Applying an OpenIdConnect Identity Provider

Roland Guijt

MICROSOFT MVP, CONSULTANT, AUTHOR AND SPEAKER

@rolandguijt rolandguijt.com

Module Overview

OpenIdConnect identity providers

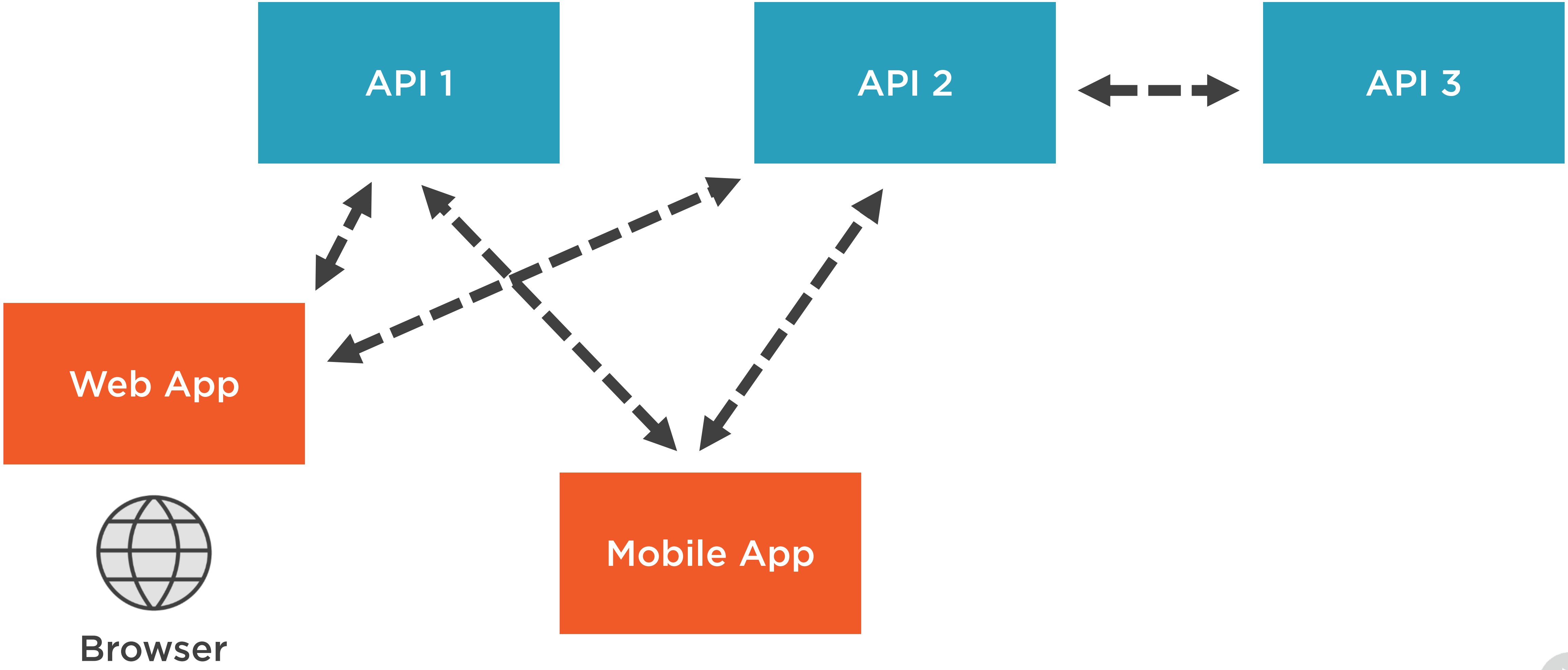
Clients and tokens

Flows

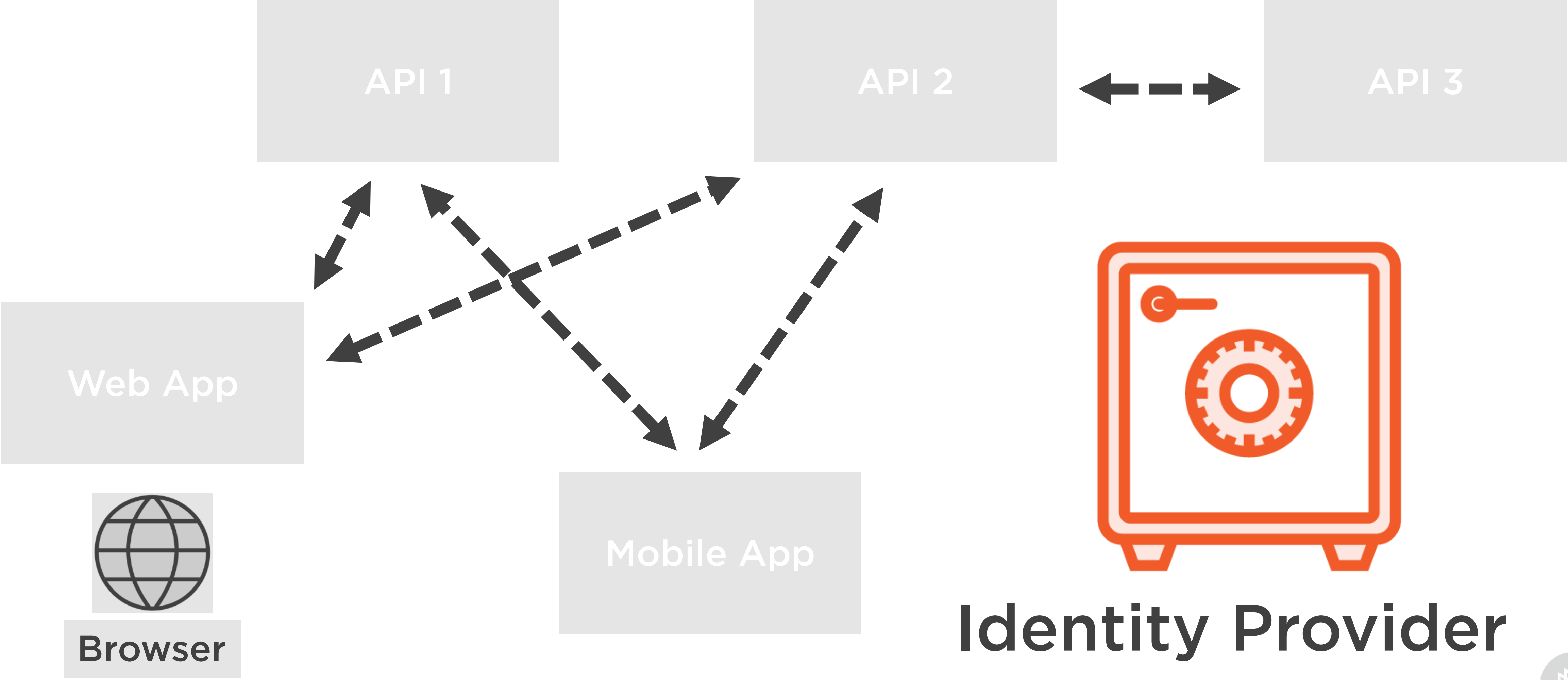
