~~WebID~~

FedCM

dsinclair@chromium.org

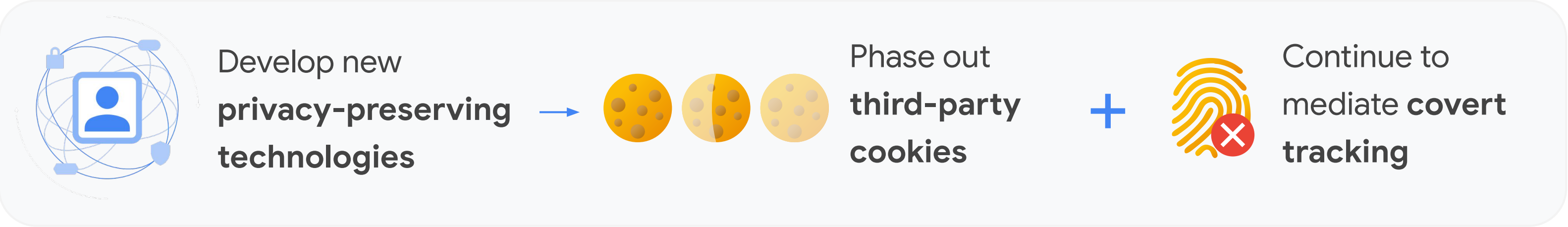
2021.11.17

PUBLIC

The why, when, what, and how

Why

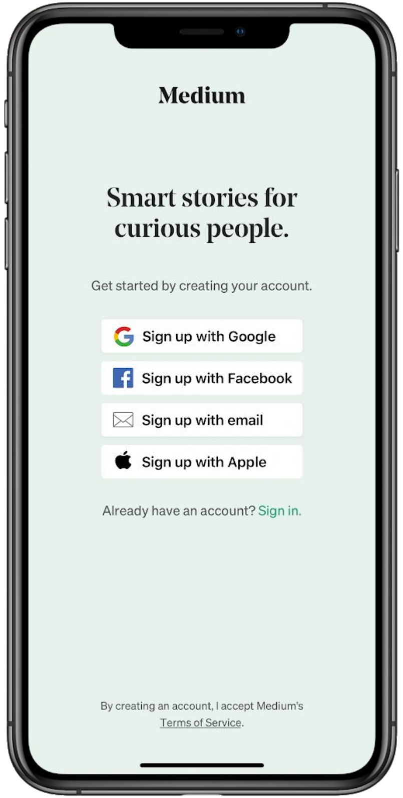
Privacy Sandbox



Develop new **privacy-preserving technologies** → Phase out **third-party cookies** + Continue to mediate **covert tracking**

https://privacysandbox.com/

Wait, federated identity?



Medium

Smart stories for curious people.

Get started by creating your account.

Sign up with Google

Sign up with Facebook

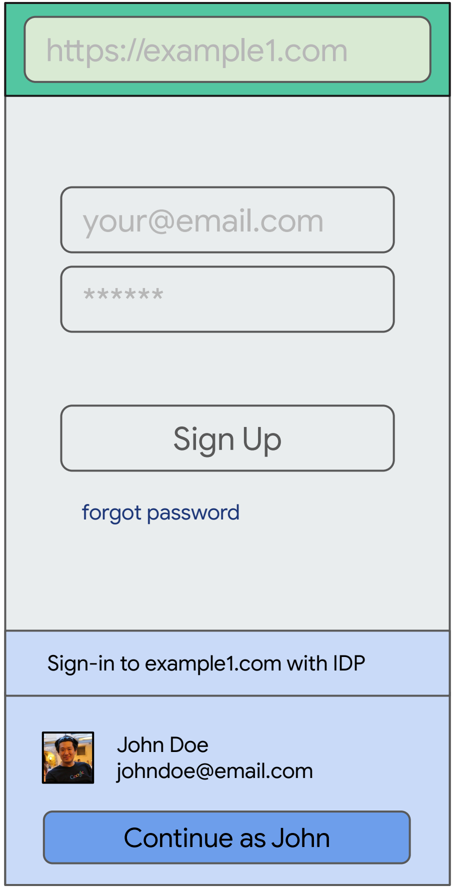
Sign up with email

Sign up with Apple

Already have an account? Sign in.

