Authorization: Securing the API

Kevin Dockx

ARCHITECT

@KevinDockx   www.kevindockx.com

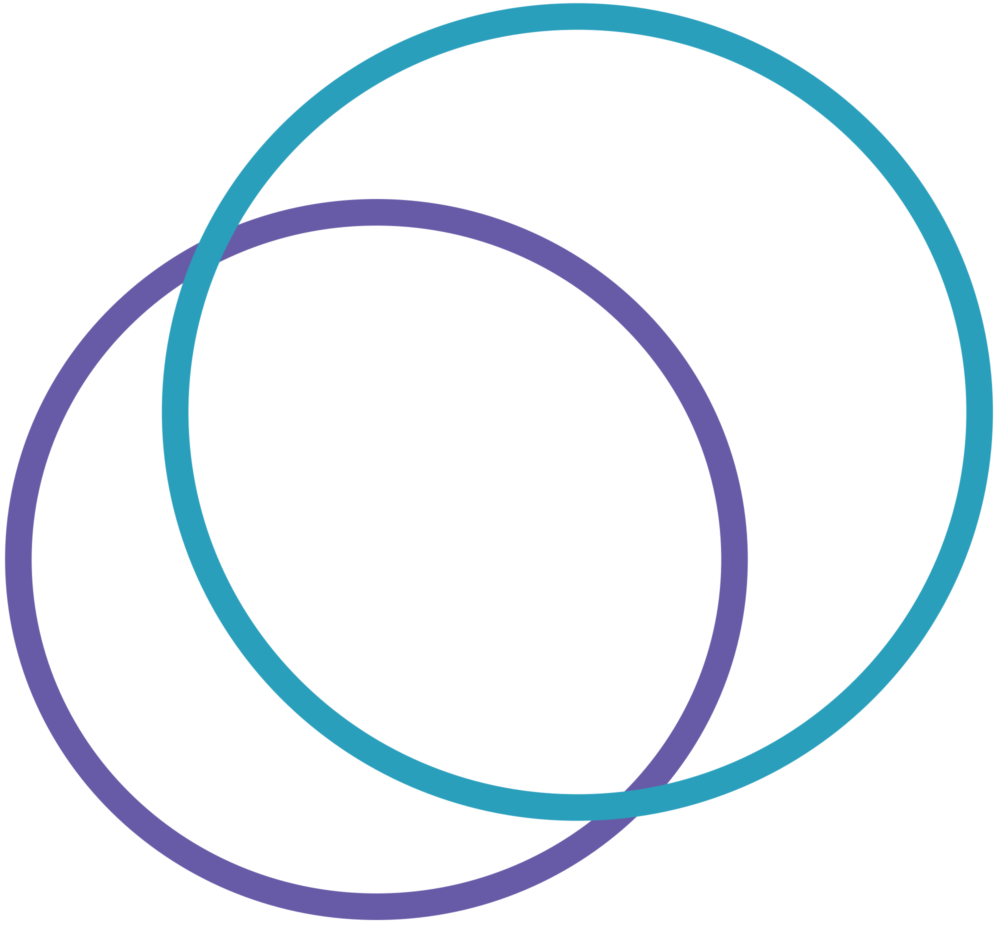
Coming Up

Using OpenID Connect for Authentication and Authorization

Blocking and Gaining Access to the API

Handling Token Expiration

Using OpenID Connect for Authentication and Authorization



OpenID Connect

Authentication
Identity token

