Tweakers Developers Summit

Getting started
with OAuth2.0 and OpenID Connect
in a web API strategy

Achmea IT | Robert Jan & Michel

Utrecht, 15-02-2018

achmea

Who we are

Michel Wolvekamp
Domain architect Integration
Mobile & security

Robert Jan van Holland
Domain architect Microsoft
API & Portals

Who has an insurance at Achmea?

Achmea exists of 24 brands.
Chances are big that you have an insurance at Achmea!

Power brands	Propositie merken	Labels	Service instellingen	Buitenland
Centraal Beheer Achmea	FBTO	ZORGVERZEKERAAR OZF	Pim Mulier	Union
	avéro achmea	syntrus achmea	LEEFSTIJL TRAINING & COACHING	Friends First
Interpolis	inshared	PROLIFE zorgverzekeringen	eurocross assistance	EUREKO SIGORTA
	independer	Hagelunie	WINNOCK	INTERAMERICAN
Zilveren Kruis	DE FRIESLAND ZORGVERZEKERAAR	Woonfonds	work away	
		Staalbankiers		

achmea

Agenda

- Brief introduction to OAuth2.0 and OIDC
- High level architecture
  - Building blocks
  - Auth0
- Regular web application + DEMO
- Single Page web application + DEMO

Brief introduction to OAuth2.0 and OIDC

Oauth2.0
- Roles: Client, Resource Server, Resource Owner, Authorization Server
- Grants:
  - Authorization Code
  - Implicit
  - Resource Owner Password Credentials
  - Client Credentials

OIDC
- Roles: Relying Party (Client), OpenID Provider (Authorization Server)
- Flows:
  - **Authorization code flow**
  - **Implicit flow**
  - Hybrid flow

JWT (JSON Web Token)
- ID Token, Access Token, Refresh Token

JWT contains a header, payload and signature.

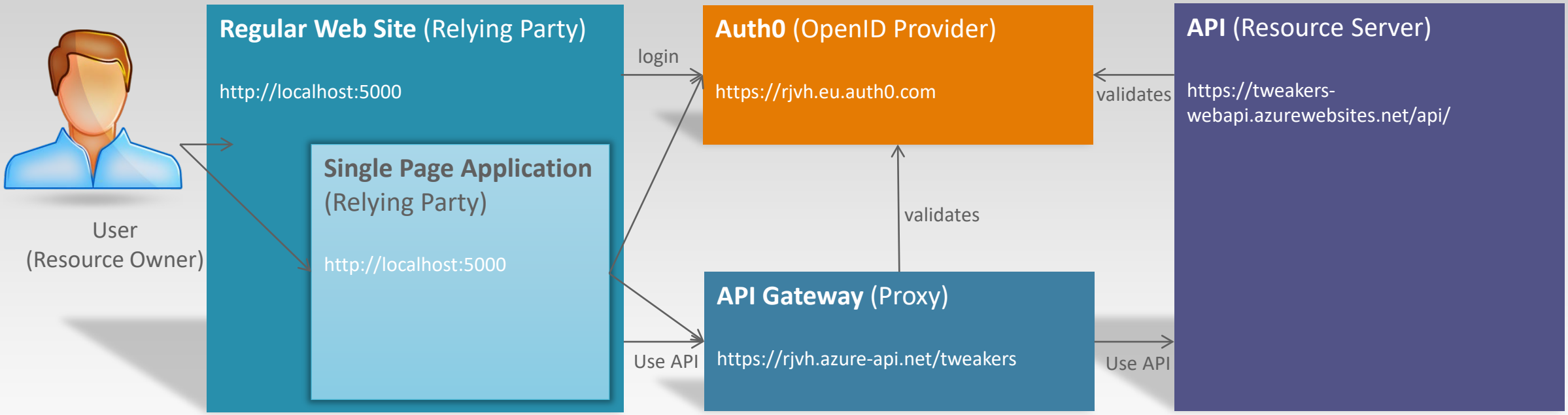
Digitally signed.

Sent as bearer token.

Access token has an audience (API it is intended for).

achmea

High level architecture



achmea

Building blocks

Regular Web Site (Client)

ASP.NET Core 2.0

SPA (Client)

JS

Auth0 (Authorization Server)

Auth0

API Gateway (proxy)

Azure API Management

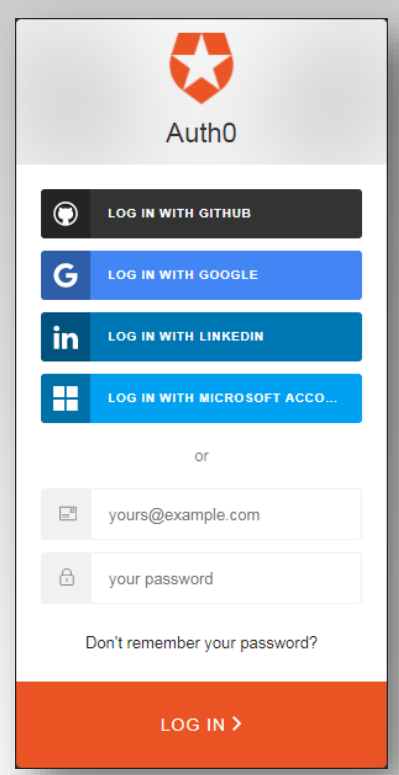
API (Resource Server)