IdentityServer

Cloud identity providers

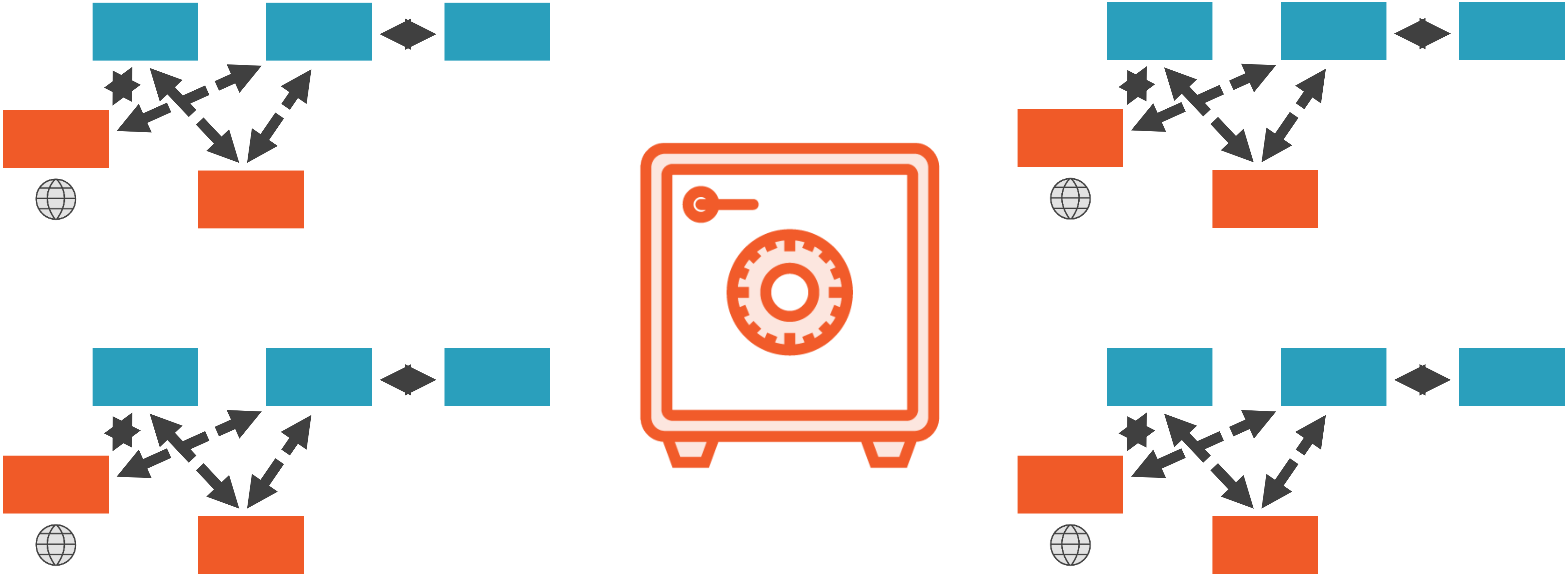
A Typical Modern Application



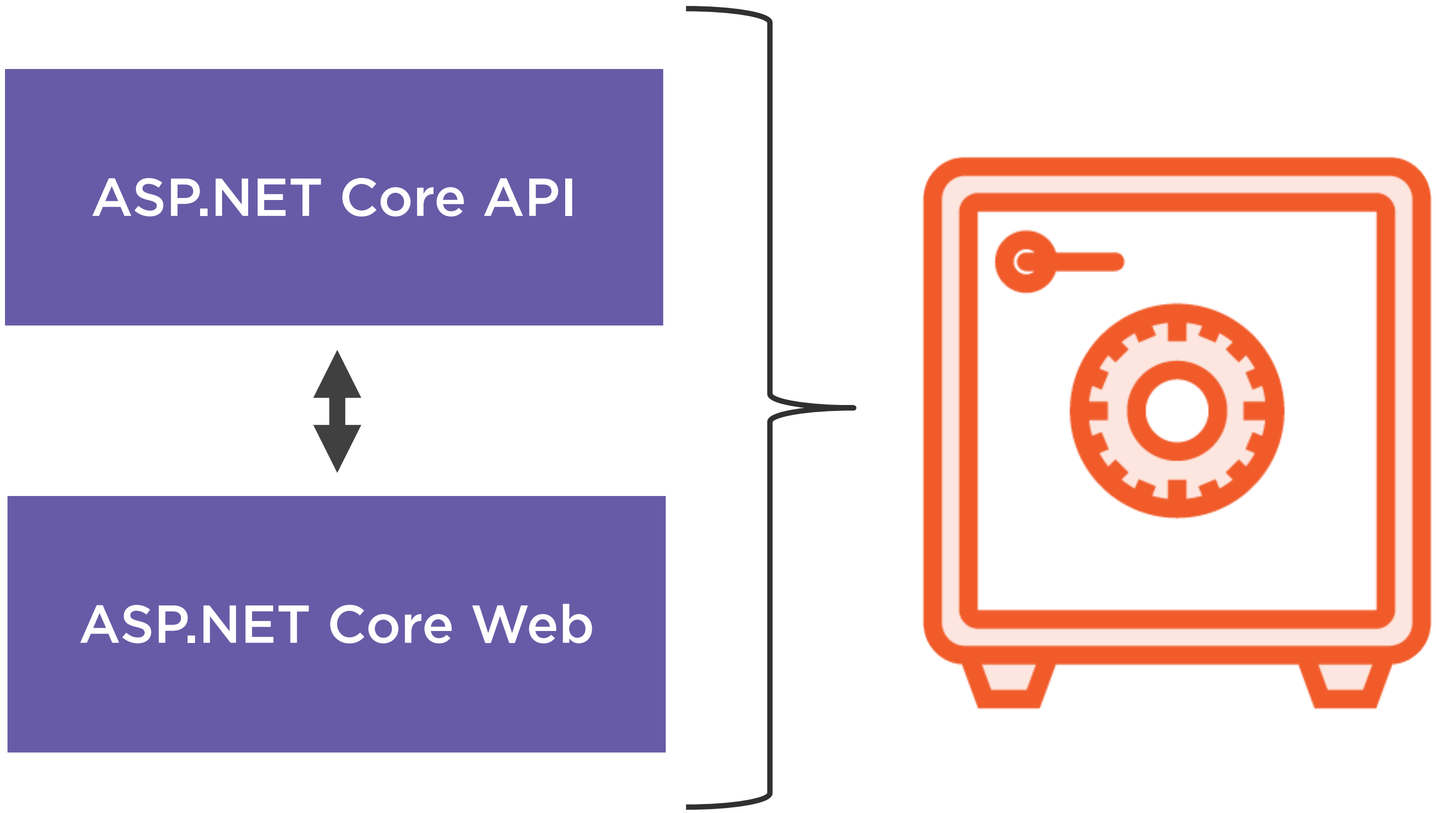
The Identity Provider



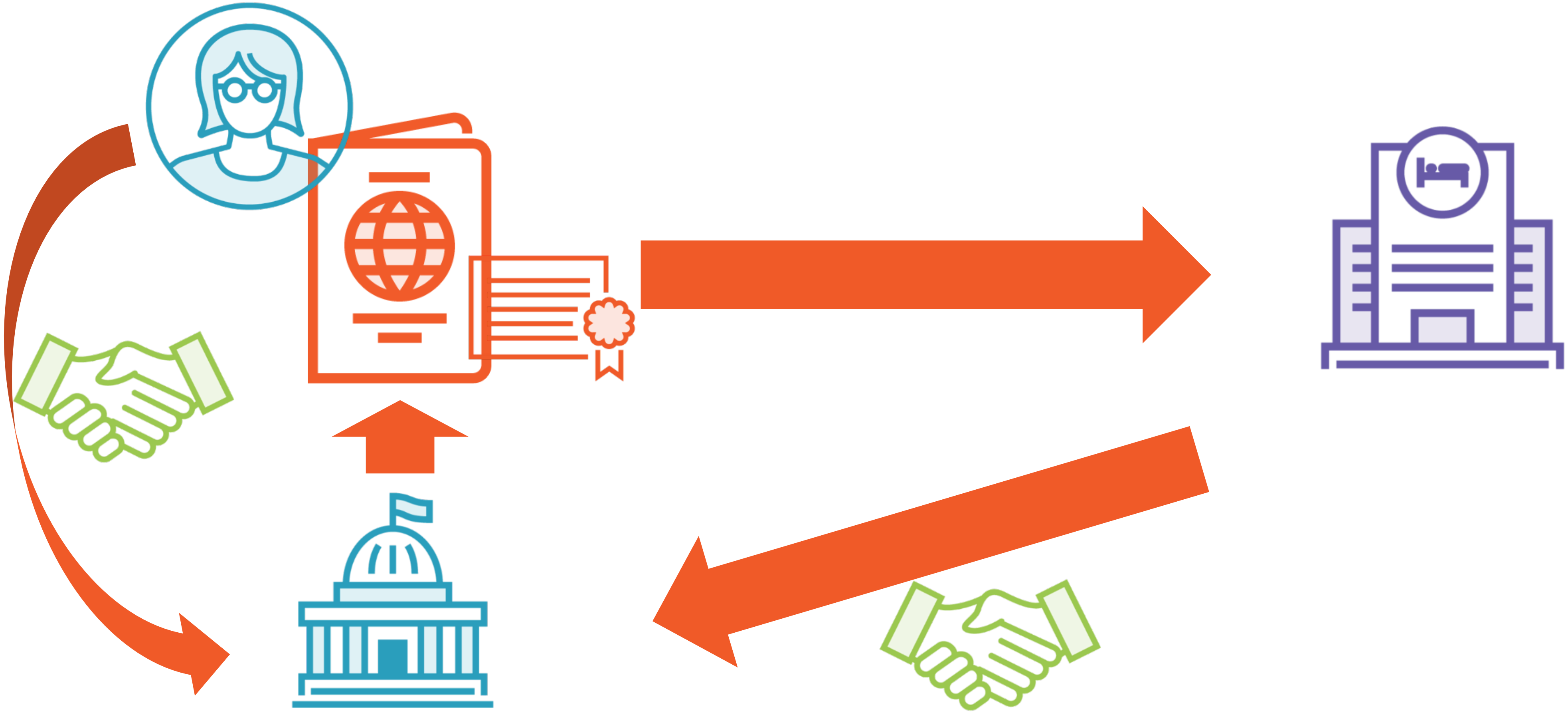
One Identity Provider to Rule Them All



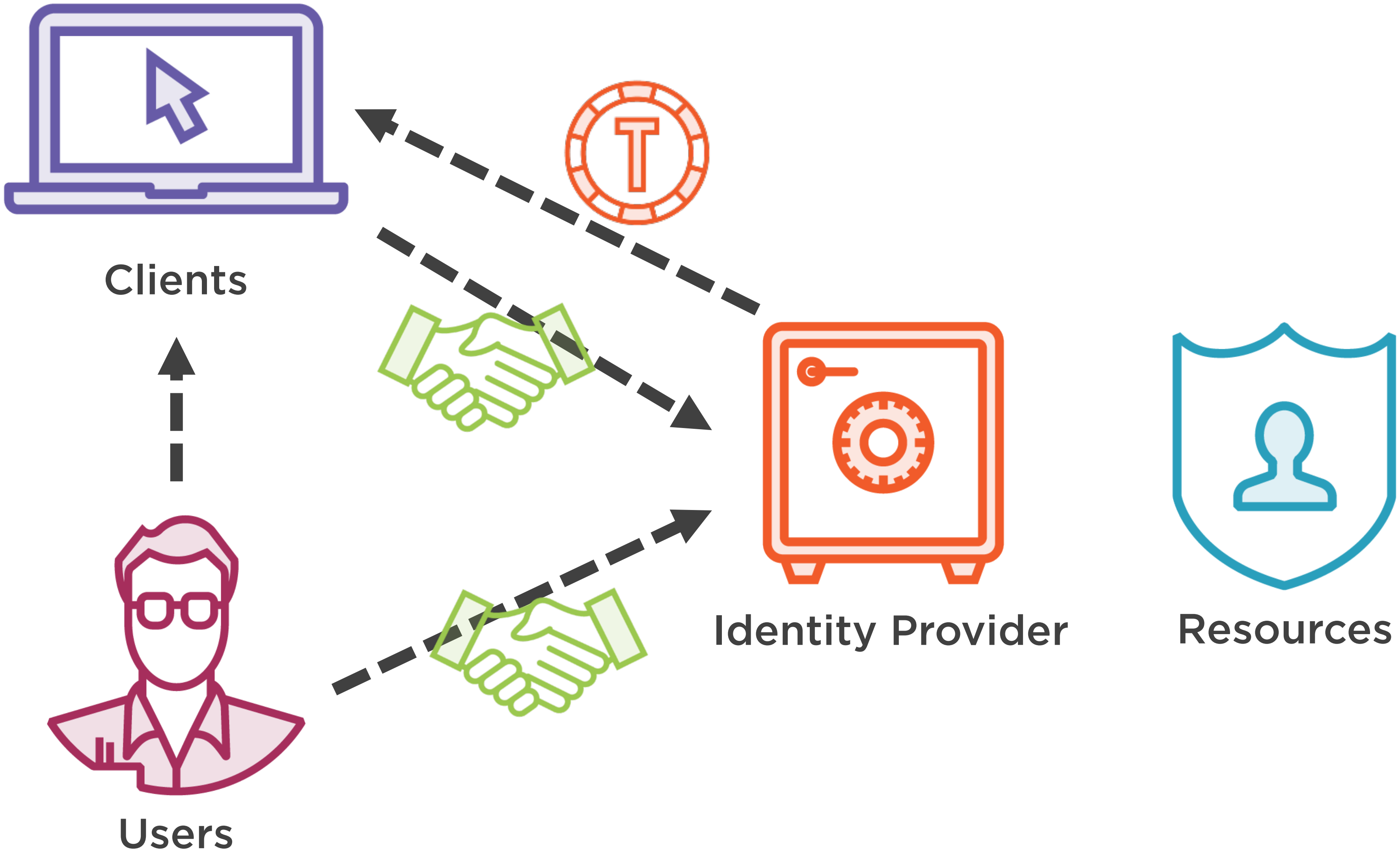
