Fermilab's experience transitioning to token-based AAI technologies

TAGPMA 2020 Workshop on Token-Based Authentication and Authorization

1 Dec 2020

Jeny Teheran, Dave Dykstra

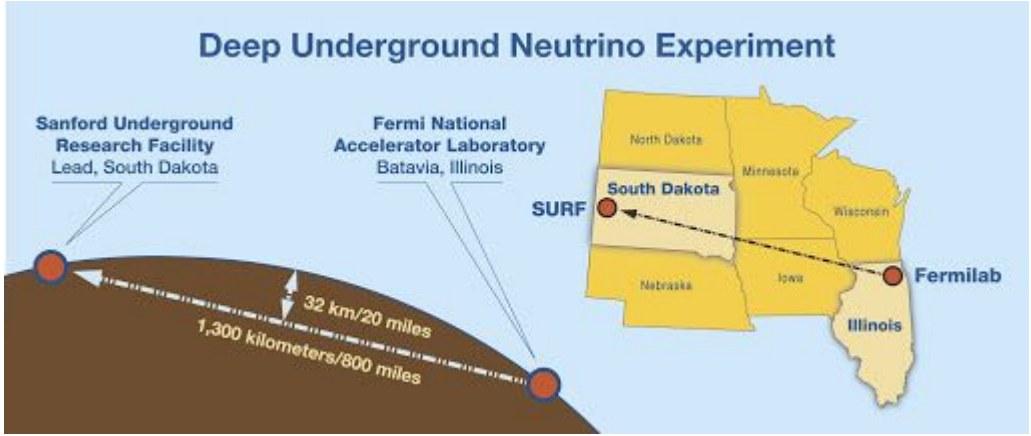
Outline

- Our current AAI architecture
- Updates on our new federated identities architecture
- Progress of our work: accomplishments and current status
- Challenges and concerns
- Q&A

Motivation

As Fermilab becomes the host laboratory for international collaborations like DUNE, it is our goal to provide transparent access to computing resources for all of our scientific user community across organizational and national boundaries.

DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT



Fermilab

Motivation

- Reduce complexity on the authentication infrastructure by moving away from X.509 user certificates.
  - X.509 certificates also add complexity to the development of scientific tools and the operation of scientific services.
- Guarantee interoperability for international collaborations moving towards token-based AAI, such as CMS.
- Align our AAI infrastructure with current technologies.

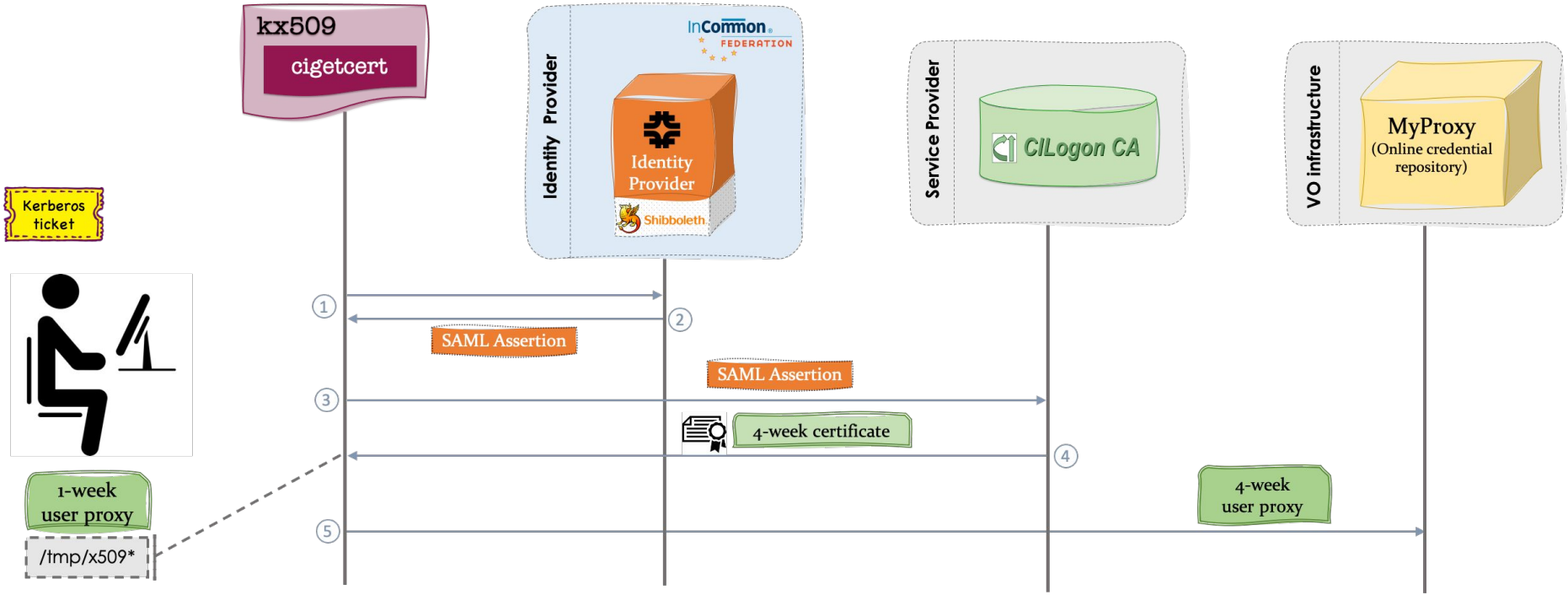
Background 1/2

- Our environment for scientific services and tools is command-line oriented.
- Users are required to authenticate using Kerberos before accessing general computing nodes.
- Users are required to authenticate using a valid IGTF-accredited X.509 certificate before accessing scientific computing resources.
- The `kx509` tool is used to translate Kerberos tickets into X.509 certificates.
  - Initially it used our own CA: FNAL Kerberos CA
- In 2016, the FNAL Kerberos CA was retired and we migrated to CILogon Basic CA.