ASP.NET Core 2.0

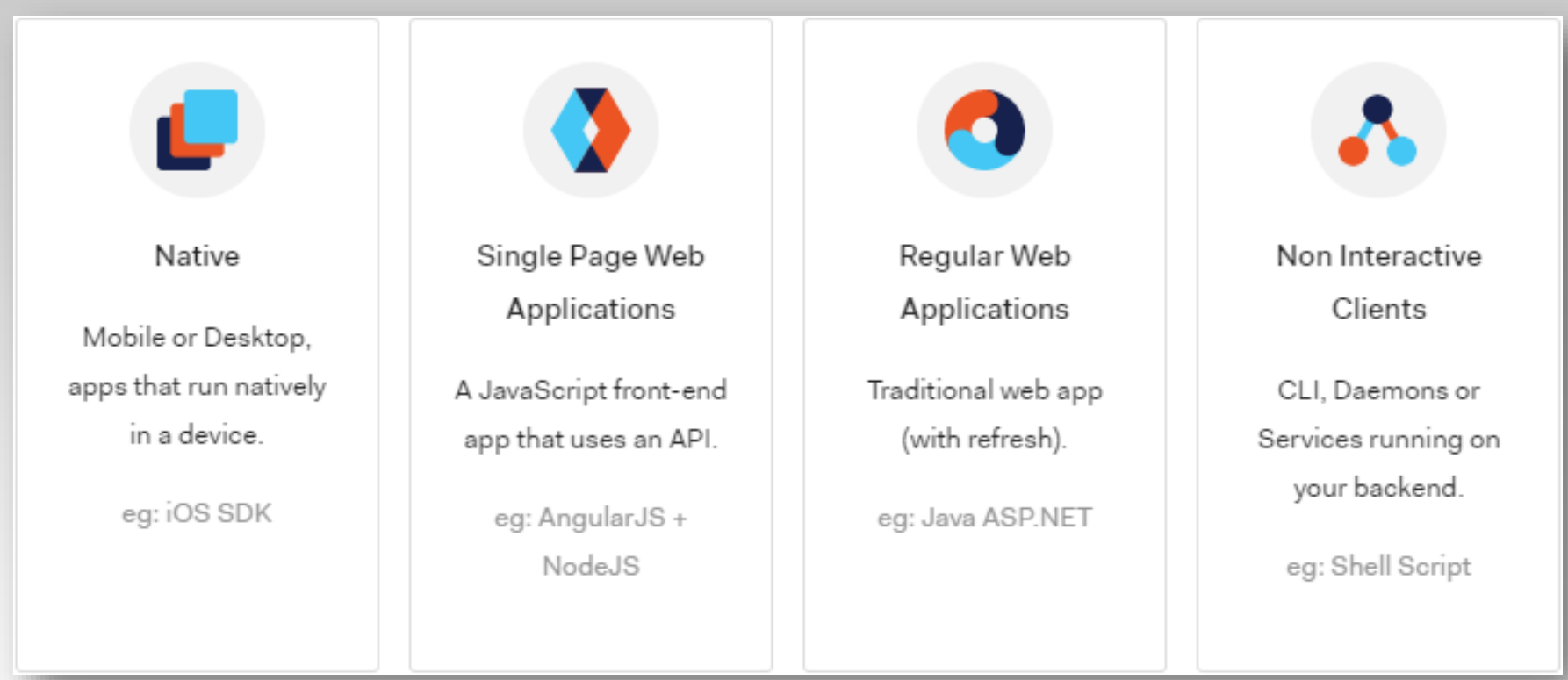
User

achmea

Auth0



OpenID Provider



Authorization code (PKCE)

