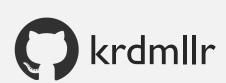
AN INTRODUCTION TO AUTHENTICATION WITH OPENID CONNECT

- Konrad Müller
- Xamarin, ASP.NET Core,
 Cloud stuff





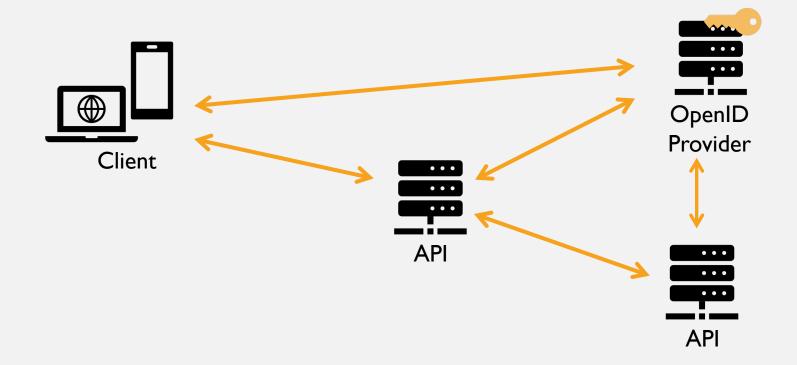


SPIRIT/21

- What is OpenID Connect and how does it work?
- How can I integrate it in my app and backend services?

A LITTLE OVERVIEW

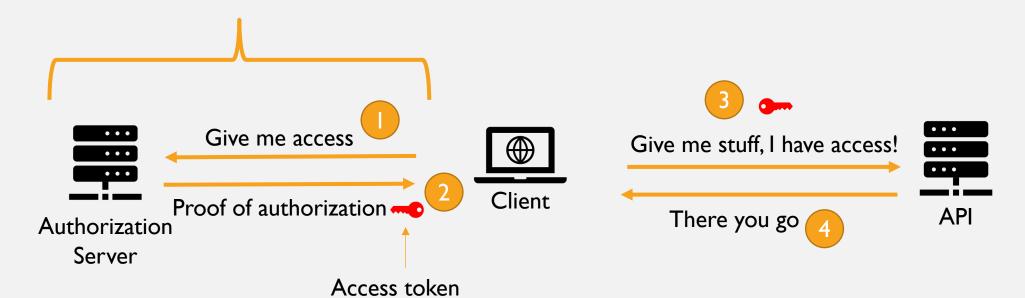
OPENID CONNECT IN YOUR SYSTEM



OAUTH 2 IN A NUTSHELL

- Standarized by IETF RFC 6749
- Authorizes resource access
- Goals:
 - Avoid sharing passwords
 - Limit access to resources
 - Ability to revoke access
- Single Sign-On ... kinda ...

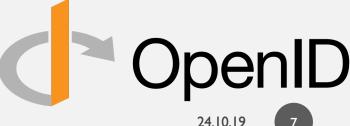
Authorization Grant/Flow



 \rightarrow You have no idea about the users identity

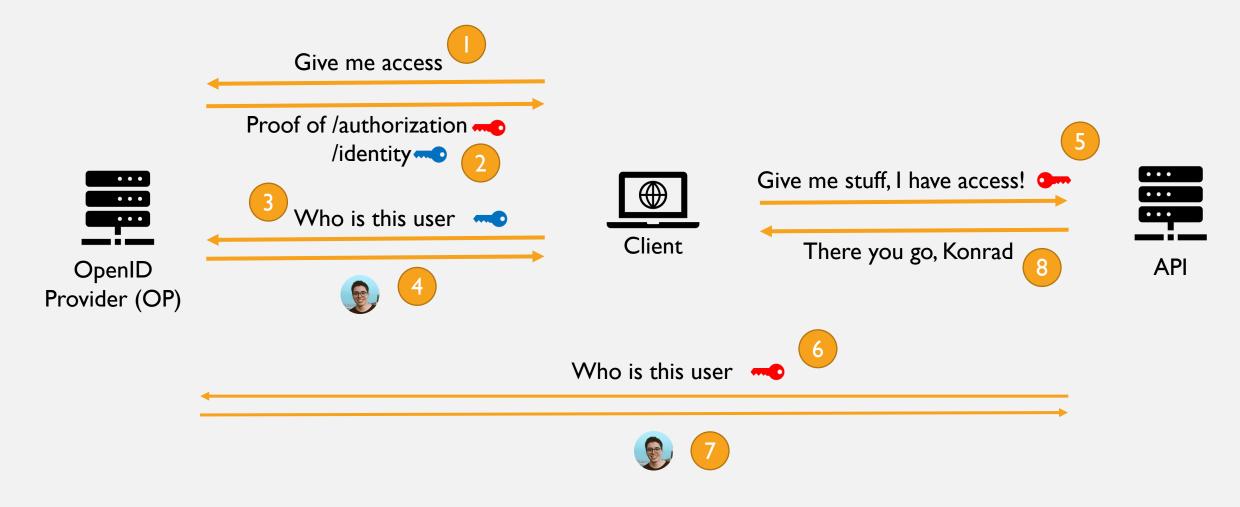
OPENID CONNECT TO THE RESCUE!

- Authentication layer on top of OAuth 2
 - Defines open specifications of Auth2
 - Identity information (Identity Token, User-Info endpoint)
 - Additional flows
- Centralized authentication and authorization
- Widely used and standardized by the OpenID foundation
- Single Sign-On!