New ConfArch Architecture



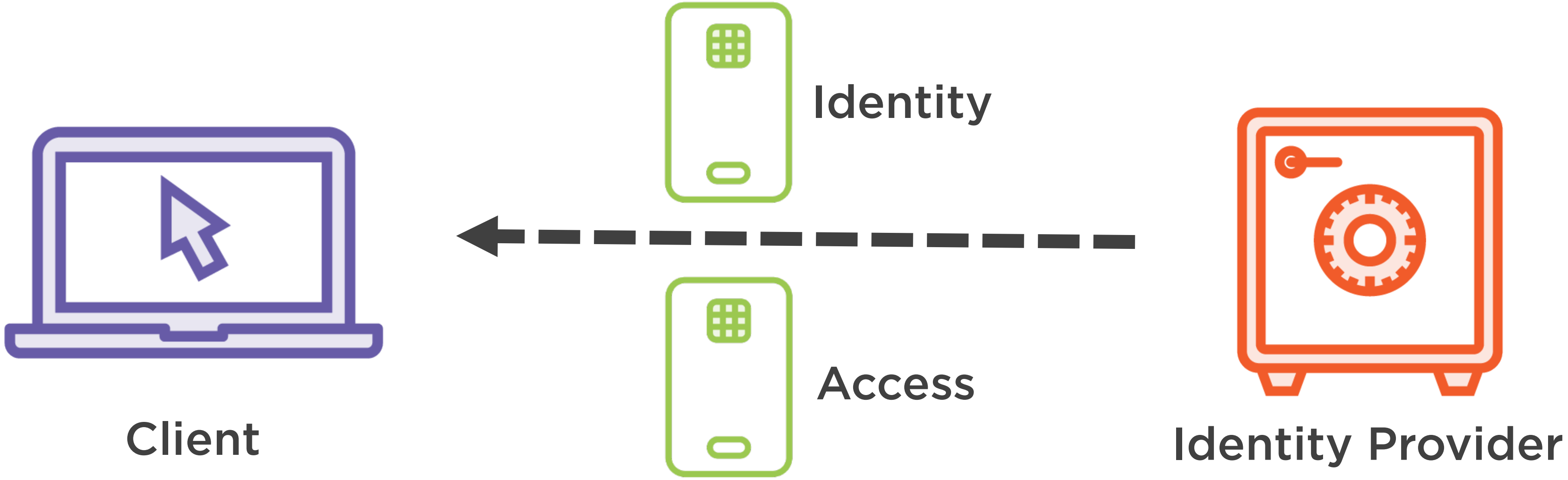
The Process of Authentication Enhanced



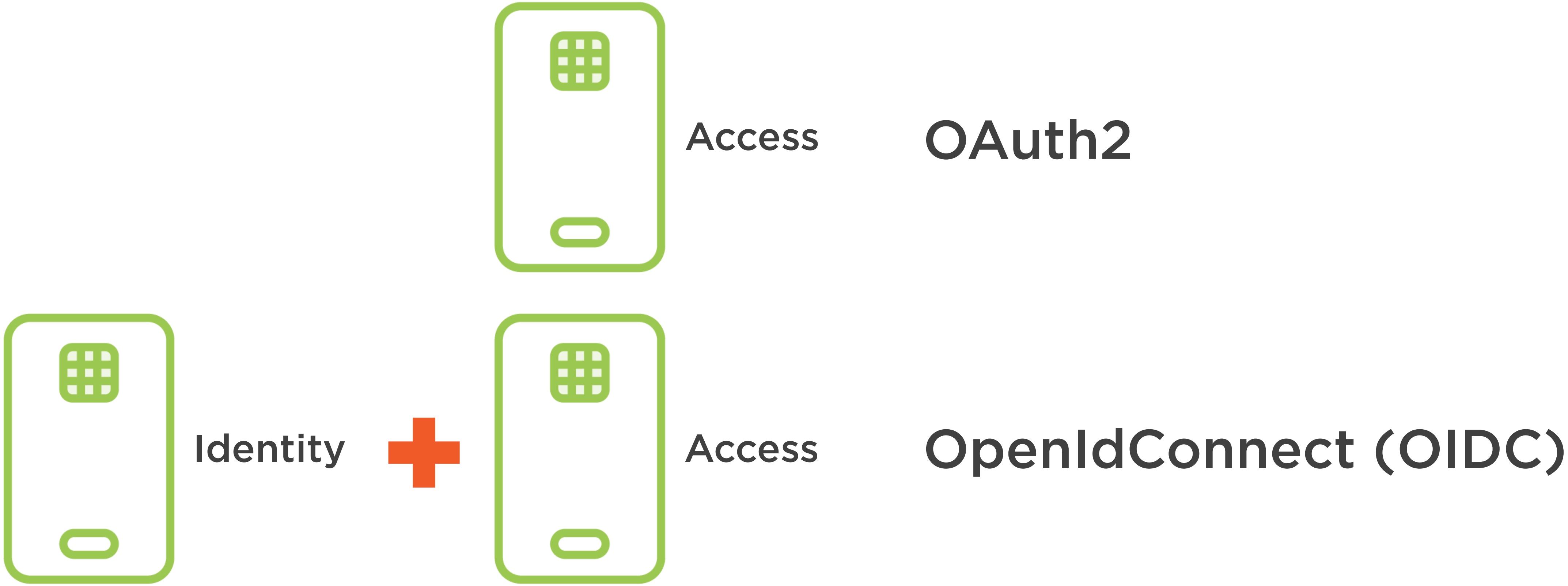
Concepts



Tokens

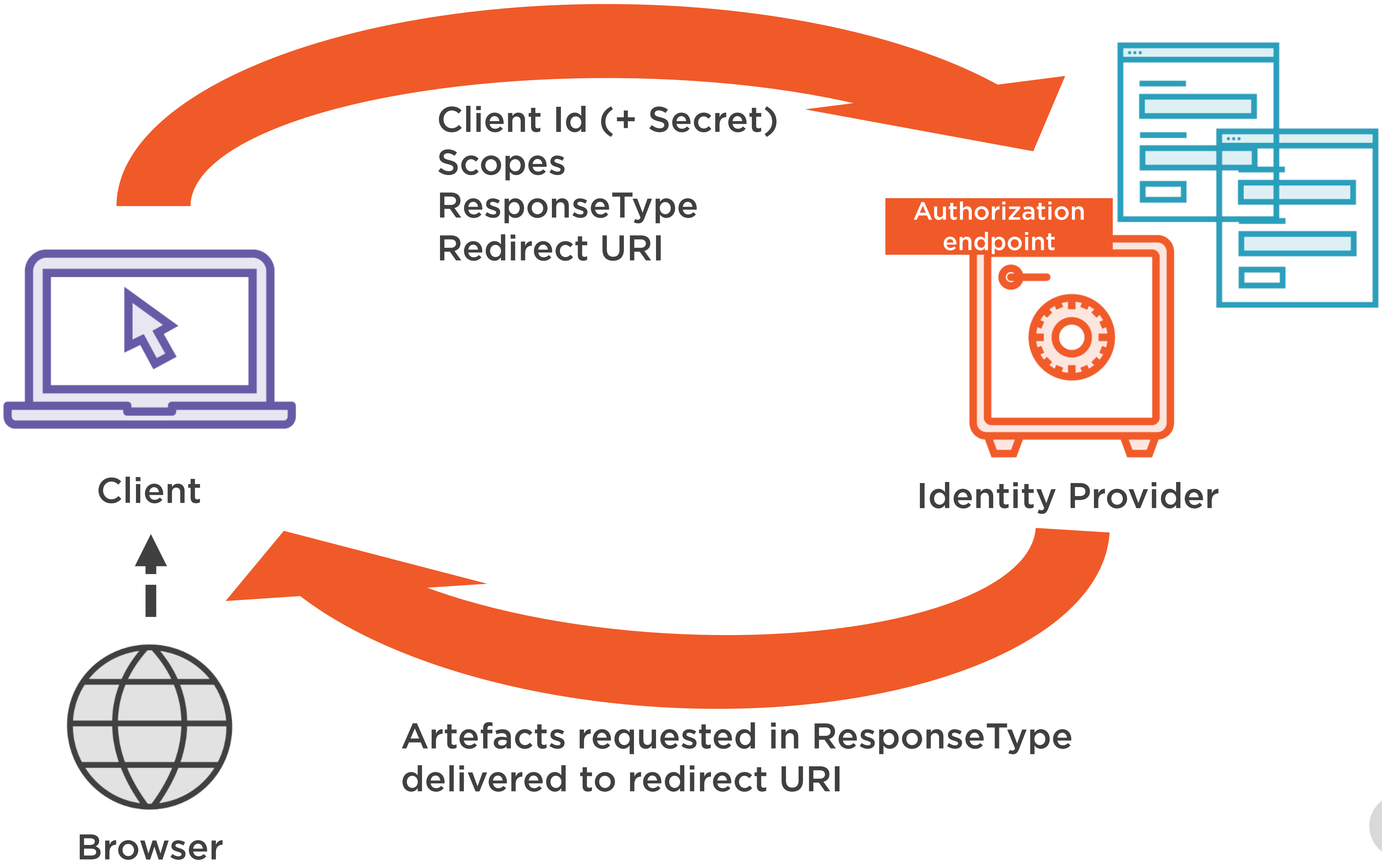


Standards



Access OAuth2

Identity + Access OpenIdConnect (OIDC)