Background 2/2

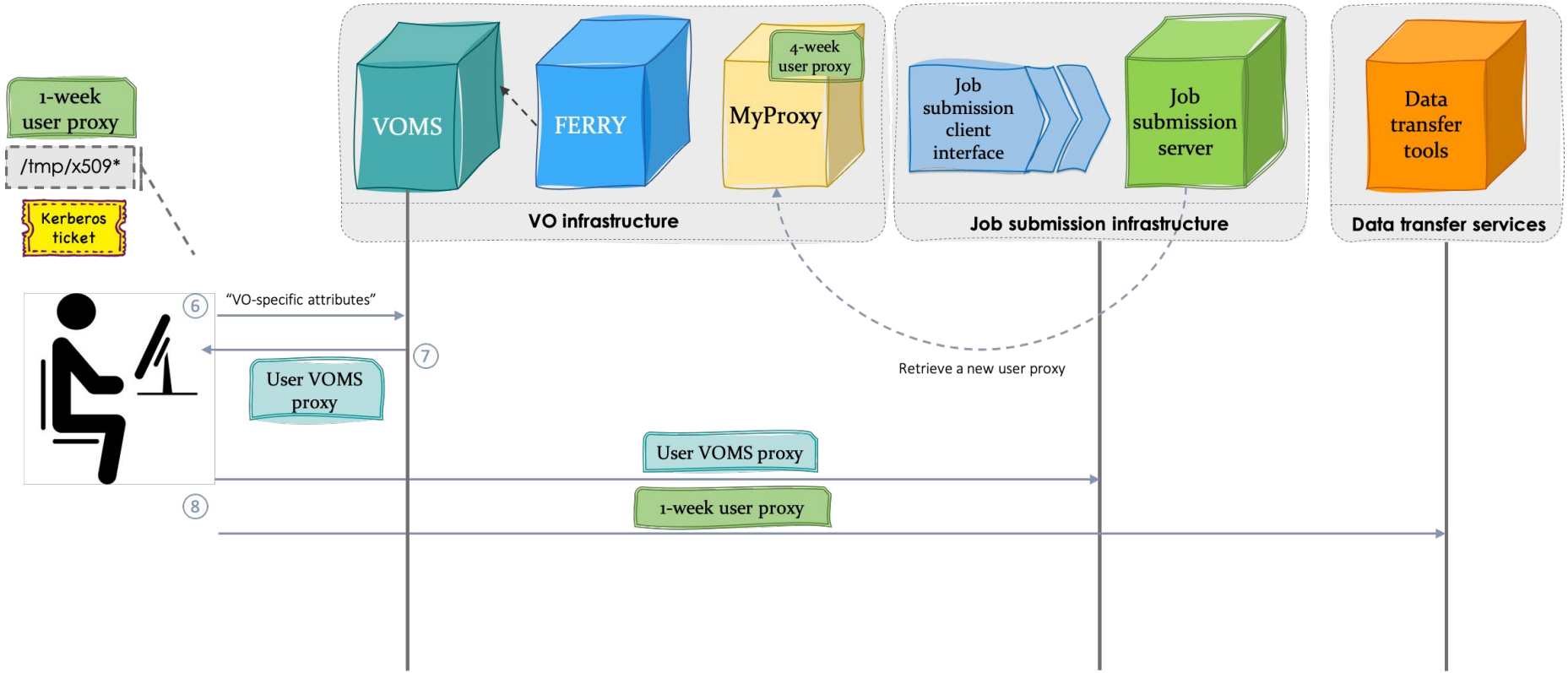
- During this transition, we added MyProxy as an online credential repository for long-lived jobs and data transfer tasks.
- An ECP-enabled Identity Provider (IdP) was put in place and registered in InCommon Trust Federation.
- A new command-line tool: `cigetcert` was developed to retrieve certificates from CILogon Basic CA using Kerberos tickets and assertions from our IdP.
  - `kx509` is now a wrapper for this command-line tool.
- In 2018, we deployed an authorization attributes repository: FERRY, which currently populates VOMS for VO-specific authorization attributes.
- Due to the need of using a higher assurance profile, we switched over to CILogon Silver CA last year.

Our current AAI infrastructure



Fermilab

Our current AAI infrastructure



Fermilab

Federated Identities project

- Integrate federation-based technologies, systems, and processes with Fermilab's distributed scientific computing infrastructure.
  - Move away from X.509 certificates in favor of JSON Web Tokens (JWTs). These tokens should include the appropriate identity fields and authorization attributes for users to access scientific computing and storage resources.
  - Scientific tools and services need to be updated to work with tokens.

