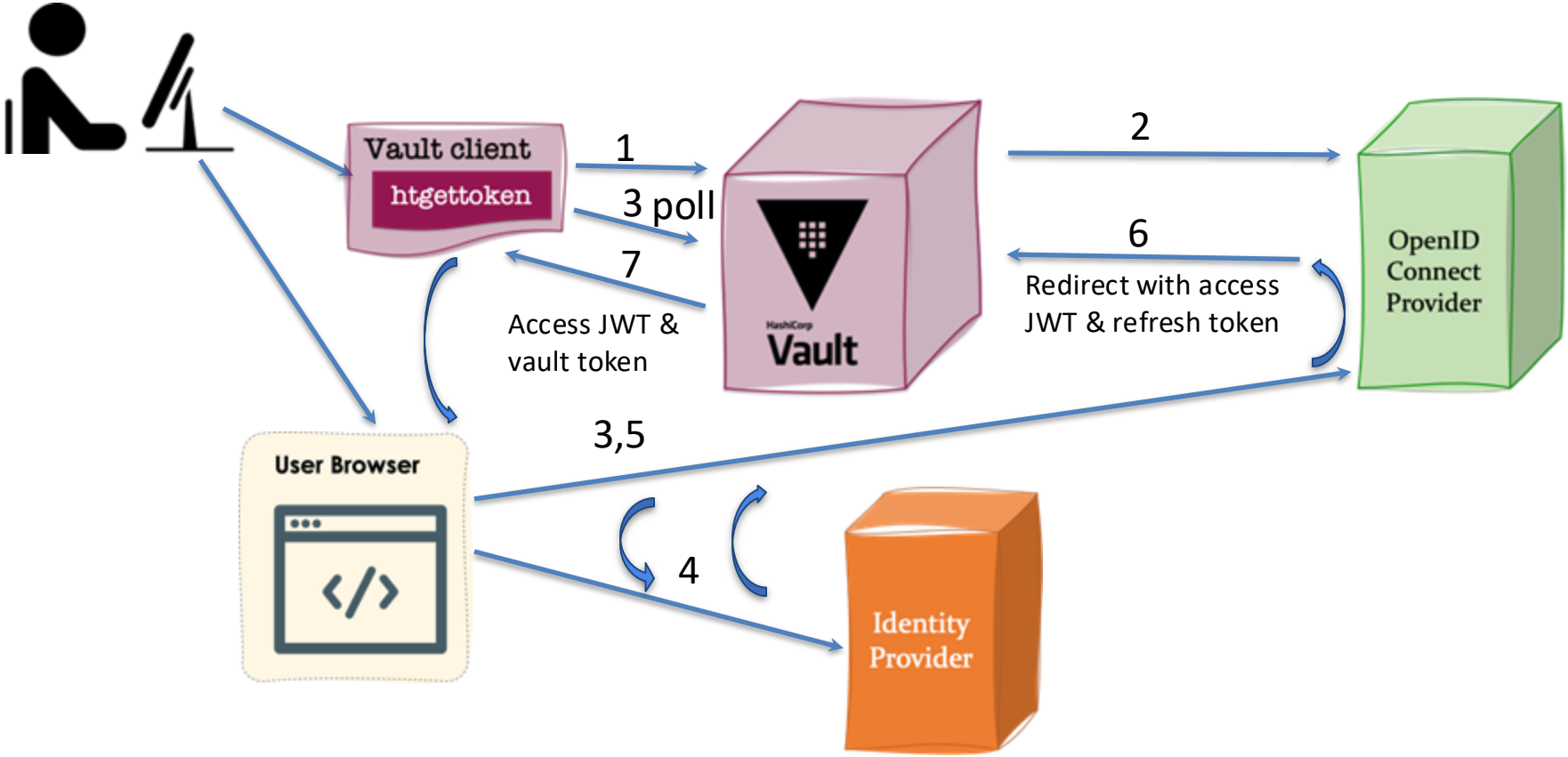
Introduction

- The X.509 proxy certificate user credentials for grid authentication never came into common use in industry, so they needed to be replaced
- All active experiments hosted at Fermilab have been sending JSON Web Tokens (JWTs) with all grid jobs for over a year
  - Tokens adhere to the WLCG Common JWT Profile
  - Tokens are issued by CILogon
- Most components of the grid infrastructure software at Fermilab have either been updated or replaced to support tokens
  - Most are available as open source for others to use when they face similar requirements
  - Protocols follow open standards wherever possible
    - Communication with CILogon uses OIDC and Oauth2, adapted to primarily command line
- Overall security goals: strongly protect long-lived credentials, and minimize power of credentials that are widely exposed
- Ease of use goal: minimize the need for end users to manage tokens

