Browsers and federation

goto@chromium.org, sso@chromium.org
This deck is shared publicly.
Agenda

1. Premise: general purpose vs special purpose APIs
2. The Problem Space
   - The Classification problem
   - The RP tracking problem
   - The IDP tracking problem
   - The Session State Opacity problem
   - The NASCAR flag problem
3. Early Exploration
   - Principles
   - Deployment considerations
4. Help?
Premise

1. Way more questions than answers.
2. We are still trying to understand the problem space
3. Federation is safer/easier than usernames/passwords
4. General Purpose Affordances, General Purpose permissions
5. Help?
The General Purpose Policy Classification Problem

Create an account with

Sign-in with A

Sign-in with B

or

your@email.com

******

Sign Up

forgot password

Create an account with

Sign-in with A

Sign-in with B

or

Pop up blocked
example.com wants to open a new window to a.com, but we blocked.

allow
The `<iframe>`'s and 3P Cookie Classification Problem

Browser

RP

IDP

your@email.com

******

Sign Up

https://example1.com

Sam Goto

samuelgoto@gmail.com

Sign-in to example.com with IDP

Continue as Sam

forgot password

Sign-in to example.com with IDP

Continue as Sam

forgot password

https://example2.com
The Top Level Navigation and Link Decoration Classification Problem

https://example.com

Create an account with

Sign-in with A

Sign-in with B

or

your@email.com

******

Sign Up

https://a.com

Welcome Sam!

Are you trying to create an account with example.com?

Navigate

Referer: https://example.com

Sam Goto
samuelgoto@gmail.com

navigation callback
?idToken=123

Browser

RP

IDP
The Unintentional RP Tracking Problem

https://example1.com

Sign-in with A

Sign-in with B

https://example2.com

Sign-in with A

Sign-in with B

Welcome Sam!
Are you trying to create an account with example.com?

Sam Goto
samuelgoto@gmail.com

Sam Goto
samuelgoto@gmail.com

global identifiers

https://a.com

Welcome Sam!
Yes
The Unintentional IDP Tracking Problem

Welcome Sam!

Here are the sites you've logged in this week:

- example.com
- a.com
- b.com
- embarrassing.com
- ugh.com
- blargh.com

Yes
The Session State Opacity Problem

Welcome Sam!
Click below to logout!

You are logged out!
(but not really, because, for real, we still have cookies set and can see what’s going on. we’ll show you a login button just to make you feel better. you can “clear cookies”, but you don’t, right?)
The NASCAR Flag Problem

Create an account with

Sign-in with A

Sign-in with B

or

your@email.com

******

Sign Up

Which one did I sign up with?

Browser

RP

IDP

forgot password
The activation intervention point: most identity providers provide an `sdk.js` library that is pulled from the $O(M)$ relying parties. Recompile that, and you'll activate $O(M)$ websites and $O(B)$ users with a flip of a switch.

```
<script src="https://signin.a.com/signin/sdk.js"></script>
```
Cameron’s 7 Laws of Identity

1) User Control and Consent
2) Minimal Disclosure for a Constrained Use
3) Justifiable Parties
4) Directed Identity
5) Pluralism of Operators and Technologies
6) Human Integration
7) Consistent Experience Across Contexts
Mitigating the RP Tracking Problem

Welcome Sam!
Are you trying to create an account with example.com?

Sam Goto
samuelgoto@gmail.com
Sam Goto
samuelgoto@gmail.com
Sam Goto
samuelgoto@gmail.com
Sam G.
asjlkd234@gmail.com
Sam G.
32wer2343@gmail.com

directed identifiers

Browser
RP
IDP

https://example1.com
Sign-in with A
Sign-in with B

https://example2.com
Sign-in with A
Sign-in with B

https://google.com
Yes
Identity-specific Browser API?

https://example.com

Welcome!

Sign-in with A

Sign-in with B

or

Continue as Sam

Identity-specific API gets called by SDKs

Identity-specific Browser UI prevents abuse outside of Auth

Directed Identifiers By Default

Backward compatible IdToken envelope

NAME: Sam G.

EMAIL: Share my email

samuelgoto@gmail.com

Hide my email

Forward to samuelgoto@gmail.com
Welcome Sam!
We got your verified email on record!

If the user grants access, the id token is passed back to the application:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
{
  "iss": "https://accounts.a.com",
  "sub": "110169484474386276334",
  "aud": "https://example.com",
  "name": "Sam",
  "given_name": "Sam",
  "family_name": "G."
  "email": "242423asf390@gmail.com",
  "email_verified": "true"
}
HMACSHA256(
  base64UrlEncode(header) + "." + 
  base64UrlEncode(payload),
  SECRET
)`
Help?

1. Way more questions than answers
2. We are still trying to understand the problem space
3. Federation is safer/easier than usernames/passwords
4. General Purpose Affordances, General Purpose permissions
5. Help?

goto@chromium.org
https://twitter.com/samuelgoto
ANNEX
Potential Data Flow

User → Sign in please?

RP → navigator.credentials.get()

Browser → Fetch .well-known/webid

IDP → Here.

IDP → Build IdToken for RP

Browser → Here.

User → Welcome Sam!

IDP → Here is a signed IdToken.