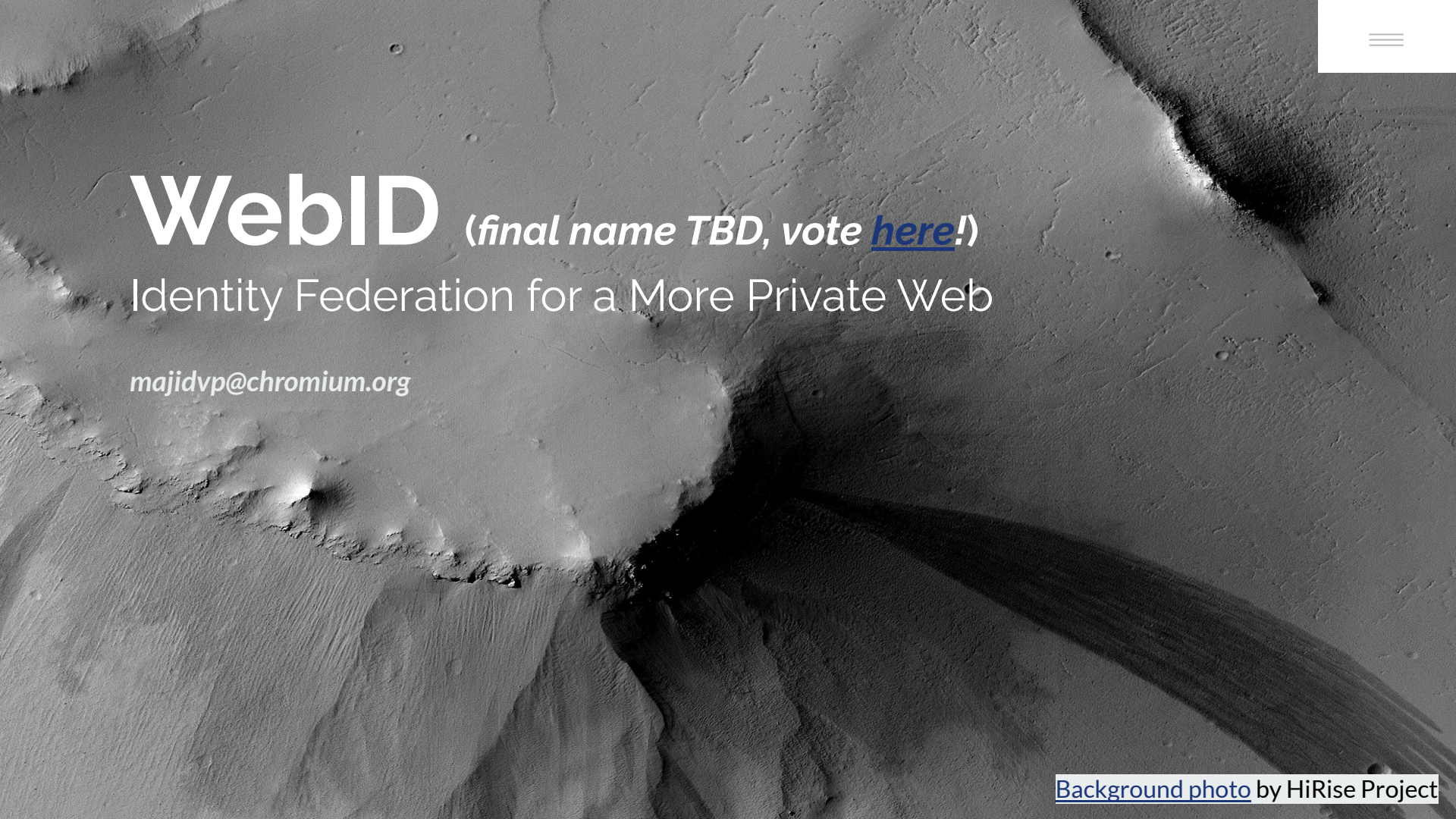


WebID *(final name TBD, vote here!)*

Identity Federation for a More Private Web

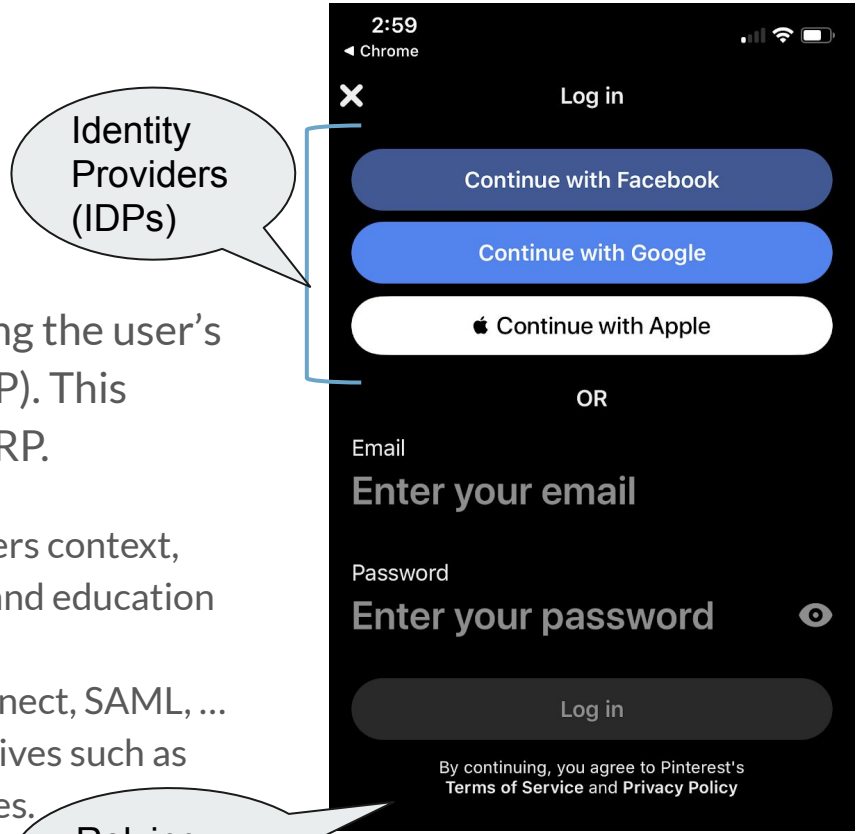
majidvp@chromium.org

Background photo by HiRise Project

Federated identity today

Users are authenticated to websites (RP) by using the user's existing credential with an identity provider (IDP). This could be account login or access to a service on RP.

