Secure Application Design

OpenID Connect: Delegated Authentication in Practice

Summer 2025

Jakob Heher, www.isec.tugraz.at
he/his

Continue with Facebook

Sign in with Apple

Sign in with Google

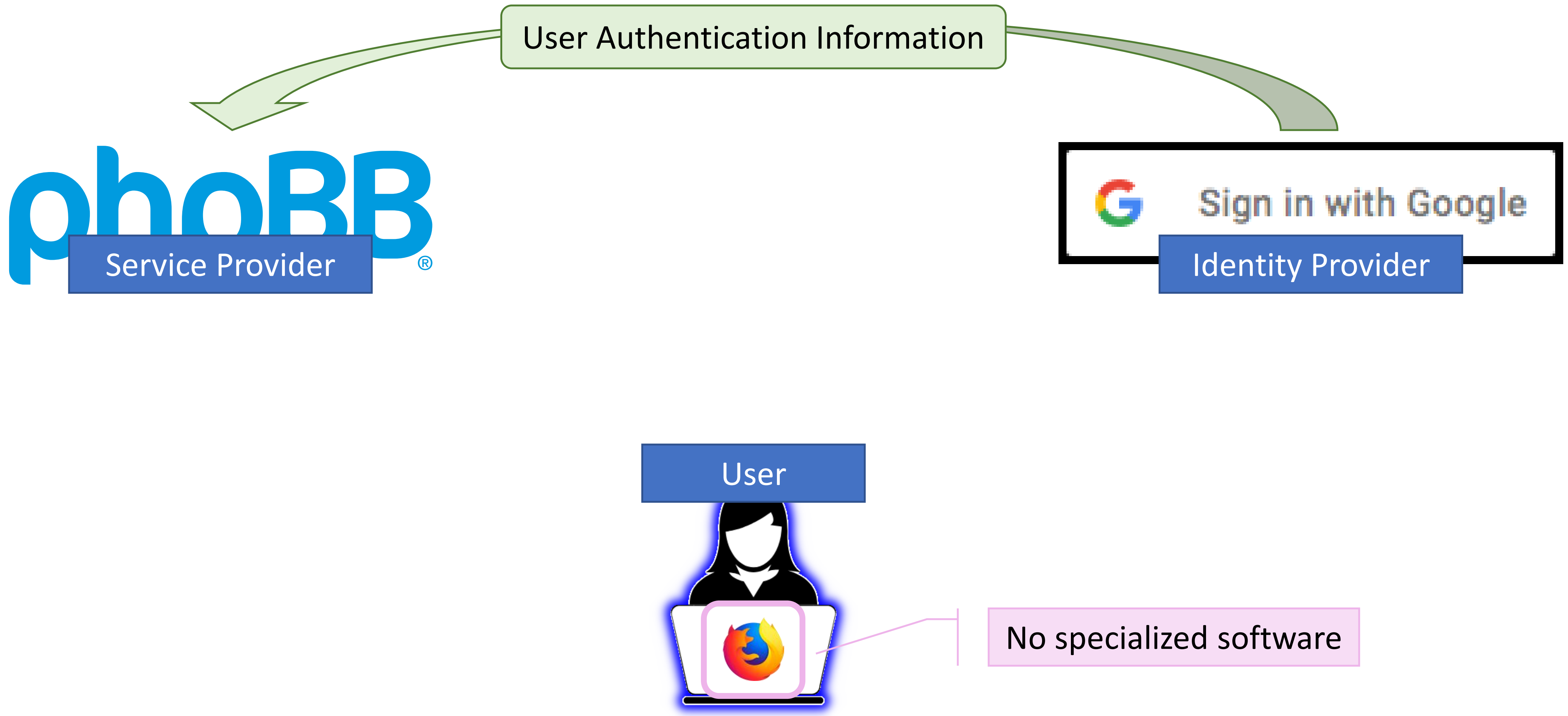
Sign up with GitHub

Login with Amazon

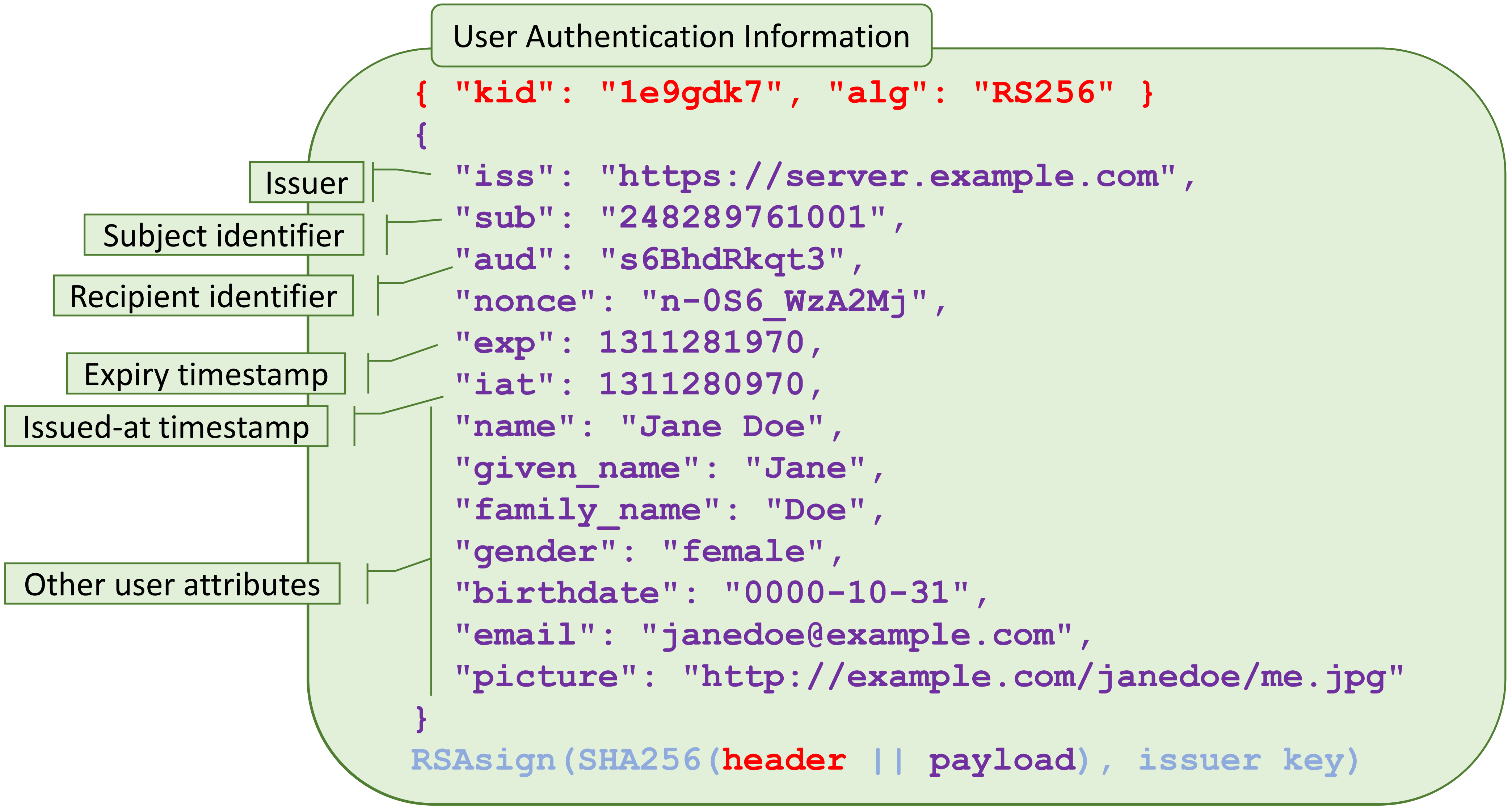
Sign in with LinkedIn

Sign in with Microsoft

Goals

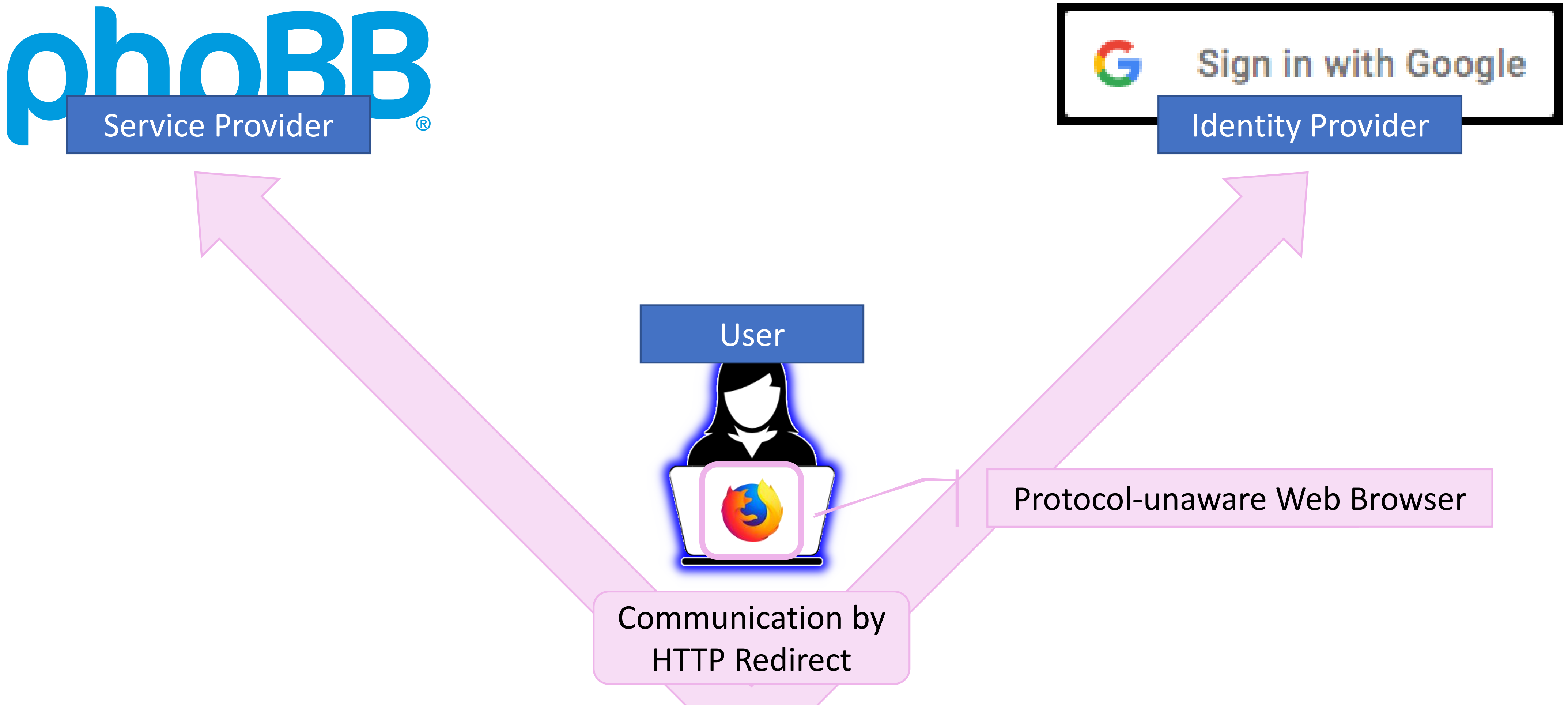

User Authentication Information
phpBB
Service Provider
Sign in with Google
Identity Provider
User