Core of the system: Vault with htgettoken

- Hashicorp Vault (or in future: fork OpenBao)
  - Popular general purpose secure secret store
  - Comes with Open ID Connect (OIDC) and Kerberos plugins
  - Integrates well with both Indigo IAM and CILogon OIDC Providers
  - Manages access with its own tokens
  - We use it to store high-value long-lived refresh tokens for many users to limit security vulnerability
  - Configured with package **htvault-config** so the combination is sometimes called **HTVault**
    - Adds modified OIDC plugin, oauth secrets plugin, and ssh-agent authentication plugin
  - Natively supports 3 servers for High Availability
- **htgettoken**
  - Command line client to automate the HTVault flows
  - Initially authenticates via OIDC & a web browser
  - Long life (~1 month, renewable) refresh tokens stay in Vault, and limited life (1 week) Vault tokens and even shorter life (3 hours) access JWTs stored unencrypted in local files
  - Follows WLCG Bearer Token discovery standard for local filename
  - Uses Vault tokens to get access tokens and renews Vault access with Kerberos (can also renew with ssh-agent)
  - Comes with helper tool **httokensh** which renews access tokens when needed for interactive use, and **htdecodetoken** and **htdestroytoken**
- Other organizations are also using these either in production or development (LIGO, JLAB, BNL, CMS)