By creating an account, I accept Medium's Terms of Service.

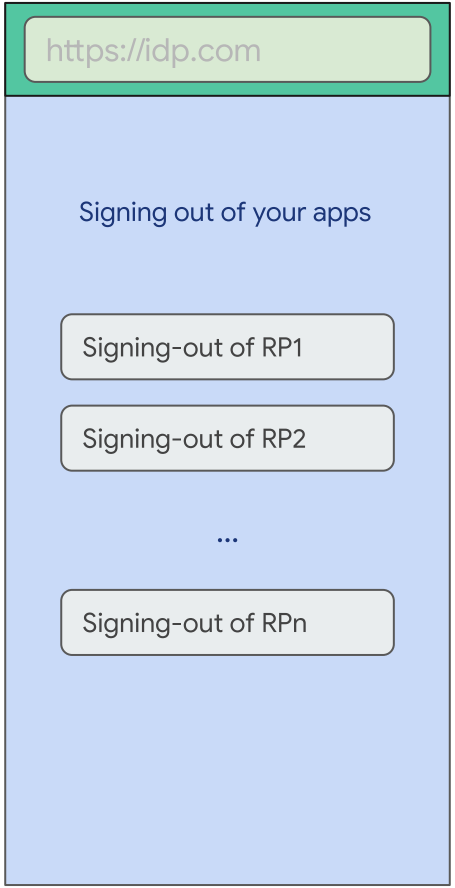
And cookies are involved?



Sign-in to example1.com with IDP

John Doe
johndoe@email.com

🟩 Browser
□ RP
🟦 IDP



https://idp.com

Signing out of your apps

Signing-out of RP1

Signing-out of RP2

...

