



An Intro to OpenID Connect

Eugenio Pace - eugeniop@auth0.com

30 min to...

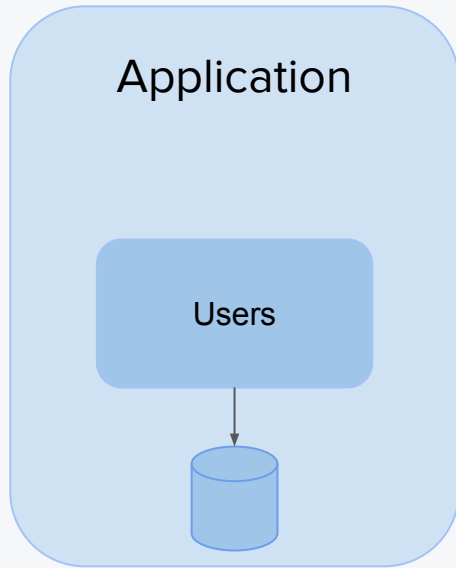
OpenID Connect

- What is it?
- How it works?
- Why should you care?

OpenID Connect

A protocol used to authenticate users of an application, and represent those users in a standard way.

A typical (web) app



Factoring out Auth

Application

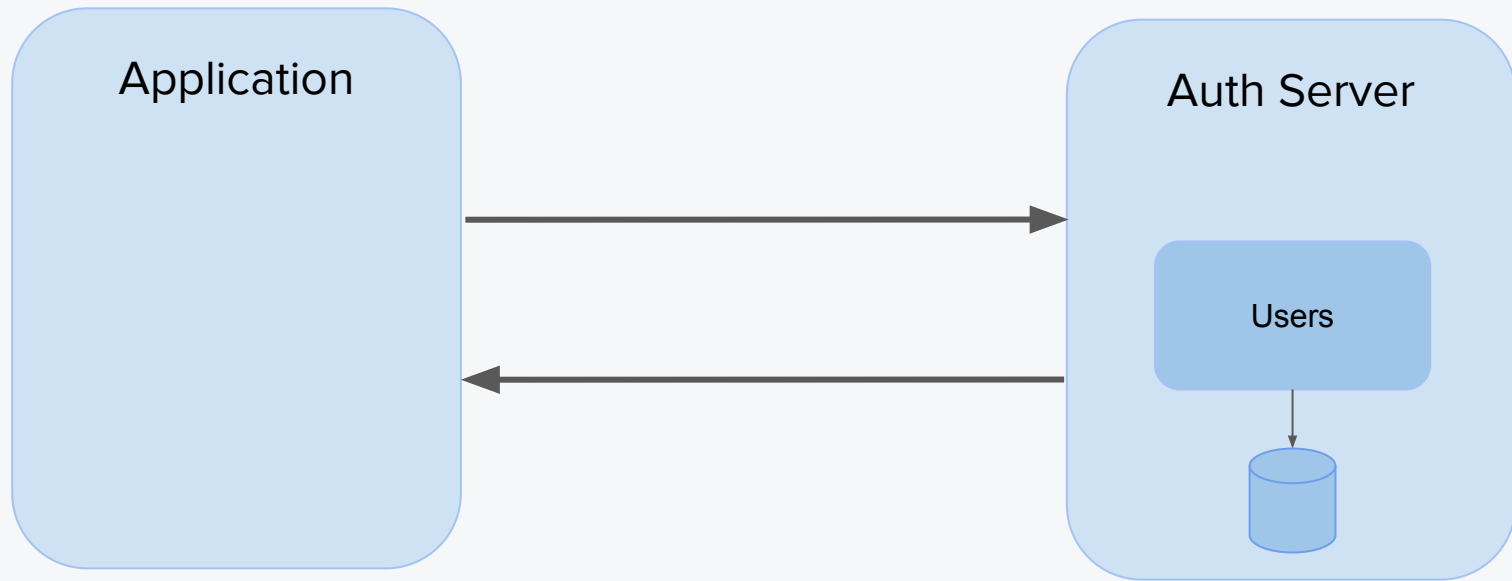


```
graph LR; subgraph Application; direction TB; Application; end; subgraph Auth_Server [Auth Server]; direction TB; Users; Database[(Database)]; Users --> Database; end;
```

