1. There is no SAML back channel, and in some situations that's a security feature. (Therefore, any solutions to issues can't be deferred to the backchannel) Example: on site at national lab, I can access servers and I can access the internet, but the servers cannot access the internet.
2. The "clientID" needs to support up to 2k in order to support the SAML request + sig; the "token" needs to support up to 100k (although perhaps that could be compressed before being sent...).
3. SAML delivers assertions to a specific list of known endpoints (ACS); FedCM asks the IdP to trust the Origin
4. SAML needs the IdP to be able to step up auth of users who may already have a session (and thus move users to a flow that may include redirects to MFA etc)
5. [maybe] the FedCM spec seems to want the IdP to keep track of the RP clientIDs that an account is associated with. If the SAML request is shoved® into that value, it will always be unique. How much does that ruin the user experience.
FedCM examined as a potential SAML binding compared to SAML2 SP initiated Web Browser SSO profile

Before FedCM begins, the RP will need to have identified a limited number of potential IdPs and generated the "clientId" - SAML AuthN request for redirect binding - for each. While there is intent for FedCM to support multiple IdPs, the number of providers R&E federations expect will unlikely be supported. Instead, R&E will likely need to do separate discovery flows.

Web page javascript invokes the browser FedCM API. This may be a list of idPs.

For each IdP, the browser confirms that the configURL corresponds to a supported IdP at that origin by inspecting the results of the origin's /.well-known file.

The config file allows specifying one of potentially many IdPs at an origin.

The well-known file contains a list of valid config URLs.

The config file contains information about the IdP.

Config file
confirm valid config location for IdP from /.well-known/ response

context-free calls to each IdP

GET /.well-known/web-identity
GET /specific-idp/config.json

URLs for IdPs

{ provider_urls: [...] }

{ accounts_endpoint: /specific-idp/accounts, client_metadata_endpoint: /specific-idp/client-md, id_assertion_endpoint: /specific-idp/assertions,0 branding: ...; }
Principal & SP specific calls to IdP

For each IdP, the browser interrogates whether the IdP recognizes the principal and whether it recognizes the RP.

GET /specific-idp/accounts
Cookie: name=value...

identify accounts from cookie

[accounts found]
{
  accounts: [ ]
}

IdP returns account(s) associated with cookie.

GET /specific-idp/client-md?
  client_id=[>1k SAML Authn Req]

IdP returns metadata associated with RP.

CHANGE POSSIBLE?
Could a signal that the IdP may need to have user interaction occur here?

if metadata is not failure,
metadata["privacy_policy_url"] is defined and
the provider’s clientid is not in the list of account["approved_clients"],
then the user agent MUST display the metadata["privacy_policy_url"] link.

if metadata is not failure,
metadata["terms_of_service_url"] is defined,
and the provider’s clientid is not in the list of account["approved_clients"],
then the user agent MUST display the metadata["terms_of_service_url"] link.

BUT ALSO
The request may contain requests to "step up" the assurance of the response. While the account is active, the IdP may still need to challenge the user.

User consent

Account chooser

Assume user selects from list of accounts.

All parties have expressed consent now:
* RP by requesting IdP
* IdP by responding on metadata end point

IdP provides SP token for user

POST /specific-idp/assertions
Cookies: name=value...
Origin: [RP origin]
account_id=123&
client_id=[>1k SAML Authn Req]&
nonce=Ct60bD&
disclosure_text_shown=true

SAML 2 SP initiated Web Browser
SSO profile redirects to out-of-band specified endpoint. Origin may not be the same party, such as a cloud service managed by a different org.

Origin check vs ACS check;

check nonce

Check cookies, account

alt [Valid session, no step up]

[up to 100k authn response]

[Failure]

[Step up needed]

need control signal

To fetch an identity assertion given a USVString accountid, a boolean disclosureTextShown, a boolean isAutoSelected, an IdentityProviderRequestOptions provider, an IdentityProviderAPIConfig config, and globalObject, run the following steps. This returns an IdentityCredential or failure.