Federated Identities project

- Allow external users to access Fermilab's scientific computing and storage resources and maintain compliance with DoE security policies.
- Enable trust relationships with IdPs:
  - Attribute release policy: specify which attributes of Fermilab users that can be released.
  - Fermilab's IdP operating procedure and assurance profile.
  - IdP trust requirements: for external token issuers trusted by Fermilab and thus users with tokens from these issuers will be allowed access Fermilab resources.
  - Expect to accept at least CERN’s IdP in addition to our own.

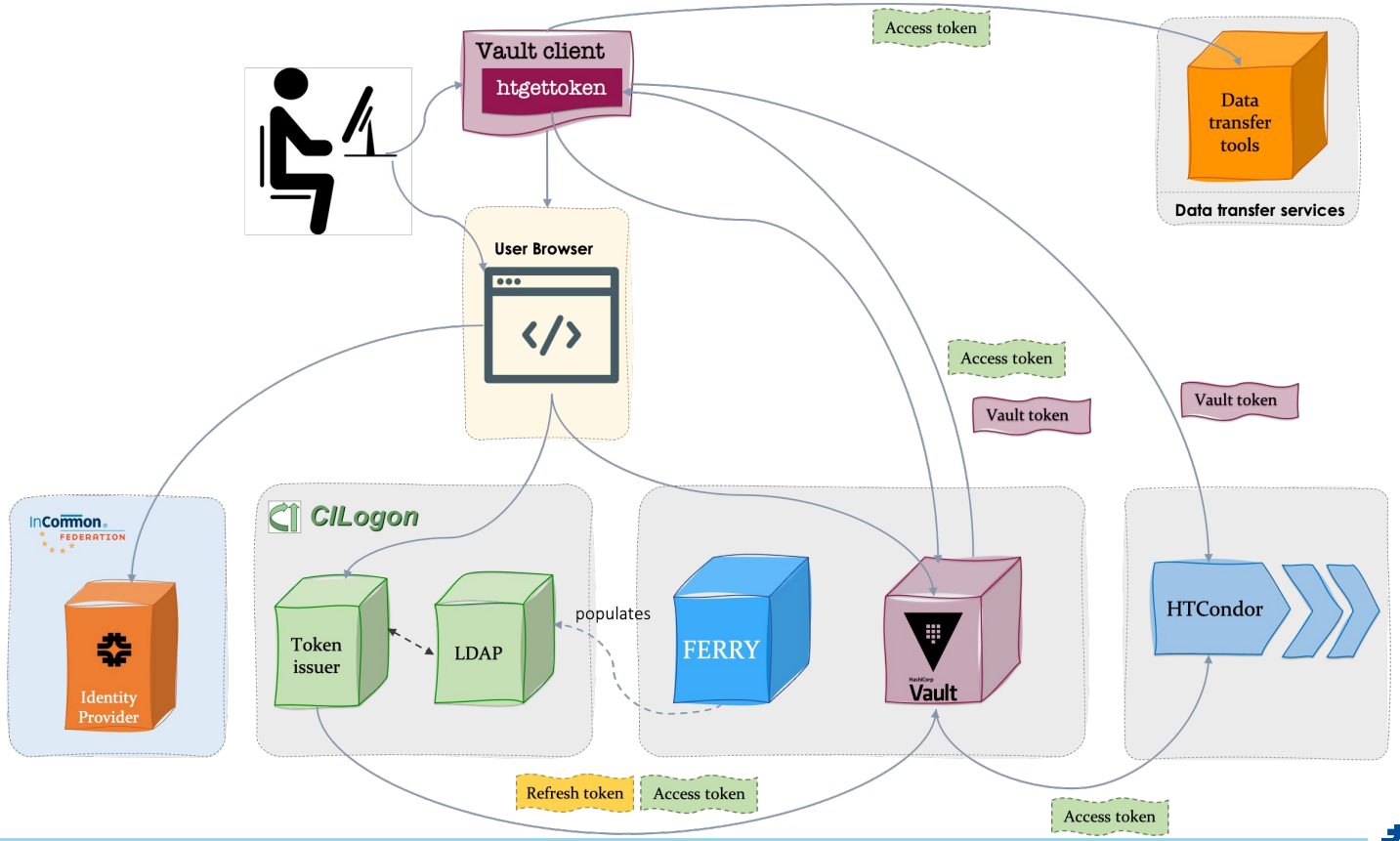
Requirements on new architecture

- Access credentials should be easy to use on a variety of computing interfaces and users should not be burdened with extra authentication credentials or a special configuration.
  - Allow inclusion of group membership and authorization attributes for different services
- Access credentials should be compatible to use with heterogeneous resources across several research infrastructures: WLCG, OSG, etc.
- Transitioning from X.509 certificates to tokens should be as transparent as possible for the end user and the scientific collaboration.

