Firefox and Federated Identity

Johann Hofmann & Peter Saint-Andre

Federation and Browsers Workshop
May 25, 2021

Some History

- We've been down this road before :-)
- From 2011 to 2016, Mozilla built a federated identity system called Persona (aka BrowserID)
- We learned a lot:
  - Decentralized identity is hard
  - UX is super-important, including both web and mobile
  - Sites need to control the UX as much as possible
  - The identity system should be invisible to users
  - Keep it simple (we shouldn't have included session management or attribute exchange)
  - See https://wiki.mozilla.org/Identity/Persona_AAR for details

Key Principles

- The user agent should be the mediator of the user's identity exchanges
- Information exchanged should be limited to the minimum necessary to complete the relevant transaction
- Identity transactions should not require cross-site tracking, collusion among relying parties (RPs), or tracking by identity providers (IdPs).
- Identity technologies should enable a wide array of IdPs to participate and should discourage centralization on a small number of IdPs
- It should not be necessary for an RP to register or have a formal relationship with an IdP in order for the RP to rely on identity information provided by the IdP
- It should be possible for the user agent itself to act as an identity provider
- Identity APIs should enable a user to choose a preferred IdP for a given RP
- Identity APIs should offer maximum user privacy and decentralization of authority

Current Status

Enhanced Tracking Protection

- Blocks third-party trackers from storing data such as cookies and thus engaging in cross-site tracking and user profiling
- Preserves core use cases such as auth through heuristics (detecting popups, redirects) + Storage Access API

Storage Access API

- Promise-based API to relax cookie & storage restrictions
- A good tool for when you have flexible third-party storage requirements and a strong relationship with the user

=> We would like to explore solutions that address authentication use cases with declarative APIs / without heuristics or Storage Access API