Signing-out of RPn

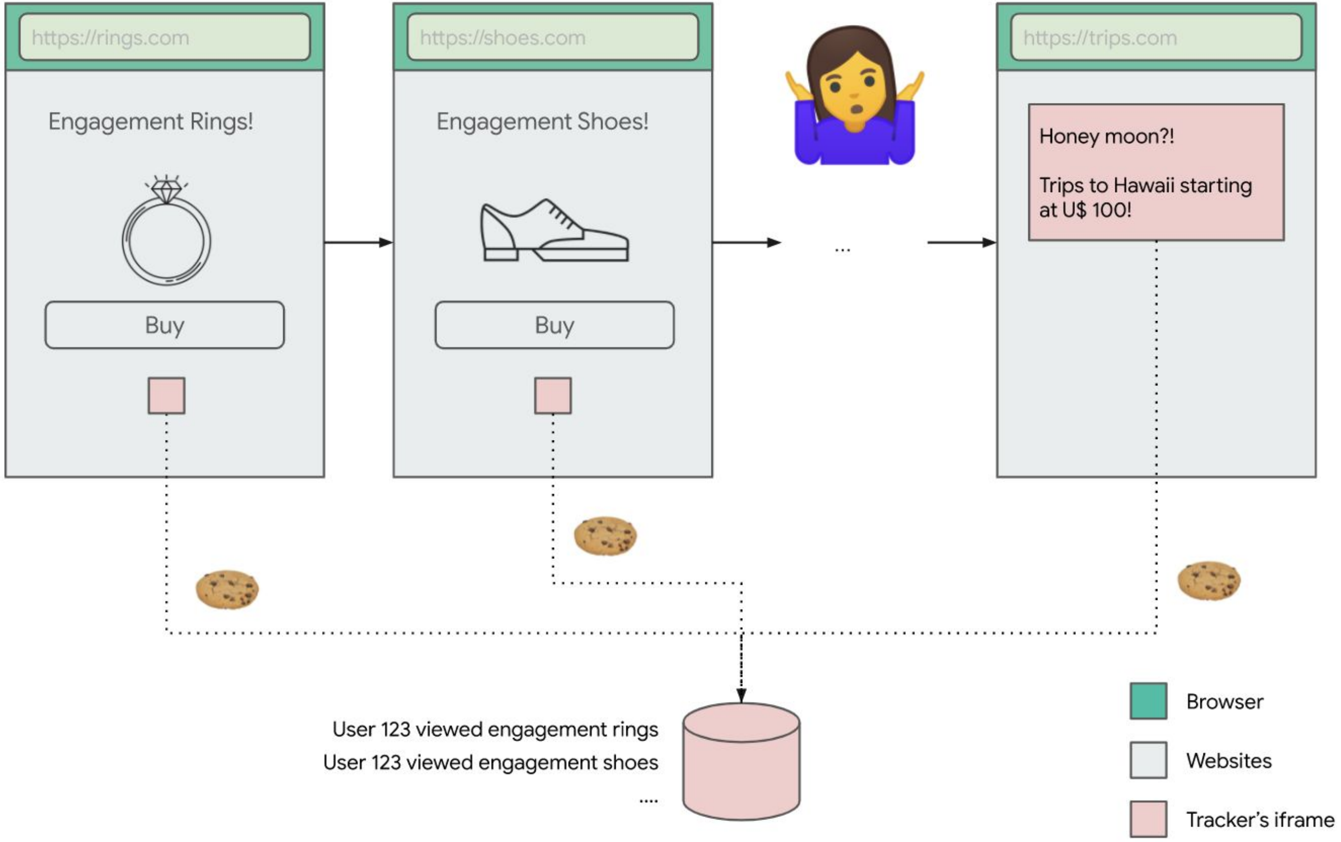
🟩 Browser

□ RP

🟦 IDP

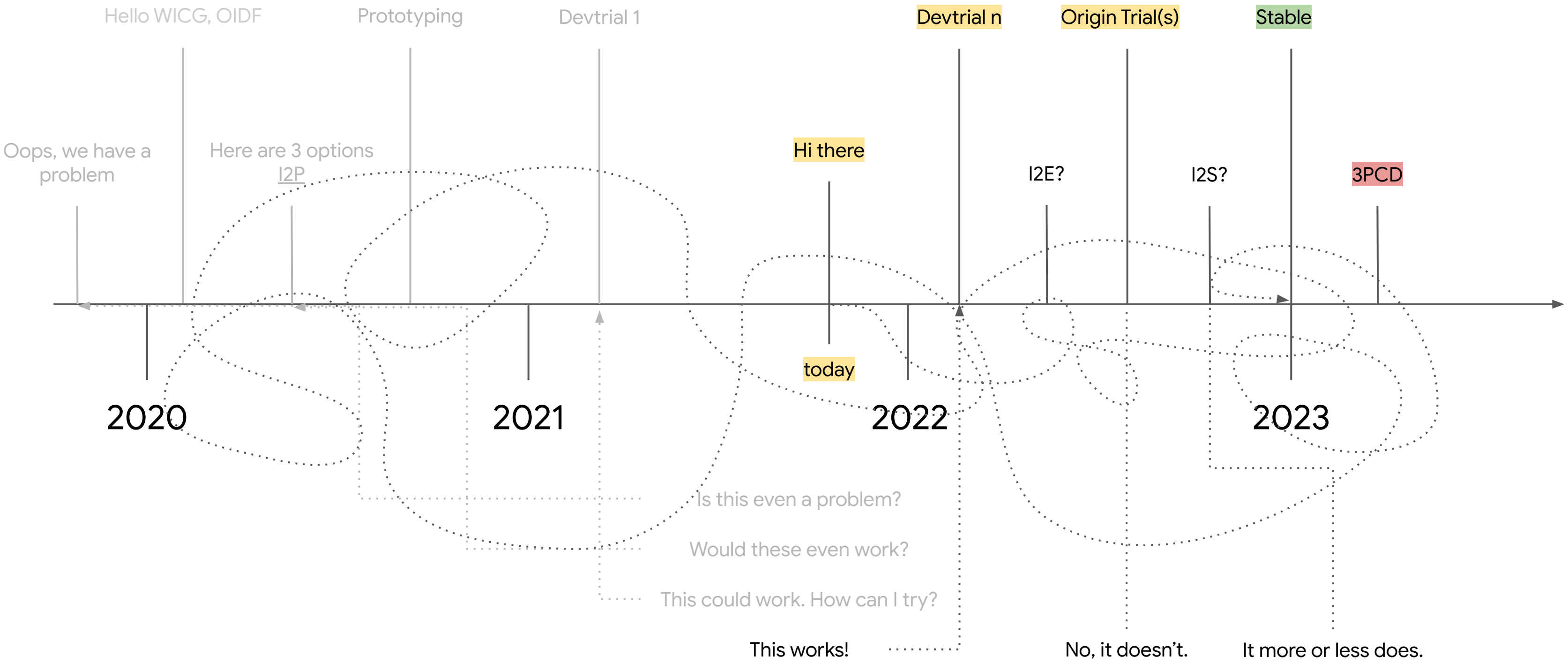
Cookies sound nutritious and delicious?