htgettoken with Vault initial OIDC flow



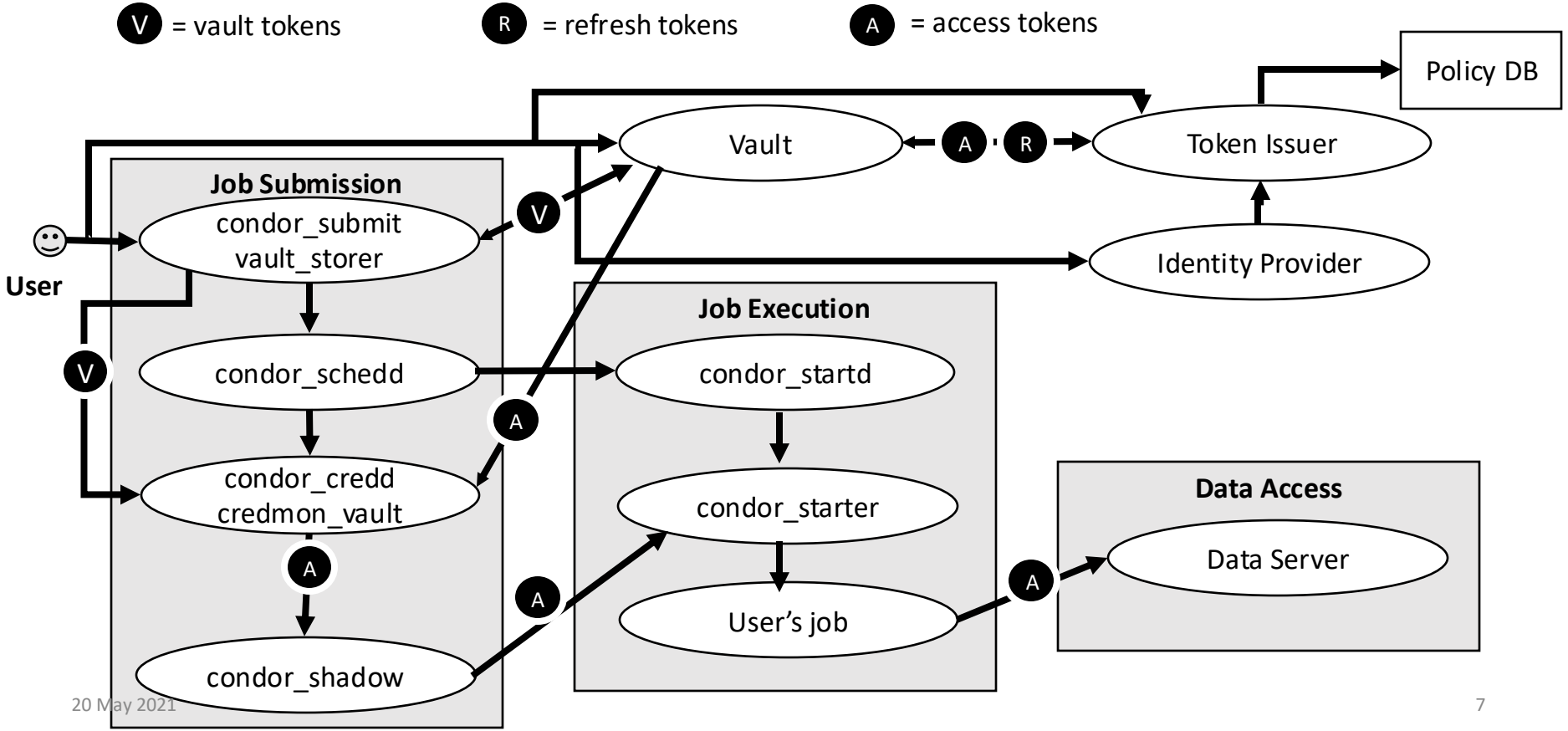
CILogon & FERRY

- CILogon provides token issuing service
  - Five larger collaboration experiments have their own issuer URL, but most share https://cilogon.org/fermilab
  - CILogon also issues tokens for many other U.S.-based customers and is very flexible in the way they do it
- A Fermilab-specific configuration system FERRY (Frontier Experiments RegistRY) contains all the information about experiment membership, roles, and token privileges
  - Writes relevant information to a CILogon-hosted database server (currently LDAP) so issuer knows who is allowed to get tokens and the scopes in those tokens (a.k.a *capability sets*)
  - Another Fermilab-specific script **htvault-gen** reads experiments and roles from FERRY to auto-generate almost all the configuration for htvault-config

Jobsub_lite, GlideinWMS, and HTCondor

- **Jobsub_lite** is our lightweight wrapper around HTCondor for a unified job submission interface
  - Open source but so far Fermilab is the only user
  - Designed for use with tokens and with multiple HTCondor schedds on separate servers
    - Also supports X.509 proxy certificates for now
  - Submits jobs to batch cluster of both local nodes and remote resources added by **GlideinWMS**
    - GlideinWMS was updated to use WLCG-profile access tokens to submit pilot jobs to site Computing Elements
- The **condor-credmon-vault** subpackage of HTCondor client & server integrates with htgettoken & HTVault
  - Supplies **condor_vault_storer** as a plugin to condor_submit and **condor_credmon_vault** as a plugin to credd
  - condor_vault_storer obtains and sends a 4-week vault token to condor_credmon_vault
  - condor_credmon_vault regularly obtains new access tokens and then credd pushes them to batch jobs via the condor shadow process