- A variety of use cases e.g., social login in consumers context, single sign on and resource access in enterprise and education contexts.
- Common protocols include: OAuth, OpenID Connect, SAML, …
- These protocols are layered on top of web primitives such as link decoration (query parameters) and 3P cookies.



A more private web is coming

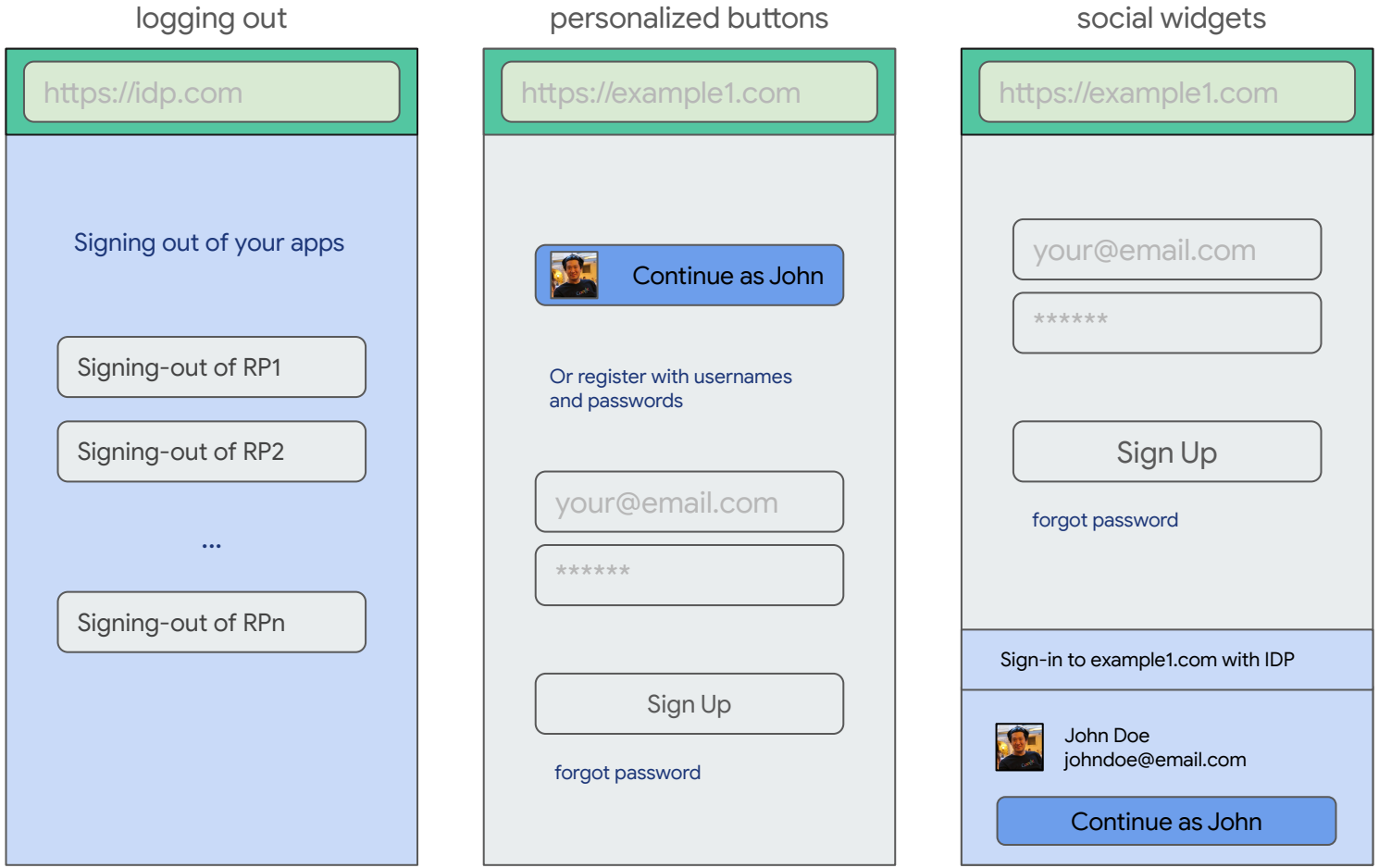
Privacy Model for the Web & Chrome Privacy Sandbox

Identity is being increasingly **sharded by first party** and not **linkable** by default.