Integrating token-based technologies

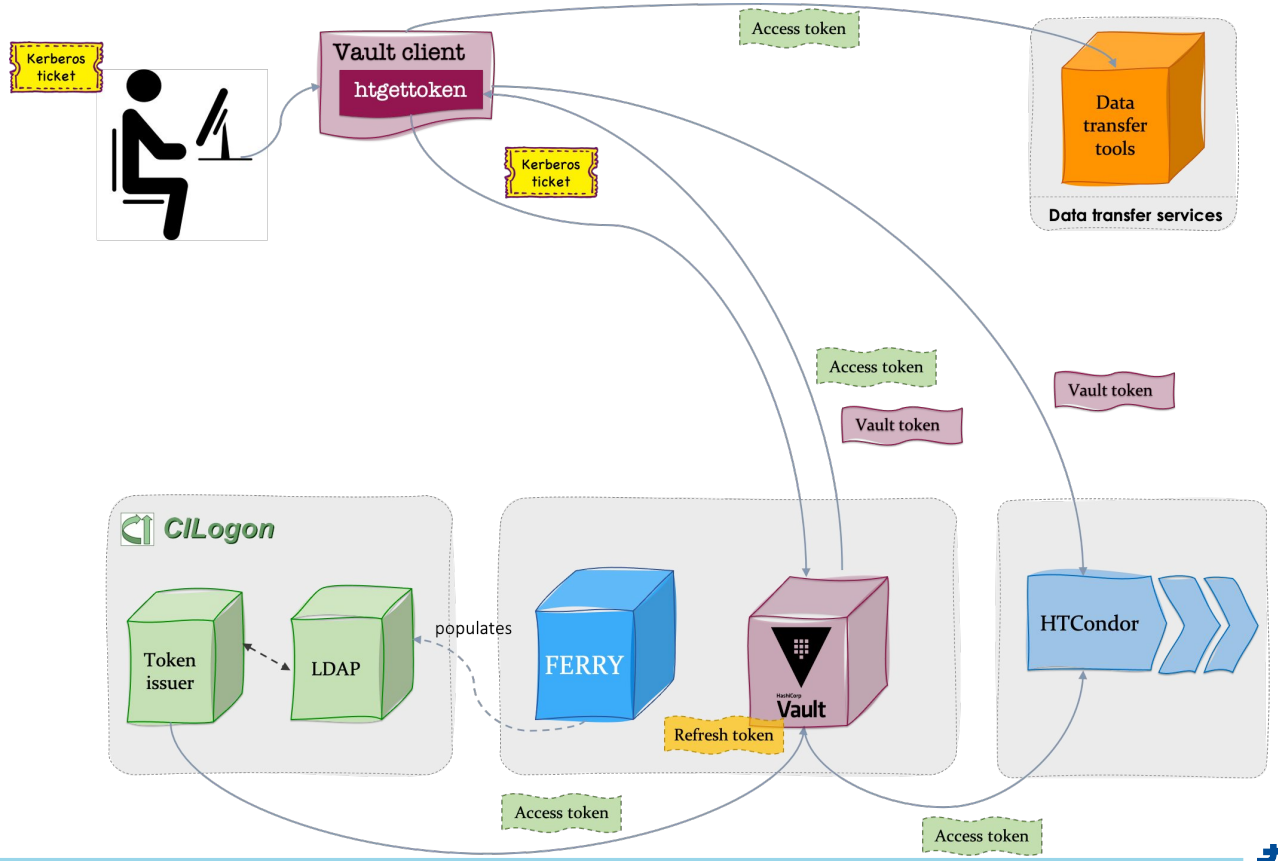
- OAuth 2.0/OpenID Connect/JSON Web Tokens.
  - Most popular modern authentication scheme, much software available.
  - Becoming accepted in the scientific computing community.
  - Credentials can be much more fine-grained than with X.509, using scopes; more secure.
- Capability-based WLCG token schema, SciTokens library.
  - Agreed upon with our major collaborators.
- CILogon as token issuer for Fermilab VOs.
  - Natural evolution of current X.509-based architecture, although departure from WLCG.
- Hashicorp Vault as OIDC client.
  - Replaces MyProxy in our architecture; HTCondor reads tokens for jobs from it.
  - Unlike MyProxy, however, all the token retrieval flows go through Vault.
- Make a new general-purpose script tool `htgettoken` for user interaction with Vault.
  - Replaces `cigetcert`.