Client Id (+ Secret)
Scopes
ResponseType
Redirect URI

Authorization endpoint

Client

Identity Provider

Browser

Artefacts requested in ResponseType delivered to redirect URI

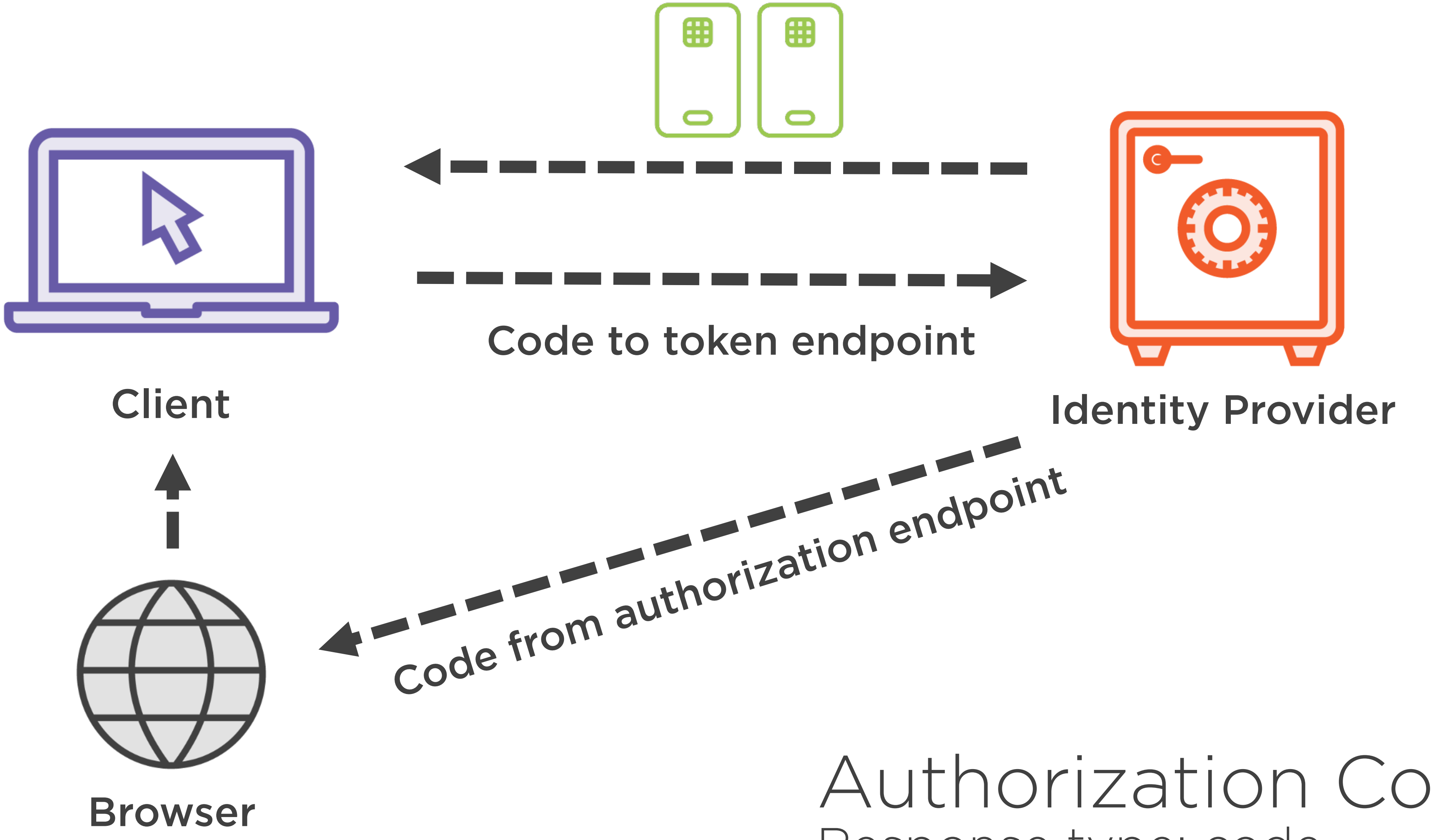
Front Channel

Interaction with authorization endpoint via browser

Redirects

Form post rather than query string

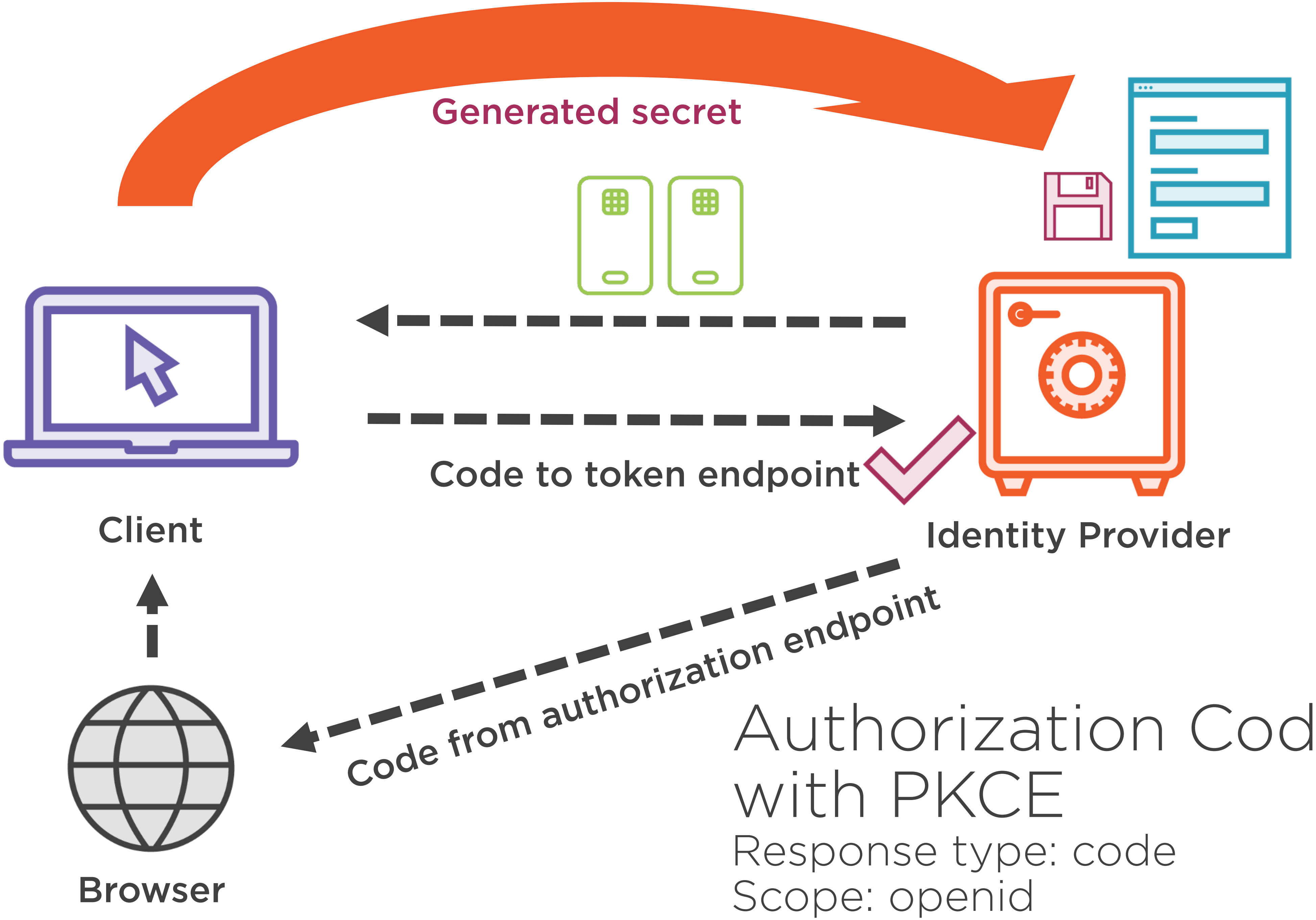
Front channel considered unsafe



Authorization Code

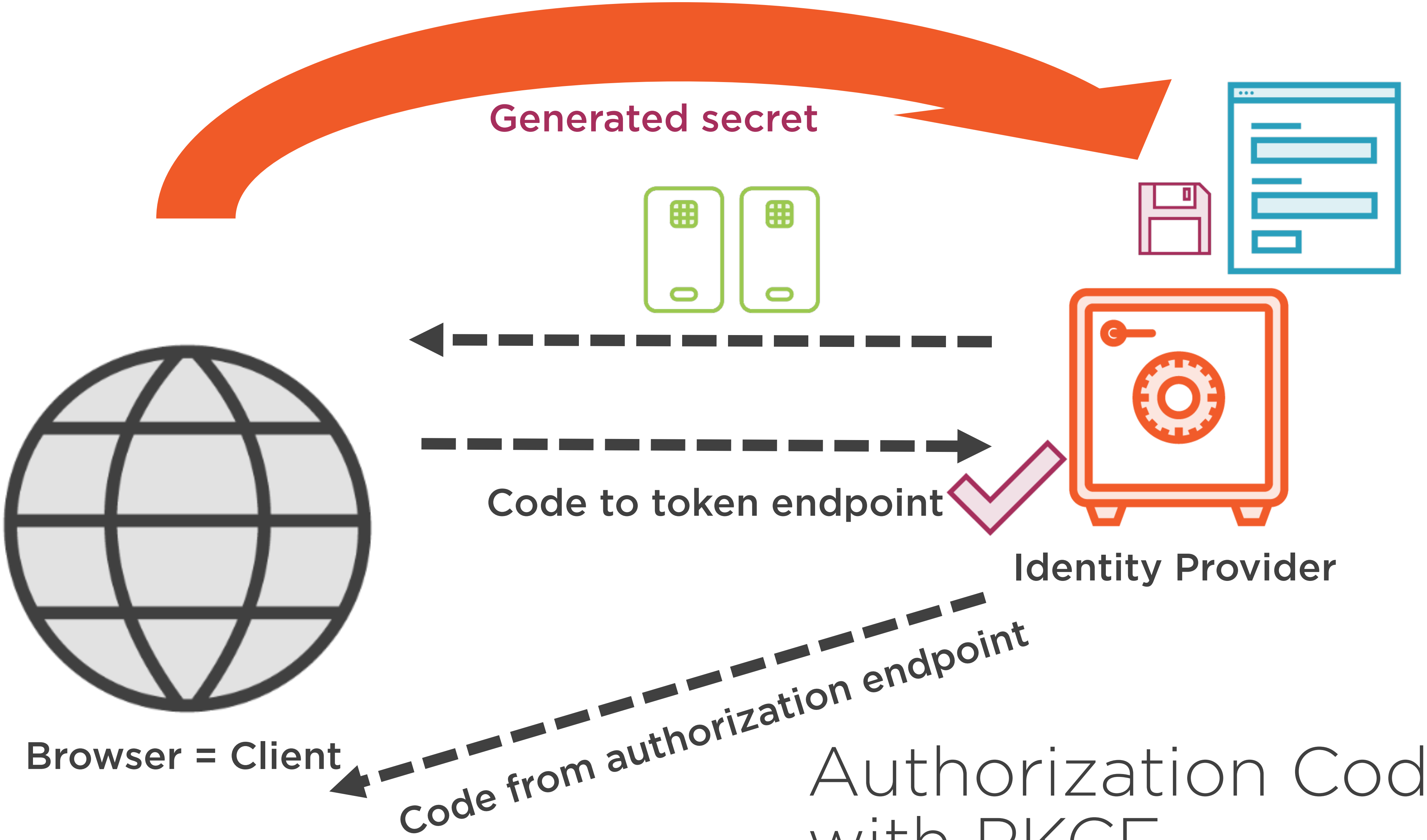
Response type: code
Scope: openid

https://4sh.nl/PkceSpec



Authorization Code with PKCE