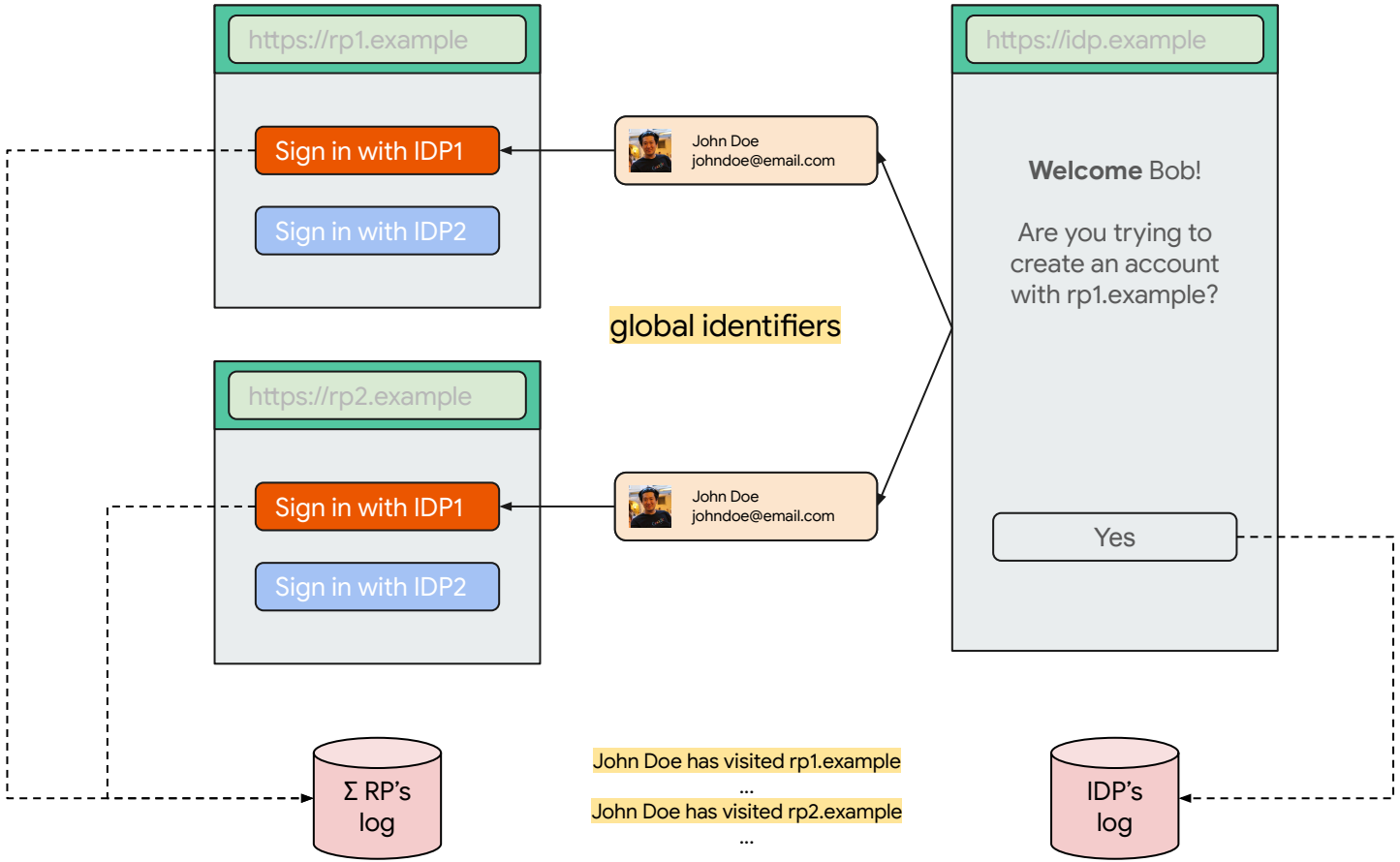
Relevant implications for federation:
3P cookie limitations reduce access to cookies in cross-site context. For federation this impacts session management, social buttons, and personalized widgets.

Photo by Jeremy Bishop on Unsplash

3P cookies use in Federation



The IDP and RP Tracking Problems



WebID Project objectives

To preserve and elevate federated identity for a more private web.

WebID approach: Identity-specific APIs

Introducing **higher-level identity functionality** to the web platform enables the browser to have visibility into identity exchanges and reduce privacy restrictions imposed on the more basic primitives.

Forward Chaining (today➟future) : Preserve identity federation while **meeting required privacy bar**. Starting with use cases that are impacted by **3P cookie limitations**.

Backward Chaining (future➟today) : Make identity federation **free of unintended tracking consequences.**

Priority of Constituencies

Web platform priority of constituencies can be paraphrased as users first, developers second, browser engineers third, technical purity last.

Users	1 |	**Maximize tracking prevention** (intentional or unintentional) and minimizing **user behavior change**.
Websites	2 |	**Optimize for backward compatibility** with RP deployed infrastructure.
Identity Providers	3 |	Meet **existing key use cases** for federation.
Browsers	4 |	Meet **current privacy sandbox requirements** but also has **a clear path** to the end state.

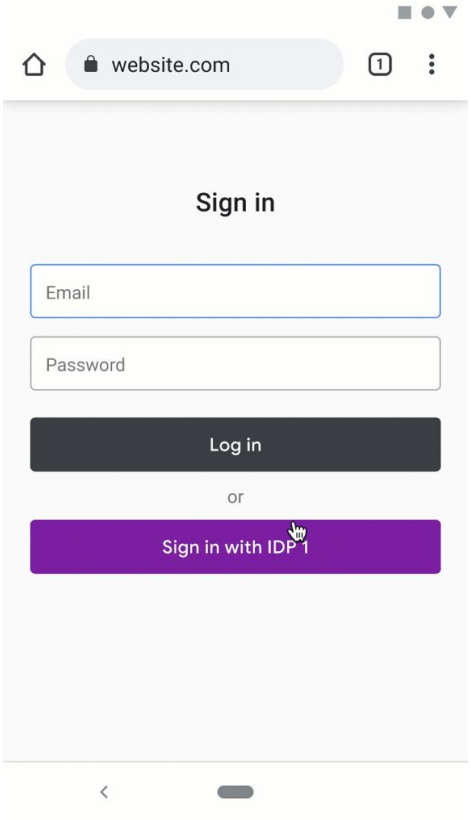
Proposals

Photo by Amin Safaripour on Unsplash

Permission Oriented

Enable cross-origin identity-specific data pathways if the user accepts a prompt warning them of the tracking risks. These prompts can be applied to existing identity protocol interactions (heuristically identified by the browser) or attached to a new API.

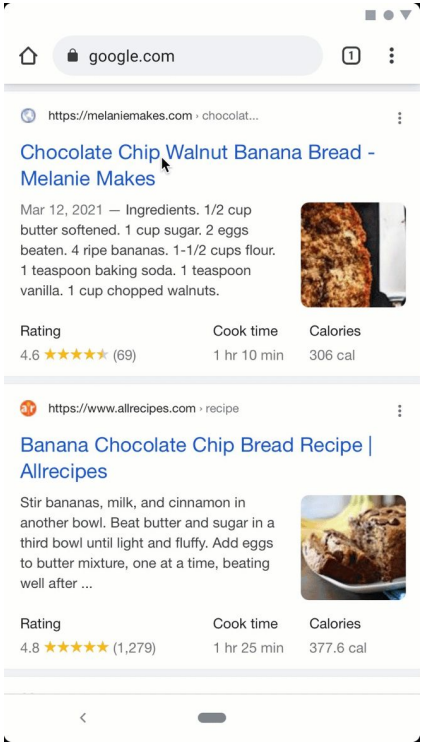
- **Pros:**
  - Most backward-compatible approach.
  - Outside permissions, UX is still under control of RP and IDPs.
- **Cons:**
  - Reliance on prompts contributes to **warning fatigue.**
- **Open questions:**
  - **Heuristic effectiveness:** Heuristically recognizing all identity transactions is difficult inevitably missing corner cases.
  - **Abuse:** frequent prompts leading to user permission blindness leaves room for trackers to hide as identity transactions.



