Browsers, Privacy and Federation

goto@chromium.org, sso@chromium.org

This deck is shared publicly.

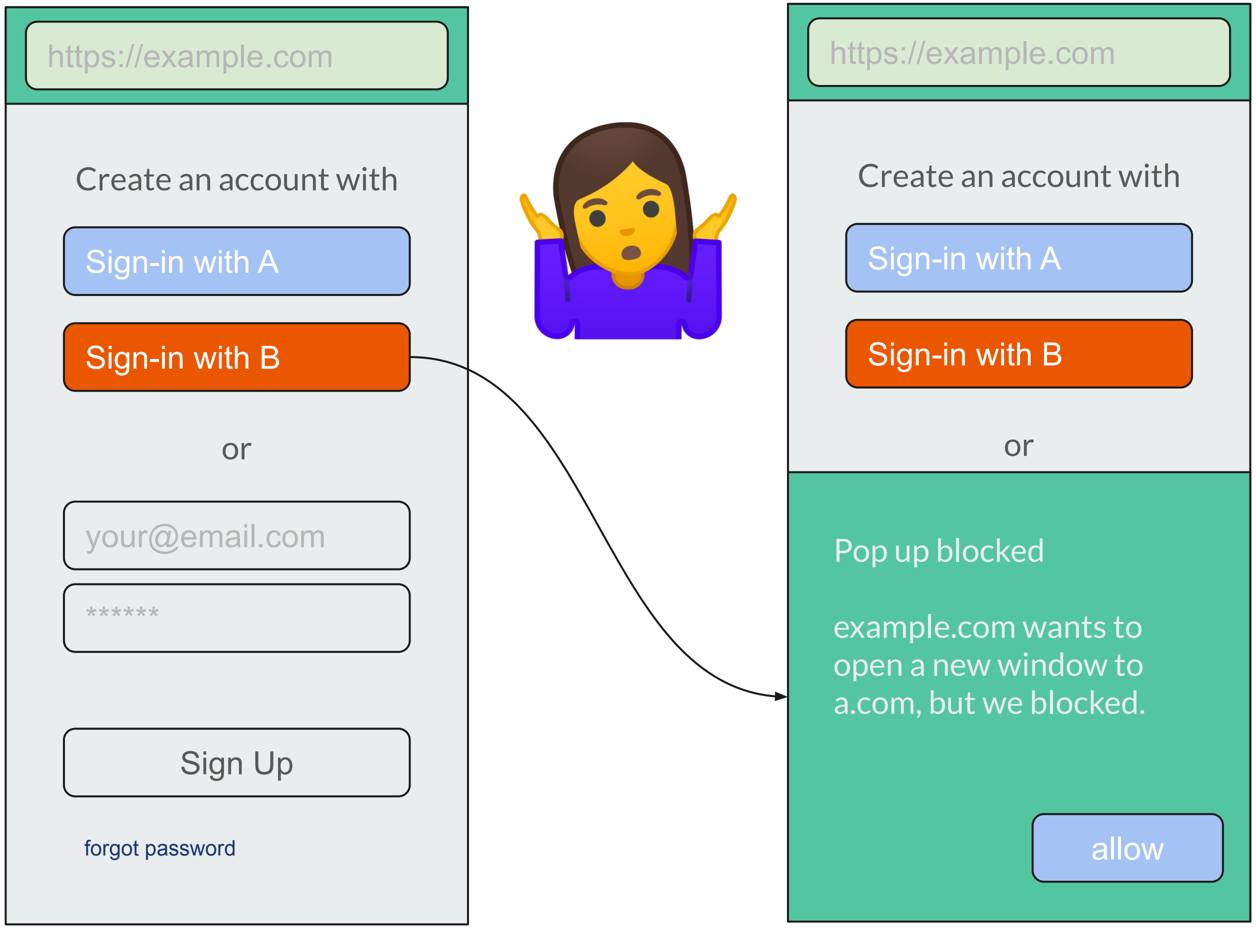
Agenda

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

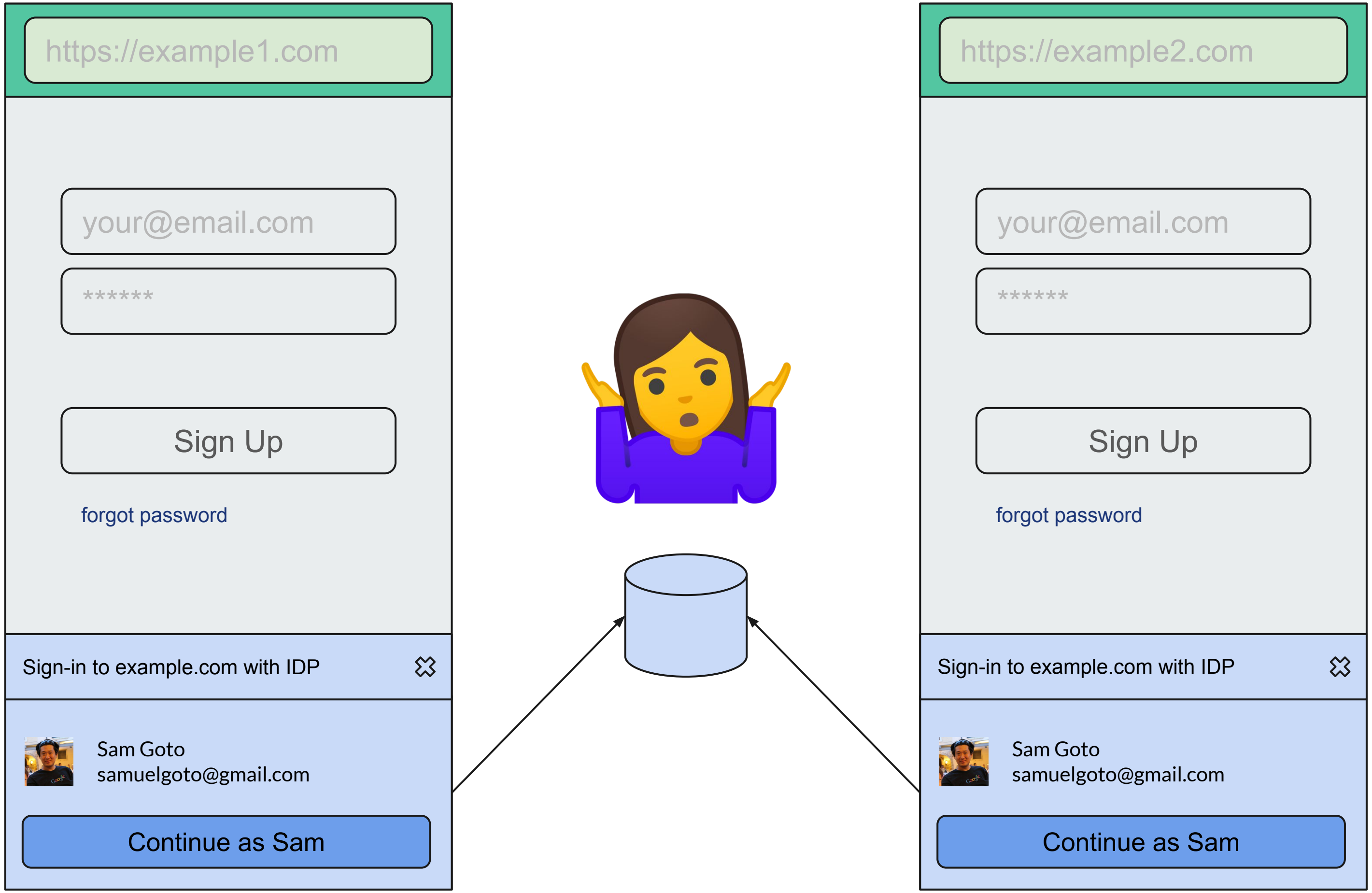
1. Way more questions than answers.
2. We are still trying to understand the problem space
3. Federation is safer/easier than usernames/passwords
4. General Purpose Affordances, General Purpose permissions
5. Help?

