



# AccountChooser



Problem:  
Passwords are broken  
for users



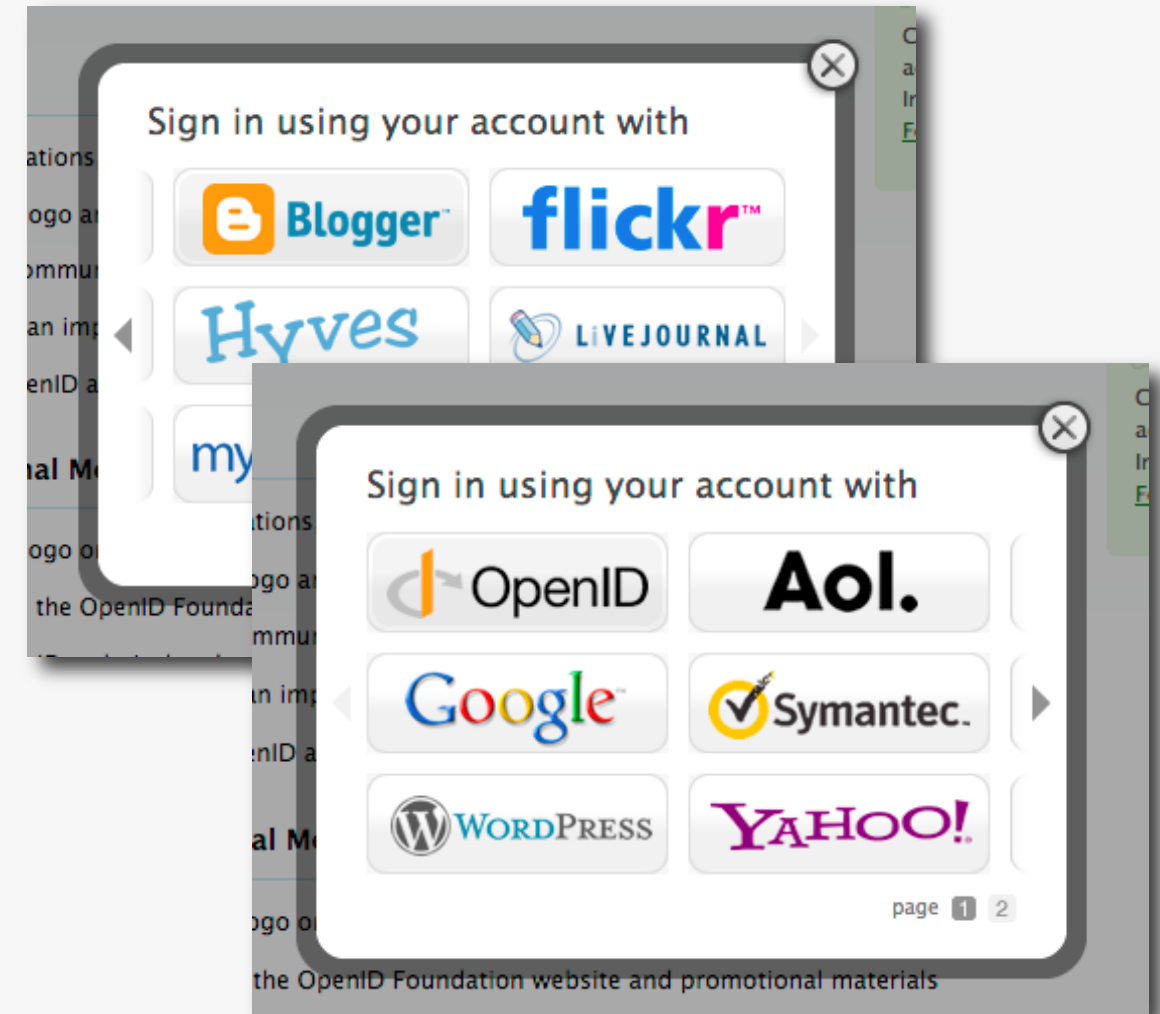
Problem:  
Passwords are broken  
for site operators



# Solution: Enable Federated Identity



The Pinterest login page features the red 'Pinterest' logo at the top. Below it are two buttons: 'Login with Facebook' (dark blue with a white 'f') and 'Login with Twitter' (light blue with a white bird). A horizontal line separates these from the standard login fields: an 'Email' input box, a 'Password' input box, and a 'Login' button. A link for 'Forgot your password?' is located to the right of the 'Login' button.



