



# Browsers and federation

[goto@chromium.org](mailto:goto@chromium.org), [sso@chromium.org](mailto:sso@chromium.org)

**This deck is shared publicly.**





# Agenda

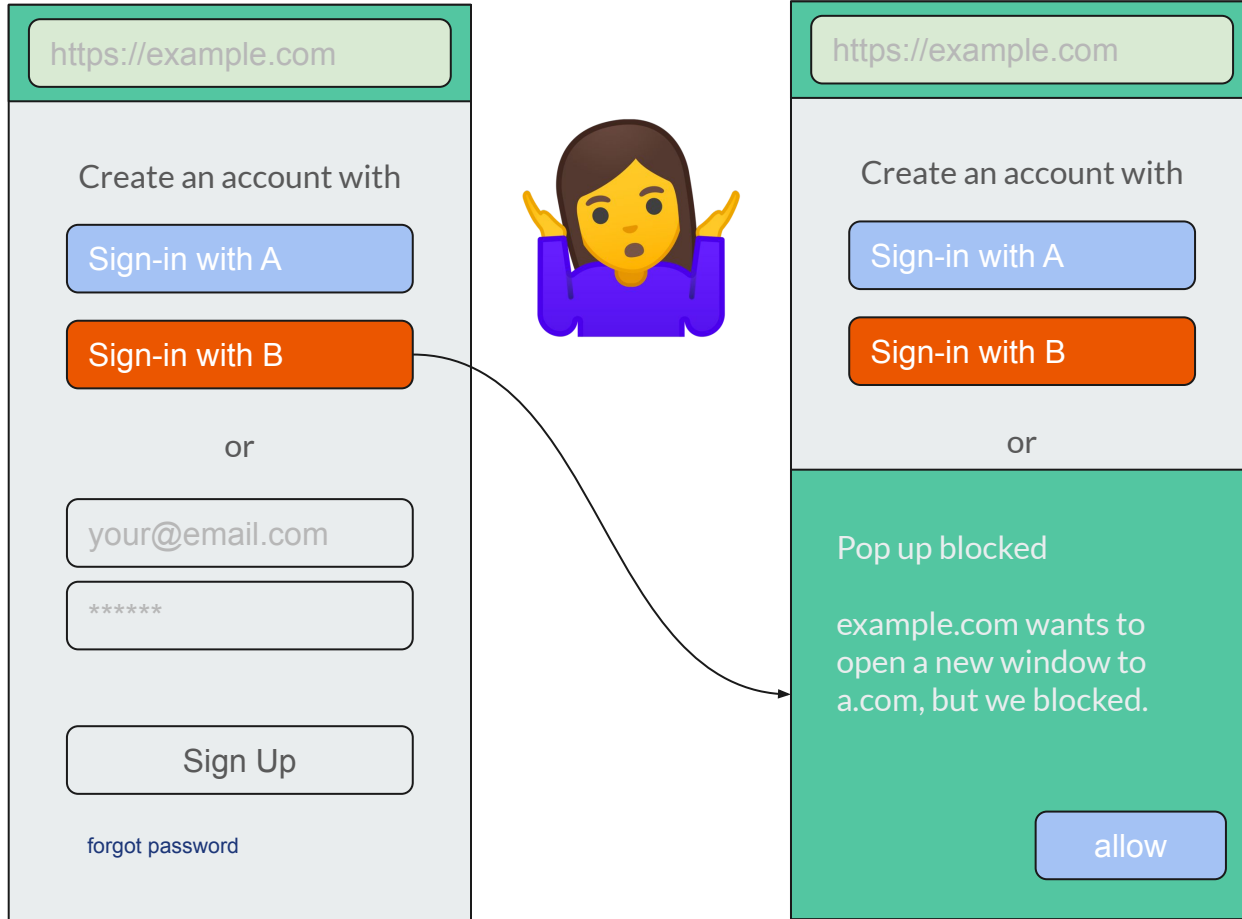
1. Premise: general purpose vs special purpose APIs
2. The Problem Space
  - The [Classification](#) problem
  - The [RP tracking](#) problem
  - The [IDP tracking](#) problem
  - The [Session State Opacity](#) problem
  - The [NASCAR flag](#) problem
3. Early Exploration
  - Principles
  - Deployment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