The General Purpose Policy Classification Problem



🟩 Browser
⬜ RP
🟦 IDP

The <iframe>s and 3P Cookie Classification Problem

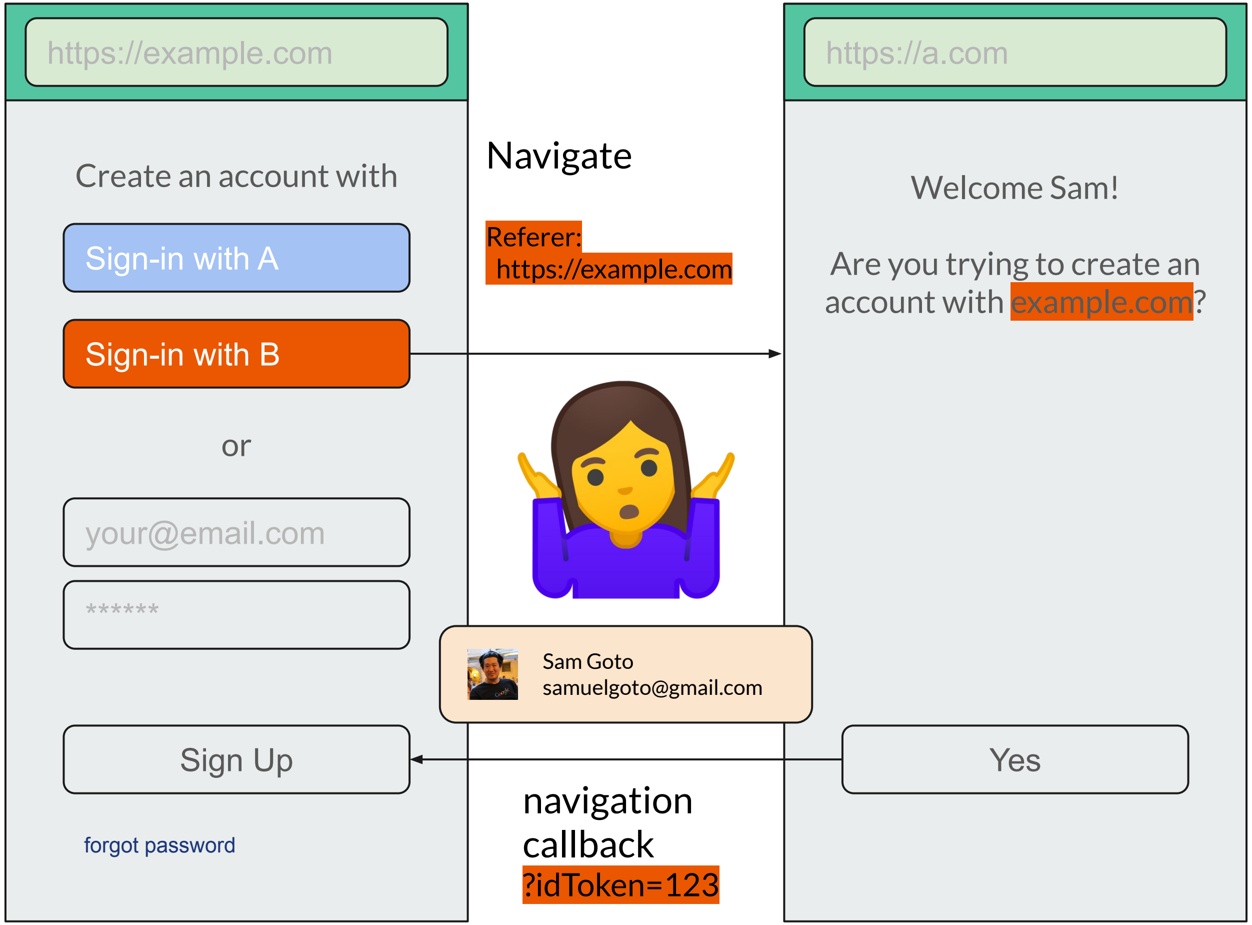


🟩 Browser

⬜ RP

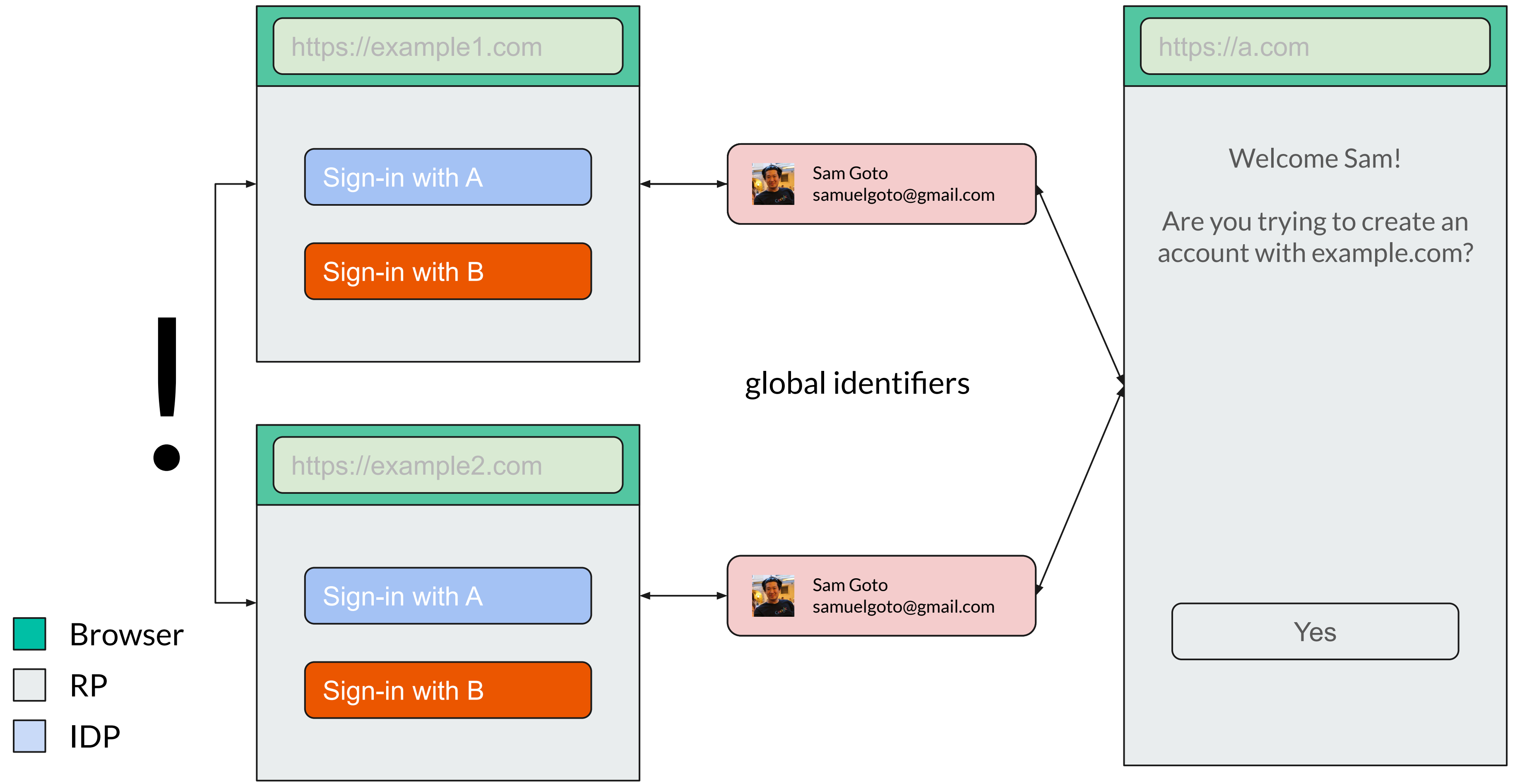
🟦 IDP