Credential flow with OIDC authentication



Fermilab

Credential flow with Kerberos authentication



Fermilab

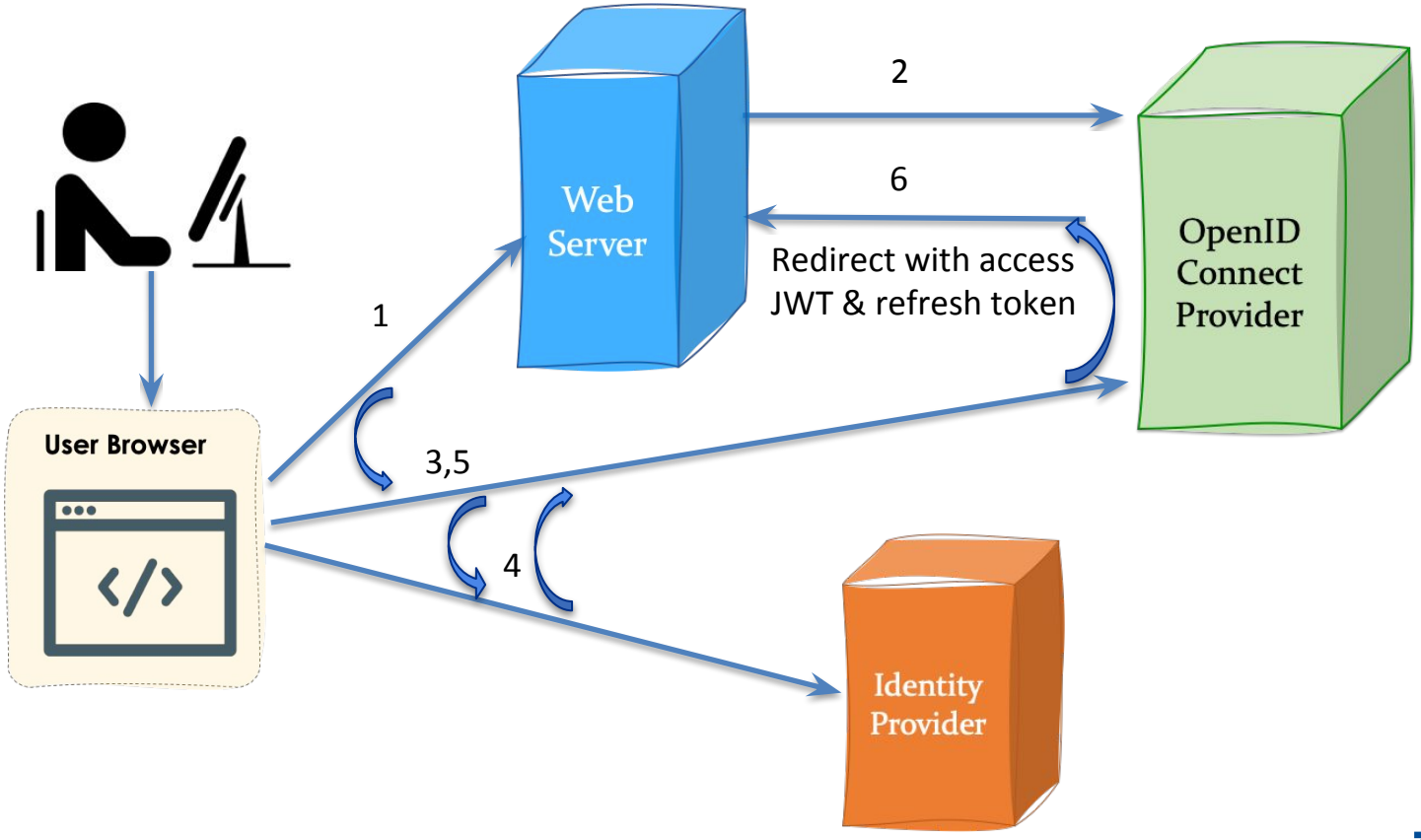
Progress on implementing the architecture

- Client tools mostly implemented.
- Initial information for issuing tokens being fed to CILogon.
- Tokens being issued and stored.
- Design plans made for:
  - Tokens issued for groups
  - Tokens for automated operations
  - Integration with HTCondor
  - Automating Vault configuration