Response type: code
Scope: openid



Authorization Code with PKCE

Response type: code
Scope: openid

Single Page Applications

"Back channel" still used

Tokens exposed in browser

Authorization Code flow with PKCE still safer than other options

Public client

Client secret pointless

Can be turned off

Mobile + desktop applications are also public clients

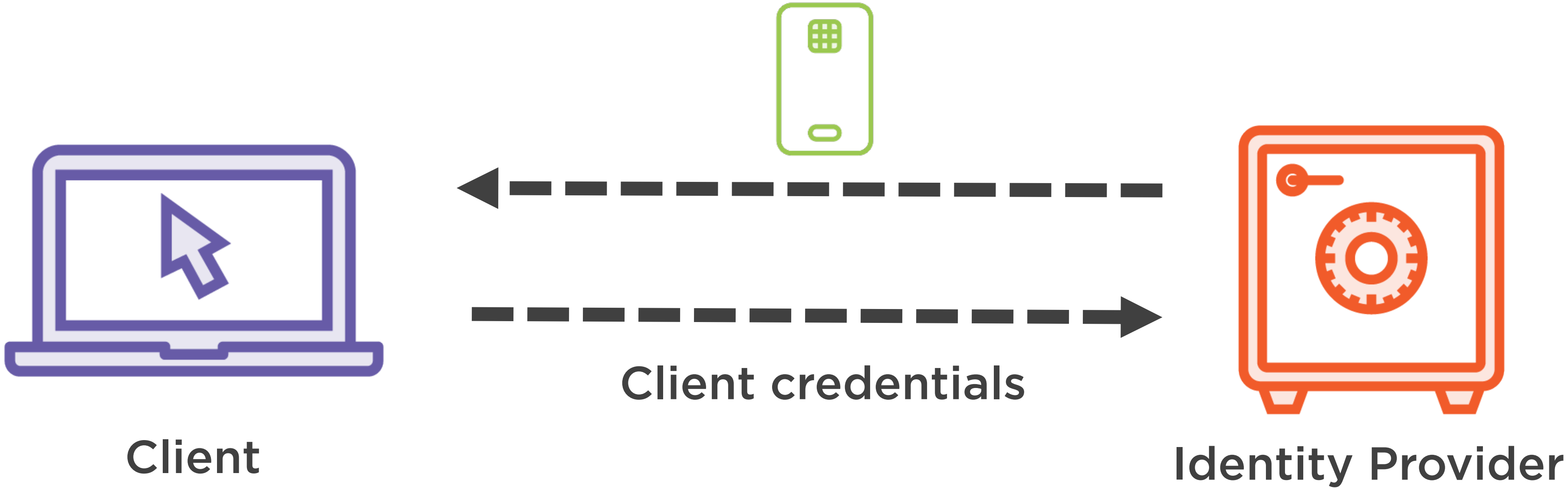
Other Flows

Implicit and hybrid

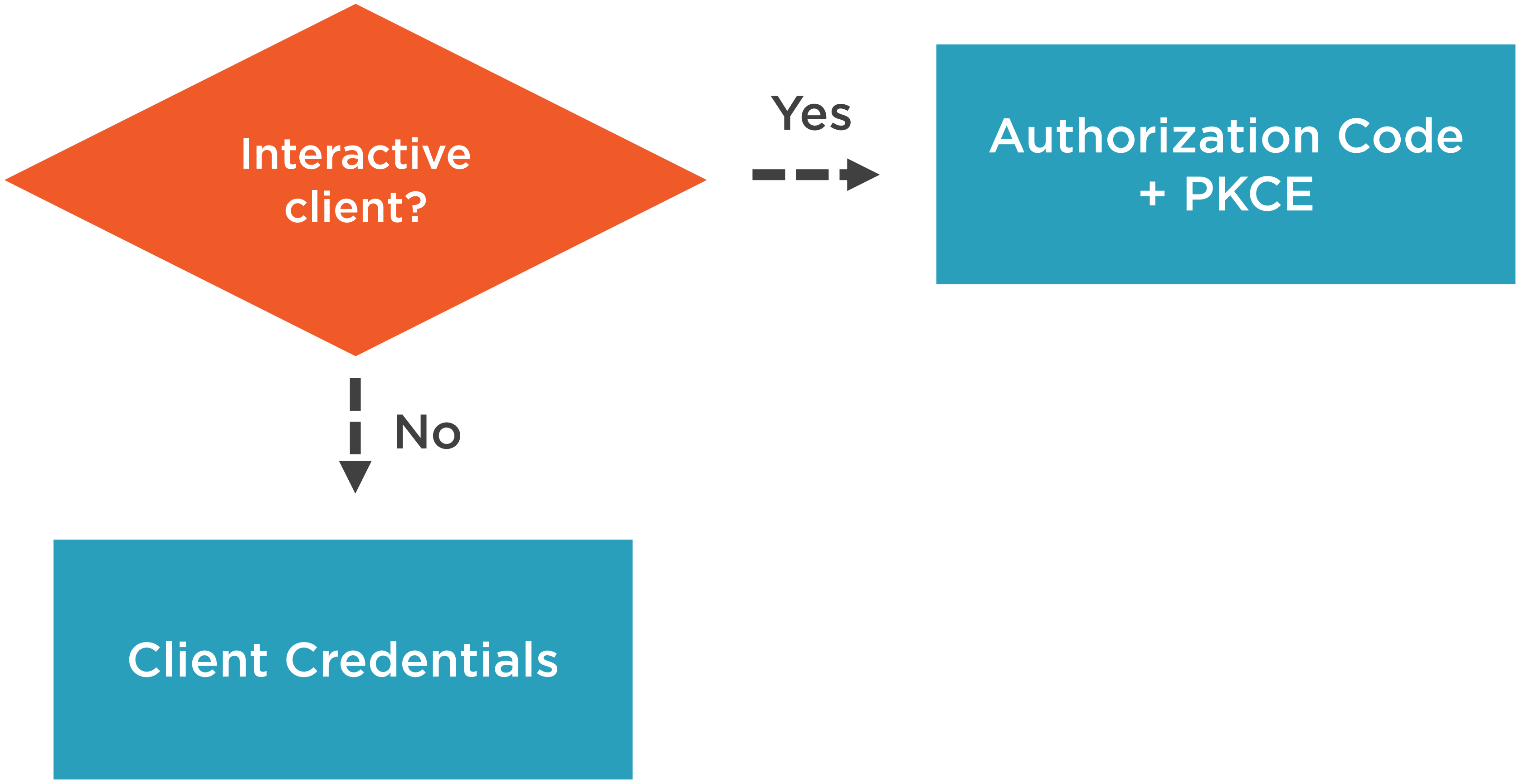
https://4sh.nl/flows

Client Credentials

Client Credentials



Which Flow?



Interactive client?