The Top Level Navigation and Link Decoration Classification Problem

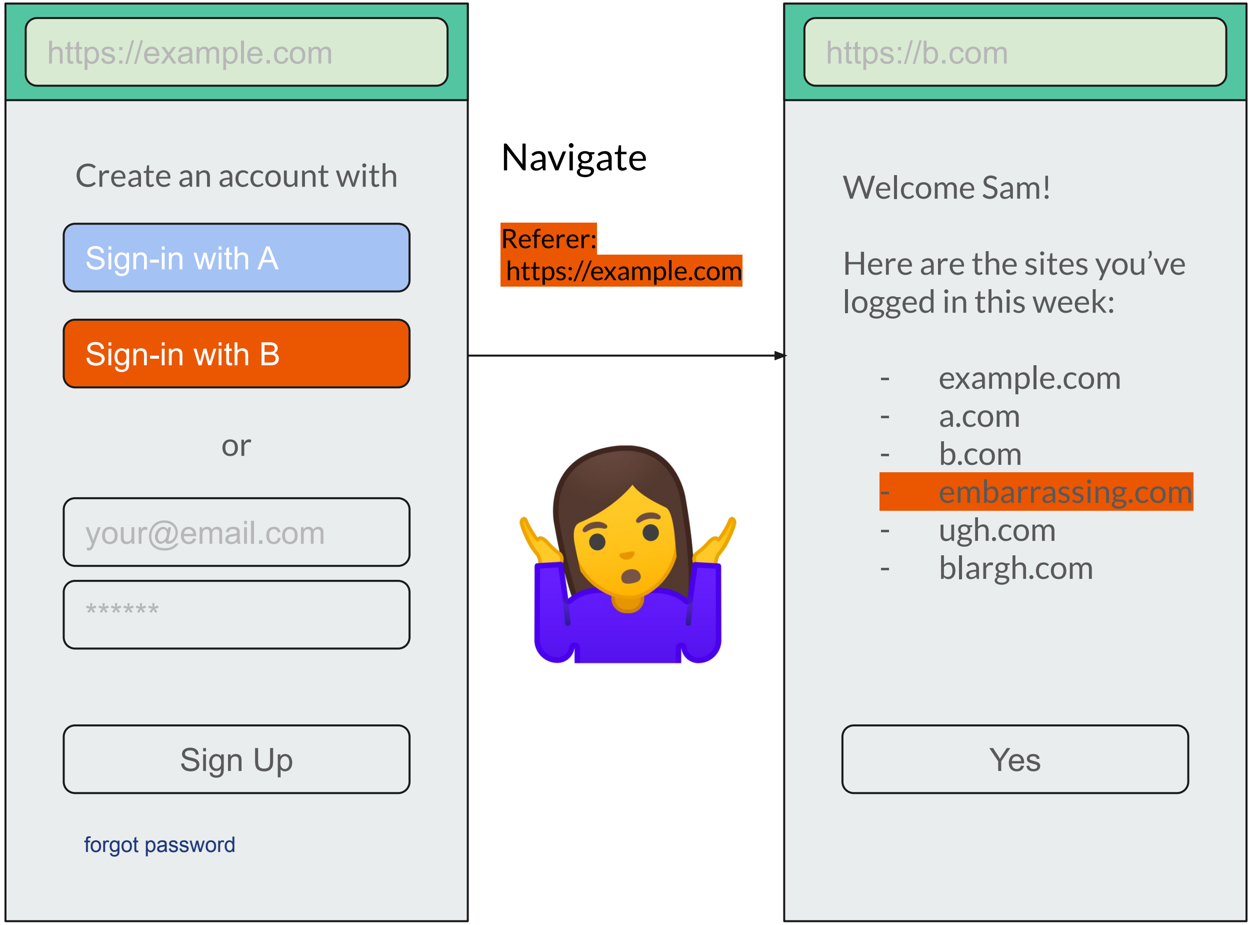


🟩 Browser
⬜ RP
🟦 IDP

The Unintentional RP Tracking Problem



The Unintentional IDP Tracking Problem



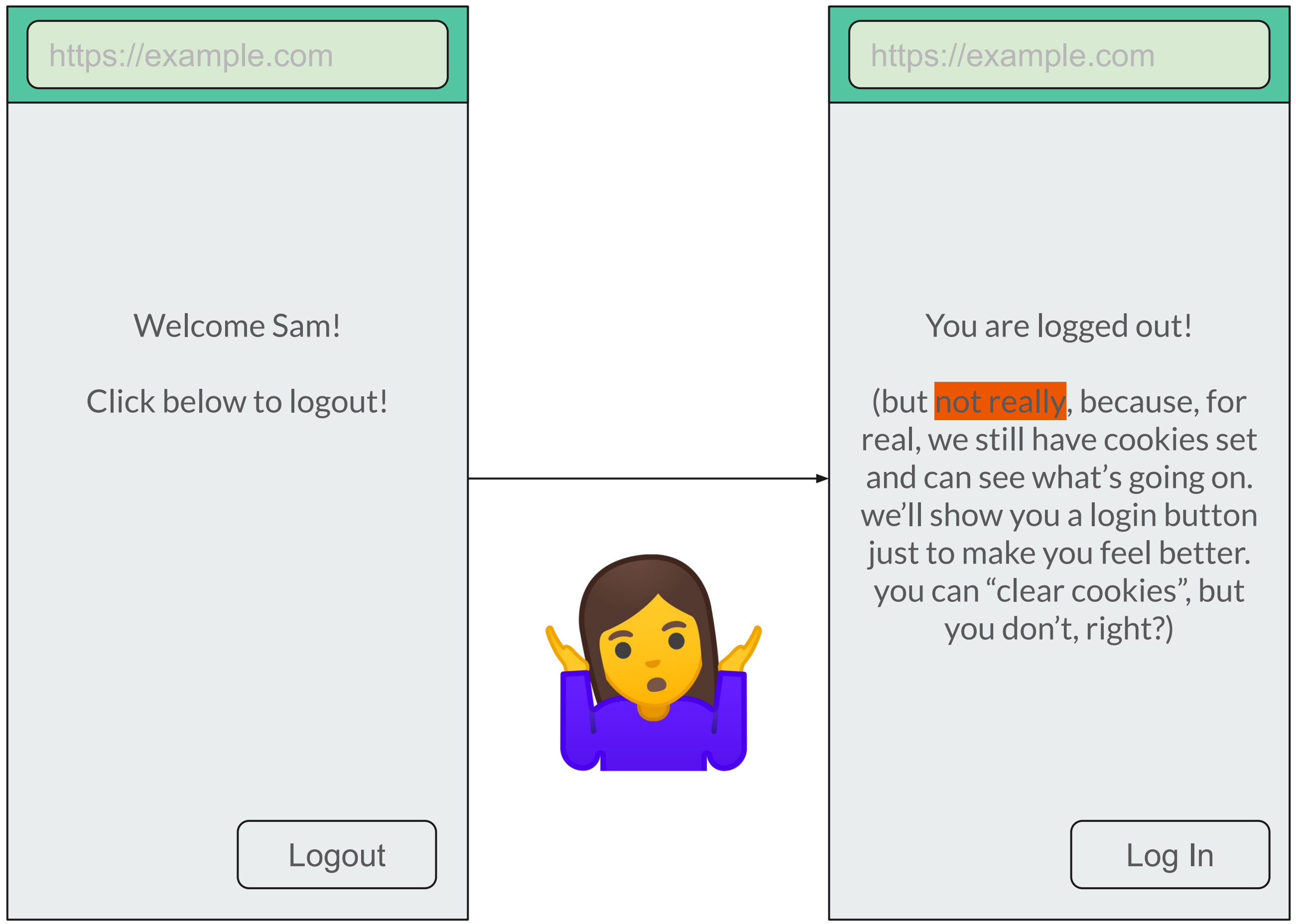
Welcome Sam!