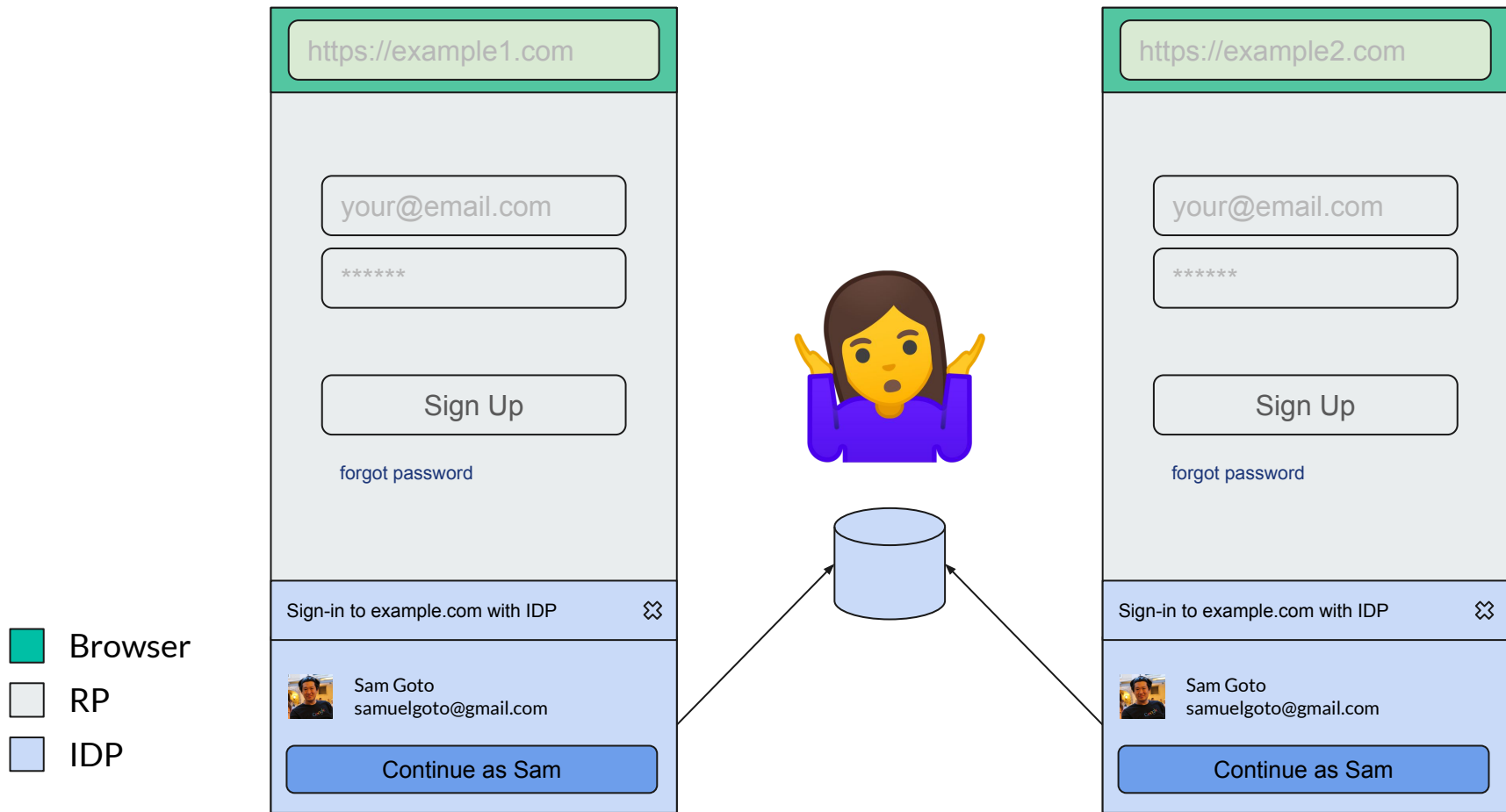
# Premise

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

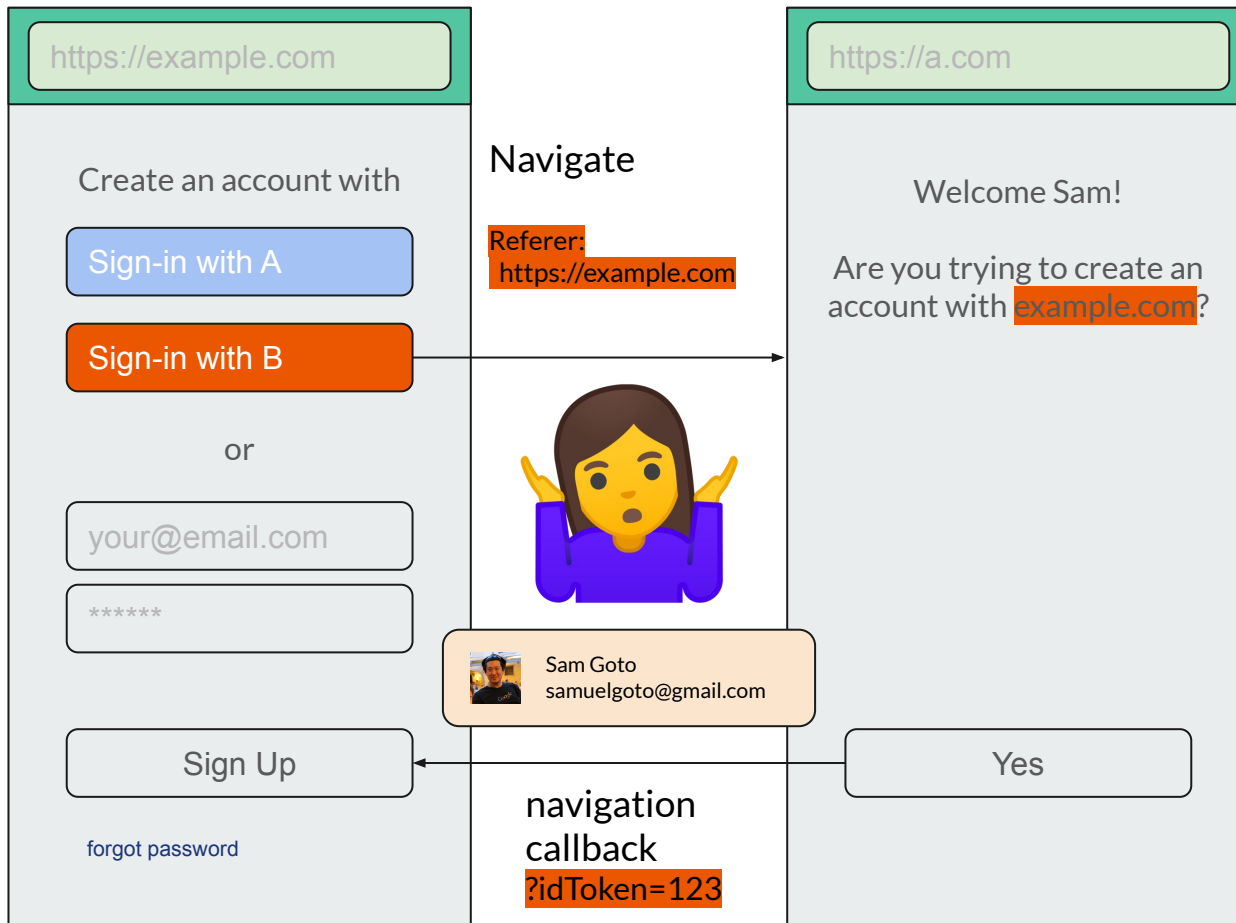