Vault with htgettoken

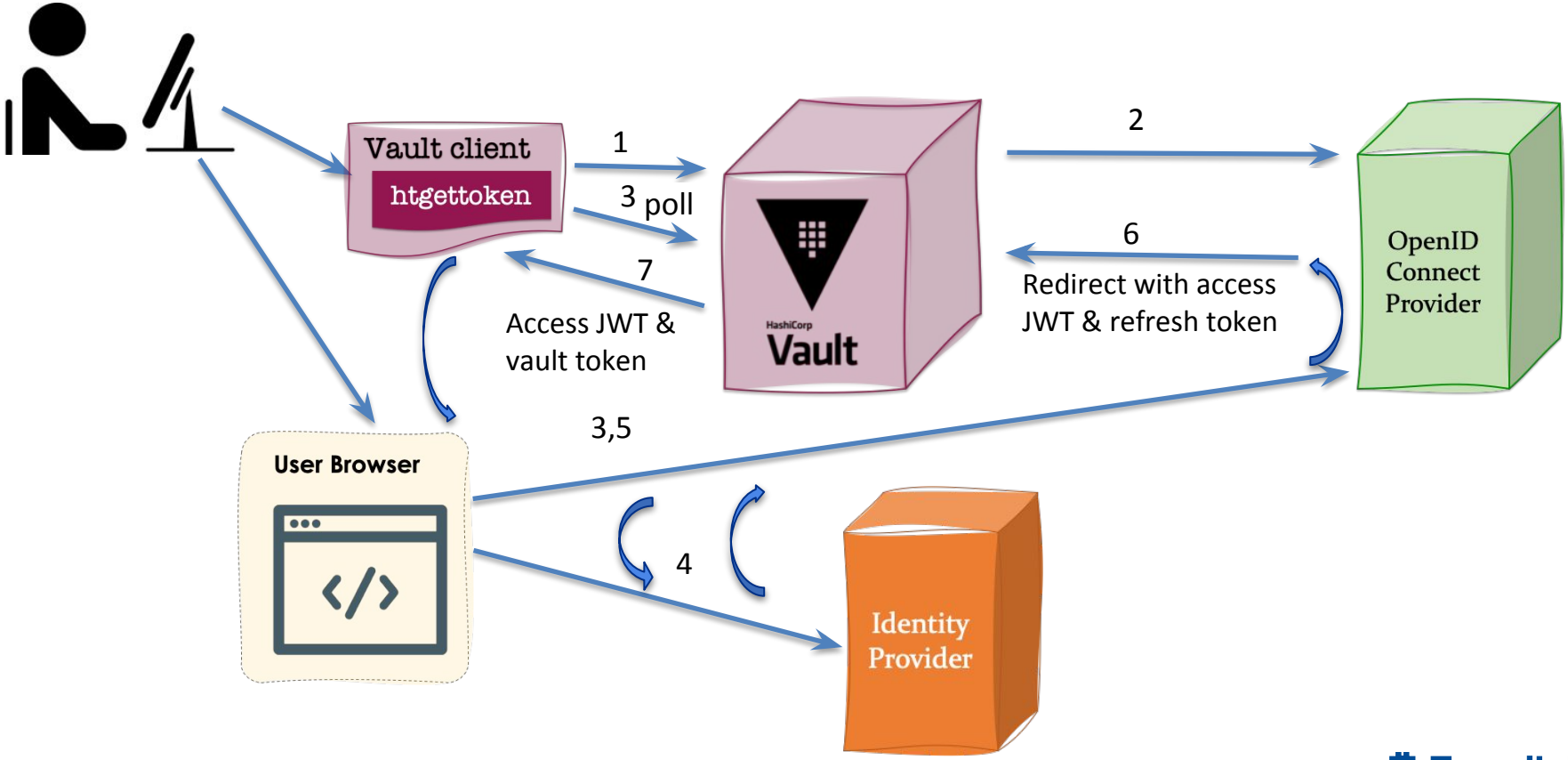
- Hashicorp Vault
  - Popular, full-featured open source secure secrets manager.
  - Very flexible plugin architecture and client/server API; accesses mapped like a filesystem.
  - Has existing OIDC authentication & storage plugins, and Kerberos authentication plugin.
    - Needed some extensions, submitted as pull requests.
  - Manages accesses with its own tokens.
- `htgettoken`
  - Relatively simple python command line client for the user to automate the flows.
  - Initially invokes Vault OIDC authentication via a web browser.
  - Long life (~1 month, renewable) refresh token stays in Vault, limited life (~1 week) Vault tokens and even shorter life (~1 hour) access JWTs stored unencrypted in local file.
  - New access tokens retrieved from OIDC token issuer via Vault using Vault token.
  - Automatically renews expired Vault authentication with Kerberos.