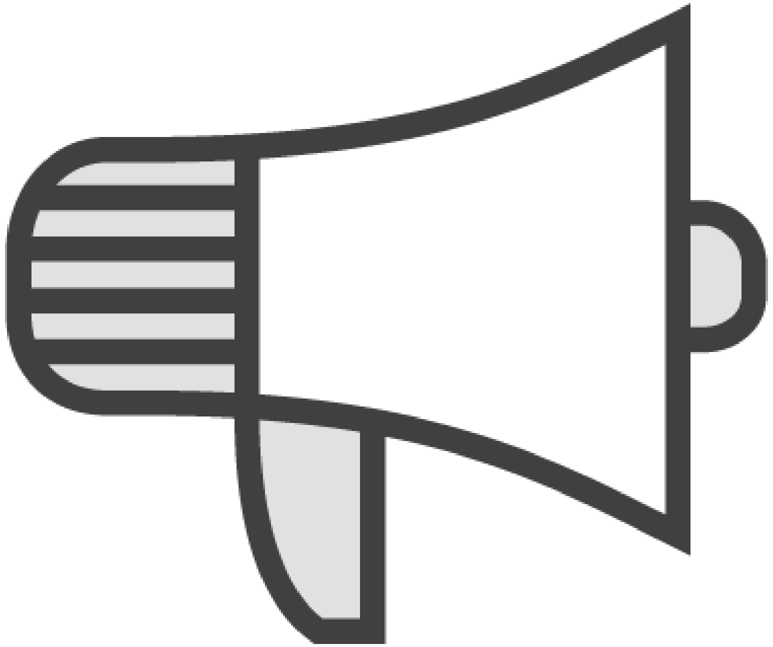
Gaining identity information and signing in

OAuth 2.0

Authorization
Access token

Securing resources (API)

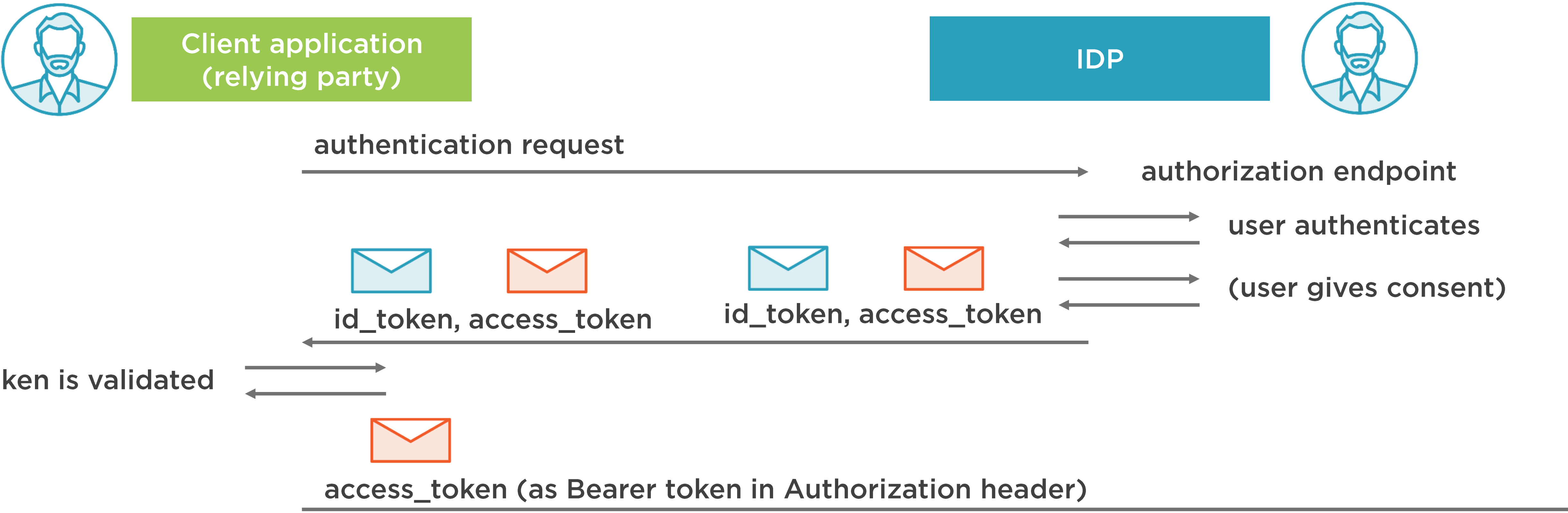
Using OpenID Connect for Authentication and Authorization



OpenID Connect is the superior protocol

- Identity token can be linked to access token (at_hash)
- Identity token can be verified first