The Cross-Site Tracking Problem

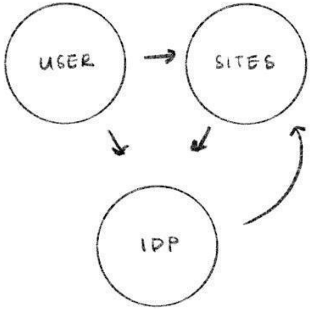


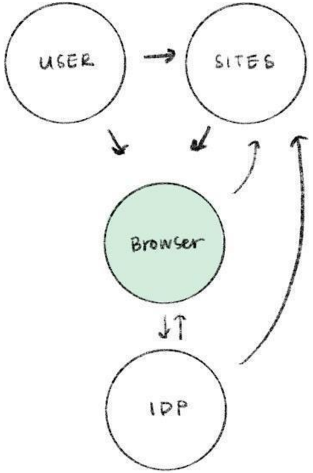
To preserve and elevate
federated identity
for a more private web.

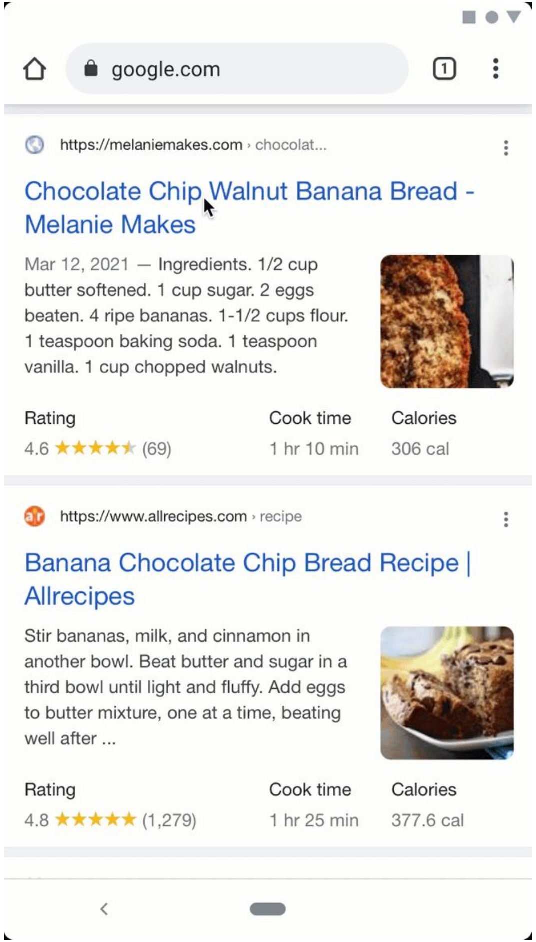
When


Hello WICG, OIDF
Prototyping
Devtrial 1
Devtrial n
Origin Trial(s)
Stable
Oops, we have a problem
Here are 3 options
I2P
Hi there
I2E?
I2S?
3PCD
today
2020
2021
2022
2023
Is this even a problem?
Would these even work?
This could work. How can I try?
This works!
No, it doesn't.
It more or less does.