Normal Web-based Federated OIDC flow



Fermilab

htgettoken with Vault initial OIDC flow



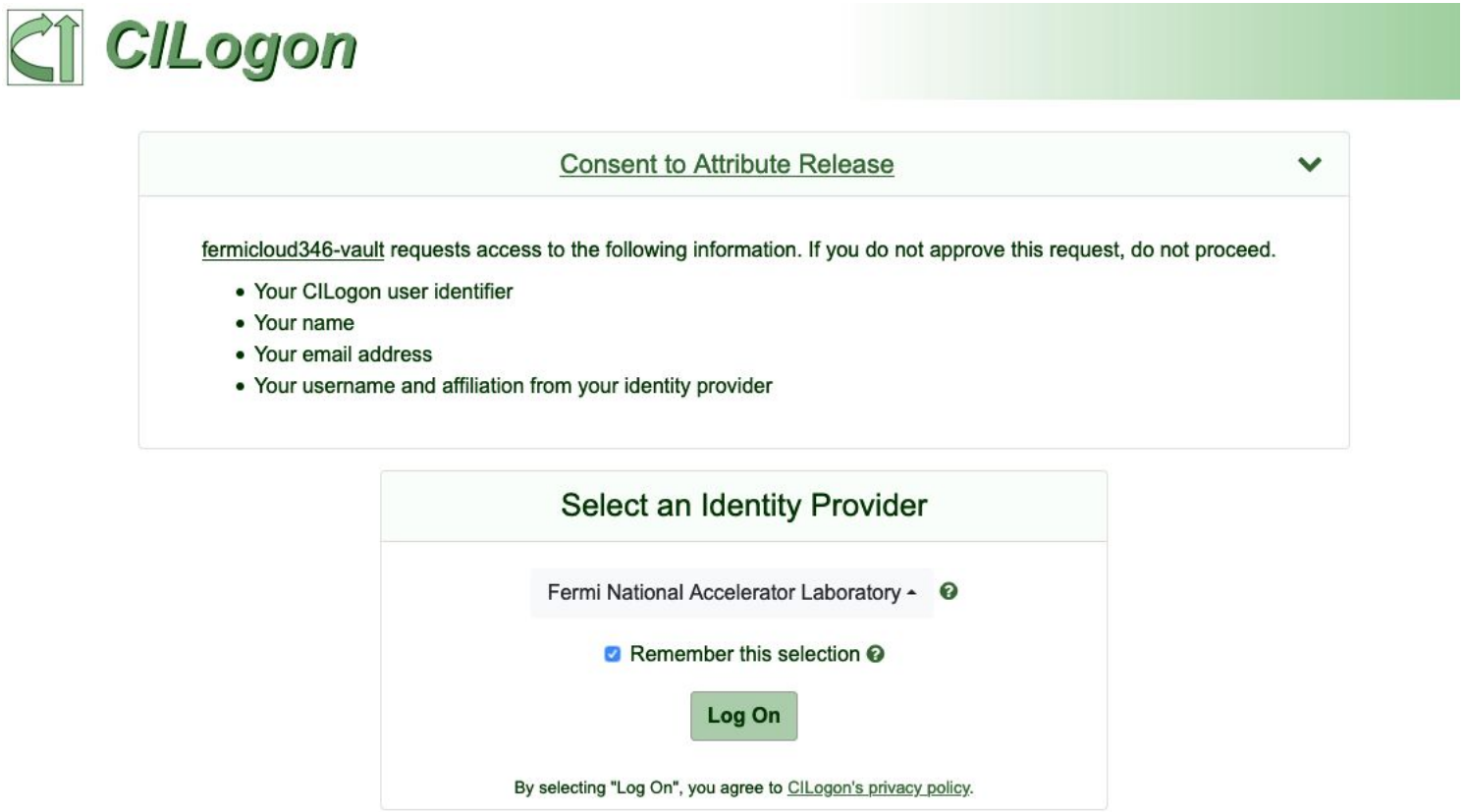
Fermilab

Demo -- initiate htgettoken

```
$ htgettoken -v -a fermicloud346.fnal.gov -i cilogon
Attempting OIDC authentication with https://fermicloud346.fnal.gov:8200

Complete the authentication via web browser at:
    https://test.cilogon.org/authorize?client_id=cilogon%3A%2Fclient_id%2F3b024f
44b076d36c4c7e3a0f86ad5677&nonce=d3e3f4998e808020af7790884b562e1f9756dc8b&redire
ct_uri=https%3A%2F%2Ffermicloud346.fnal.gov%3A8200%2Fv1%2Fauth%2Foidc-cilogon%2F
oidc%2Fcallback&response_type=code&scope=openid+profile+email+org.cilogon.userin
fo+storage.read%3A%2F+storage.create%3A%2F&state=2c601a29b30da146fcc6f79ce7e8c0f
946c19e77
```

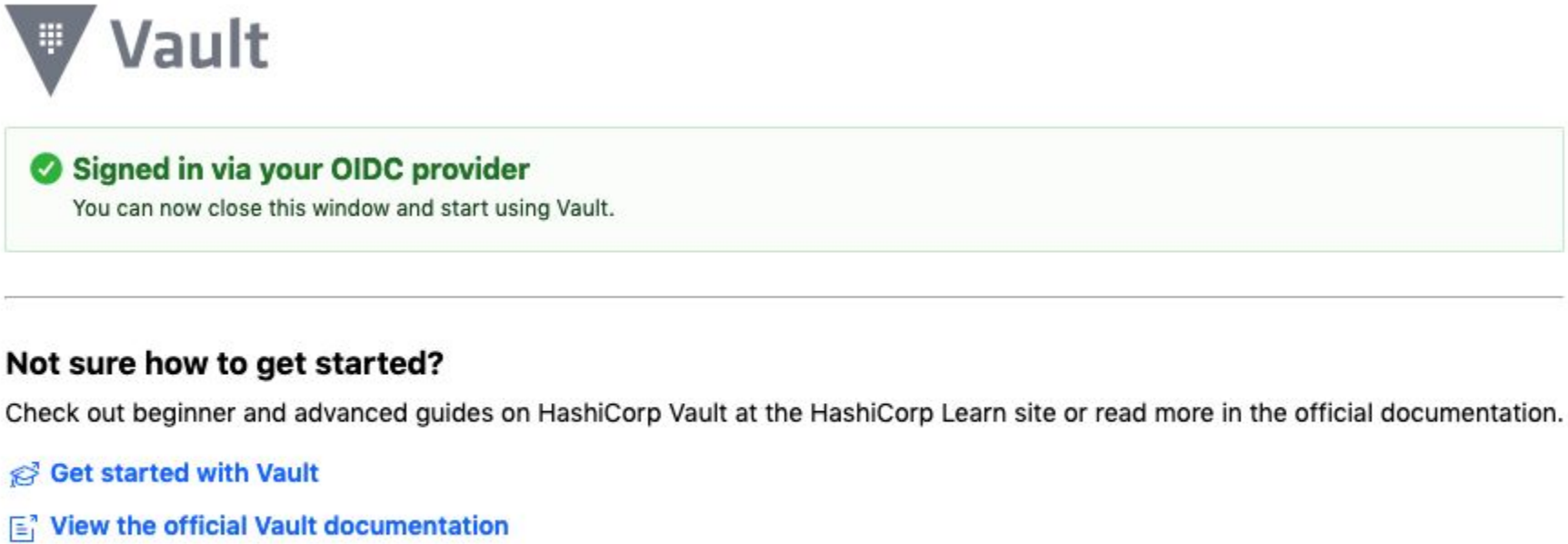
Demo -- token issuer consent, IdP selection