ABOUT TOKENS

- Json Web Token (JWT) Bearer Token
 - Meta
 - Body
 - Signature
- Base64 encoded

eyJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWd kazcifQ.ewogImlzcyI6ICJodHRwOi8vc2V ydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOi AiMjQ4Mjq5NzYxMDAxIiwKICJhdWQiOiAic zZCaGRSa3F0MyIsCiAibm9uY2Ui0iAibi0w UzZfV3pBMk1qIiwKICJleHAiOiAxMzExMjq xOTcwLAogImlhdCI6IDEzMTEyODA5NzAKfQ .qqW8hZ1EuVLuxNuuIJKX V8a OMXzR0EHR 9R6jqdqr00F4daGU96Sr P6qJp6IcmD3HP9 90bi1PRs-cwh3L0p146waJ8IhehcwL7F09JdijmBqkvPeB2T9C JNgeGpegccMg4vfKjkM8FcGvnzZUN4 KSP0aAp1tOJ 1zZwgjxqGByKHiOtX7TpdQyHE5lcMiKPXfE IQILVq0pc E2DzL7emopWoaoZTF m0 N0Yz FC6q6EJb0EoRoS

```
Header
     "typ": "JWT",
     "alg": "RS256",
     "kid": "1e9gdk7"
Body/Payload
     "iss": "http://server.example.com",
     "sub": "248289761001",
     "aud": "s6BhdRkqt3",
     "nonce": "n-0S6 WzA2Mj",
     "exp": 1311281970,
     "iat": 1311280970
Signature: Signed (header + payload + secret)
```

Konrad Müller 24.10.19

SCOPES

Specifies ...

- I. Which resources can be accessed
- 2. Which information about the identity are available
- Optional: Consent
- Not all scopes must be granted
- "openid" is required

OpenID Proof of /authorization Client //dentity

FLOWS

- Resource Owner Password (don't do that)
- Client Credentials
- Implicit (deprecated)
- Authorization Code
- Hybrid Flow

Interactive

Authentication request Microsoft Sign in Authentication/Consent Email, phone, or Skype No account? Create one! Authorization Code Code -OpenID Client Provider Token request (Code —) Token Access Token **ID** Token Konrad Müller 24.10.19 Refresh Token

WHY AN INTERACTIVE FLOW?

- Single Sign-On
- One trusted entity
- Consistent login experience
- Password leaks
- Safe storage

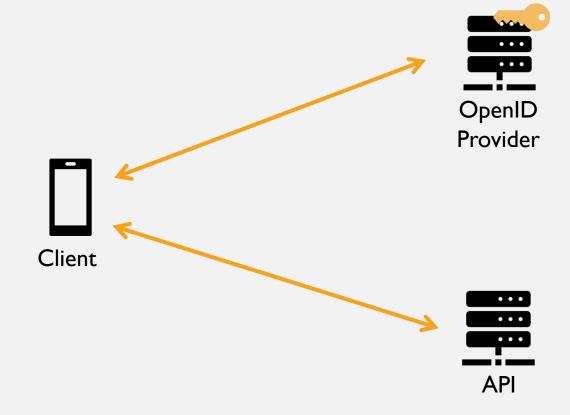
PROVIDERS, FRAMEWORKS, LIBRARIES

- Standard is widely supported
- Provider:
 - Microsoft, Google, Twitter, Facebook, GitHub, Apple
 - Auth0, Google Firebase, Azure AD B2C
 - Do it yourself: IdentityServer4 (.NET Core) and other certified provider frameworks
- Certified client and middleware libraries

IMPLEMENTING A MOBILE OIDC CLIENT WITH XAMARIN

OVERVIEW

- Identity Server 4 Demo
- Xamarin.Forms 4, iOS/Android with IdentityModel.Client library
- ASP.NET Core 3.0 API



SAMPLE TIME!

VISUAL STUDIO APP CENTER AUTH

VISUAL STUDIO APP CENTER AUTH

- OpenID provider build in Visual Studio App Center
- Easy to setup
- Support for Azure AD B2C Auth0 and Firebase in the future
- SDK for Xamarin Android/iOS
- Preview

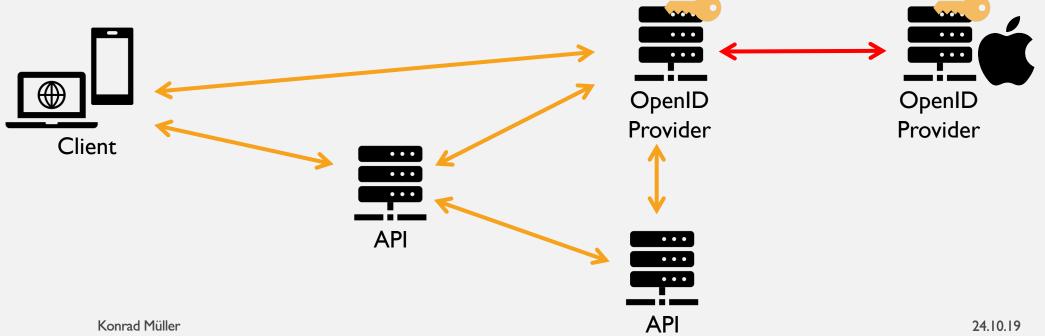
SIGN IN WITH APPLE



SIGN IN WITH APPLE

- Follows OpenID Connect standards not fully compliant!
- Certificate based ...
- 2 Factor required
- Required for "social logins"
- ASP.NET Core middleware available
- Federation possible

FEDERATION?



- Konrad Müller
- Xamarin, ASP.NET Core,
 Cloud stuff







SPIRIT/21