What


USER
SITES
IDP


USER
SITES
Browser
IDP



google.com

https://melaniemakes.com › chocolat...

Chocolate Chip Walnut Banana Bread - Melanie Makes

Mar 12, 2021 — Ingredients. 1/2 cup butter softened. 1 cup sugar. 2 eggs beaten. 4 ripe bananas. 1-1/2 cups flour. 1 teaspoon baking soda. 1 teaspoon vanilla. 1 cup chopped walnuts.

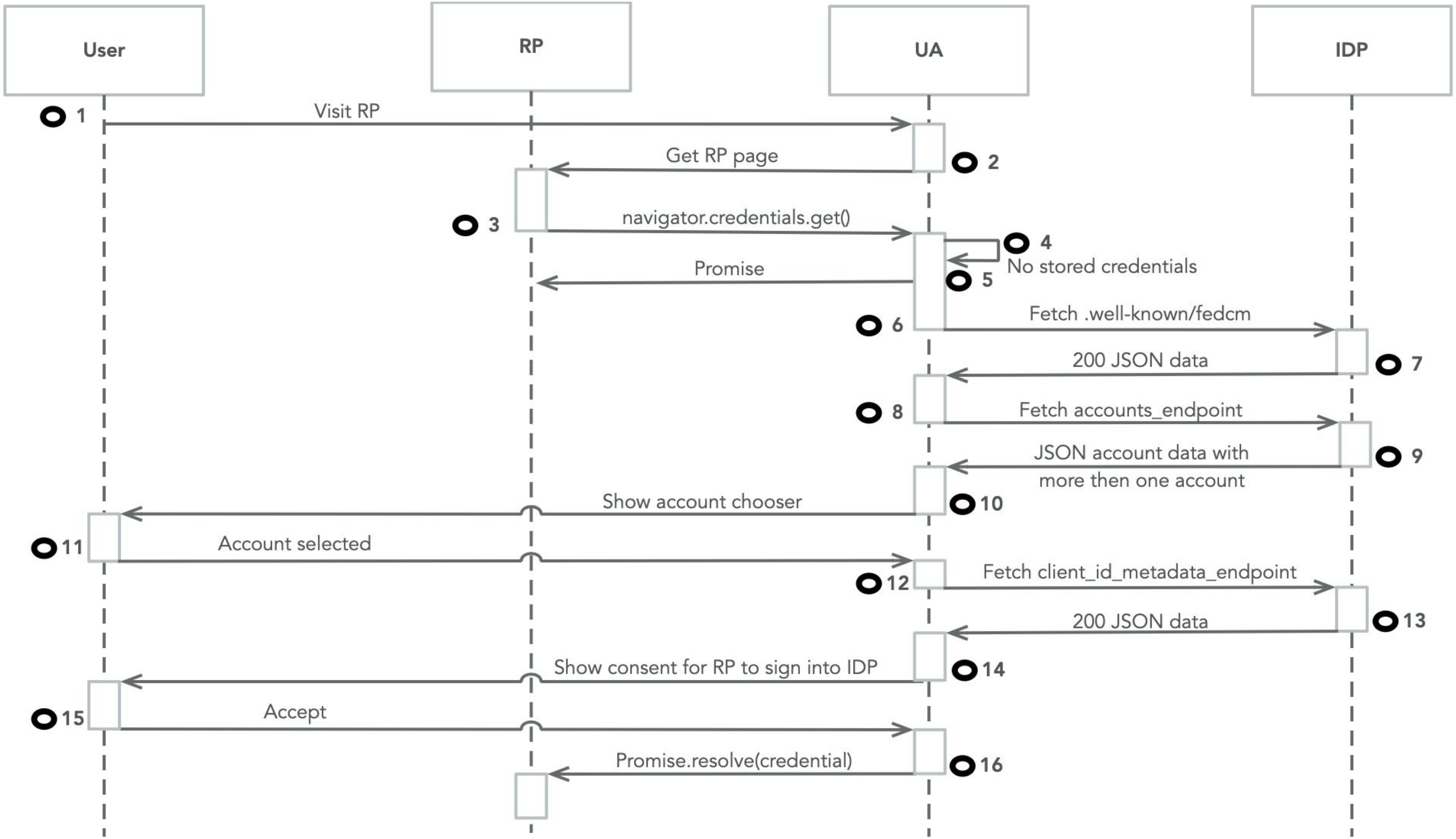
Rating	Cook time	Calories
4.6 ★★★★☆ (69)	1 hr 10 min	306 cal