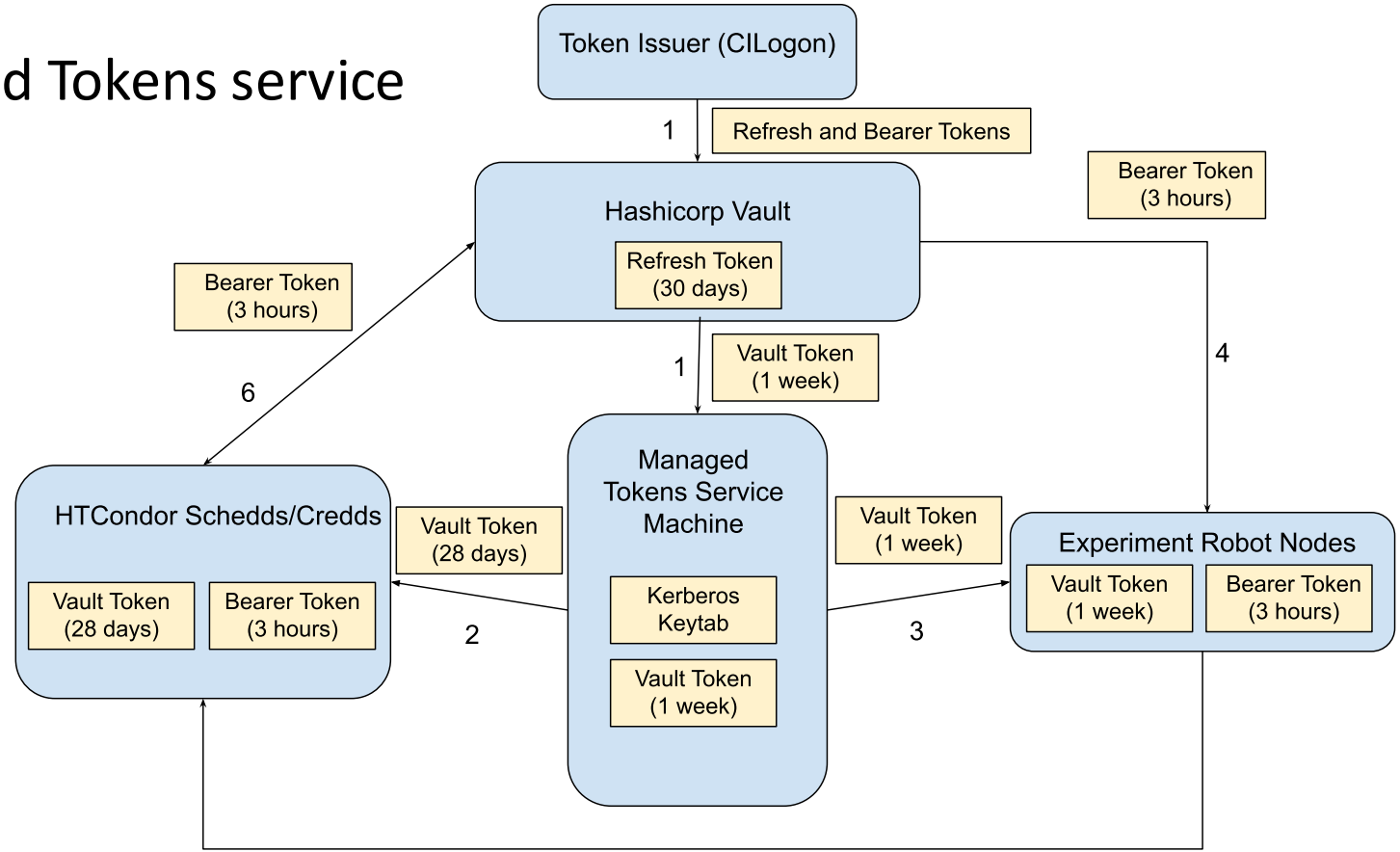
Token flow with HTCondor and Vault



Managed Tokens service

- htgettoken and HTVault were designed primarily with the interactive user use case in mind, but there are also many unattended “robot” operations needed
- These require long-lived credentials, which should be protected on a secured “bastion” host
- The **Managed Tokens** service handles this for all the experiments’ robot operations
  - Initial OIDC authentication handled by the operator
  - Maintains Kerberos keytabs for each credential needed
  - Pushes vault tokens to the robot machines
  - Pushes vault tokens to all HTCondor credds by using condor_vault_storer
  - Code is open source, intended for any user of HTVault with HTCondor
  - See poster 489 for more details

Managed Tokens service



Other updated components

- We use **dCache** to store experiment data
  - dCache was a pioneer in support of WLCG tokens, we just enabled it
  - Handling of "sub-VO"s under the fermilab VO required non-trivial configuration
  - Used directory ownership inheritance
- **ifdh** is Fermilab's internal "Intensity Frontier Data Handler"
  - A command line front end for several other tools for moving data around
  - Updated to locate tokens and send them along to the backend tools
- **POMS** is Fermilab's "Production Operation Management Service"
  - Web & command-line based system for managing large experiment job campaigns such as Monte Carlo production
  - Updated to handle tokens in similar ways as the Managed Tokens service
- RCDS is the "Rapid Code Distribution System"
  - Fermilab installation of open source **cvmfs-userpub** which publishes individual user analysis code on demand into CVMFS
  - Updated to accept JWTs with *compute.create* scope for user authentication from any configured acceptable token issuer

Concluding remarks

- System working reliably after fixing early issues
- Getting credentials almost as hidden from users as they can be
  - Users with Kerberos only need to approve on web browser once
- Long-lived credentials stored only on minimal, secured hosts
- Configuration is managed by server operators, nothing for end users
  - Reducing scopes for best security managed by per-experiment experts
- JWTs are better supported and more secure than X.509 proxies
  - Can be much more purpose-specific
- X.509 proxy certs are still sent with most jobs in case needed, but some experiments have converted to use only tokens

Links

- WLCG JWT profile
  - https://github.com/WLCG-AuthZ-WG/common-jwt-profile
- WLCG Bearer token discovery:
  - https://github.com/WLCG-AuthZ-WG/bearer-token-discovery
- CILogon: https://cilogon.org
- htvault-config: https://github.com/fermitools/htvault-config
- htgettoken: https://github.com/fermitools/htgettoken
- HTcondor: https://htcondor.readthedocs.io
- Jobsub_lite: https://github.com/fermitools/jobsub_lite
- GlideinWMS: https://github.com/glideinwms/glideinwms
- Managed Tokens: https://github.com/fermitools/managed-tokens
- dCache: https://dcache.org
- cvmfs-userpub: https://github.com/cvmfs-contrib/cvmfs-userpub