No specialized software

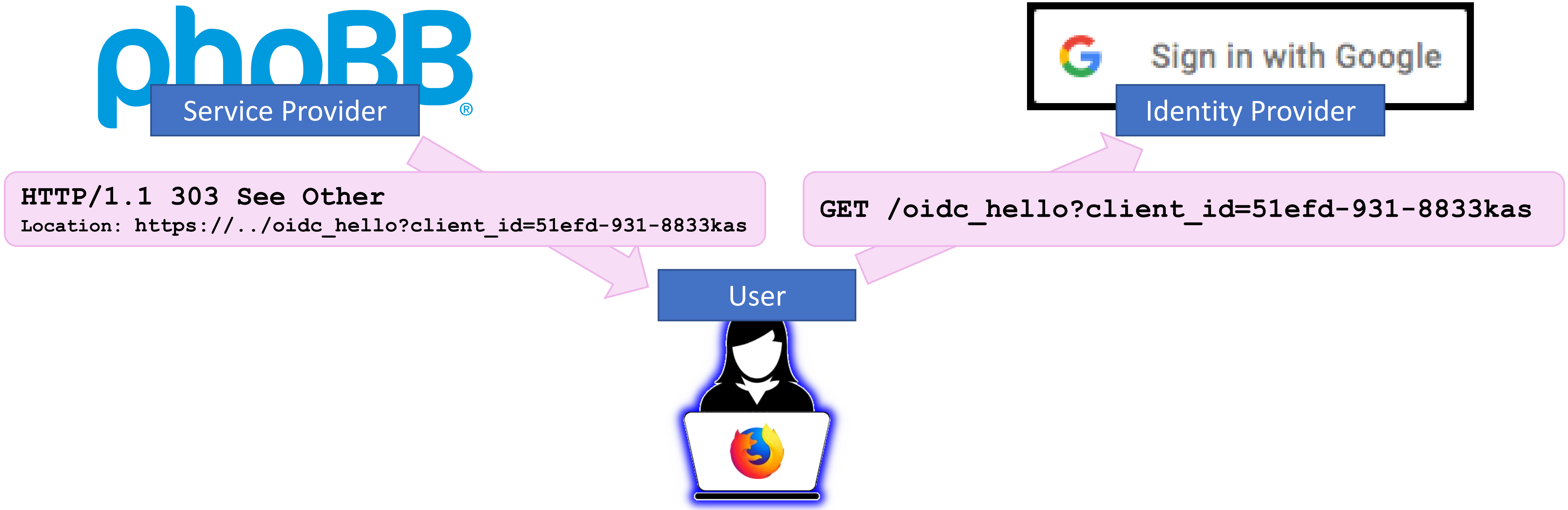
ID Token



```
{ "kid": "1e9gdk7", "alg": "RS256" }
{
  "iss": "https://server.example.com",
  "sub": "248289761001",
  "aud": "s6BhdRkqt3",
  "nonce": "n-0S6_WzA2Mj",
  "exp": 1311281970,
  "iat": 1311280970,
  "name": "Jane Doe",
  "given_name": "Jane",
  "family_name": "Doe",
  "gender": "female",
  "birthdate": "0000-10-31",
  "email": "janedoe@example.com",
  "picture": "http://example.com/janedoe/me.jpg"
}
RSAsign(SHA256(header || payload), issuer key)
```