https://www.allrecipes.com › recipe

Banana Chocolate Chip Bread Recipe | Allrecipes

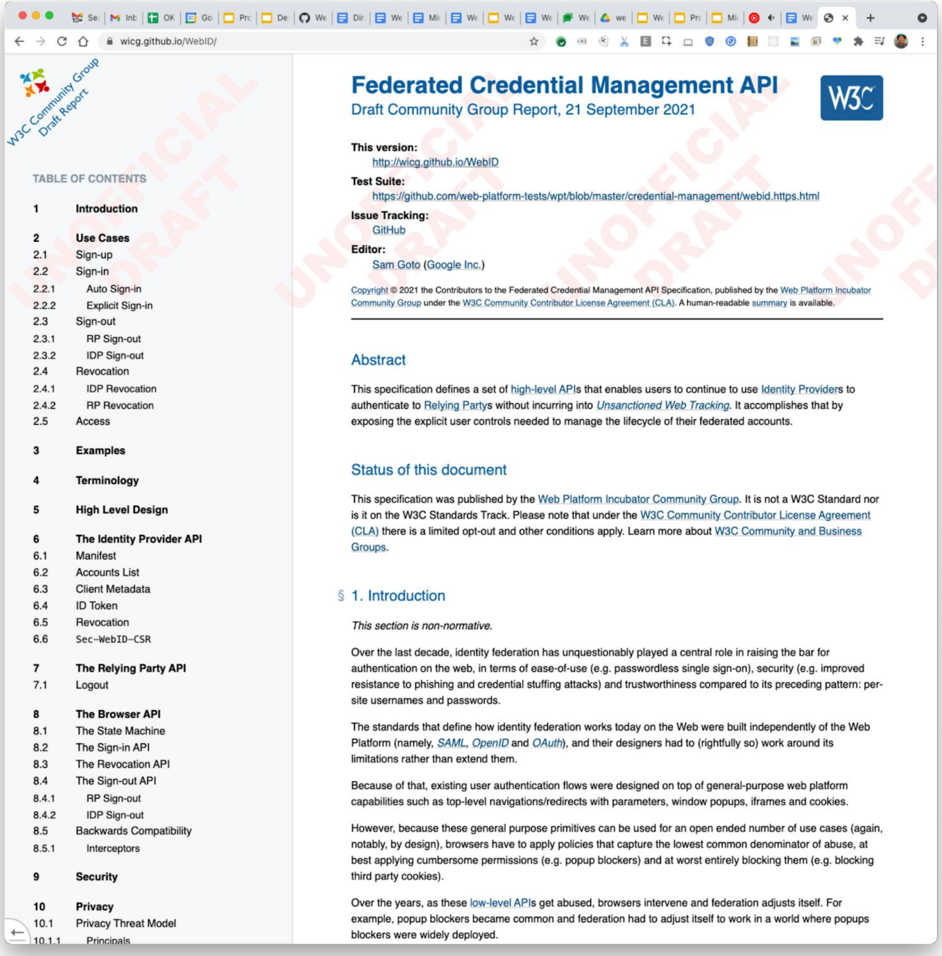
Stir bananas, milk, and cinnamon in another bowl. Beat butter and sugar in a third bowl until light and fluffy. Add eggs to butter mixture, one at a time, beating well after ...