Yes

Authorization Code + PKCE

No

Client Credentials

Identity Provider Choices

Build your own

Not required

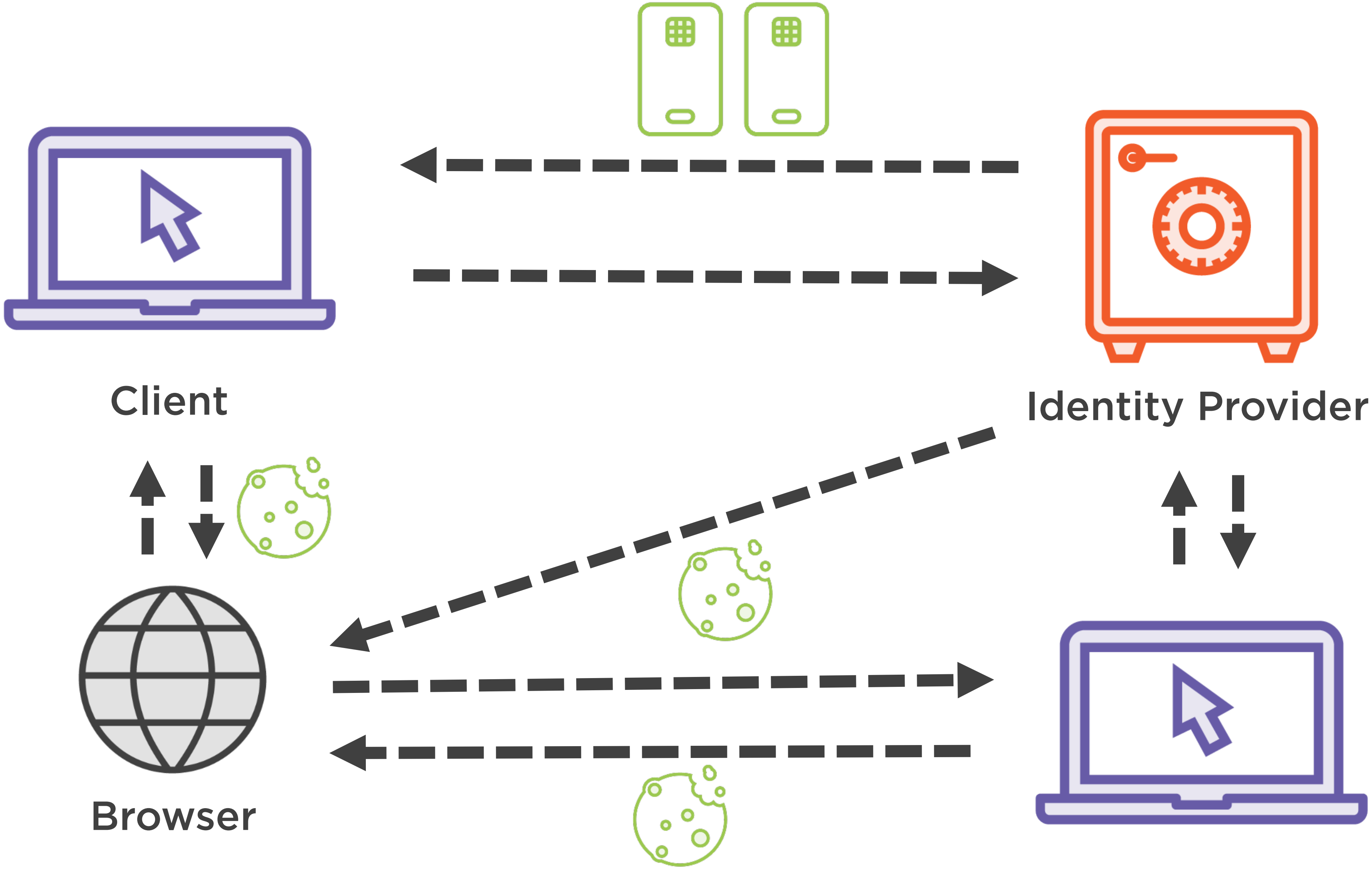
Plenty of cloud identity providers

Building == understanding

Web application leaves authentication to the identity provider https://4sh.nl/IdentityServerTemplates

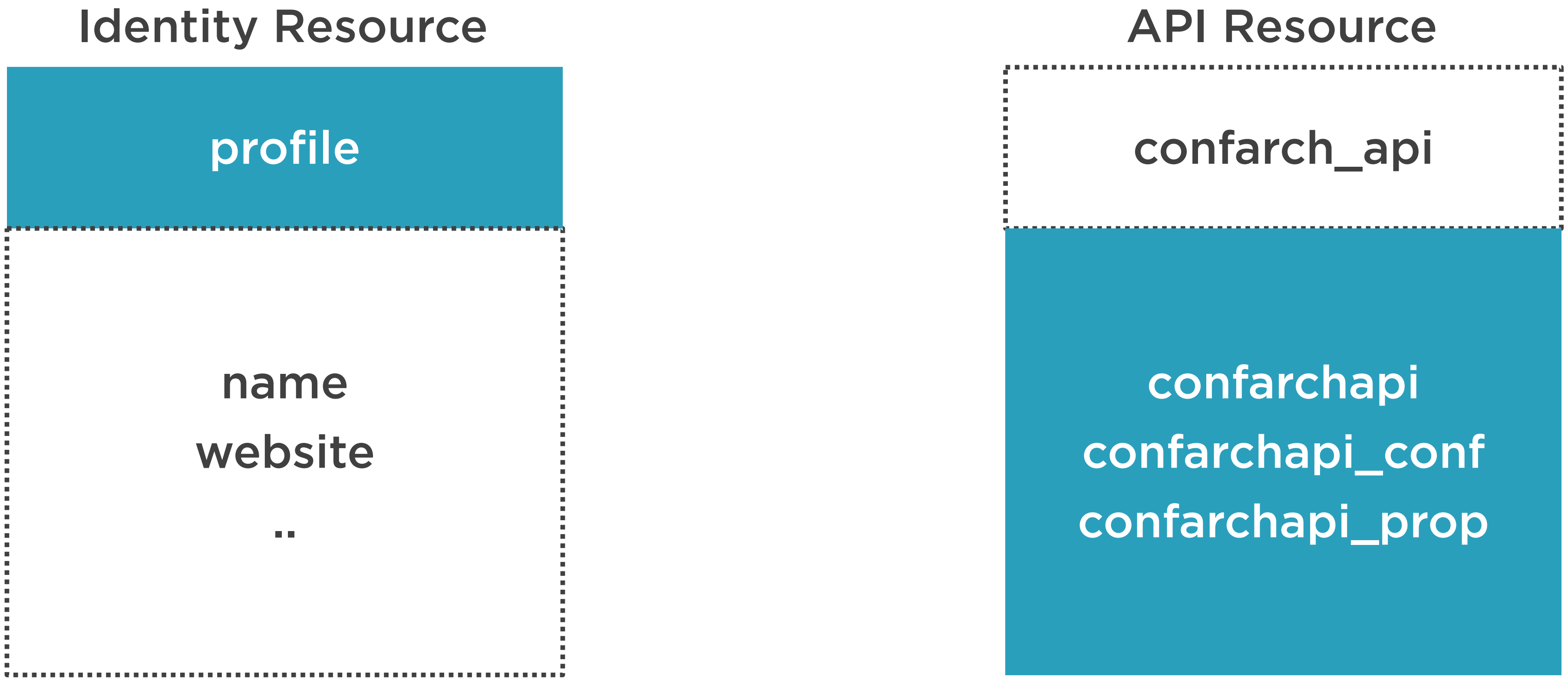
https://4sh.nl/IdentityServerDatabase

Identity Providers, Clients, and Cookies



https://4sh.nl/ClientConfig

Resources and Scopes



Identity Resource

profile

name
website
..

API Resource

confarch_api

confarchapi
confarchapi_conf
confarchapi_prop

The blue is what a client requests

Often-used Client Settings

RequireConsent

AlwaysIncludeUserClaimsInIdToken

RequireClientSecret

Client Credentials Example

Outside organization sells tickets

Each time ticket is sold it calls the ConfArch API

Any modern application platform can do the call

For a .NET application: IdentityModel

Extension methods on HttpClient

Token Verification by the Client

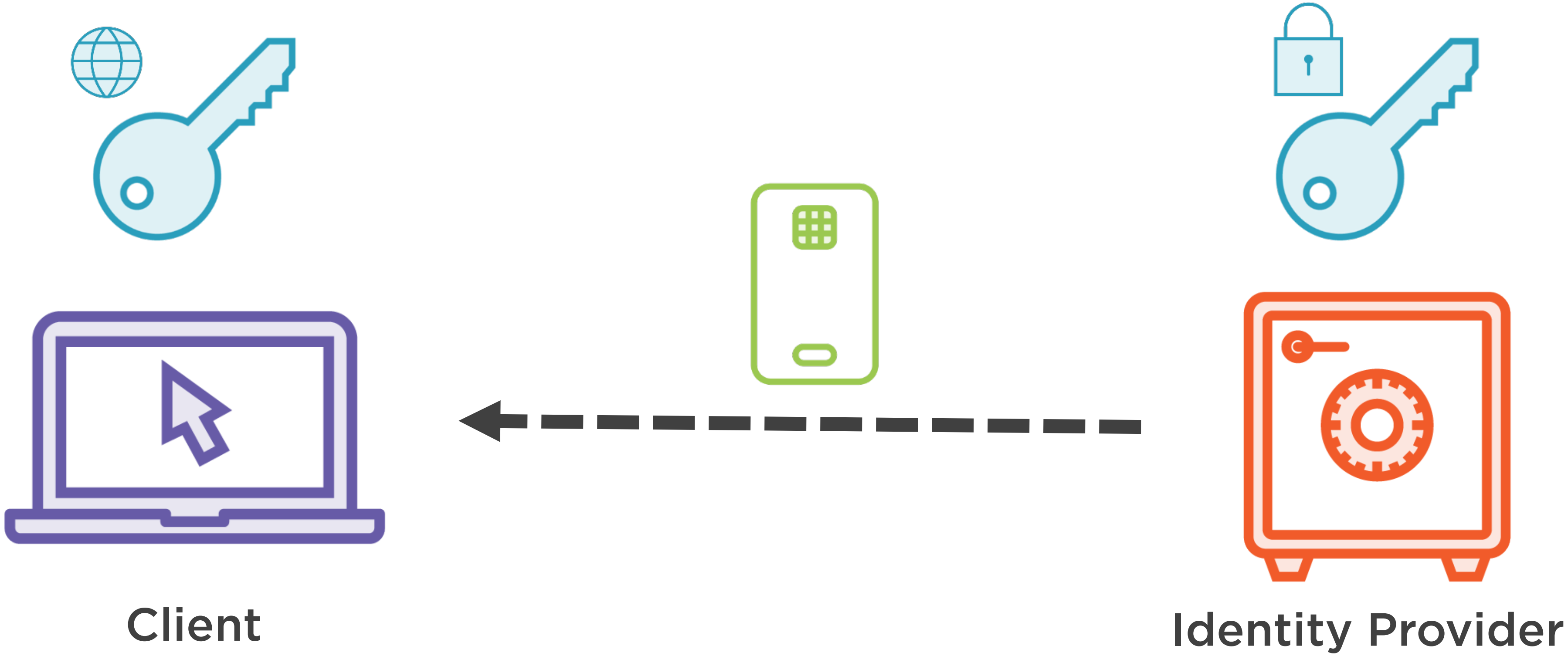
Did the token come from the trusted authority?