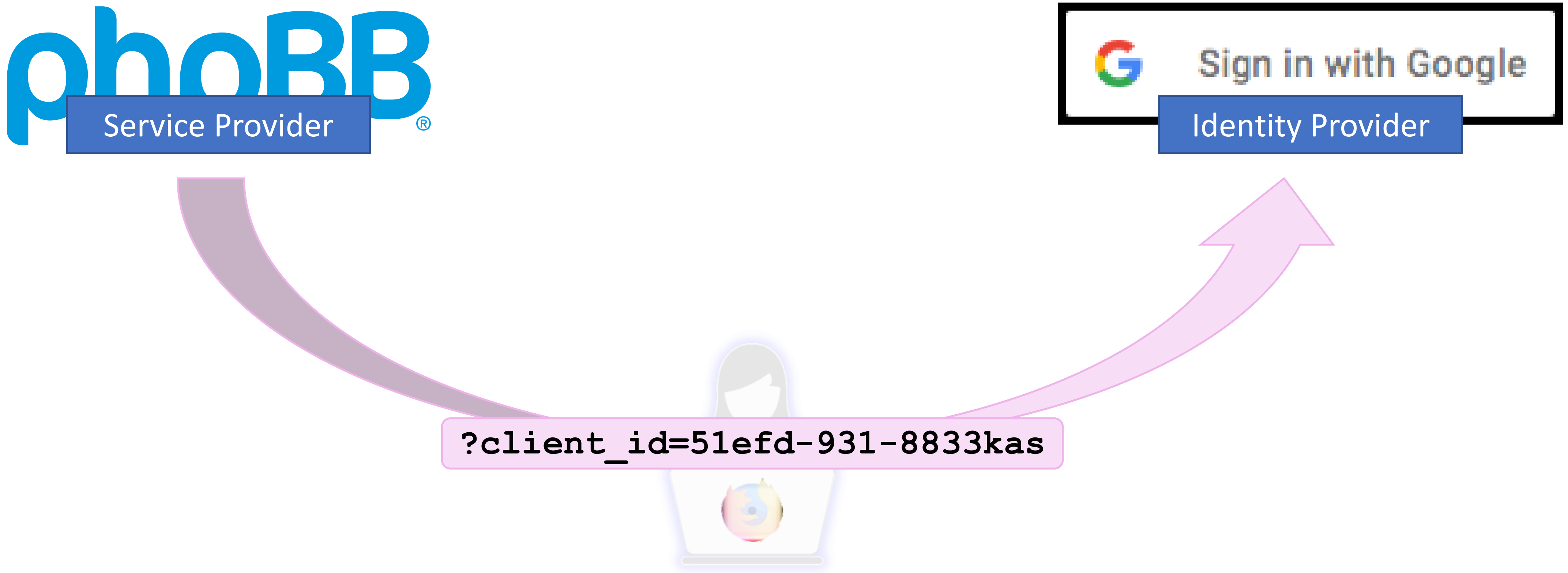
Goals

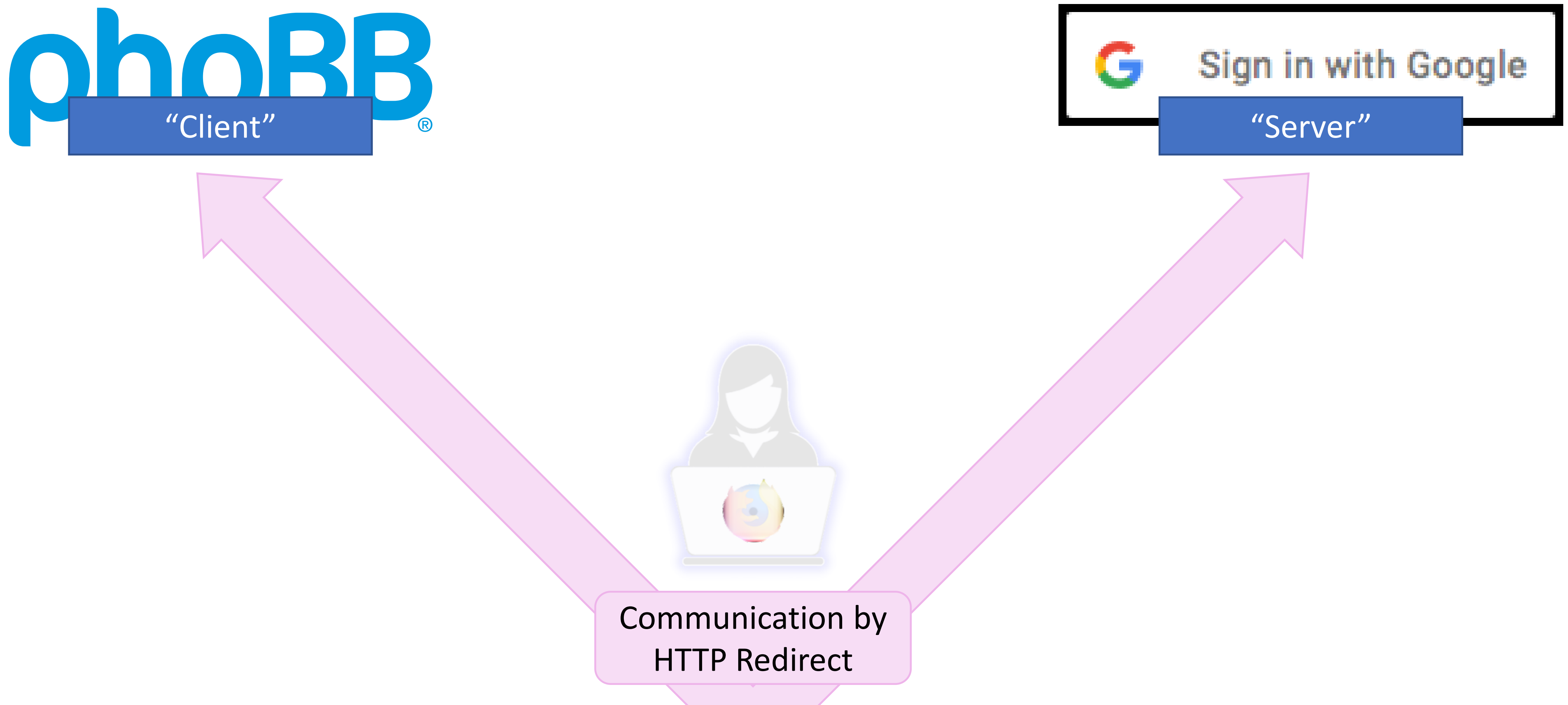

phpBB
Service Provider
Sign in with Google
Identity Provider
User
Protocol-unaware Web Browser
Communication by HTTP Redirect

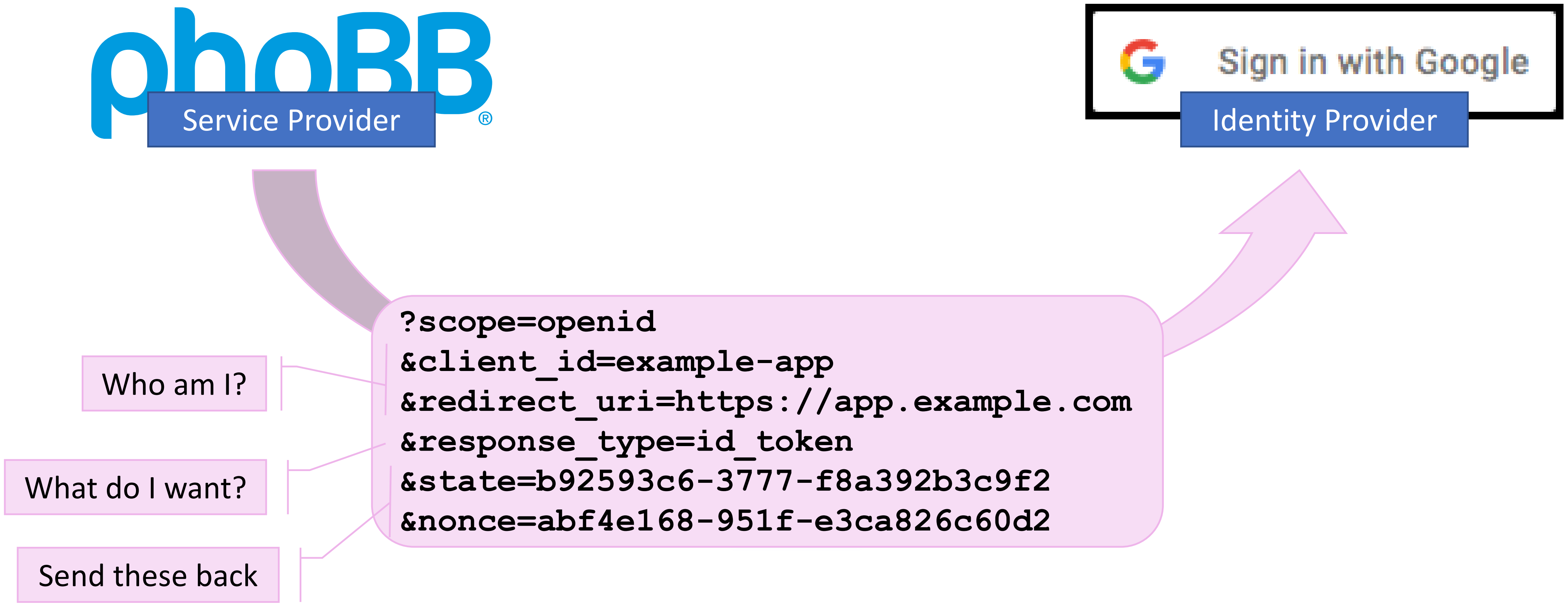


```
HTTP/1.1 303 See Other
Location: https://../oidc_hello?client_id=51efd-931-8833kas
```