Mediation Oriented

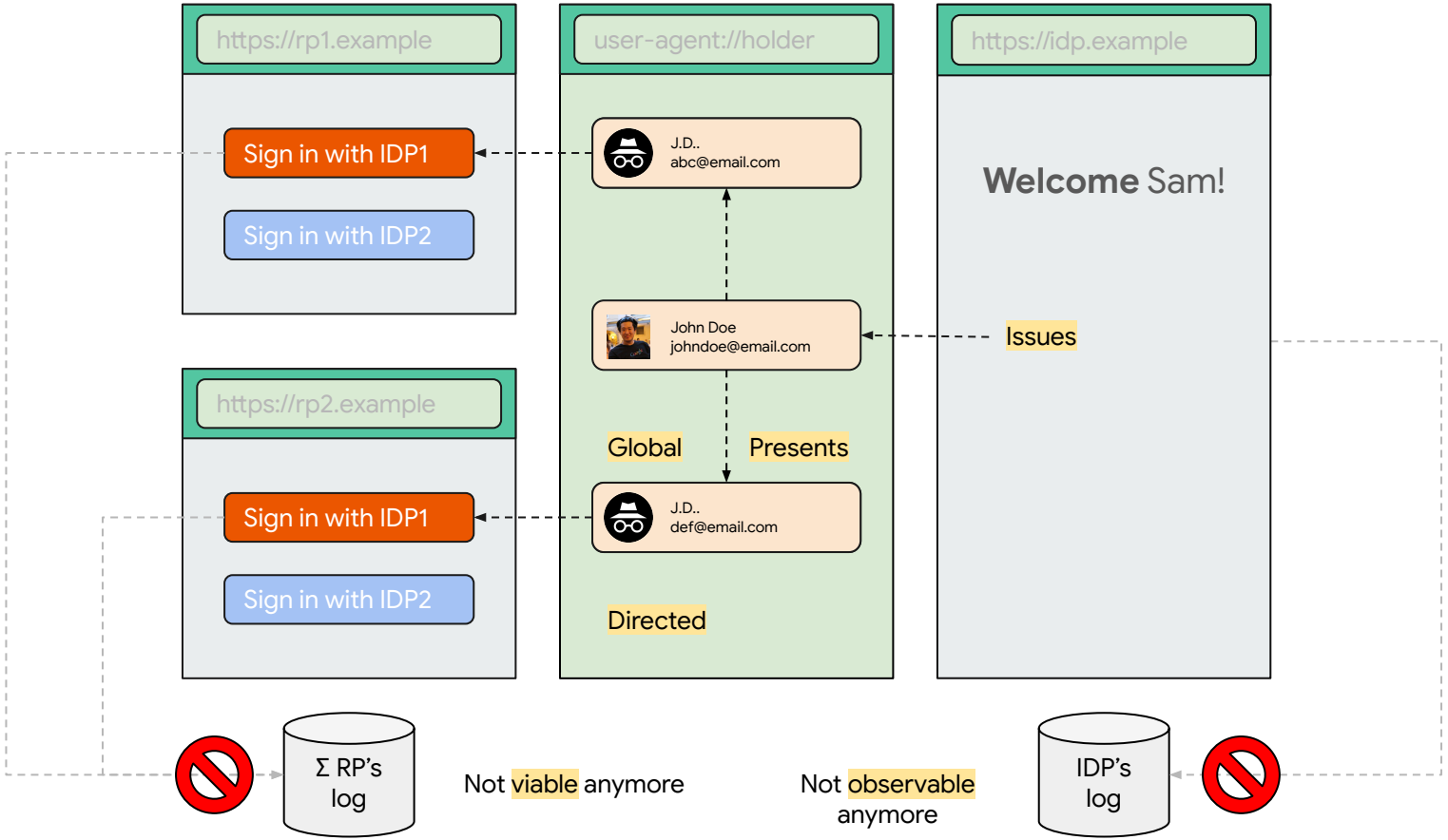
In this formulation the browser takes a more active role. In particular it is able to show relevant login and account selection UI to the user without revealing the RP identity to the IDP until after user decides to login.

- **Pros:**
  - **Improves UX** by reducing prompts, trustworthy and helpful in context UI.
  - Mediation is a **step toward** the consequence free end state.
- **Cons**
  - The browsers **ossifying** a particular sign-in UI hampers identity providers innovation on that front in the user land.
  - Inherently limited to a **narrow scope** (e.g., basic login) and does not cover the more varied and IDP specific scopes used for authorization.



Mediation Oriented - Demo

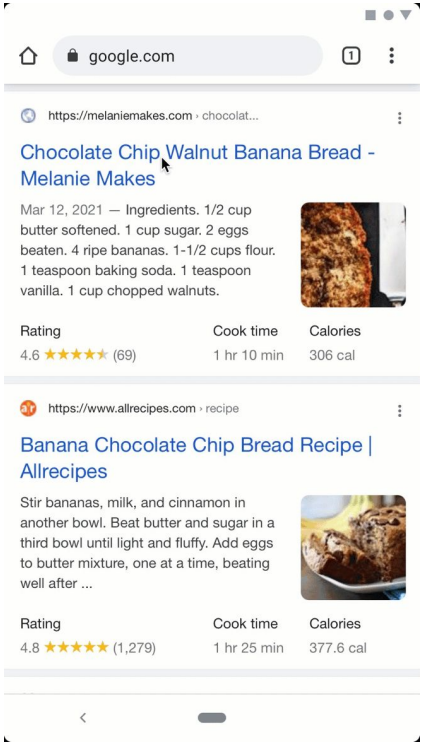
The **unbundling** of global identification into directed and The **unbundling** of issuing and presentation



Delegation Oriented

In this formulation the browsers intermediates the identity exchange to make **directed per-site identities default** and to **unbundle presentation** of credentials from its issuing.