# The <iframe>s and 3P Cookie Classification Problem

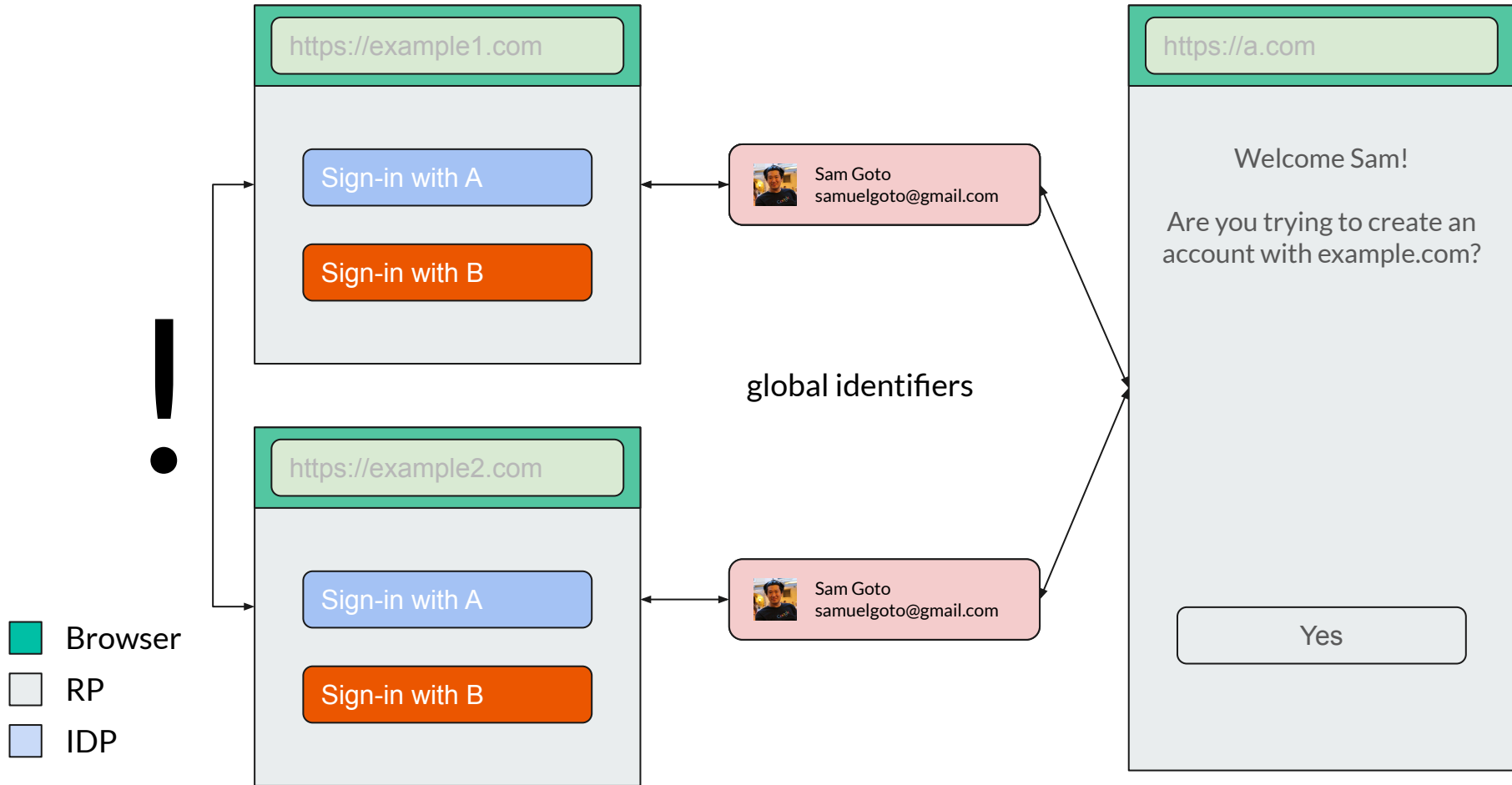


# The Top Level Navigation and Link Decoration Classification Problem

-  Browser