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

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

🟩 Browser
⬜ RP
🟦 IDP

The Session State Opacity Problem



Welcome Sam!

Click below to logout!

You are logged out!

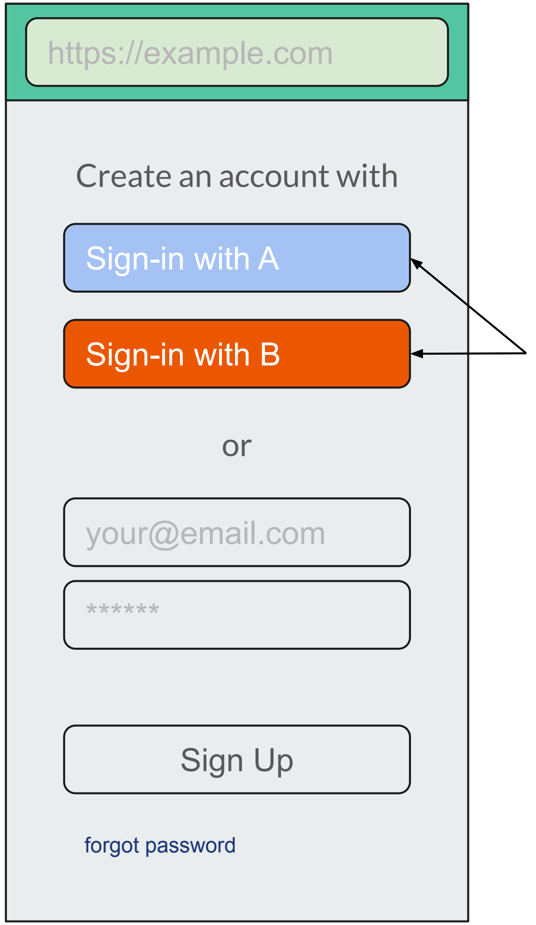
(but not really, because, for real, we still have cookies set and can see what's going on. we'll show you a login button just to make you feel better. you can "clear cookies", but you don't, right?)

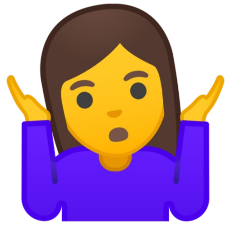
🟩 Browser

⬜ RP

🟦 IDP

The NASCAR Flag Problem





Which one did I sign up with?