Problem: The “NASCAR page”












Problem:  
Switching to Federated  
Identity is hard





Sign in to  
favcolor.net

	<b>Tim Bray</b> f tbray@textuality.com	>
	<b>Tim Bray</b> g timbray@gmail.com	>
	<b>Tim Bray</b> g tbray@textuality.com	>
	<b>Tim Bray</b> twbray@live.com live.com	>
	<b>T. Example Bray</b> tim@example.com	>

[Add account](#)

# Solution: AccountChooser





# People Who Use the Web

- ▶ You might sail right through on one click, with an IDP.
- ▶ You might have to tell the IDP “OK”.
- ▶ You might land on an ordinary login page with your email filled in.



# People Who Run Web Sites

- ▶ Fewer keystrokes to sign up new arrivals.  
(Sometimes, one-click sign-up).
- ▶ Get people to the right IDP without any NASCAR visuals.
- ▶ Move to an IDP with less friction.
- ▶ Enterprise people: Less picking the wrong account.



# Demo



# For Developers

```
<script  
  type="text/javascript"  
  src="https://www.accountchooser.com/ac.js">  
  // options go here  
</script>
```



# For Developers

1. *.ac.js* redirects from landing page to [accountchooser.com](https://www.accountchooser.com).

- send along a list of IDPs you like:

```
<script
  type="text/javascript"
  src="https://www.accountchooser.com/ac.js">
  providerIds: [
    "facebook.com", "aol.com", "google.com"
  ]
</script>
```



# For Developers

*2.ac.js* POSTs to your account-status:

- email address
- IDP (from your list, if provided)
- Photo URL (if AccountChooser has it)
- Display-name (if AccountChooser has it)



# For Developers

## 3. Your account-status returns JSON:

- {"registered":true} - sends them to the login page.
- {"registered":false} - sends them to the signin page.
- {"authUrl":<url>} - sends them to an IDP.





# For Developers

## 4. After signin, update *ac.js*:

```
<script
  type="text/javascript"
  src="https://www.accountchooser.com/ac.js">
storeAccount: {
  "email":      ...
  "providerId": ...
  "photoUrl":   ...
  "displayName": ...
}
</script>
```



# For Developers

## 5. There is no step 5!



# For Developers

1. *.ac.js* redirects from landing page to [accountchooser.com](http://accountchooser.com),
2. *.ac.js* POSTs to your account-status,
3. Your account-status returns JSON,
4. After signin, update *ac.js*.



# First step to Federated ID

- ▶ AccountChooser looks like a central site that remembers the ways you log in.
- ▶ Actually it's HTML5 storage in your browser; no data leaves your computer.
- ▶ It remembers emails, display names, picture URLs, and identity providers.
- ▶ That's all!



So... we won, right?



# The AccountChoser Problem

- ▶ Today: Everyone's AccountChooser is empty
- ▶ So: Developer tries it, and nothing happens
- ▶ Theory: It fills up organically as RPs use it
- ▶ But: The first RP to use it gets no UX gain
- ▶ So: Nobody wants to be the first



# Solution

- ▶ Pre-load AccountChooser!
- ▶ Goal: Google & Facebook & many other IDPs and ISPs jam entries in silently
- ▶ Result: When an RP tries it, they get results first time
- ▶ (This is work in progress)





# Thanks!