-  RP
-  IDP

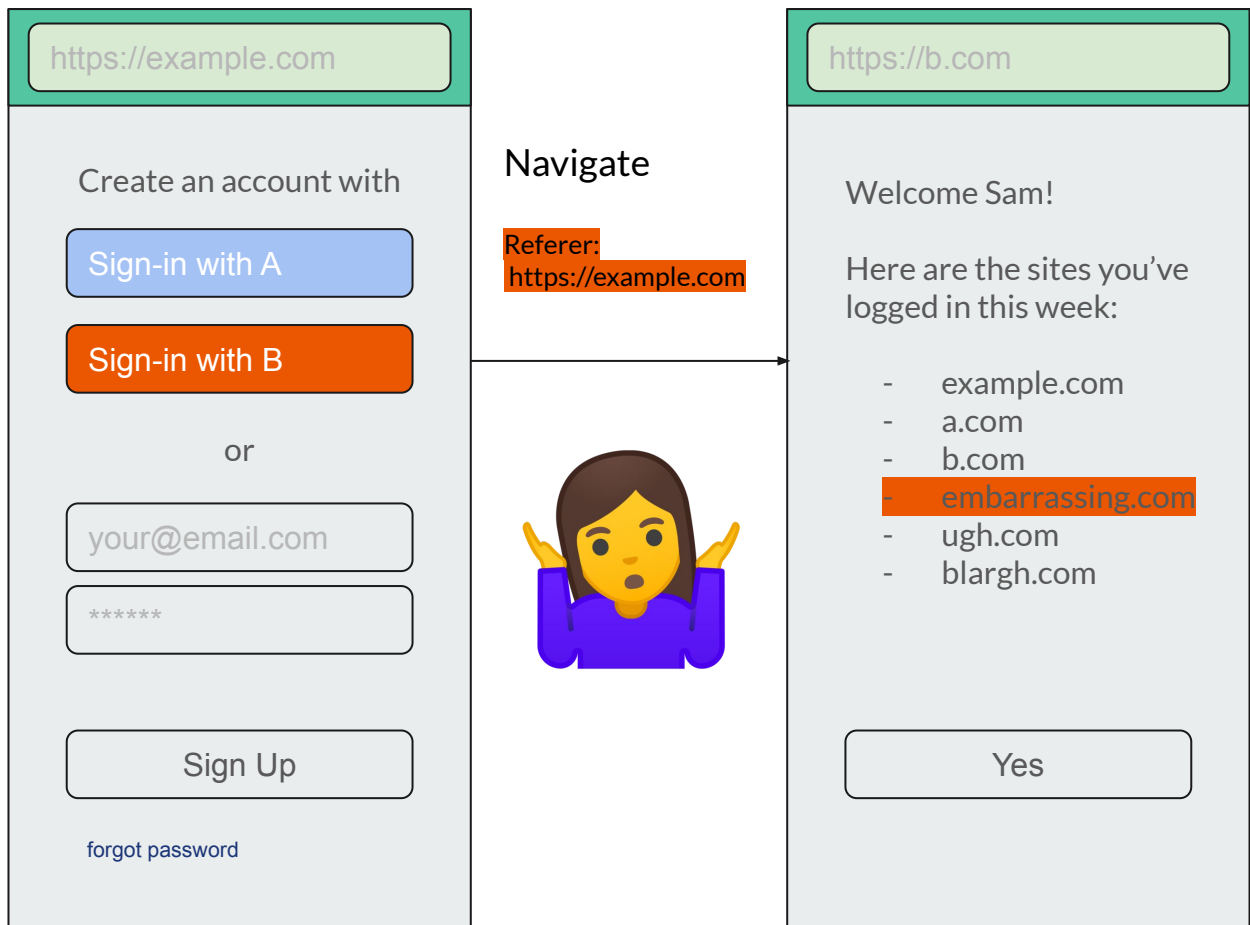


# The Unintentional RP Tracking Problem

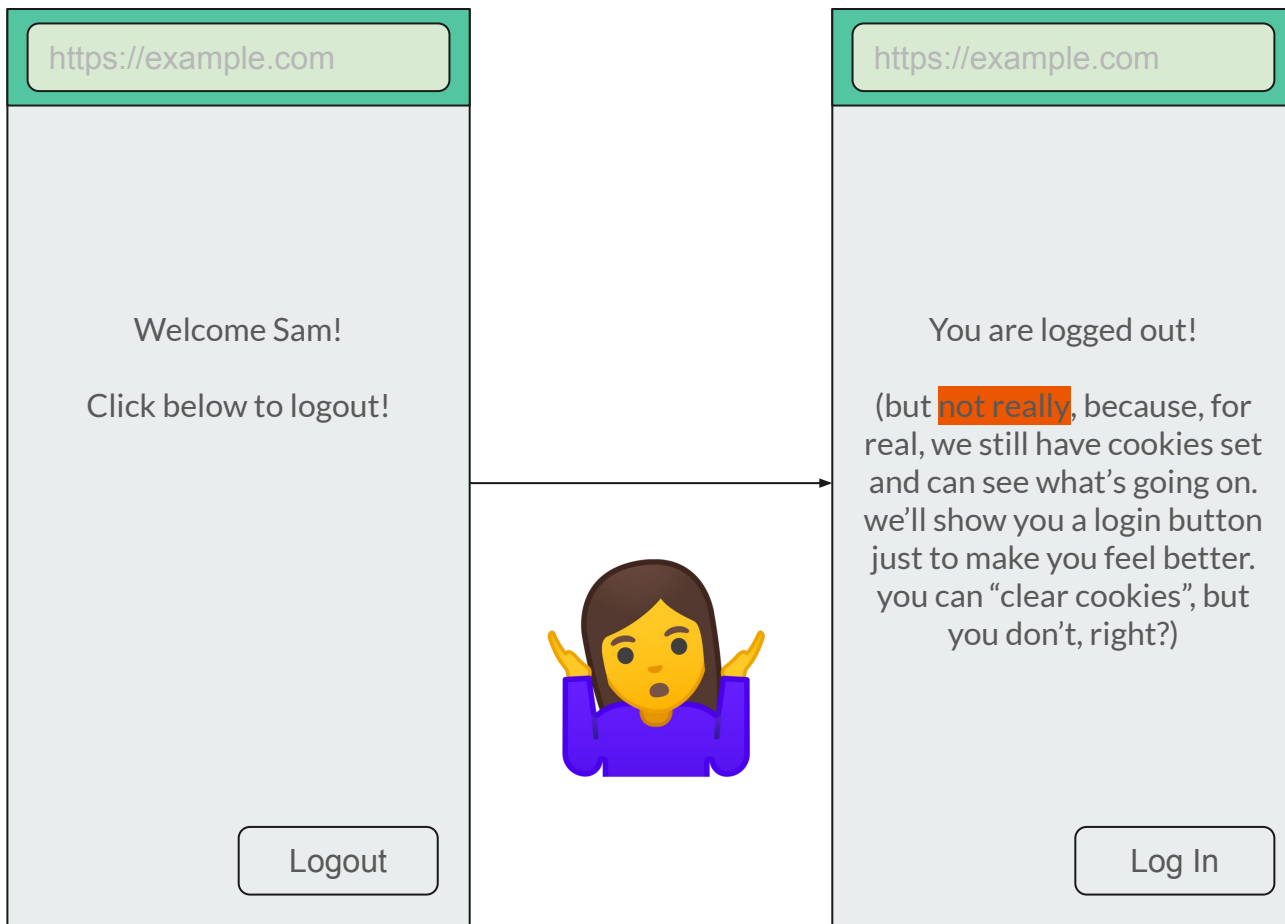




# The Unintentional IDP Tracking Problem



# The Session State Opacity Problem



# The NASCAR Flag Problem

https://example.com

Create an account with

Sign-in with A

Sign-in with B

or

your@email.com




\*\*\*\*\*

Sign Up

forgot password



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.

O(B)

Users

O(M)

Relying Parties



O(6)