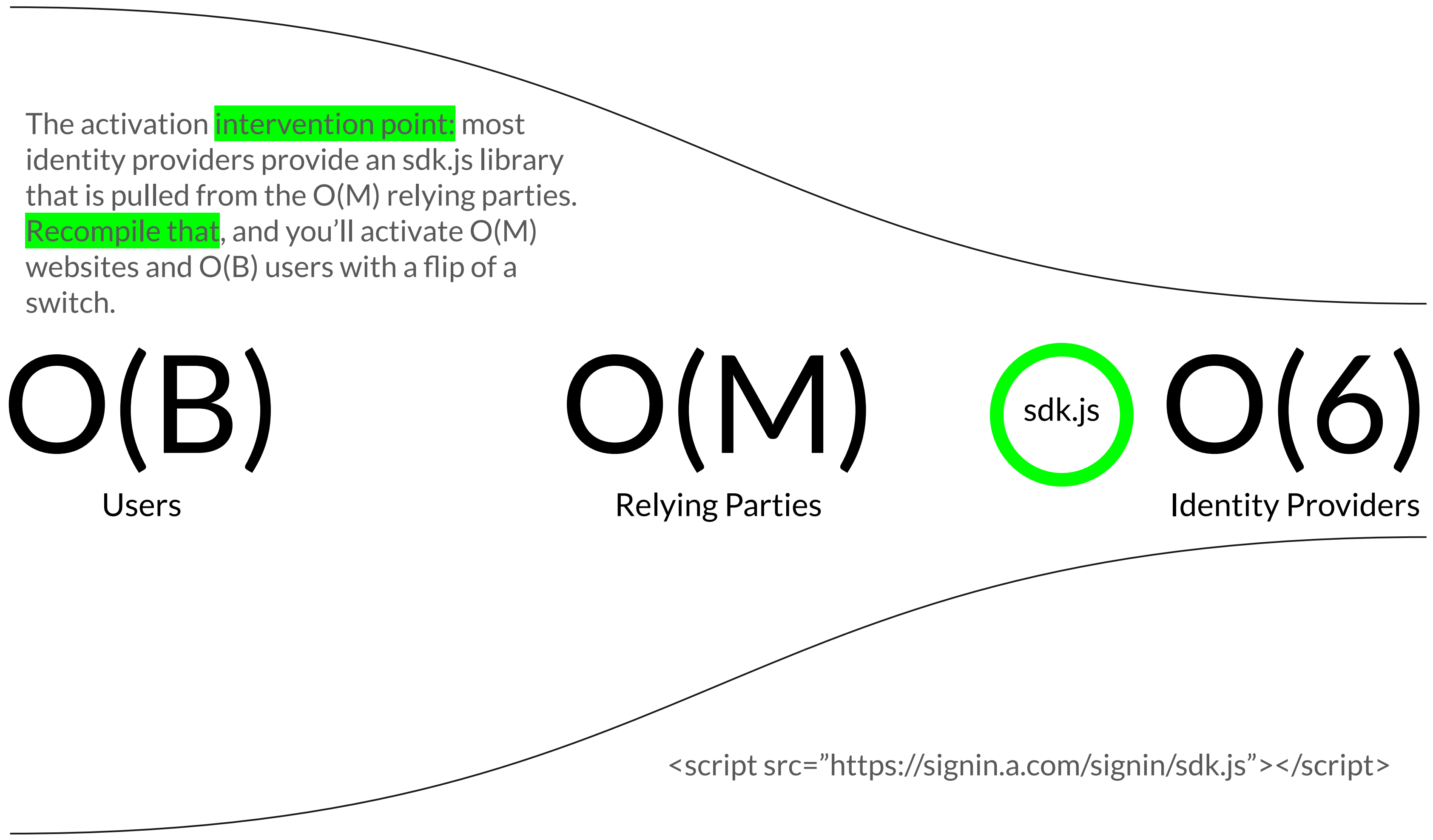
🟩 Browser

⬜ RP

🟦 IDP

Activation Vehicle

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.

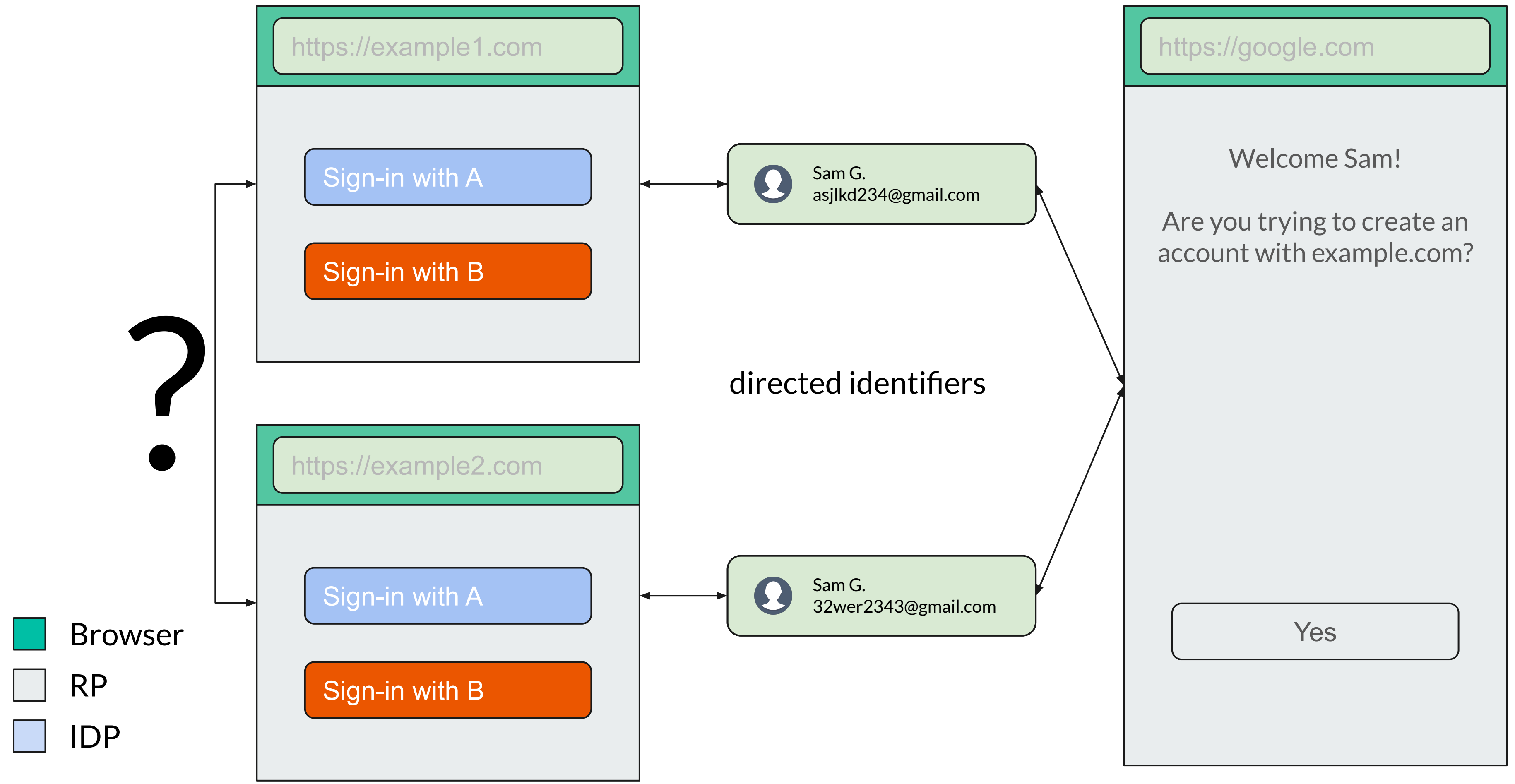


```
<script src="https://signin.a.com/signin/sdk.js"></script>
```