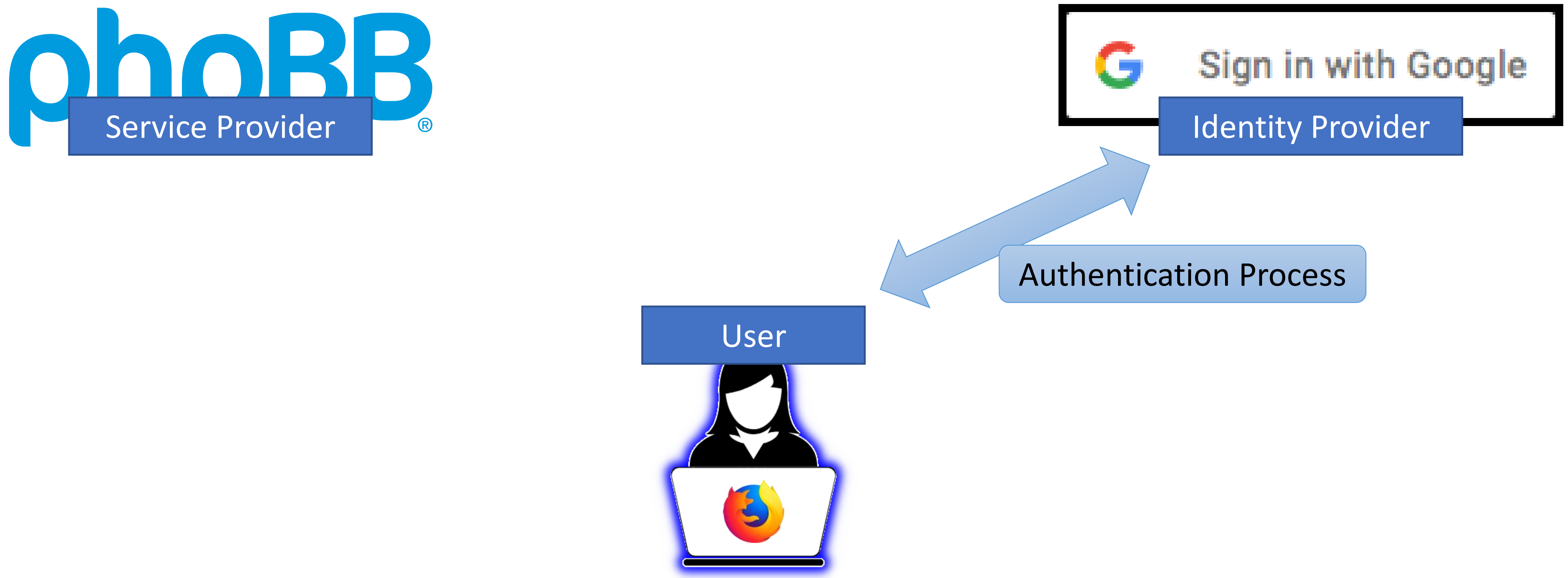
```
GET /oidc_hello?client_id=51efd-931-8833kas
```

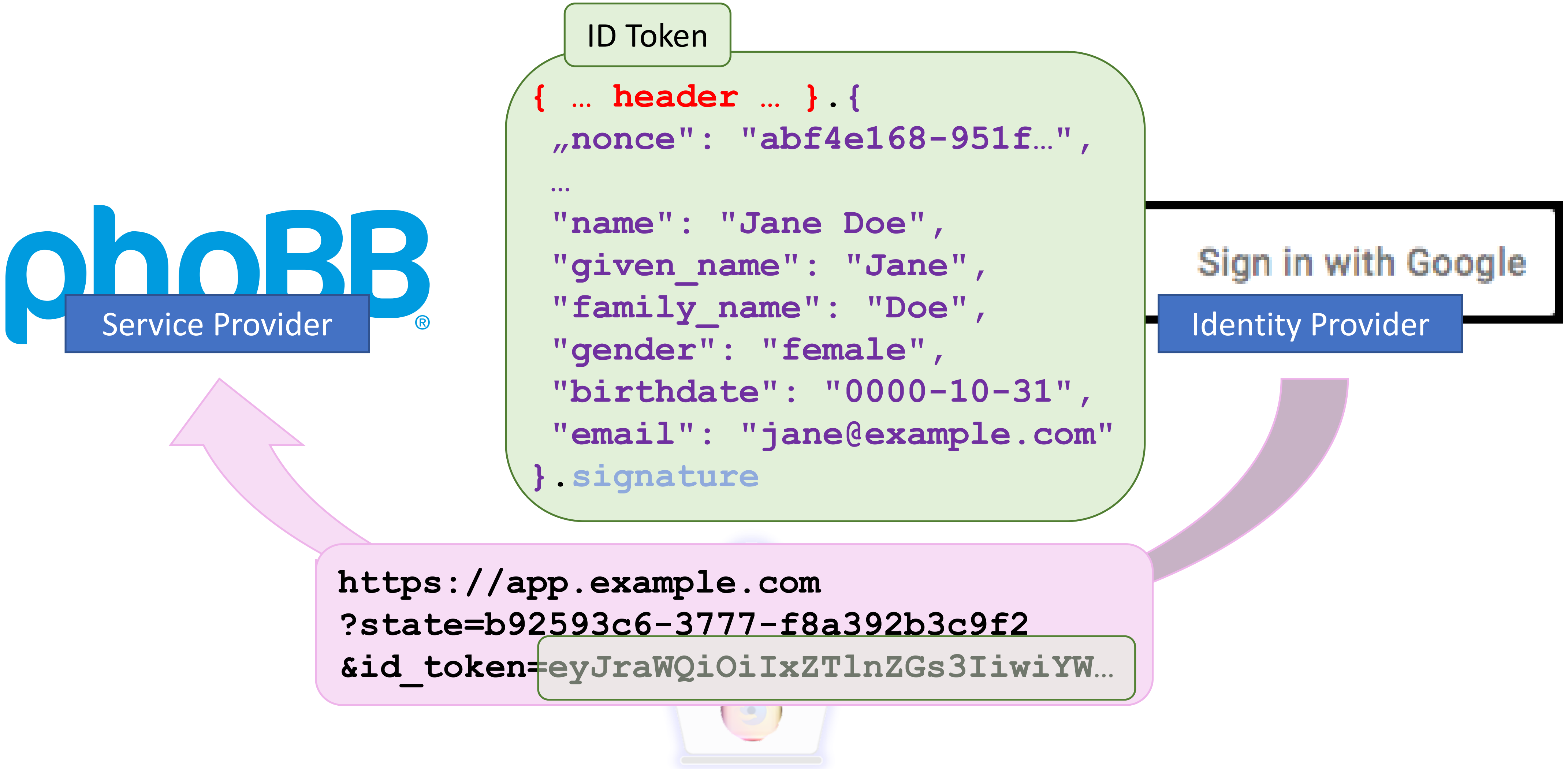

phpBB
Service Provider
Sign in with Google
Identity Provider
?client_id=51efd-931-8833kas


phpBB®
"Client"
Sign in with Google
"Server"
Communication by HTTP Redirect



```
?scope=openid
&client_id=example-app
&redirect_uri=https://app.example.com
&response_type=id_token
&state=b92593c6-3777-f8a392b3c9f2
&nonce=abf4e168-951f-e3ca826c60d2
```


phpBB
Service Provider
Sign in with Google
Identity Provider
Authentication Process
User