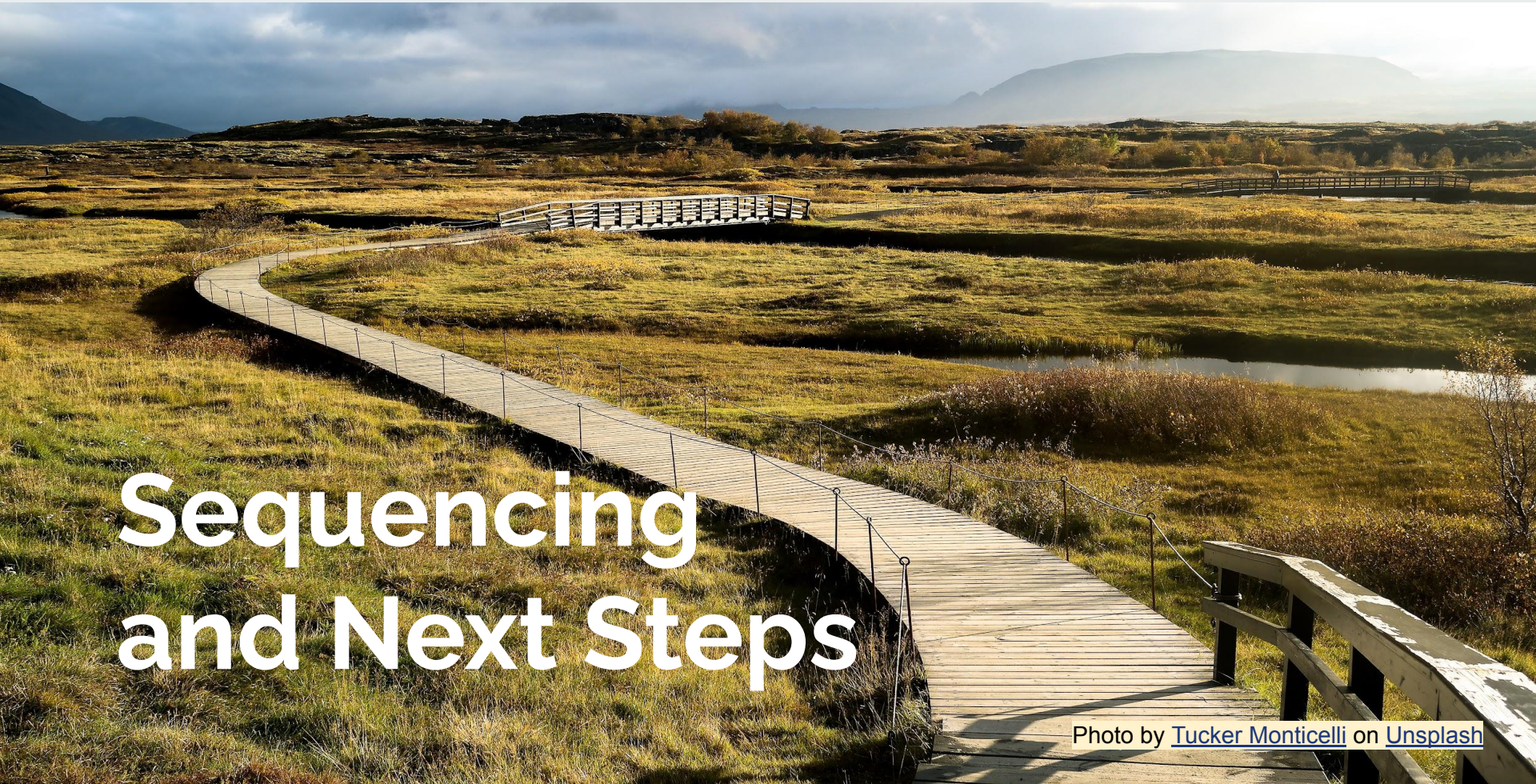
- **Pros**
  - Best UX.
  - Maximum prevention of IDP tracking.
- **Cons**
  - Not backward compatible with current deployments.
- **Open questions**
  - Economic equilibrium.



No Silver Bullets

These variations are not mutually exclusive. They make different trade offs. We believe they all have a role to play and would coexist long term to address a variety of usecases.

Variation	UX	Privacy Improvements	Ossification	Backward Compat
Permission				
Mediation				
Delegation				



Sequencing and Next Steps

Photo by Tucker Monticelli on Unsplash

Sequencing

1

Validate

3P Cookie APIs

2021 - Start dev trial in Q2 aiming for origin trial in Q3-4.