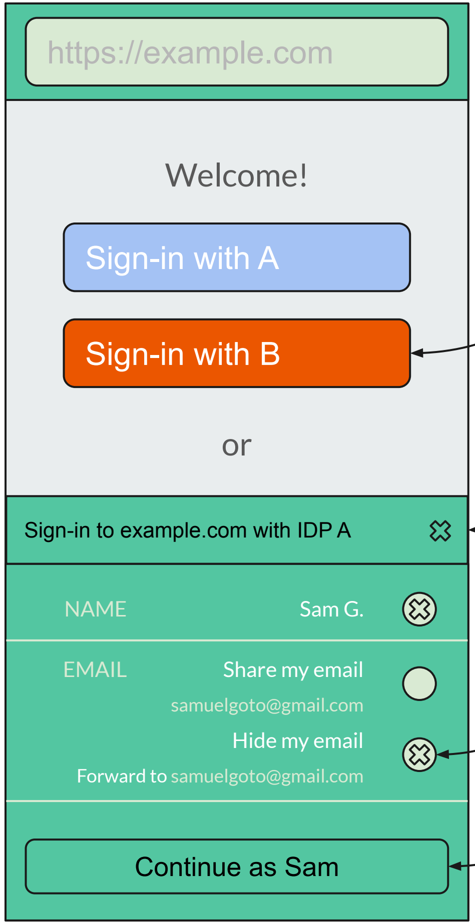
Cameron’s 7 Laws of Identity

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



Identity-specific Browser API?