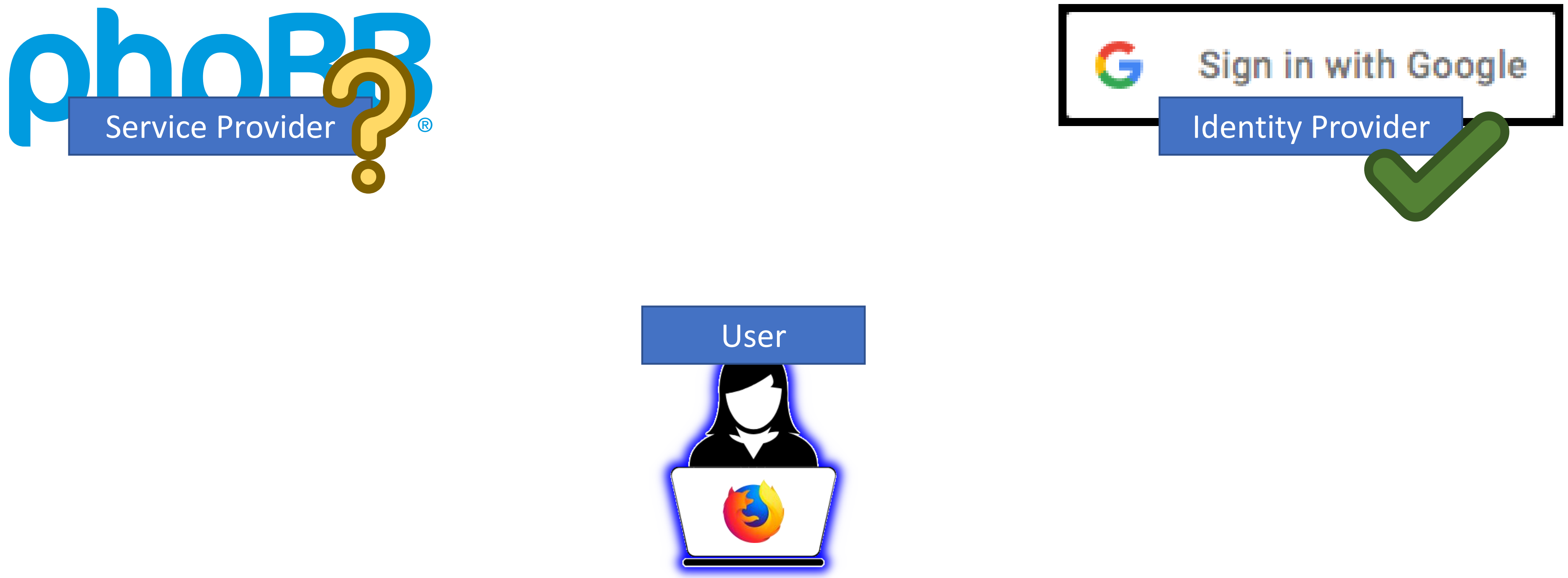


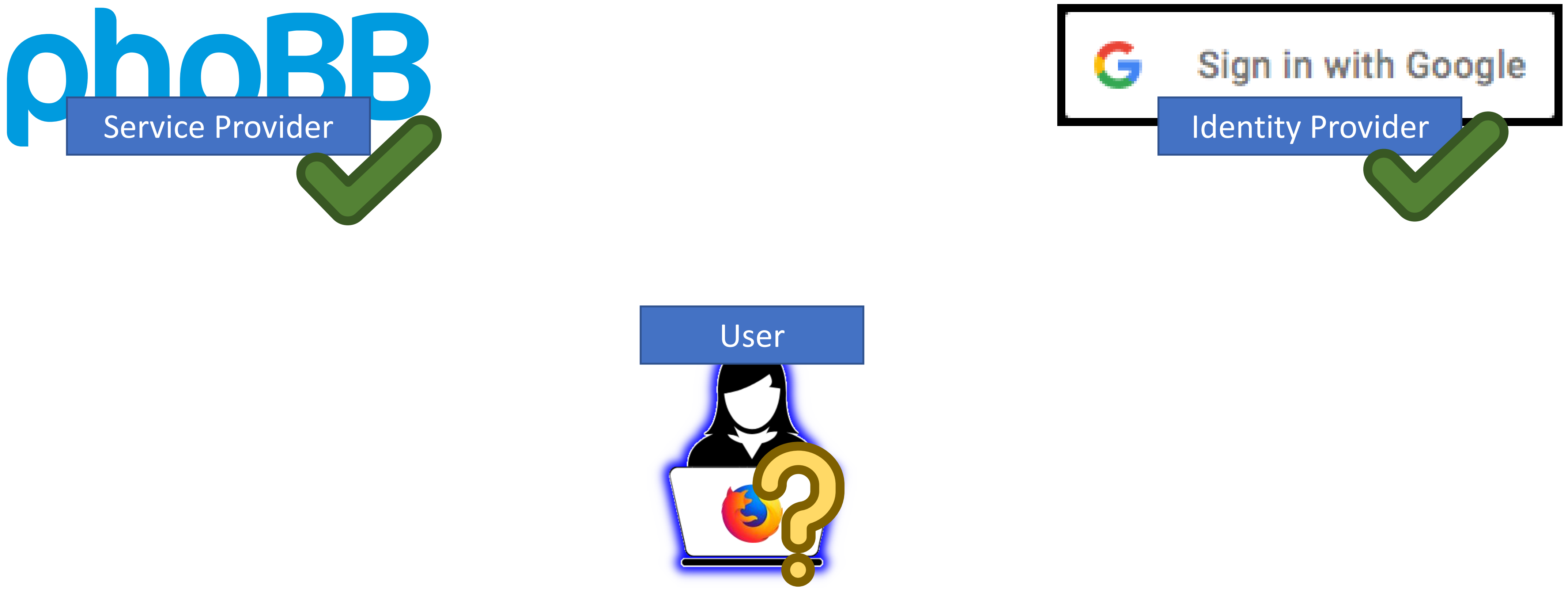
```
{ … header … }.{
 „nonce": "abf4e168-951f…",
 …
 "name": "Jane Doe",
 "given_name": "Jane",
 "family_name": "Doe",
 "gender": "female",
 "birthdate": "0000-10-31",
 "email": "jane@example.com"
}.signature
```

```
https://app.example.com
?state=b92593c6-3777-f8a392b3c9f2
&id_token=eyJraWQiOiIxZTlnZGs3IiwiYW…
```

OpenID Connect: Implicit Flow

- Service Provider redirects to *authorization endpoint* at IdP
- Identity Provider performs authentication
- Identity Provider redirects to *redirect URI* at SP with ID Token