Is the contents as the authority issued it?

Token Verification by the Client

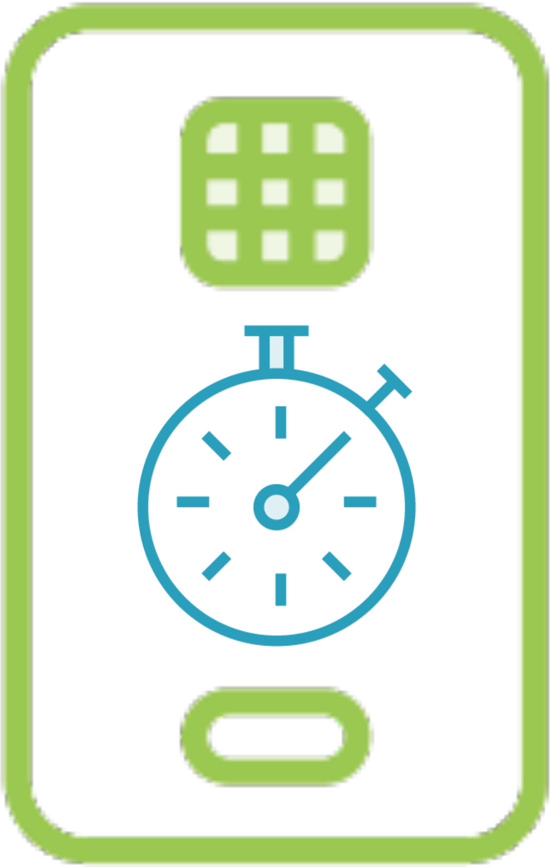
1. Identity Provider creates hash of contents
2. Hash is encrypted using private key
3. Attaches result (== signature) to token
4. Client uses public key to decrypt hash
5. Readable contents is hashed
6. Compares own hash with decrypted hash