Identity-specific API gets called by SDKs

Identity-specific Browser UI prevents abuse outside of Auth

Directed Identifiers By Default

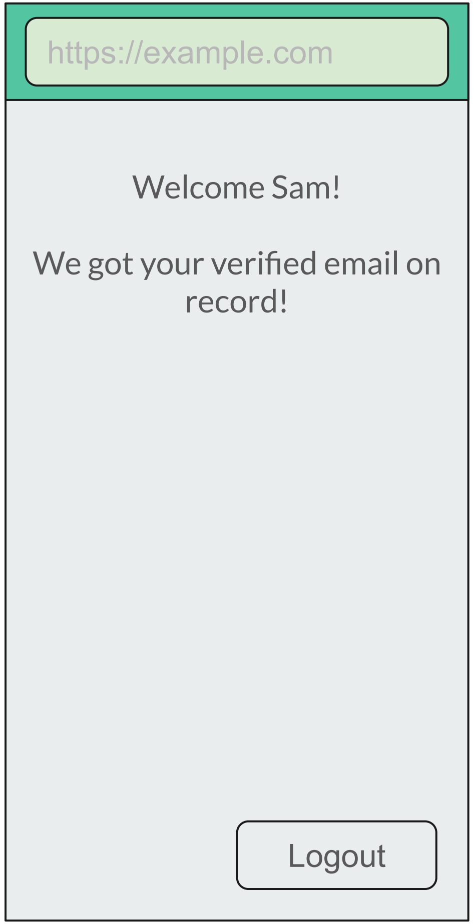
Backward compatible IdToken envelope

🟩 Browser

⬜ RP

🟦 IDP

Backwards Compatibility



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
)
```

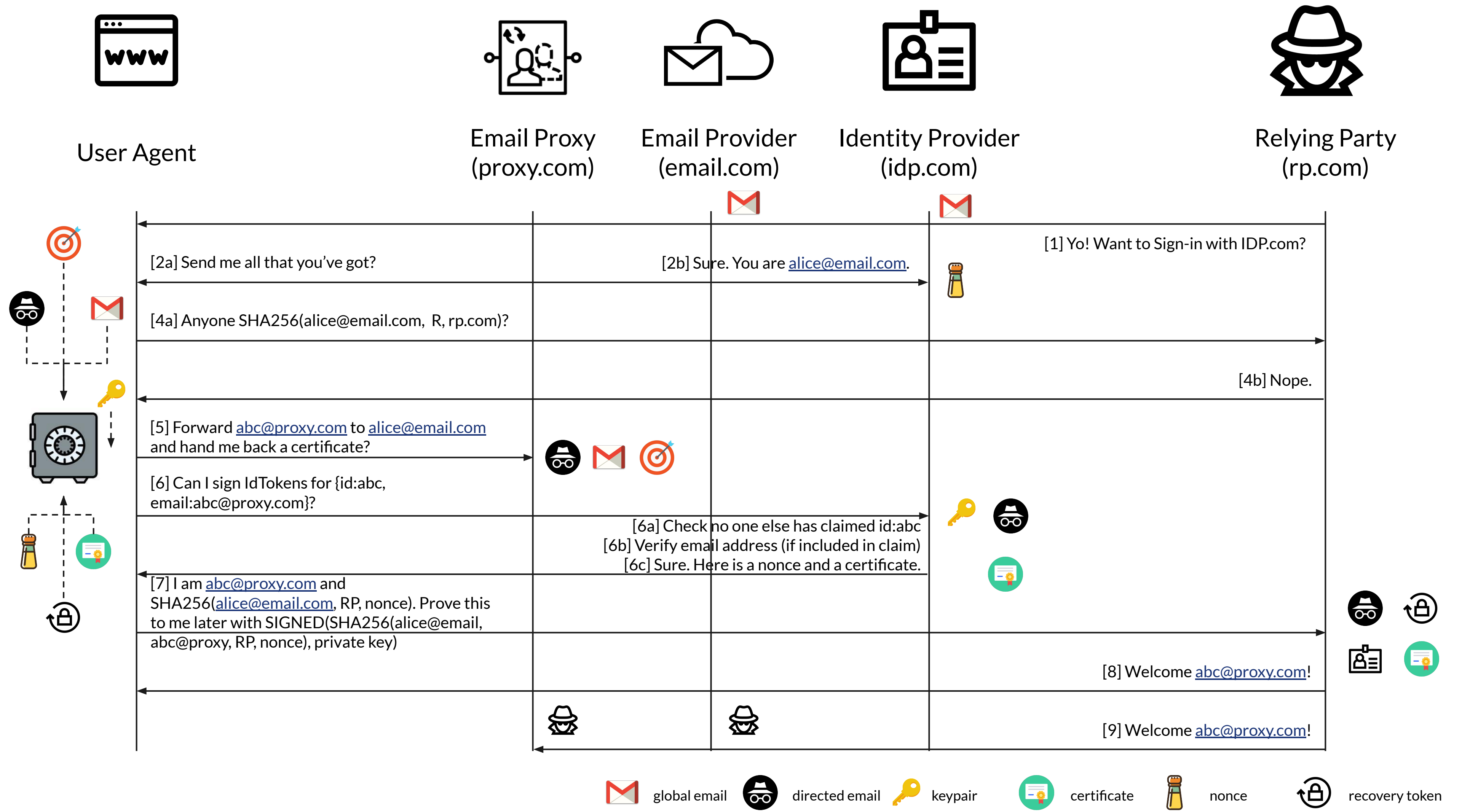
🟩 Browser
⬜ RP
🟦 IDP

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



[1] Yo! Want to Sign-in with IDP.com?

[2a] Send me all that you've got?

[2b] Sure. You are alice@email.com.

[4a] Anyone SHA256(alice@email.com, R, rp.com)?

[4b] Nope.

[5] Forward abc@proxy.com to alice@email.com and hand me back a certificate?