The Implicit Flow



Client application (relying party)

IDP

authentication request

authorization endpoint

user authenticates

(user gives consent)

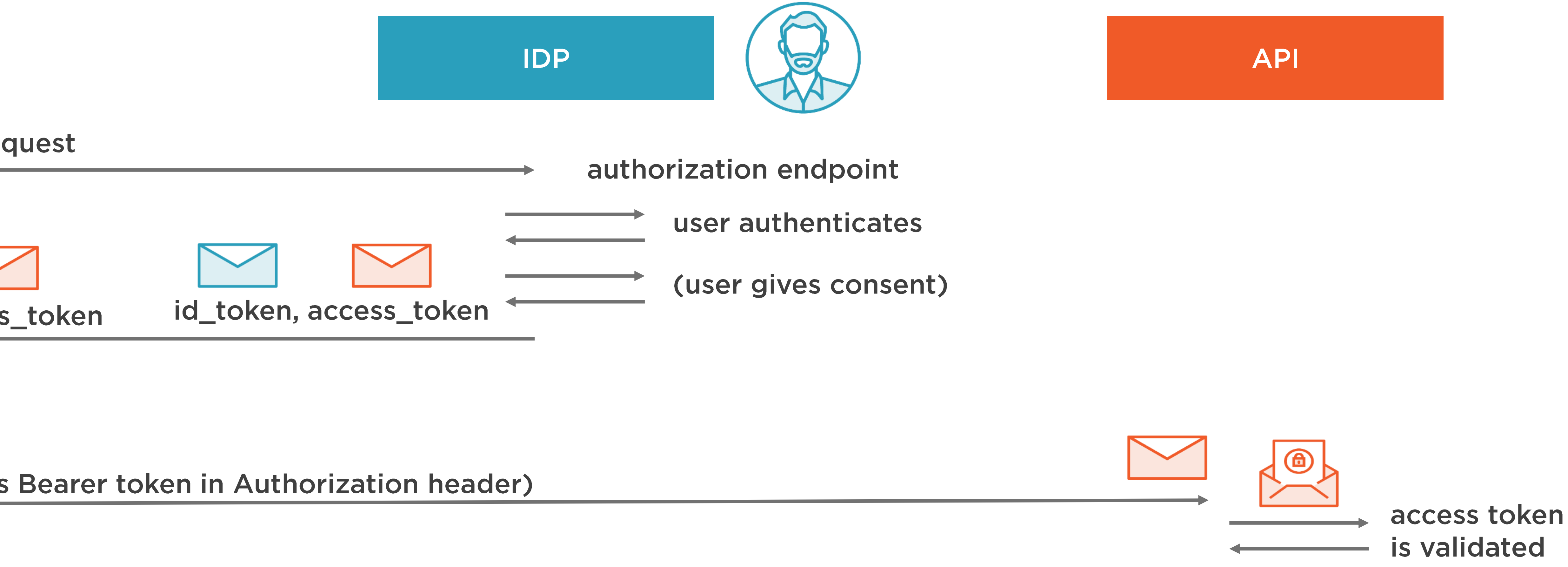
id_token, access_token

id_token, access_token

token is validated

access_token (as Bearer token in Authorization header)

The Implicit Flow



IDP

API

quest

authorization endpoint

user authenticates

(user gives consent)

s_token

id_token, access_token

s Bearer token in Authorization header)

