FedCM Gaps, Questions and Concerns

Gaps, Questions and Concerns after reviewing the FedCM specifications through the lens of SAML mesh-federations and Higher Ed.

Discovery

Mesh federations RPs can have thousands of OPs. Many RPs redirect users to a discovery service for making this selection (SAML disco spec).

- The discovery service may be on a 3rd party domain.
- Discovery may use cookies to remember previous selections
- Discovery may not prompt the user to make a choice if a previous selection is remembered.
- Discovery service choices are customized per service provider/RP.
- Discovery based on email address is not well suited for higher ed.
  - A university may have common email domains for schools in a system, but users must authn at their school's OP

Similar to OAuth2 authorized redirect urls, the discovery specification has service providers (RPs) define a discovery response url in their saml metadata.
Examples:

- https://met.refeds.org/ Click 'Access through your institution' uses https://service.seamlessaccess.org for discovery

Questions:

- How will discovery work with FedCM?
    - Pass FedCM a discovery URL and SP entityId (RP client ID) and have it render the page? Do a redirect?
    - Lots of FedCM is privacy focused on hiding the RP from the OP until a choice is made. Will they be opposed to a discovery service knowing the RP in advance of choosing the OP?

Resources:

- Nicole is doing work on discovery https://github.com/nckroy/fedcm-rne-hacking/ and initial diagram https://github.com/nckroy/fedcm-rne-hacking/blob/main/documentation/fedcm-rne-sequence.png
- https://github.com/fedidcg/FedCM/issues/319
- https://github.com/fedidcg/meetings/blob/main/2022/2022-10-03-notes.md#issue-319-make-sure-the-spec-allows-multiple-idps-in-the-get-call

Initial Login

If the RP knows the OP then it can initiate a login to the OP's signin_url.
FedCM then follows these steps

```
To Sign-in to the IDP given an IdentityProviderAPIConfig manifest and an IdentityProviderConfig provider:
    1. Let configUrl be the result of running url parser with provider's configURL.
    2. Let idpOrigin be the origin corresponding to configUrl.
    3. Assert that the Sign-in Status of the idpOrigin is signed-out.
    4. In parallel, wait until one of the following tasks returns to continue:
        1. Open a dialog that directs the user to the manifest's signin_url.
        2. Wait until the Sign-in Status of the idpOrigin becomes signed-in
            1. Close the dialog
            2. Return the result of the fetch the accounts list algorithm
        3. Wait until the user explicitly cancels the dialog
            1. Close the dialog
            2. Return empty list
```

Questions:

- How much control of the dialog does the IdP have? Can it prompt for MFA, passkeys, etc?
- How does the RP request authentication contexts? (See RP Authentication Requirements/Request MFA)

Reference

- https://github.com/fedidcg/FedCM/issues/380 Clarify use case of not-yet-logged-in user
- https://github.com/fedidcg/FedCM/issues/377 IdP requires knowledge of RP to allow a session

- https://fedidcg.github.io/FedCM/#dom-identityproviderapiconfig-signin_url

Idp Proxying/Authentication Chaining/Protocol Bridging/

We often refer to this as a SAML proxy. In FedCM terms, an RP makes a request to an OP. The OP in turn may then be an RP for another OP, or it may first show discovery to allow the user to select the next OP. Authentication protocols may change

Example:

- https://stackoverflow.com/ -> "Login with Google" -> OIDC flow to Google -> enter swl@stanford.edu -> SAML redirect to Stanford

Questions:

- Can the OP in the 'IdP Sign in flow' support triggering login to another OP?
- Can the OP in the 'IdP Sign in flow' show discovery?
  - It may need to know the RP's identity to show the correct discovery

Resources:

RP Authentication Requirements/Request MFA

Higher Ed Mesh federations allow RPs to request that MFA is performed by the IdP by, when redirecting the browser to the IdP with a SAML authnRequest, requesting a SAML RequestedAuthnContext value of https://refeds.org/profile/mfa

Questions:

- How does RP send its desired authn context (e.g. ensure user has performed MFA)?

User without Email

The specification Account List API requires that each account has an email address. Not all of our accounts have email addresses.

Questions:

- Why is email required?
- What do we do when there is no email address for an account?

How to handle more than one RP or OP per origin

We see use cases with multiple SAML service providers within a single HTTP origin (protocol + domain name + port), and see it when interacting with Azure AD (all Azure SAML IDPs are on the origin https://login.microsoftonline.com/). This presents a few issues with FedCM.

RP impersonation:
From

> An IDP MUST check the referrer to ensure that a malicious RP does not receive an ID token corresponding to another RP. In other words, the IDP MUST check that the referrer is represented by the client id. As the client ids are IDP-specific, the user agent cannot perform this check.