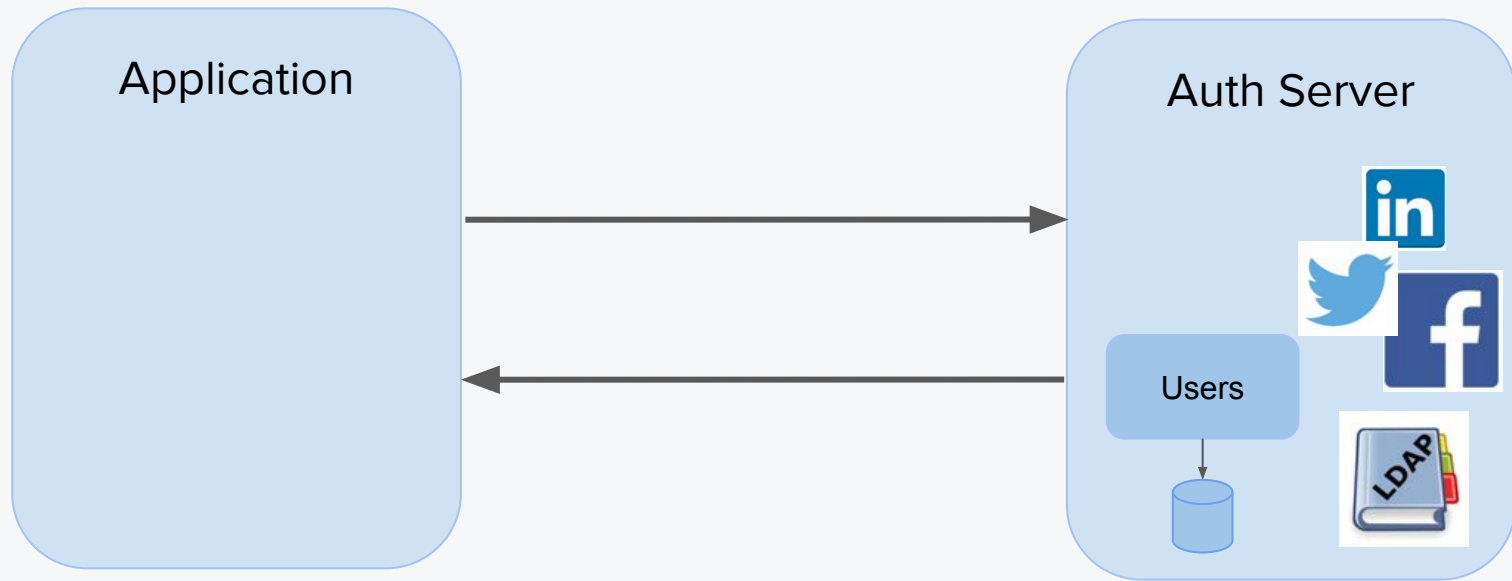
Auth Server

Users

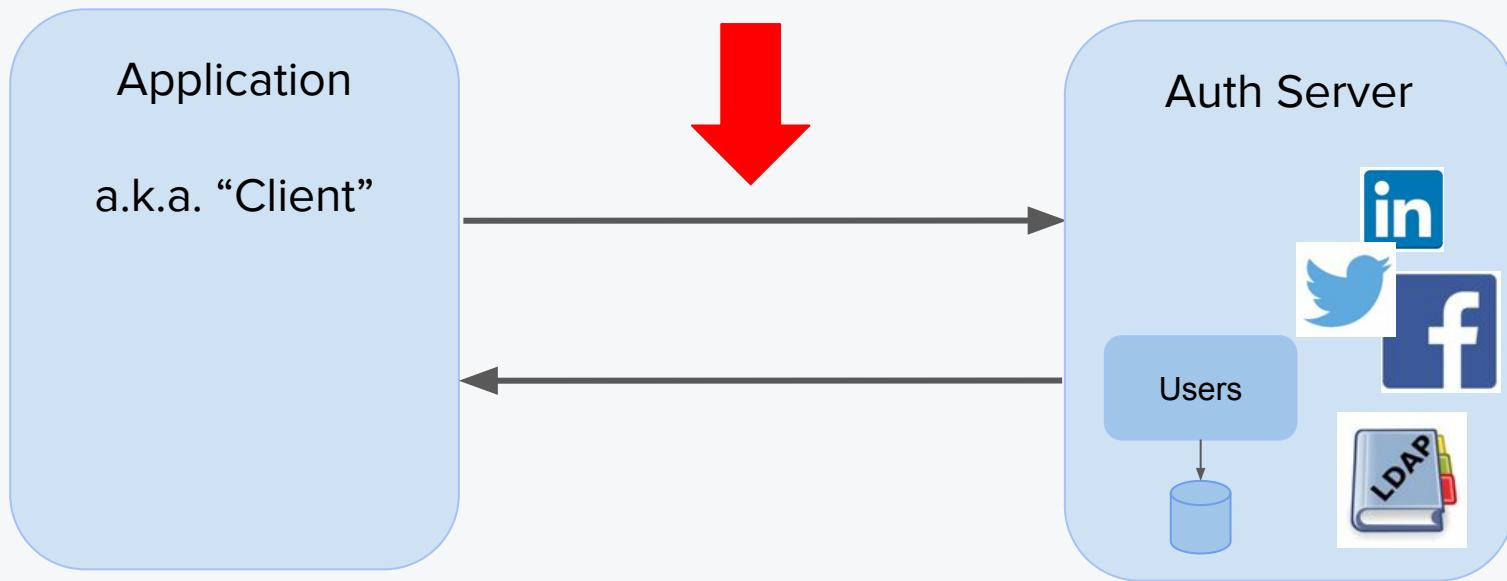
Factoring out Auth



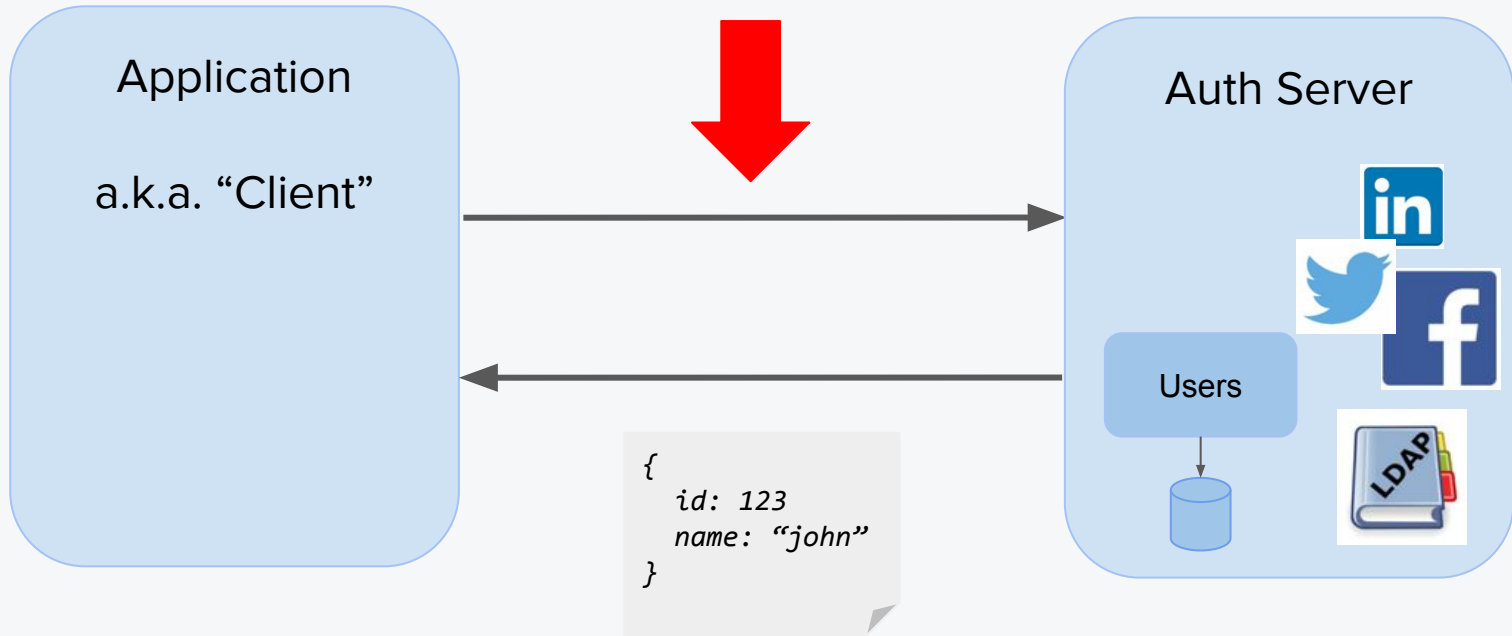
Factoring out Auth



OIDC



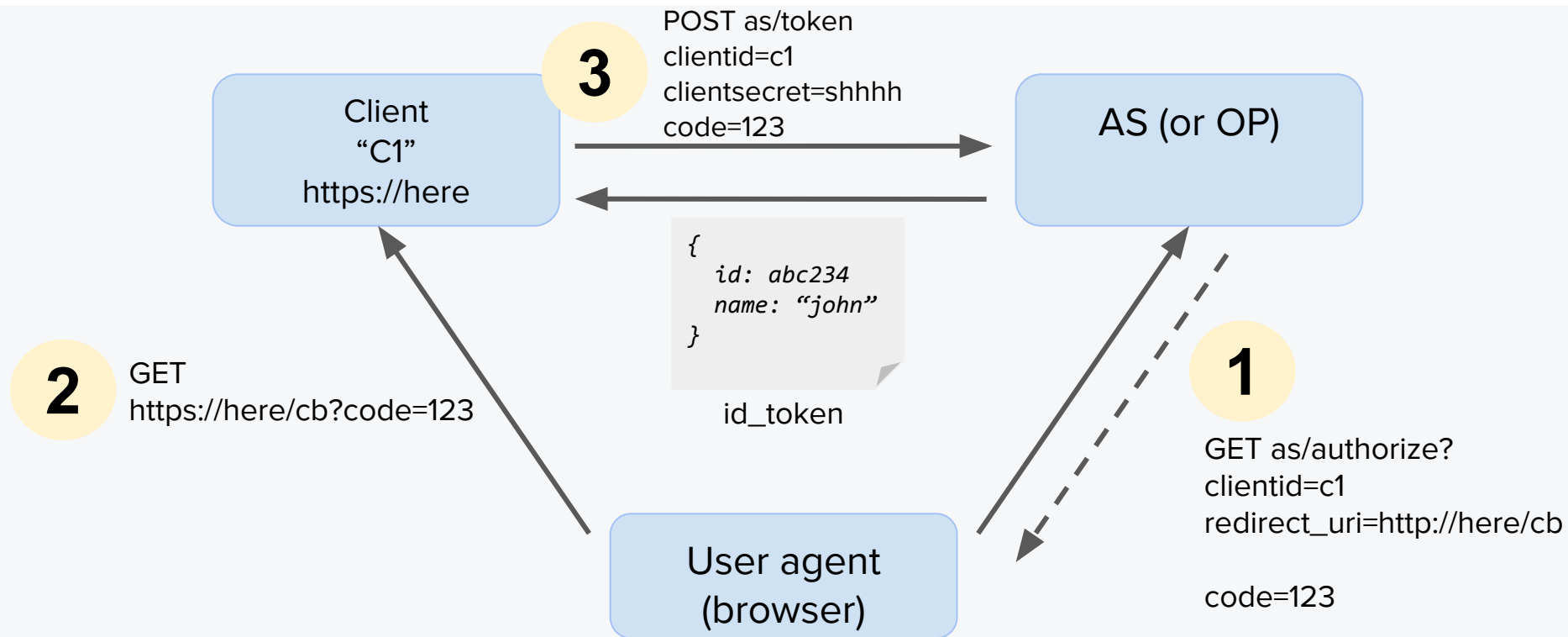
OIDC



How it works?

<https://openidconnect.net>

Authorization Code Flow - review



OIDC & OAuth2

- OIDC is built on top of OAuth2
 - Same flows
- User information is encoded in an **id_token**
- $\text{OIDC} = \text{OAuth2} + \text{JWT}$
- You can still get *access_tokens* to call APIs

Why?

- Industry standard:
 - Wide availability of SDKs, simplifies implementation
 - Interop: connect stuff with less effort
 - “Modern” → lightweight / cloud friendly
 - Avoid lock-in

Why?

- Industry standard:
 - Wide availability of SDKs, simplifies implementation
 - Interop: connect stuff with less effort
 - “Modern” → lightweight / cloud friendly
 - Avoid lock-in
- Standing on the shoulder of giants:
 - Battle tested (e.g. Google, FB, and many many more)
 - Security proofed

Resources / Learn more

- **Standard:**
 - <http://openid.net>
 - Core: http://openid.net/specs/openid-connect-core-1_0.html
- **OIDC Playground**
 - <https://openidconnect.net>
- **JWT Playground**
 - <http://jwt.io>
- **Overview of Protocols**
 - <https://docs.auth0.com/protocols>



<https://auth0.com>