[6] Can I sign IdTokens for {id:abc, email:abc@proxy.com}?

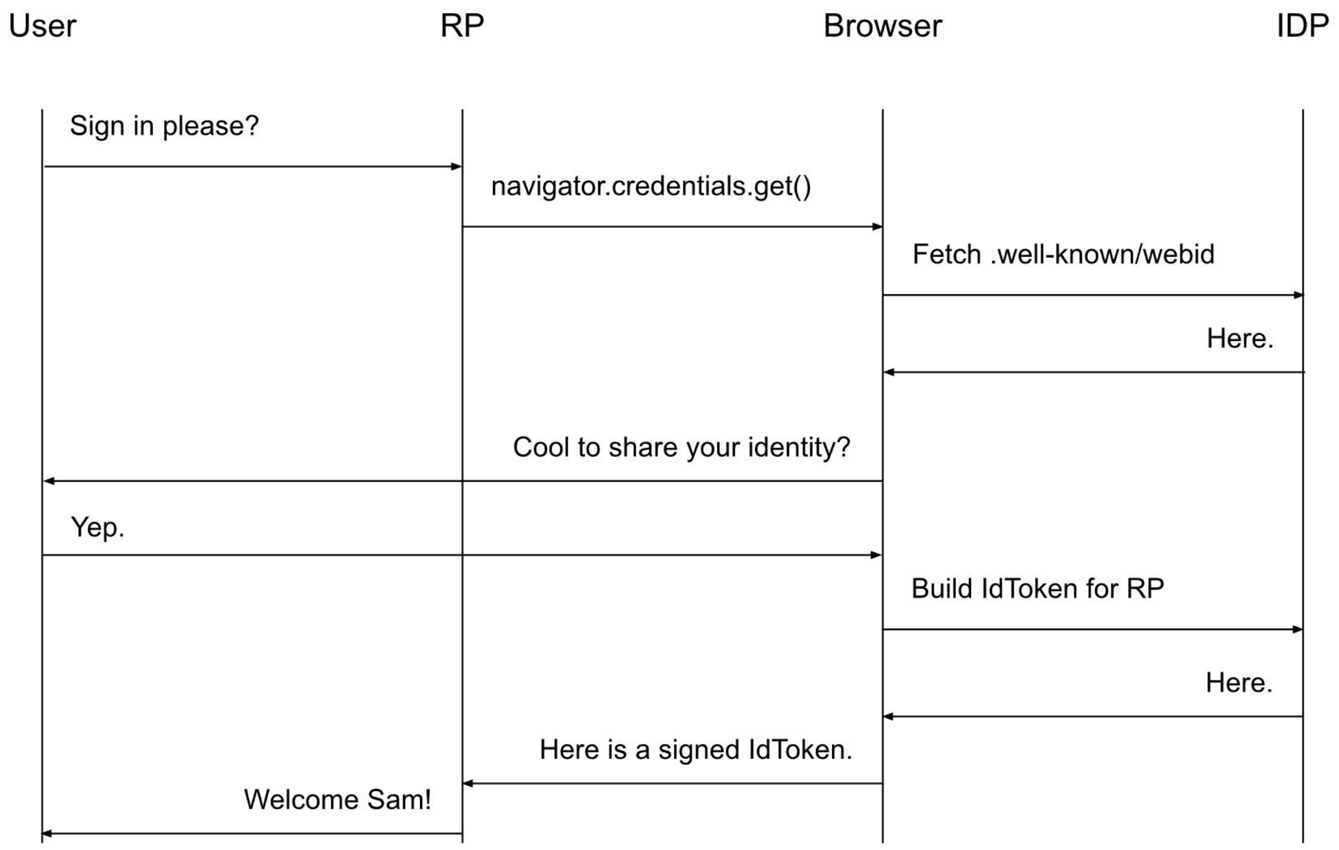
[6a] Check no one else has claimed id:abc
[6b] Verify email address (if included in claim)
[6c] Sure. Here is a nonce and a certificate.

[7] I am abc@proxy.com and SHA256(alice@email.com, RP, nonce). Prove this to me later with SIGNED(SHA256(alice@email, abc@proxy, RP, nonce), private key)

[8] Welcome abc@proxy.com!

[9] Welcome abc@proxy.com!

Potential Data Flow


User
RP
Browser
IDP
Sign in please?
navigator.credentials.get()
Fetch .well-known/webid
Here.
Cool to share your identity?
Yep.
Build IdToken for RP
Here.
Here is a signed IdToken.
Welcome Sam!