access token
is validated

Demo

Blocking Access to the API

Demo

Requesting an Access Token with the Correct Audience

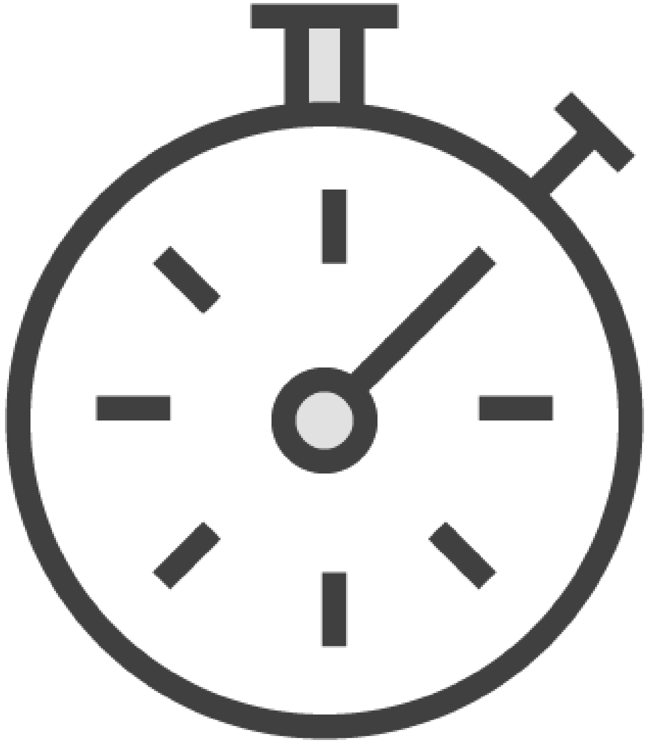
Demo

Passing an Access Token to the API

Demo

Using Claims from the Access Token

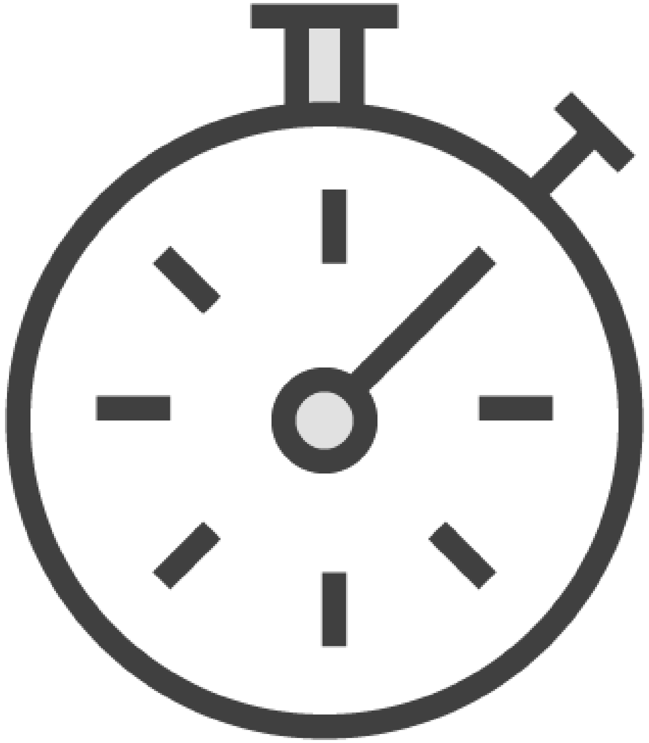
Renewing an Expired Access Token



Tokens have a limited lifetime

Validation of an expired token will fail

Renewing an Expired Access Token



The Authorization Code and Hybrid flows allow refresh tokens, which can be used to gain long-lived access

The Implicit flow does not allow refresh tokens, as that would be unsafe

Renewing an Expired Access Token

When a token expires, we can send a new authentication request to get new tokens

Renewing an Expired Access Token

Issue: user shouldn’t always have to input credentials

As long as we’re logged in to the IDP, we can get new tokens without authenticating again

Issue: visible redirection to the IDP should be avoided

We can avoid this by using hidden IFrames

Renewing an Expired Access Token

This approach is described in the Session Management standard

- http://bit.ly/2COOS2x

This allows renewing tokens as long as the user is still logged in to the IDP, while refresh tokens also allow renewing them when the user isn’t logged in anymore

Demo

Renewing an Expired Access Token

Summary

Authorization to an API is granted through an access token with a matching audience

Access tokens are passed to the API as Bearer tokens on each request

Summary

The implicit flow doesn’t allow long-lived access through refresh tokens

Tokens can be renewed as long as there’s an active user session at level of the IDP