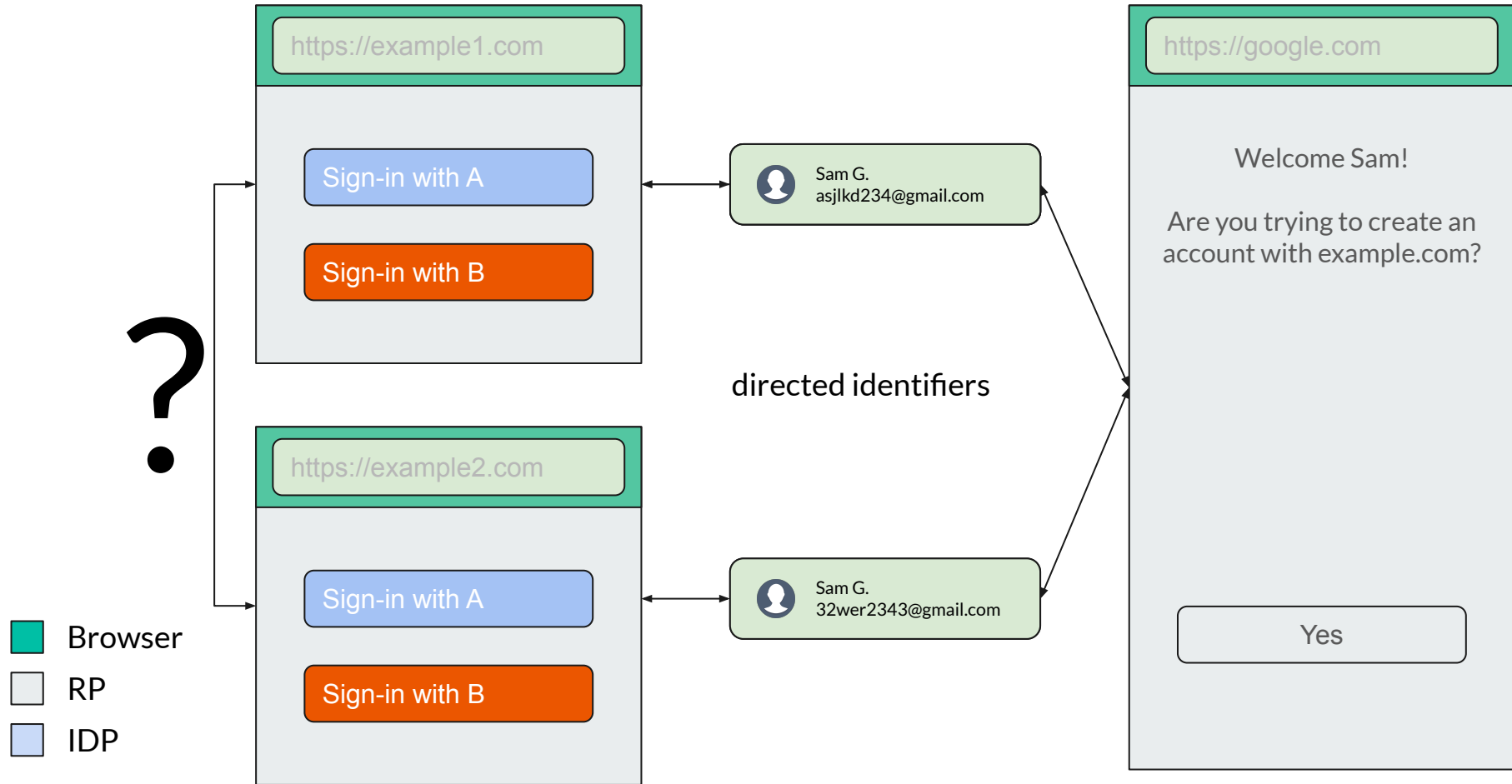
Identity Providers

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

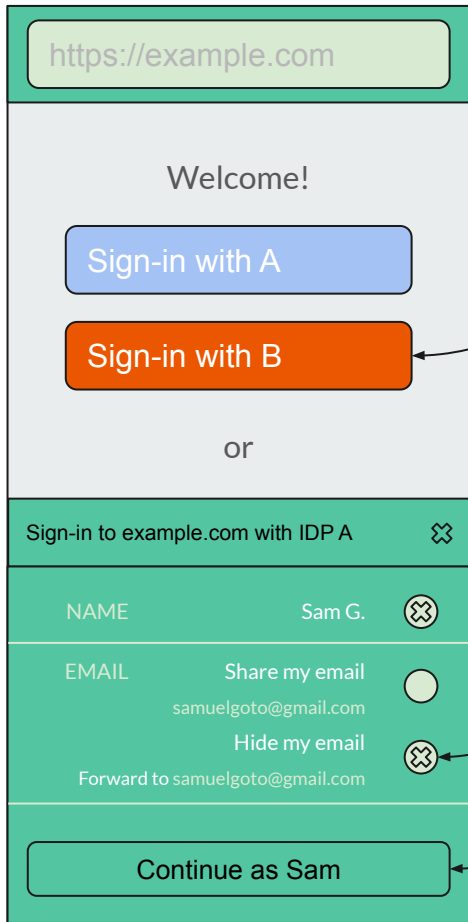
## Cameron's 7 Laws of Identity

- 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