2

Launch

3P Cookie APIs

2021+

3

...Toward End State

2021+++

Call to action

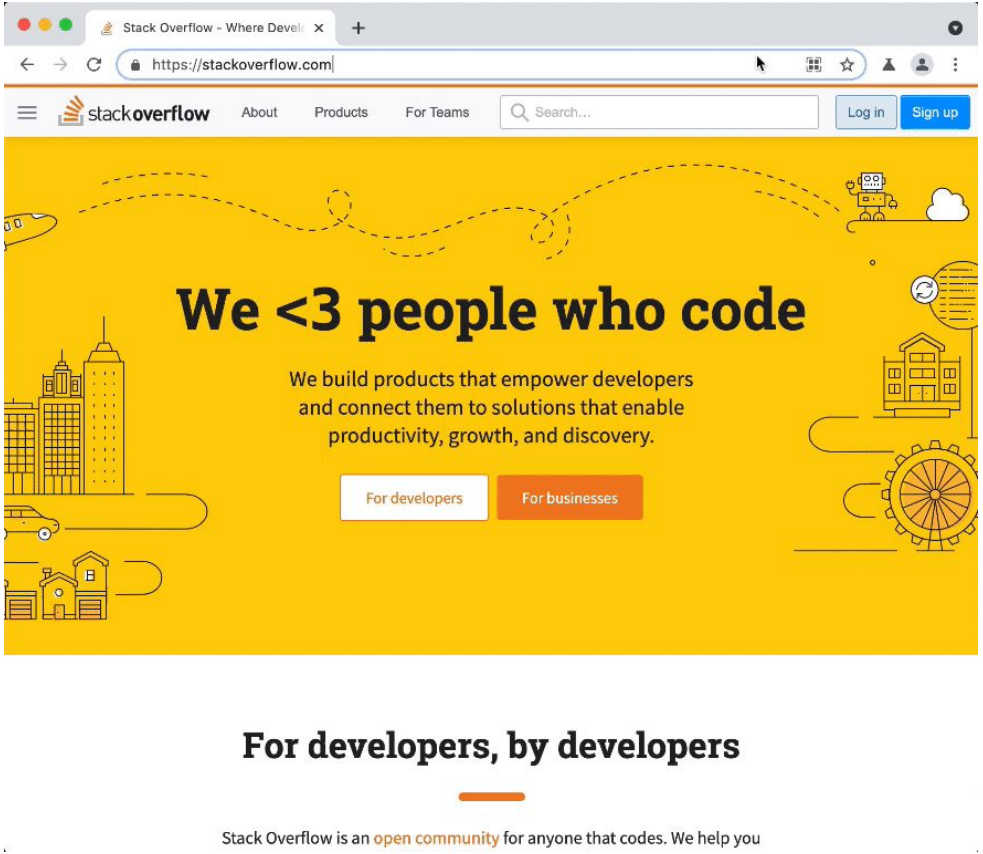
Help us:

- Attend our upcoming WICG workshop for WebID
    - Save the date: May 25th-26th. Details to be shared soon but please contact us if interested.
- Determine requirement by contributing your use case that may be impacted.
- Discover alternative options by evaluate current ones (for cookies and navigations).
- Understand what works / doesn't work by trying the APIs under development.

Thank you.

Annex

Permission Oriented - Demo



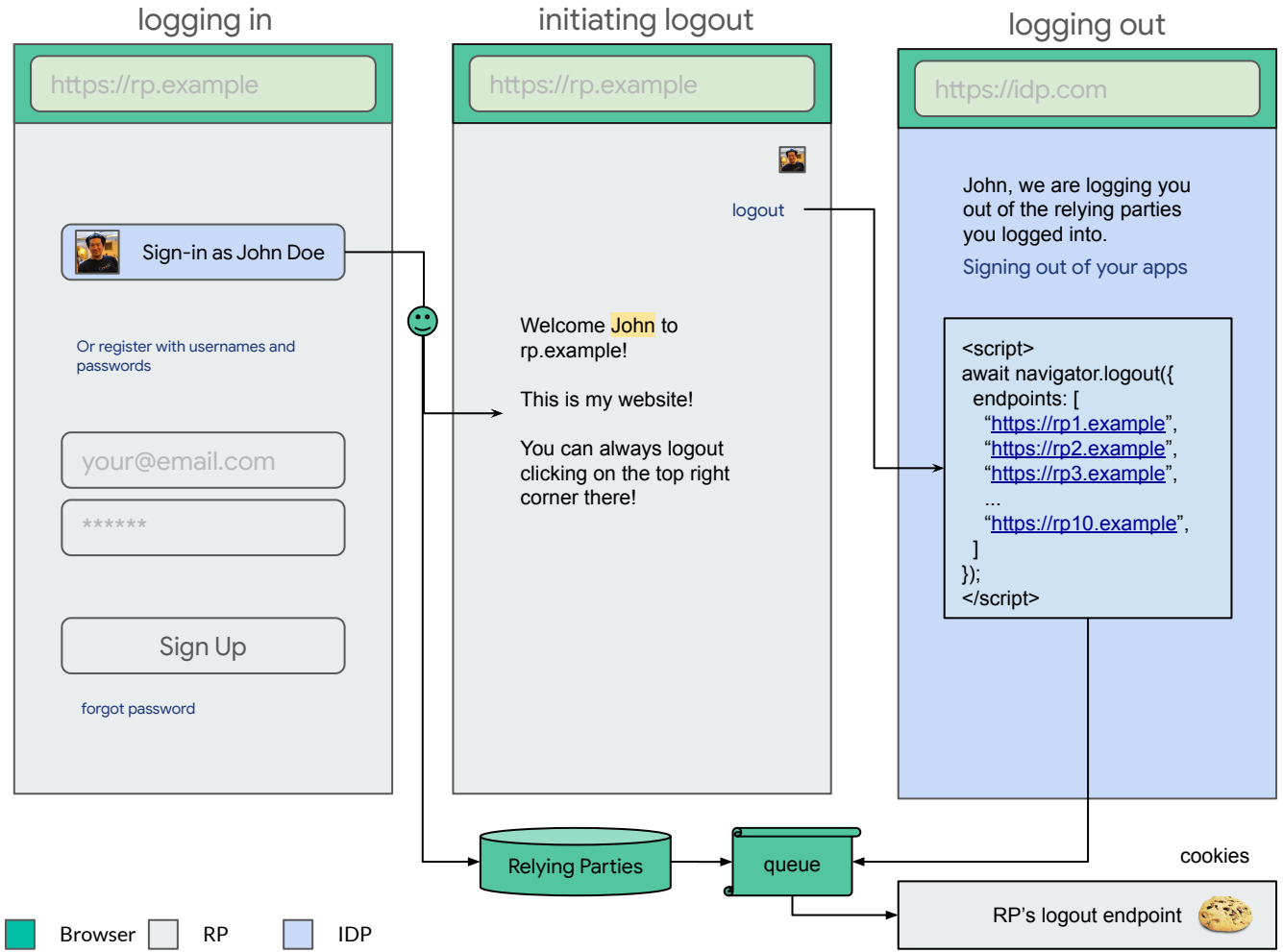
This is Chrome Canary with `--enable-features=WebID` flag demoing actual OIDC logins without any RP/IDP modification.

Session Management

Some features of session managements in the existing federated sign-in protocols depends on using iframes and 3P cookies. Two that we have identified in OIDC are: Front-channel logout and token renewal.