Fermilab

Demo -- authenticate with IdP

Demo -- redirect back via token issuer to Vault



Fermilab

Demo -- htgettoken completes

```
Storing vault token in /tmp/vt_u3382
Saving credkey to /cloud/login/dwd/.config/htgettoken/credkey-cilogon-default: d
wd@fnal.gov
Saving refresh token to https://fermicloud346.fnal.gov:8200
  at path secret/oauth-cilogon/creds/dwd@fnal.gov:default
Getting bearer token from https://fermicloud346.fnal.gov:8200
  at path secret/oauth-cilogon/creds/dwd@fnal.gov:default
Storing bearer token in /run/user/3382/bt_u3382
```

Demo -- look in the JWT

```
$ decode_jwt <$XDG_RUNTIME_DIR/bt_u`id -u`
{
  "wlcg.ver": "1.0",
  "sub": "dwd@fnal.gov",
  "aud": "https://wlcg.cern.ch/jwt/v1/any",
  "nbf": 1606770828,
  "scope": "storage.read:/dune/ storage.create:/dune/scratch/users/dwd",
  "iss": "https:cilogon.org",
  "exp": 1606774433,
  "iat": 1606770833,
  "jti": "https://test.cilogon.org/oauth2/accessToken/37963d89d68dbac924fc74e695
ff4d66/1606770833597"
}
```

Demo -- htgettoken with kerberos

```
$ htgettoken -v -a fermicloud346.fnal.gov -i cilogon
Credkey from /cloud/login/dwd/.config/htgettoken/credkey-cilogon-default: dwd@fn
al.gov
Initializing kerberos client for host@fermicloud346.fnal.gov
Negotiating kerberos with https://fermicloud346.fnal.gov:8200
Storing vault token in /tmp/vt_u3382
Attempting to get bearer token from https://fermicloud346.fnal.gov:8200
  at path secret/oauth-cilogon/creds/dwd@fnal.gov:default
Storing bearer token in /run/user/3382/bt_u3382
```

Token issuer configuration

- We have arranged with CILogon for their OIDC token issuer to make its decisions based on information we send them.
  - They host an LDAP server containing information about each person authorized to get tokens.
  - We populate LDAP based on our custom system FERRY that maintains authorization information for everyone that uses scientific computing.
    - We include the list of JWT scopes that are authorized for each person.
    - Changes to FERRY information is largely done through Service Now processes.
  - We plan to also specify the project/experiment/Virtual Organizations in LDAP so new projects can be added without any intervention by CILogon people.

Group tokens

- We need tokens for groups too, not just individuals.
  - For example, a group of people manage “production” jobs for each experiment.
  - However, only individuals can do OIDC authentication.
  - Our plan for this:
    - List the authorized individuals for group in LDAP.
    - Have any individual in the group select a Vault “role” in order to get a refresh token for the group stored into Vault.
    - Have CILogon verify that the individual is an authorized group member.
    - Use a group Kerberos credential to renew Vault tokens for the group.

Automated operations

- Some users also have a need to do automated operations such as job submission and file movement from scripts.
  - We can get special long-lived Kerberos principals for this purpose of form “user/method/hostname”, for example “dwd/cron/fnalu.fnal.gov”.
    - Those work with the Vault Kerberos plugin and can be used as part of the Vault secrets path for a refresh token, while preventing access to other refresh tokens of that user.
    - The “user” can also be a group login for example “dunepro”.
  - For deployments that do not use such Kerberos principals, a Vault administrator could issue an indefinitely renewable Vault token with access to a limited secrets path.

Implementation summary/status

- Getting credentials is almost as hidden as in the old system.
  - Users with Kerberos only have to authenticate with web browser once.
  - For those without Kerberos, a Vault authentication plugin for ssh-agent could be written.
- All components are open source and follow widely-used protocols.
- `htgettoken` is feature-complete, expected to be distributed by OSG rpm soon.
- Some Vault pull requests merged, some still pending.
- Access JWTs can now be obtained from the CILogon test server, using initial scopes for each person specified by us through a CILogon-hosted LDAP server.
- Plan to make an HTCondor credmon plugin that gets new access tokens from Vault to refresh job tokens.
- Plan to make Vault configurator rpm.

Challenges & Concerns

- Transitioning from certificates to tokens: a hybrid environment.
  - Plan to keep both infrastructures in place for a transition period, then phase out the old one.
  - Services will need to support both styles for a while.
- Trust between Identity Providers is currently established on a one-to-one organizational relationship.
  - The Security Incident Response Trust Framework for Federated Identity (SIRTFI) aims to enable the coordination of incident response across federated organizations.
  - There are not assurance frameworks regarding operations of Identity Providers (IdP) to enable broader trust amongst organizations and increase participation in a federation of research infrastructures.
  - Our plan is to only slowly enable additional trusted IdPs, beginning with CERN.

Links

- Vault & plugins
  - https://www.vaultproject.io/
  - https://github.com/hashicorp/vault-plugin-auth-jwt
  - https://github.com/puppetlabs/vault-plugin-secrets-oauthapp
- htgettoken
  - https://github.com/fermitools/htgettoken
- CILogon OIDC
  - https://www.cilogon.org/oidc
- SciTokens
  - https://scitokens.org/