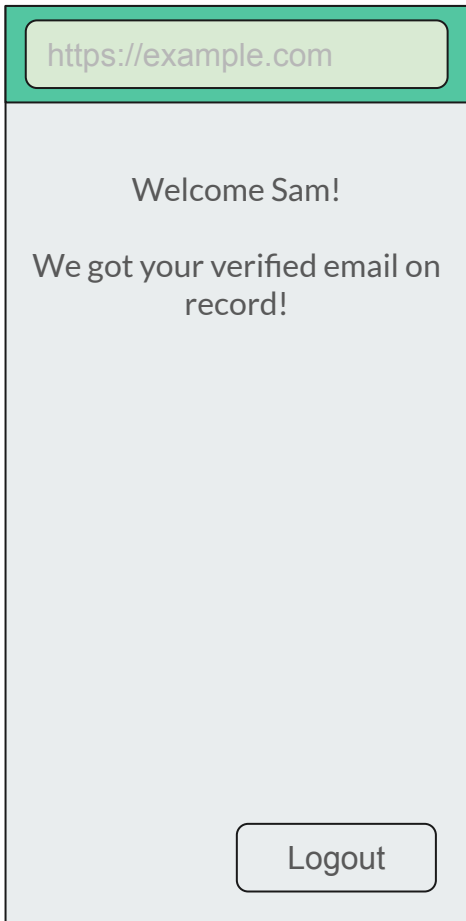
Backward compatible IdToken envelope

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](mailto:goto@chromium.org)

<https://twitter.com/samuelgoto>

**ANNEX**

—

# Potential Data Flow