In the case of multiple RPs per origin this may not be sufficient. (Or perhaps if you are running multiple RPs on a origin you are already trusting all your RPs to behave)

State Machine

From https://fedidcg.github.io/FedCM/#browser-api-state-machine

> The keys in the state machine map are triples of the form (rp, idp, account) where rp is the origin of the RP, idp is the origin of the IDP, and account is a string representing an account identifier

Not sure how the state machine would handle a non 1 to 1 mapping between RP and IdPs

Questions:

- Can there be more than one RP per origin?
- Can there be more than one IdP per origin?
  - https://github.com/fedidcg/FedCM/issues/333

User Interactions during RP sign in

We have multiple use cases of needing to interact with the user during login request from the FedCM (this assumes the user already has a session at IdP).

- Stepup MFA
- Attribute release consent
- Picking an single attribute value from a multi-valued attribute
  - Parent logs in through idp, and idp has them choose which of their children's student ids to assert

- Yes, it seems stupid that software targeted to high ed would assume a parent only has one student at a school but we’ve had to do this at two universities.

Questions:

- How can the IdP interact with the user when the RP instructs FedCM to sign the user in

Assumption: Token is Opaque

From the identity assertion spec:

> The content of the token is opaque to the user agent and can contain anything that the Identity Provider would like to pass to the Relying Party to facilitate the login.

This would indicate that it would be possible to pass a SAML assertion through FedCM to the RP.

However later it says

> If all goes well, the Relying Party receives back an IdentityCredential which contains a token in the form of a signed JWT which it can use to authenticate the user.

It seems like the JWT reference is old language.

Reference:
https://github.com/fedidcg/FedCM/issues/318
https://github.com/fedidcg/FedCM/issues/82

Accounts endpoint: signed-in accounts only?

Should the `accounts` endpoint return all accounts the user has at the IdP, regardless of signed-in status, or only those that are currently signed-in?

In 5.3. Accounts List:

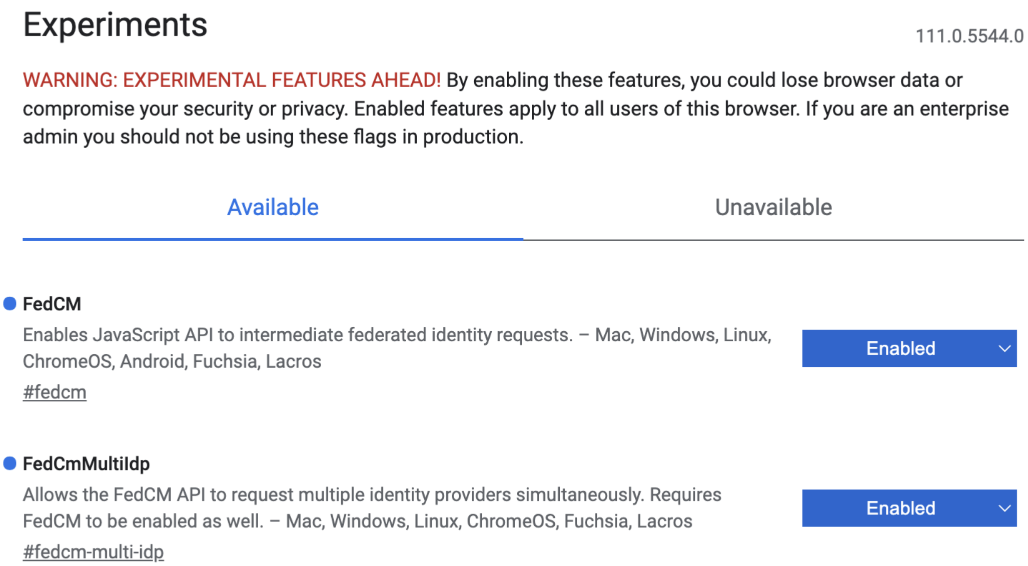
> The accounts list endpoint provides the list of accounts the user has at the IDP.
>
> The accounts list endpoint is fetched (a) with IDP cookies, (b) with the Sec-Fetch-Dest header set to webidentity, (c) without a Referer header, and (d) without following HTTP redirects.

However, step (4) of the sequence diagram in High Level Design states: “the browser proceeds to fetch the list of accounts *that the user is logged-in to*”. The current FedCM demo at https://fedcm-rp-demo.glitch.me/ also exhibits this behavior (does not prompt to use accounts that are logged-out).