Rating	Cook time	Calories
4.8 ★★★★★ (1,279)	1 hr 25 min	377.6 cal


User
RP
UA
IDP
1
Visit RP
Get RP page
2
3
navigator.credentials.get()
4
No stored credentials
Promise
5
Fetch .well-known/fedcm
6
200 JSON data
7
8
Fetch accounts_endpoint
JSON account data with
more then one account
9
Show account chooser
10
11
Account selected
Fetch client_id_metadata_endpoint
12
200 JSON data
13
Show consent for RP to sign into IDP
14
Accept
15
Promise.resolve(credential)
16

How

https://github.com/fedidcg/use-case-library/issues https://wicg.github.io/FedCM



wicg.github.io/WebID/

W3C Community Group Draft Report

TABLE OF CONTENTS

Federated Credential Management API

Draft Community Group Report, 21 September 2021

This version:
http://wicg.github.io/WebID

Test Suite:
https://github.com/web-platform-tests/wpt/blob/master/credential-management/webid.https.html

Issue Tracking:
GitHub

Editor:
Sam Goto (Google Inc.)

Abstract

This specification defines a set of high-level APIs that enables users to continue to use Identity Providers to authenticate to Relying Partys without incurring into *Unsanctioned Web Tracking*. It accomplishes that by exposing the explicit user controls needed to manage the lifecycle of their federated accounts.

Status of this document

This specification was published by the Web Platform Incubator Community Group. It is not a W3C Standard nor is it on the W3C Standards Track. Please note that under the W3C Community Contributor License Agreement (CLA) there is a limited opt-out and other conditions apply. Learn more about W3C Community and Business Groups.

§ 1. Introduction

This section is non-normative.