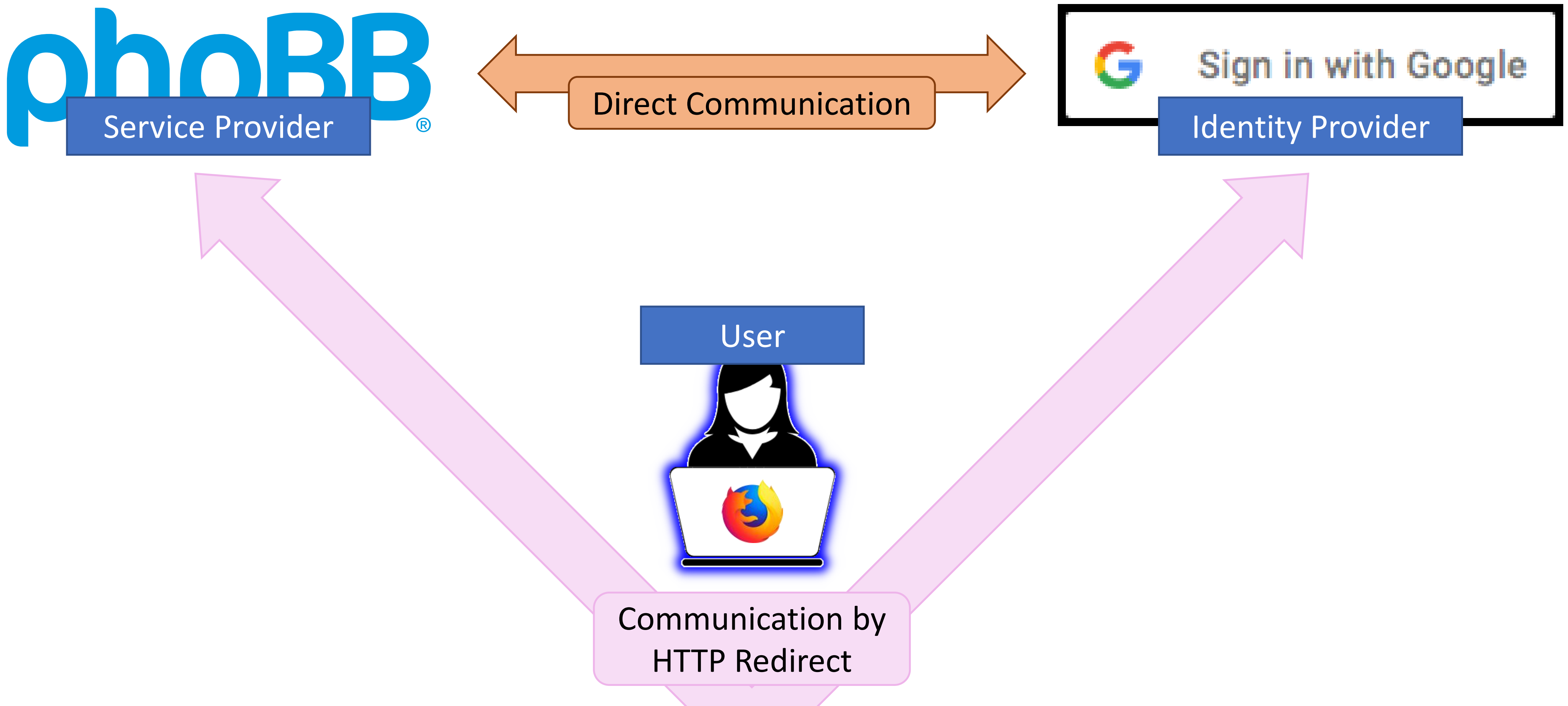

phpBB
Service Provider
Sign in with Google
Identity Provider
User

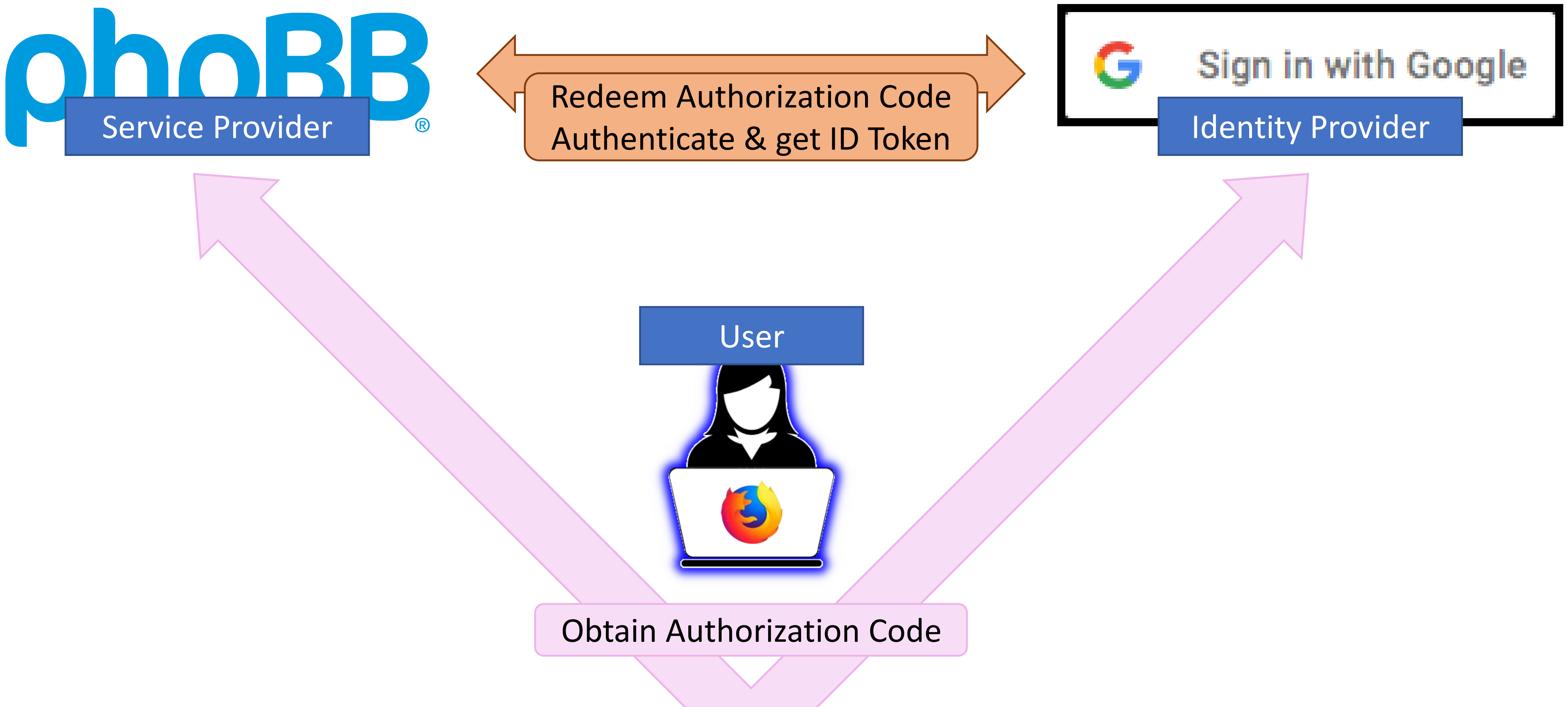

phpBB
Service Provider
Sign in with Google
Identity Provider
User

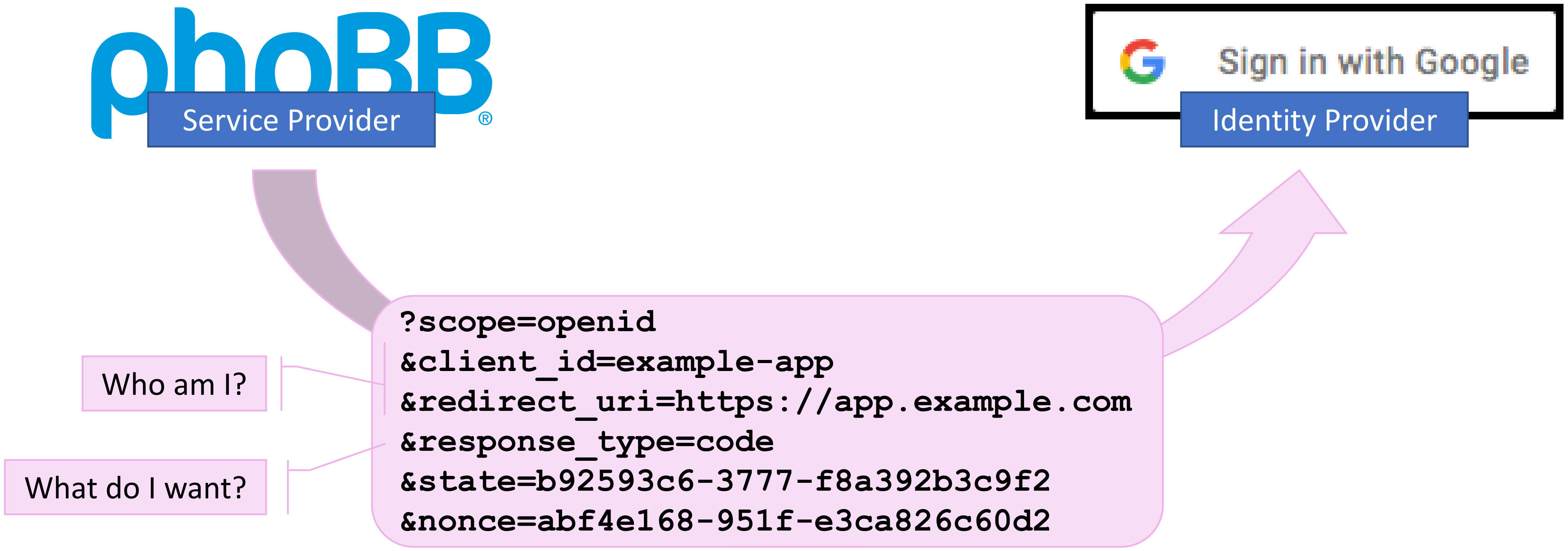
OpenID Connect: Implicit Flow

- Service Provider redirects to *authorization endpoint* at IdP
- Identity Provider performs authentication
- Identity Provider redirects to *redirect URI* at SP with ID Token