We are actively exploring solution space here but our current idea is to have **dedicated APIs** for these **behind permission prompts** that maybe skipped if the browsers have already observed the relevant login moment (either via mediation or permission).

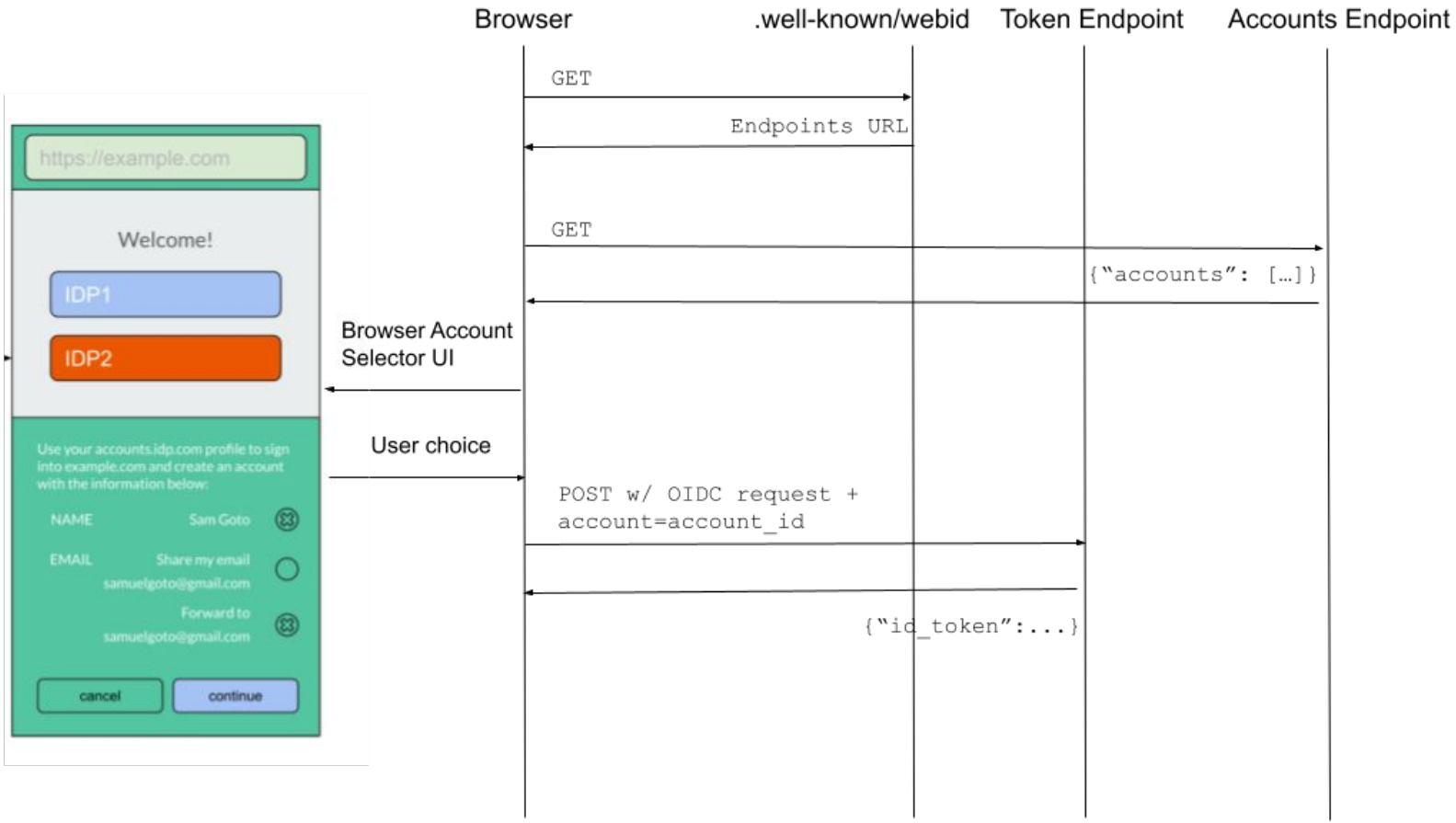
```
async function logoutAllSessions() {
  navigator.[tbd].logout(
      {
            endpoints: [
              'https://rp1.example',
              'https://rp2.example',
              'https://rp3.example',
              'https://rp4.example' ],
      });
}
```


logging in
initiating logout
logging out
https://rp.example
https://rp.example
https://idp.com
Sign-in as John Doe
Or register with usernames and passwords
your@email.com
******
Sign Up
forgot password
logout
Welcome John to rp.example!
This is my website!
You can always logout clicking on the top right corner there!
John, we are logging you out of the relying parties you logged into.
Signing out of your apps
<script>
await navigator.logout({
endpoints: [
"https://rp1.example",
"https://rp2.example",
"https://rp3.example",
...
"https://rp10.example",
]
});
</script>
Relying Parties
queue
cookies
RP's logout endpoint
Browser
RP
IDP

API examples

```
async function signInWithIDP() {
  navigator.[tdb].get(
      {
            provider: [ ‘https://accounts.idp.example’ ],
            request: { response_type: code,
                     client_id: 4e8sj09jj105,
                     state: 0ac879ed,
                     destination: https://rp.example },
           mode: ‘mediated’
       })
    .then(response => validateIDToken(response))
    .catch(err => handleError(err));
}
```