Reference:
https://github.com/fedidcg/FedCM/issues/218

Misc. Notes

- Can’t get Chrome Canary (downloaded on 1/16/2023 – Version 111.0.5544.0 (Official Build) canary (arm64)) to work with https://fedcm-rp-demo.glitch.me/ demo.
    - Following Chrome flags are set:



Experiments

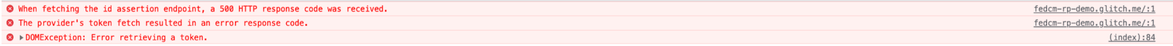
111.0.5544.0

WARNING: EXPERIMENTAL FEATURES AHEAD! By enabling these features, you could lose browser data or compromise your security or privacy. Enabled features apply to all users of this browser. If you are an enterprise admin you should not be using these flags in production.

● **FedCM**
Enables JavaScript API to intermediate federated identity requests. – Mac, Windows, Linux, ChromeOS, Android, Fuchsia, Lacros
#fedcm
Enabled

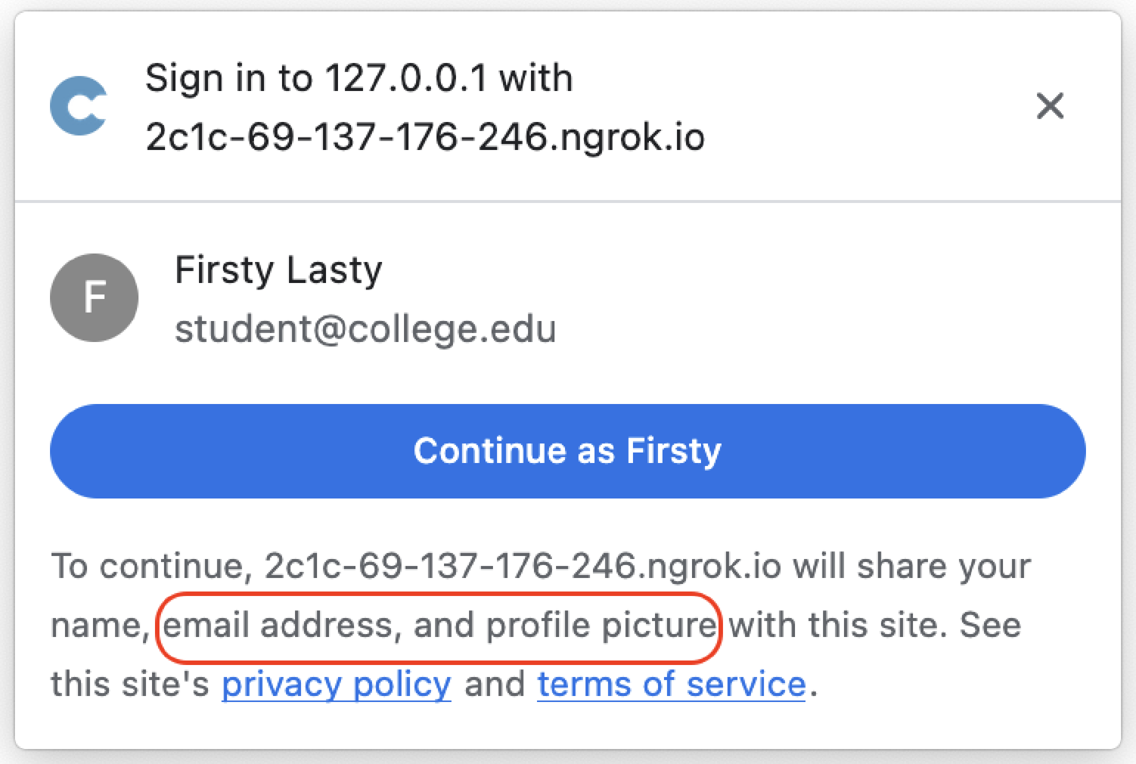
● **FedCmMultiIdp**
Allows the FedCM API to request multiple identity providers simultaneously. Requires FedCM to be enabled as well. – Mac, Windows, Linux, ChromeOS, Fuchsia, Lacros
#fedcm-multi-idp
Enabled

    - Running demo works for account selection, but fails after account is selected:



```
When fetching the id assertion endpoint, a 500 HTTP response code was received.    fedcm-rp-demo.glitch.me/:1
The provider's token fetch resulted in an error response code.                      fedcm-rp-demo.glitch.me/:1
▸DOMException: Error retrieving a token.                                                      (index):84
```

- This verbiage will be problematic for many R&E SPs. It is currently not configurable.



Sign in to 127.0.0.1 with 2c1c-69-137-176-246.ngrok.io

Firsty Lasty
student@college.edu

Continue as Firsty

To continue, 2c1c-69-137-176-246.ngrok.io will share your name, email address, and profile picture with this site. See this site's privacy policy and terms of service.

-