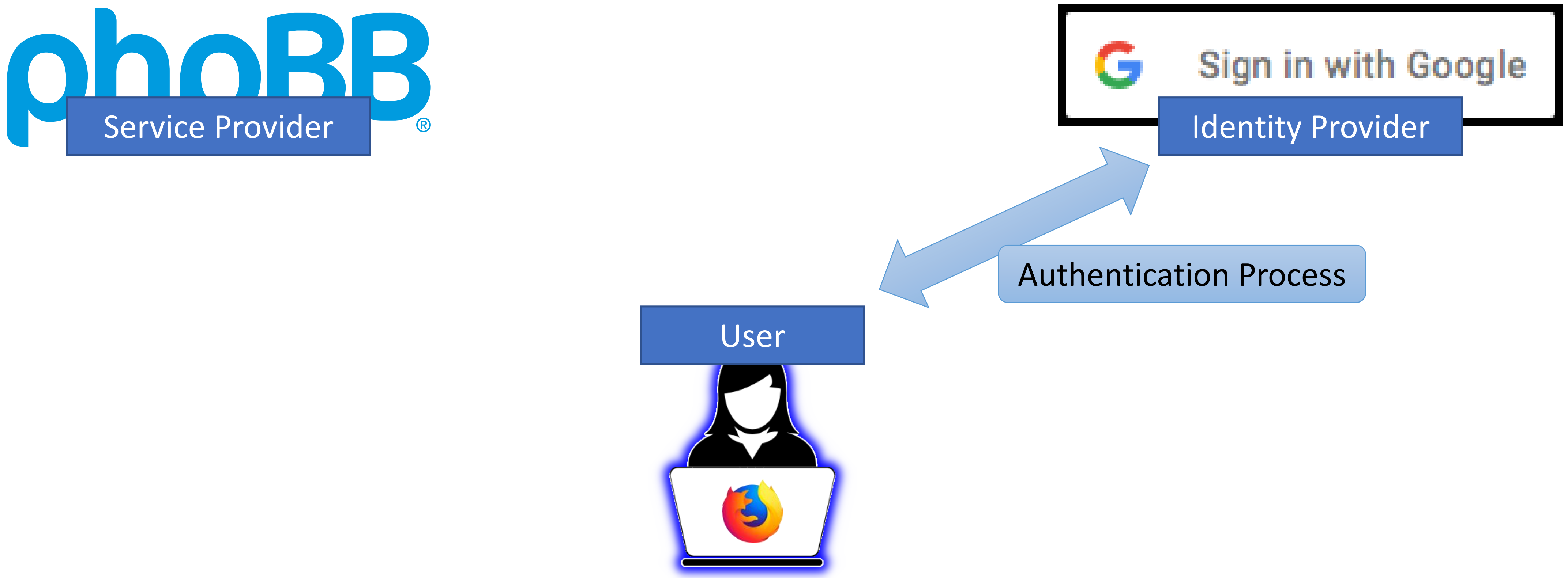
- The ID Token is exposed to the user's browser
  - Malicious extensions might capture it
  - It might be stored in browser history

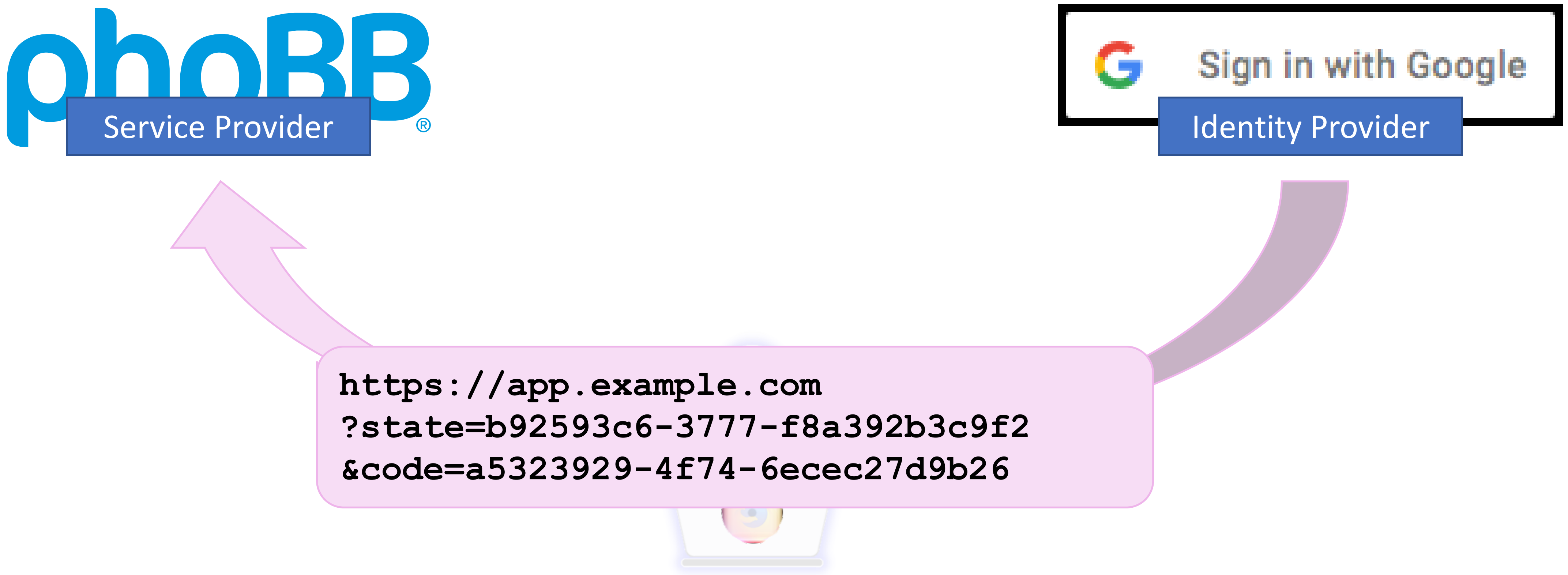

Service Provider
Direct Communication
Sign in with Google
Identity Provider
User
Communication by HTTP Redirect


Service Provider
Redeem Authorization Code
Authenticate & get ID Token
Sign in with Google
Identity Provider
User
Obtain Authorization Code

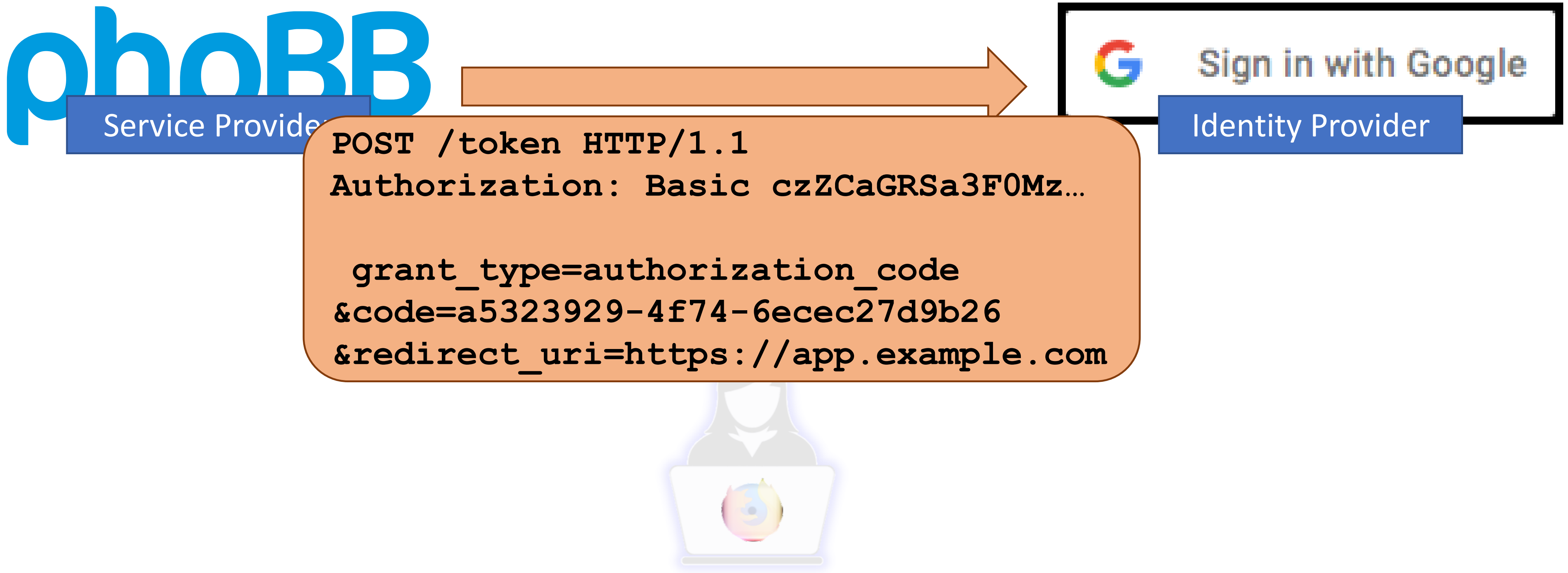


```
?scope=openid
&client_id=example-app
&redirect_uri=https://app.example.com
&response_type=code
&state=b92593c6-3777-f8a392b3c9f2
&nonce=abf4e168-951f-e3ca826c60d2
```