Mediation Oriented Data Flow

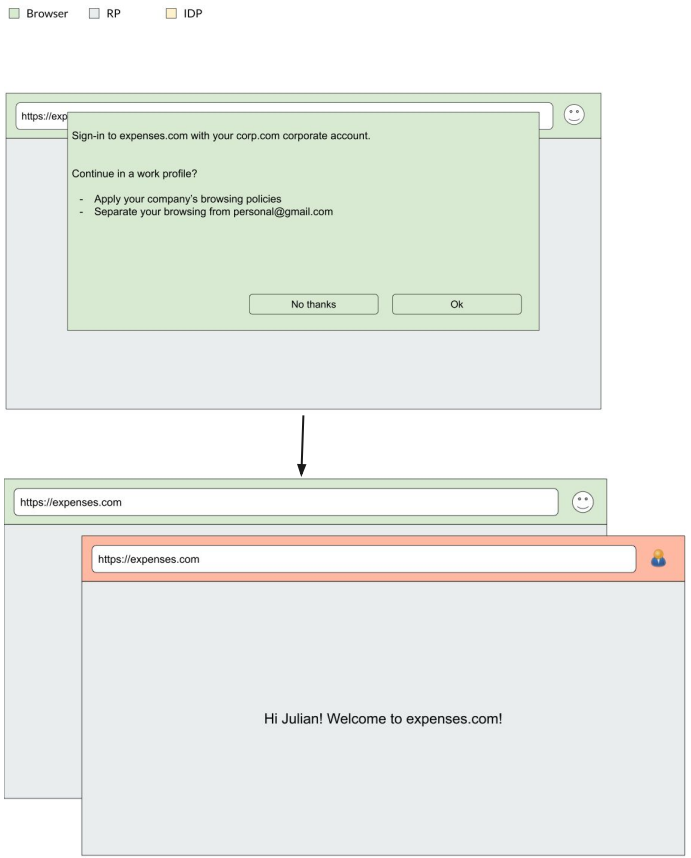


Enterprise

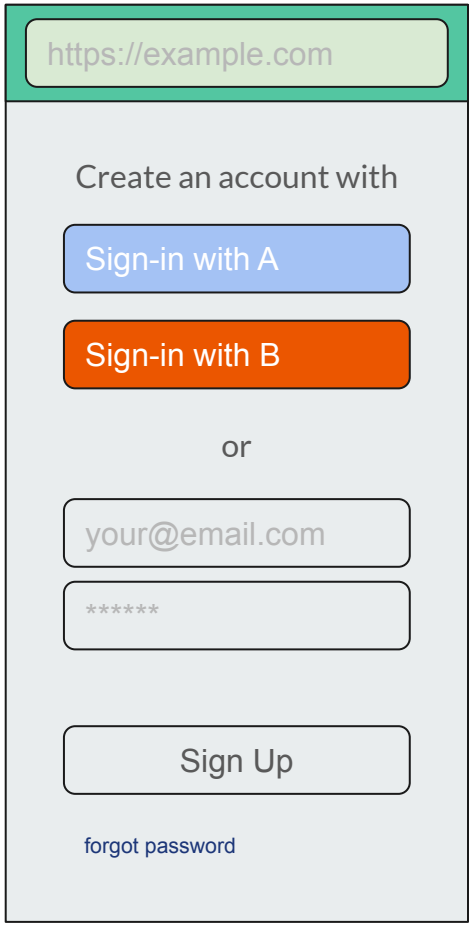
Enterprise use cases are different than consumer ones:

- Account policy is owned and controlled by employer and not employee.
- Different privacy expectations.
- More complex setup (e.g., leveraging federation as a single sign on solution across vendor sites)
- May or may not be used with employee personal device (BYOD)

This is a very active area of exploration but we are considering an approach where we separate personal and work profile where there each one has different set of policy and privacy settings.



Federation is Safer Than Usernames/Passwords

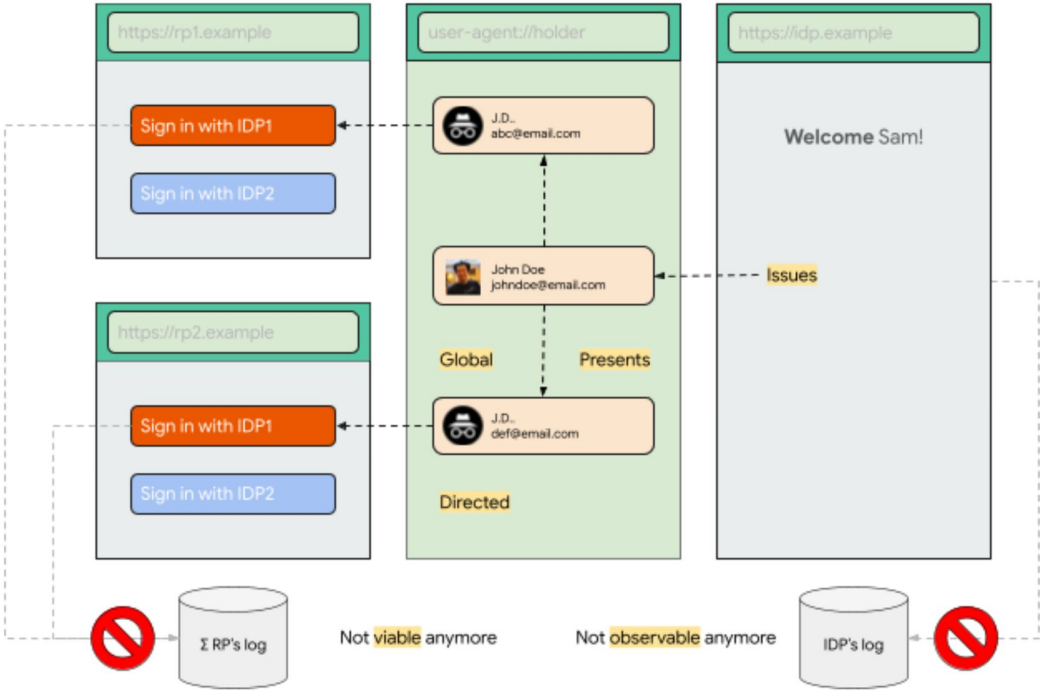


Two factor authentication,
password-less single sign-on

phishing, credential stuffing,
password reuse

Browser
RP
IDP

Unintended Tracking Mitigations



💡 **Mitigation of RP Tracking**: Unbundle your **global** identity into multiple **directed** identities per-site and progressive disclosure of identification.

💡 **Mitigation of IDP Tracking:** Unbundling the **issuing** of credentials with their **presentation**.

Unintended Tracking Risks

🤔 **Risks of RP Tracking:** Global identifiers (e.g., email) shared by default via federation allow cross-site identity joins.

🤔**Risks of IDP Tracking:** Identity providers are involved in all transactions even it is not justified.

Other related problems

- NASCAR flag problem
- Identity portability