Tokens



https://4sh.nl/IdentityServerCert

Self-contained Access Tokens



Refresh Tokens

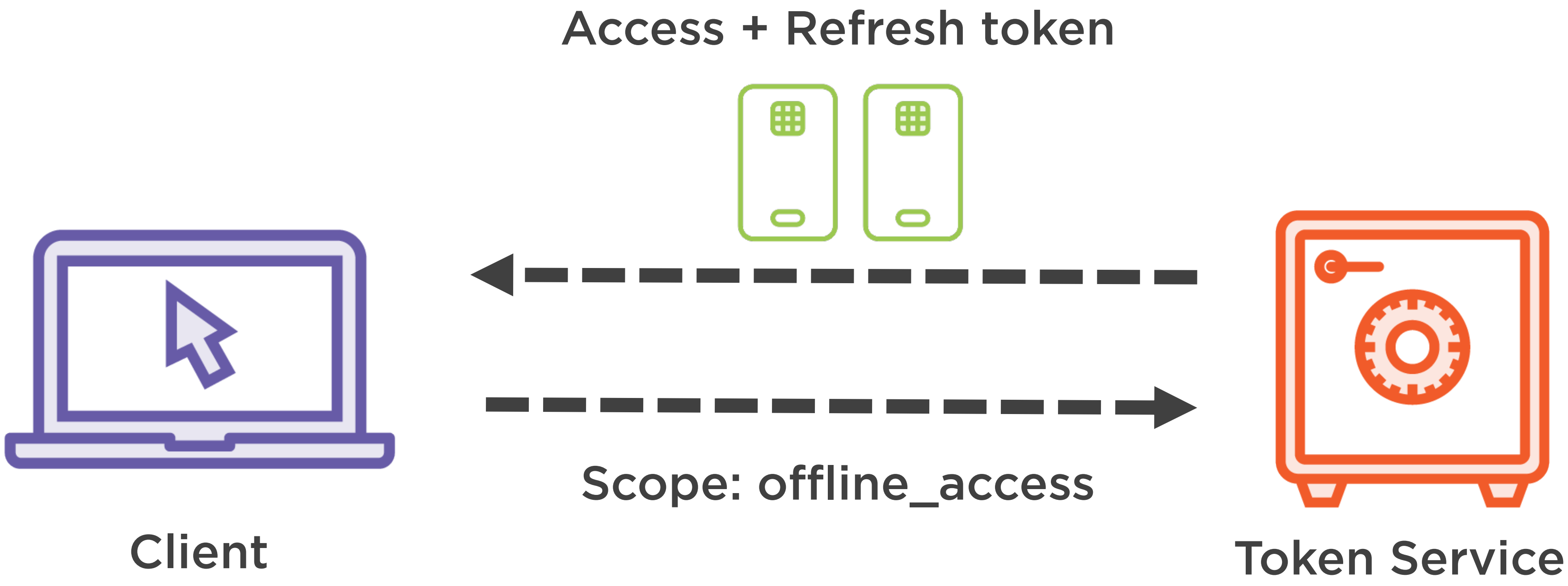
Seperate token

Used to get new access token

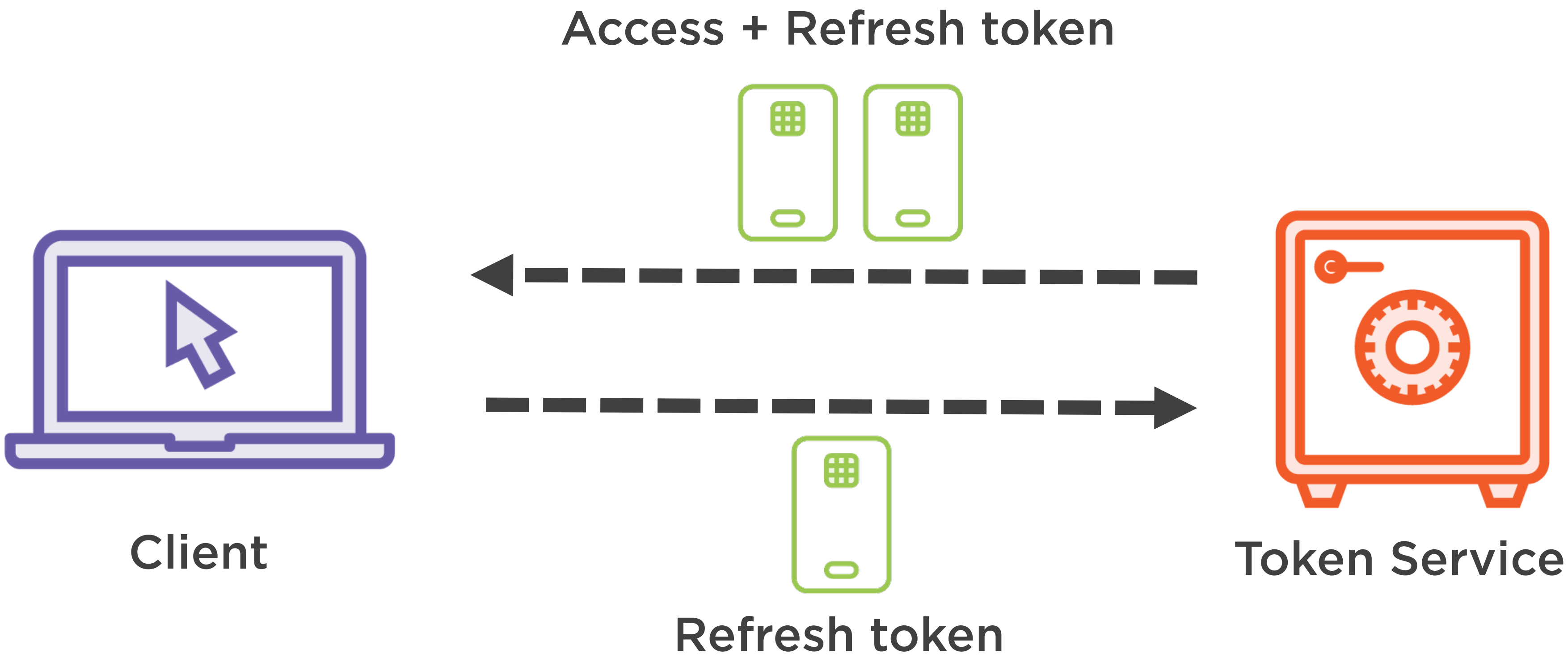
User doesn’t have to re-authenticate

Longer expiration time than access token

Refresh Tokens



Refresh Tokens



Reference Tokens

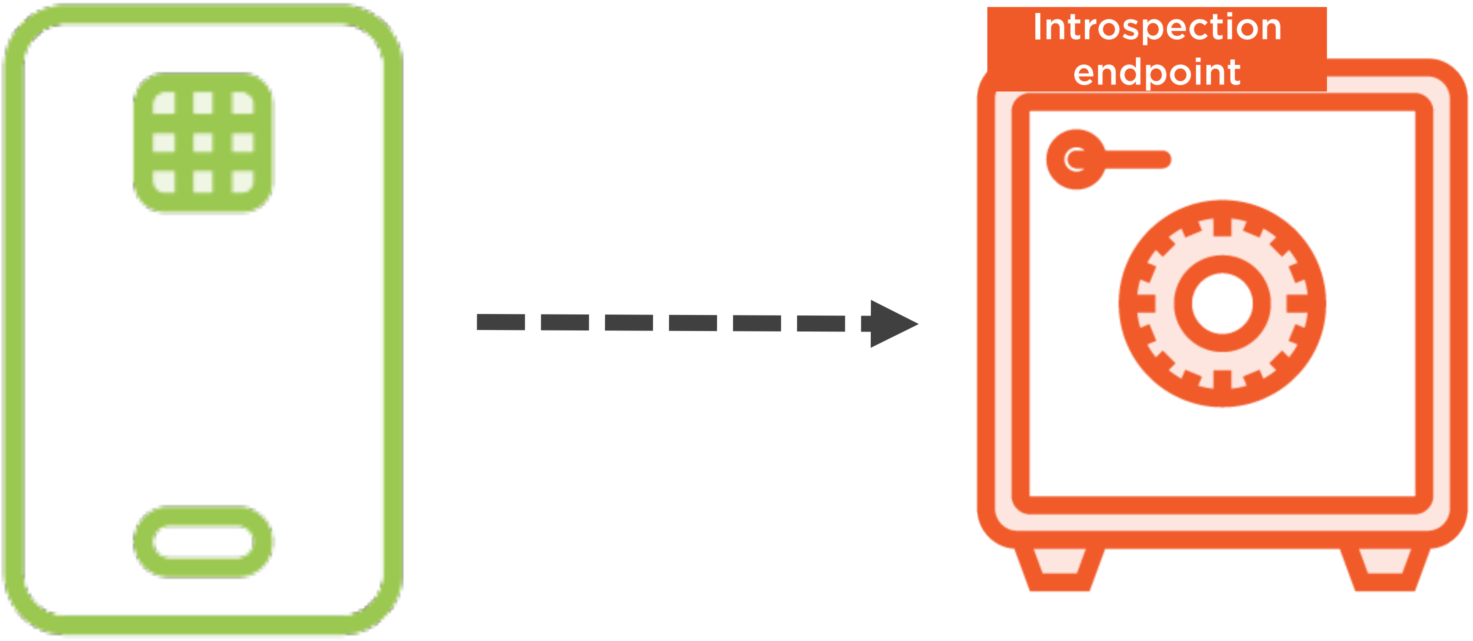
Alternative for refresh tokens

Don't contain claims

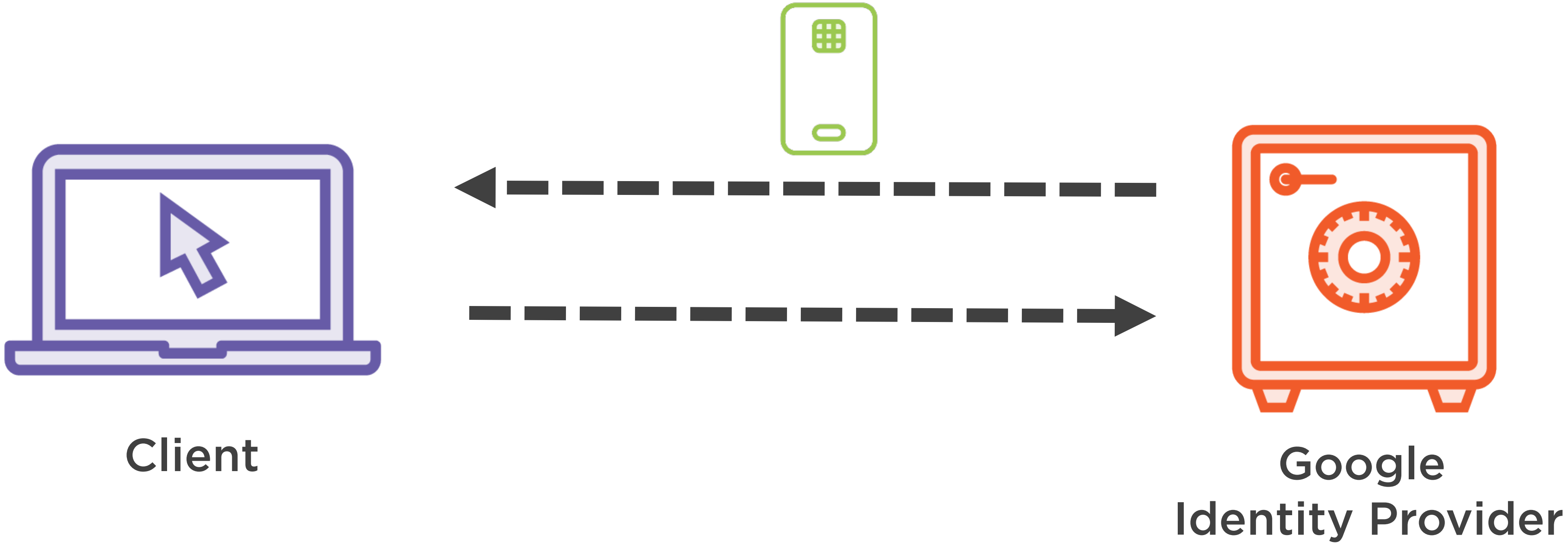
Claims are kept on the identity provider

Contain a unique ID

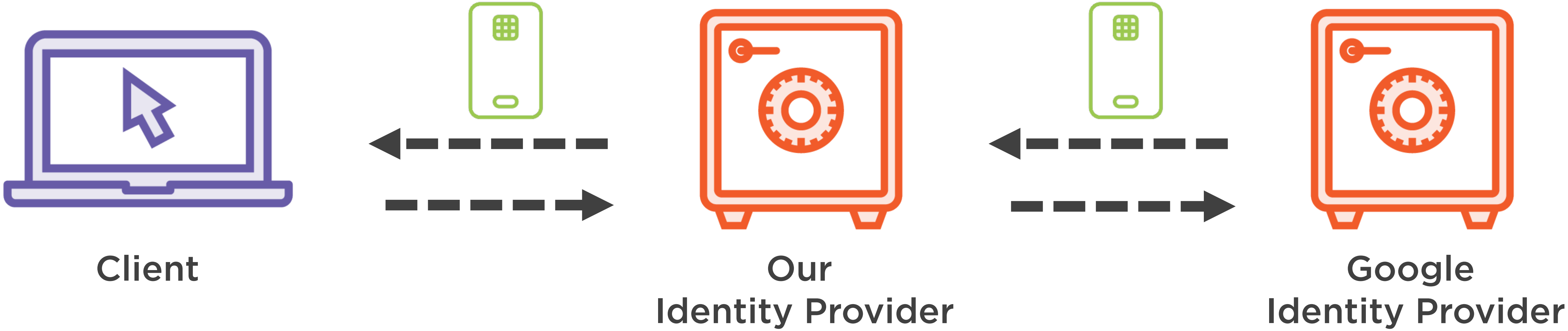
Reference Tokens



Adding Google's Identity Provider



Adding Google's Identity Provider

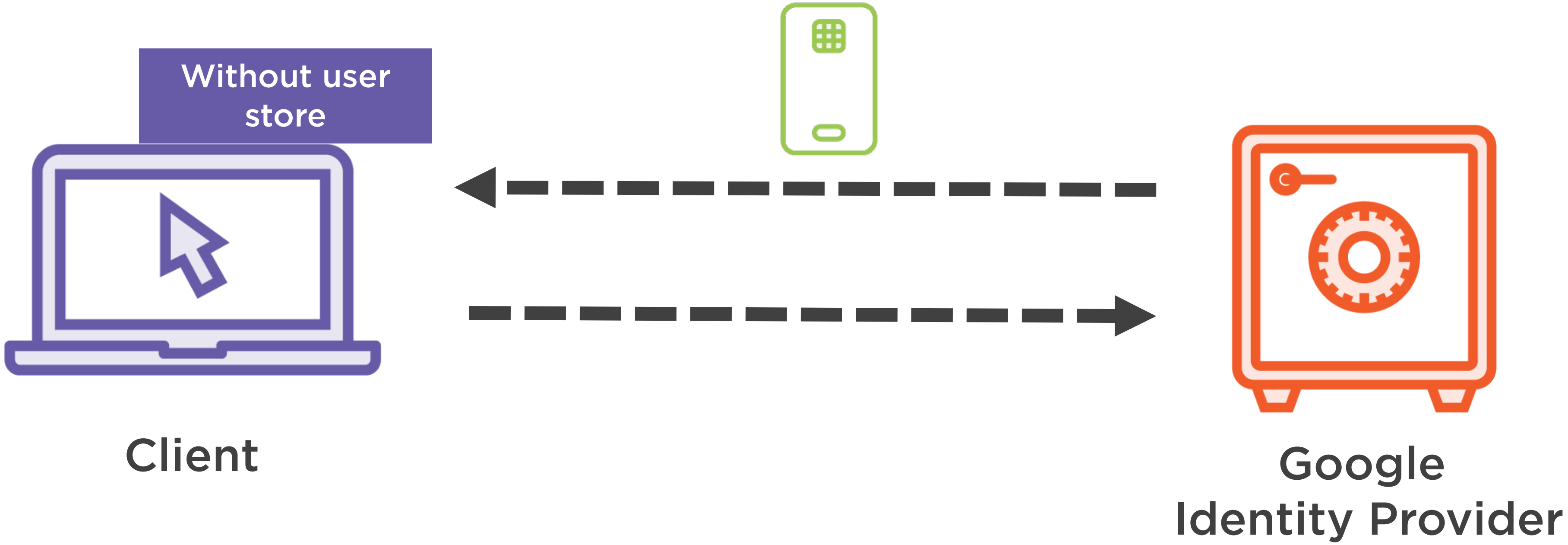


Client

Our
Identity Provider

Google
Identity Provider

Adding Google's Identity Provider



Other Cloud Identity Providers

Azure Active Directory (AAD)

Auth0

Okta

Summary

OpenIdConnect concepts

Client authentication process

Authorization code + PKCE flow

IdentityServer

Protecting APIs

Client credentials flow

Refresh and reference tokens

Cloud identity providers