phpBB
Service Provider
Sign in with Google
Identity Provider
Authentication Process
User

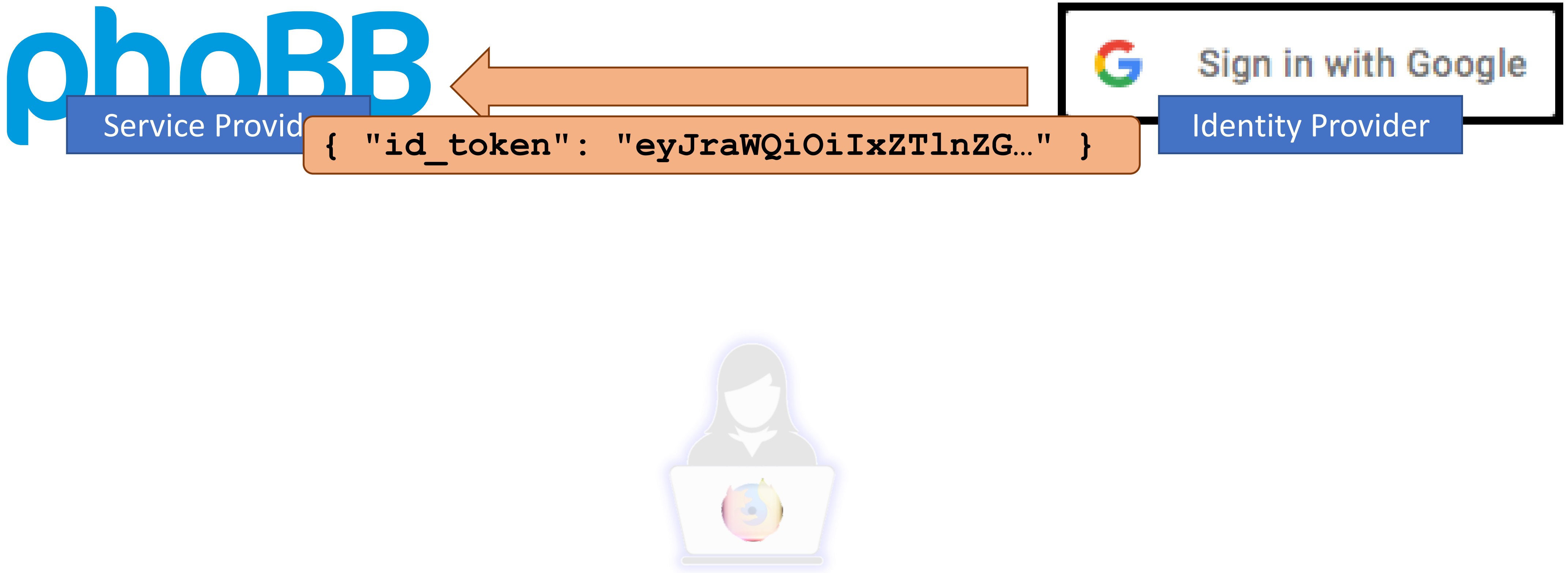


```
https://app.example.com
?state=b92593c6-3777-f8a392b3c9f2
&code=a5323929-4f74-6ecec27d9b26
```



```
POST /token HTTP/1.1
Authorization: Basic czZCaGRSa3F0Mz…

 grant_type=authorization_code
&code=a5323929-4f74-6ecec27d9b26
&redirect_uri=https://app.example.com
```



```
{ "id_token": "eyJraWQiOiIxZTlnZG…" }
```

OpenID Connect: Authorization Code Flow

- Service Provider redirects to *authorization endpoint* at IdP
- Identity Provider performs authentication
- Identity Provider redirects to SP with *authorization code*
- Service Provider redeems authorization code at IdP's *token endpoint*
    - Requires *authorization code* and *SP authentication*

Real-World Demo

Code available: https://extgit.isec.tugraz.at/A-SIT/oidc-ida-research-template

Identifiers & Pseudonyms

- *Re*-Authentication is crucial
  - How do we tell if two logins are the same person?
- Re-Authentication implies *Linkability*!
- Does every SP need the same identifier?

ID Token

```
{
 "sub": "248289761001",
 …
 "name": "Jane Doe",
 "given_name": "Jane",
 "family_name": "Doe",
 "gender": "female",
 "birthdate": "0000-10-31",
 "email": "jane@example.com"
}
```

Pairwise Pseudonyms

- Idea: different pseudonym for each service
  - Example: **`H(user secret || service provider ID)`**
  - We can still re-authenticate to the same service
  - Different services don’t know we’re the same person

- Each IdP-”user” can only have one identity at a given SP
  - Is this desirable?

Secure Application Design

FedCM: Delegated Authentication in the Future?

Summer 2025

Jakob Heher, www.isec.tugraz.at
he/his

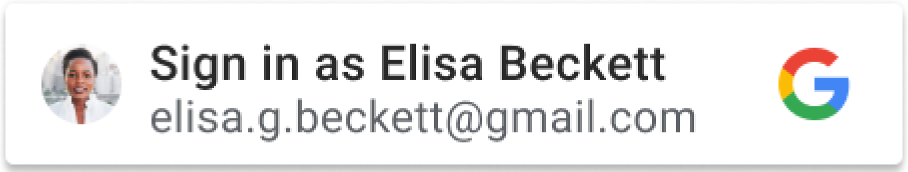
What is a third-party component?

- Component requested from a web origin that is *not* the current origin
  - Images, fonts, script files, iframes, …
- These requests trigger HTTP GET requests
  - These requests can include cookies!
- These *third-party cookies* are a powerful tracking tool
  - They allow correlating user movement across the web (e.g., “Facebook pixel”)

Why does this matter for OpenID Connect?

- For the traditional flow, it doesn't!
  - OIDC uses full-page redirects

- It matters for certain implementations…



Sign in as Elisa Beckett
elisa.g.beckett@gmail.com

  - … which are based on **`<iframe>`** s

Federated Credential Management API

- The browser is now protocol-aware!

- Here's the rough idea:
  1. The browser gets a list of logged-in accounts from the IdP
     - No reference to the requesting service provider is included
  2. The user chooses an account for log-in using browser-provided UX
     - If the user stops, the IdP never learns the requesting service provider
  3. The browser requests an ID token for the account and SP from the IdP
     - The ID token is conceptually the same as in OIDC

Demo available in Chrome: https://fedcm-rp-demo.glitch.me/

FedCM – The Future?

- Current standard only works for a specific use case
- Privacy challenges from OIDC remain unsolved
- … do we really need this?

- Browser becoming protocol-aware has great potential!
  - … if only it were used

Photograph by Yellowstone National Park staff, public domain (Flickr)