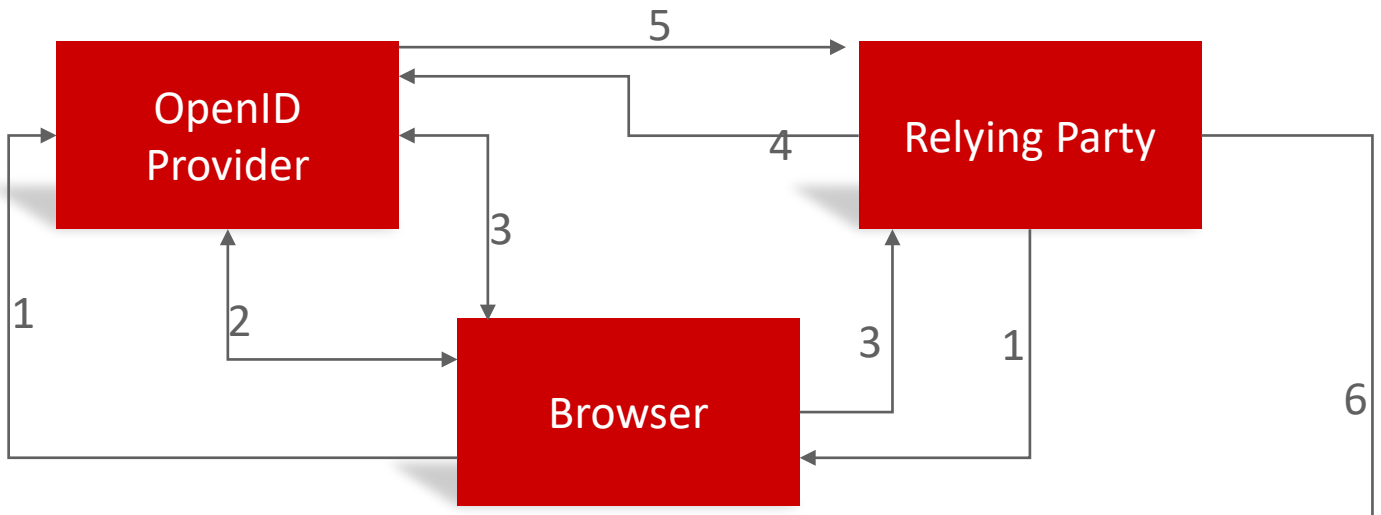
Implicit

Authorization code

Client Credentials

achmea

Regular web application



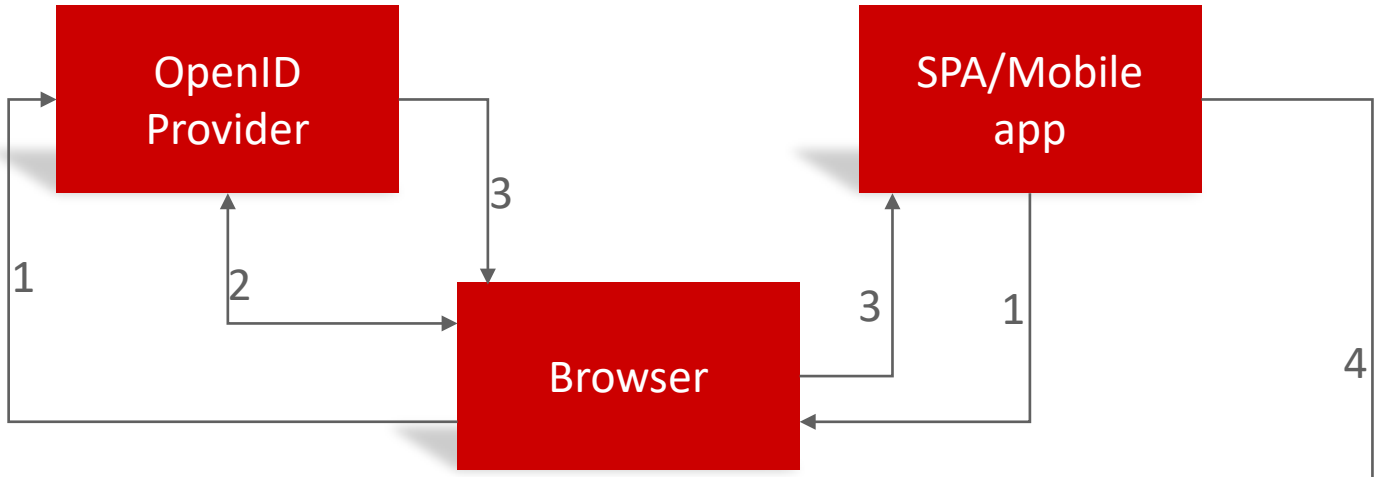
1. Web app initiates flow & redirects to Auth0 for authentication
2. Auth0 authenticates user; shows consent window
3. Auth0 redirects user to web app with authorization code
4. Web app sends authorization code and asks for access token (and id token and refresh token). Web app authenticates using clientid & secret)
5. Auth0 authenticates web app, validates Authorization Code and responds with token(s)
6. Web app uses access token to call API on behalf of user

- Relying Party (live build)
- API (hosted on Azure)
- API GW (hosted on Azure)
- OpenID Provider (SaaS)

achmea

DEMO

SPA



1. App initiates flow & redirects to Auth0 for authentication
2. Auth0 authenticates user; shows consent window
3. Auth0 redirects user to app with access token (and id token) in the URI
4. App uses access token to call API on behalf of user

- Relying Party (live build)
- API (hosted on Azure)
- API GW (hosted on Azure)
- OpenID Provider (SaaS)

achmea

DEMO

Questions?

Getting started
with OAuth2.0 and OpenID Connect
in a web API strategy

Achmea IT | Robert Jan & Michel

Utrecht, 15-02-2018

achmea