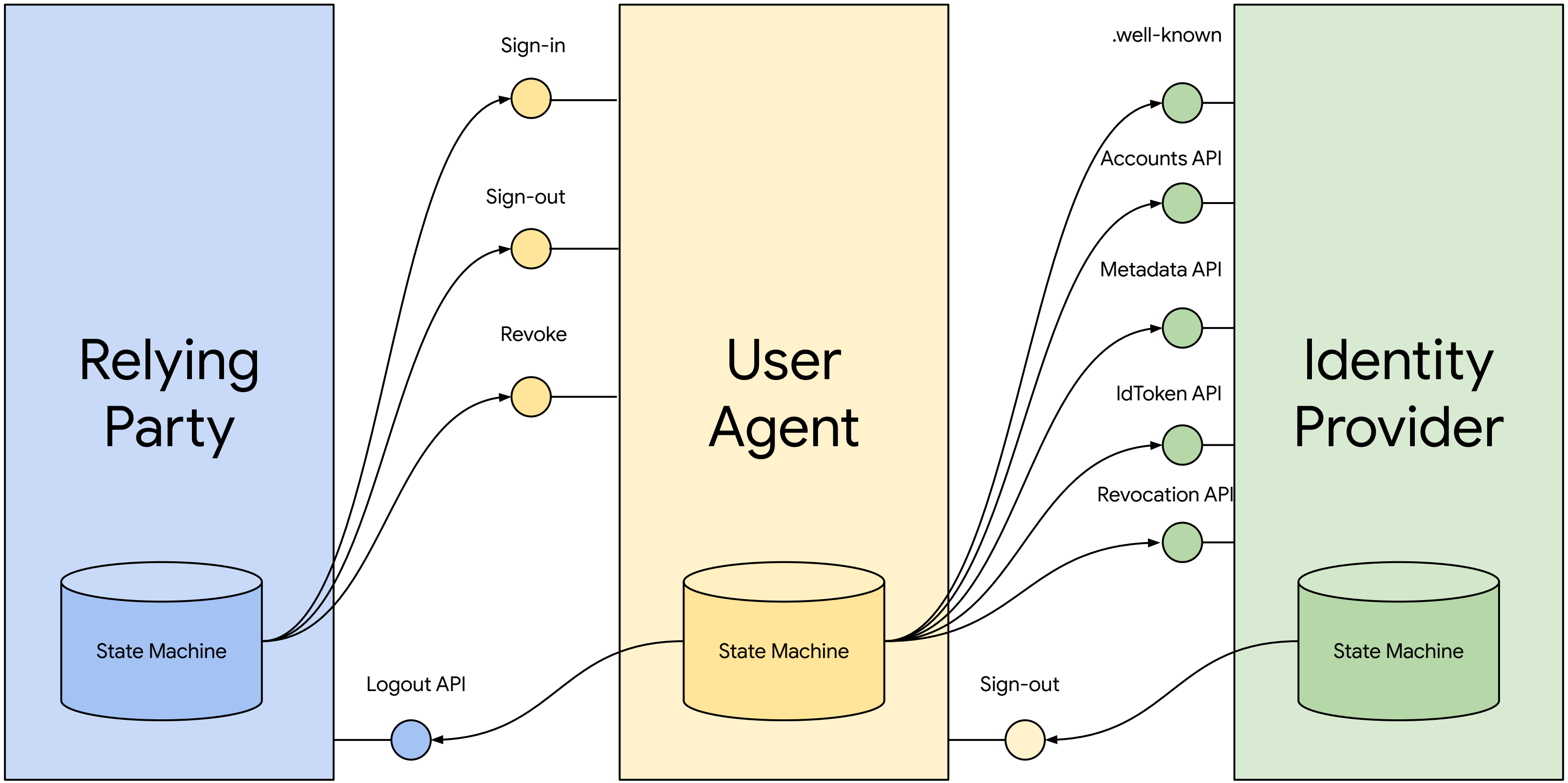
Over the last decade, identity federation has unquestionably played a central role in raising the bar for authentication on the web, in terms of ease-of-use (e.g. passwordless single sign-on), security (e.g. improved resistance to phishing and credential stuffing attacks) and trustworthiness compared to its preceding pattern: per-site usernames and passwords.

The standards that define how identity federation works today on the Web were built independently of the Web Platform (namely, *SAML*, *OpenID* and *OAuth*), and their designers had to (rightfully so) work around its limitations rather than extend them.

Because of that, existing user authentication flows were designed on top of general-purpose web platform capabilities such as top-level navigations/redirects with parameters, window popups, iframes and cookies.

However, because these general purpose primitives can be used for an open ended number of use cases (again, notably, by design), browsers have to apply policies that capture the lowest common denominator of abuse, at best applying cumbersome permissions (e.g. popup blockers) and at worst entirely blocking them (e.g. blocking third party cookies).

Over the years, as these low-level APIs get abused, browsers intervene and federation adjusts itself. For example, popup blockers became common and federation had to adjust itself to work in a world where popups blockers were widely deployed.


Relying Party
State Machine
Sign-in
Sign-out
Revoke
Logout API
User Agent
State Machine
.well-known
Accounts API
Metadata API
IdToken API
Revocation API
Sign-out
Identity Provider
State Machine

O(B) > O(M) > O(100s)

Users

Relying Parties

Identity Providers

Session Management

Curious?
chrome://flags/#webid

(mobile only for now)

TL;DR;

- Third party cookies are being deprecated
- Targeted at sometime in 2023
- We think browser meditation will work to keep federated identity working
- https://github.com/fedidcg/use-case-library/issues
- Federated Credential Management API spec
- chrome://flags/#webid

Questions

dsinclair@chromium.org