consol
Enterprising IT

WEBINAR BEGINNT IN KÜRZE:

Application Security With OAuth2 and Keycloak

Dr. Marco Bungart

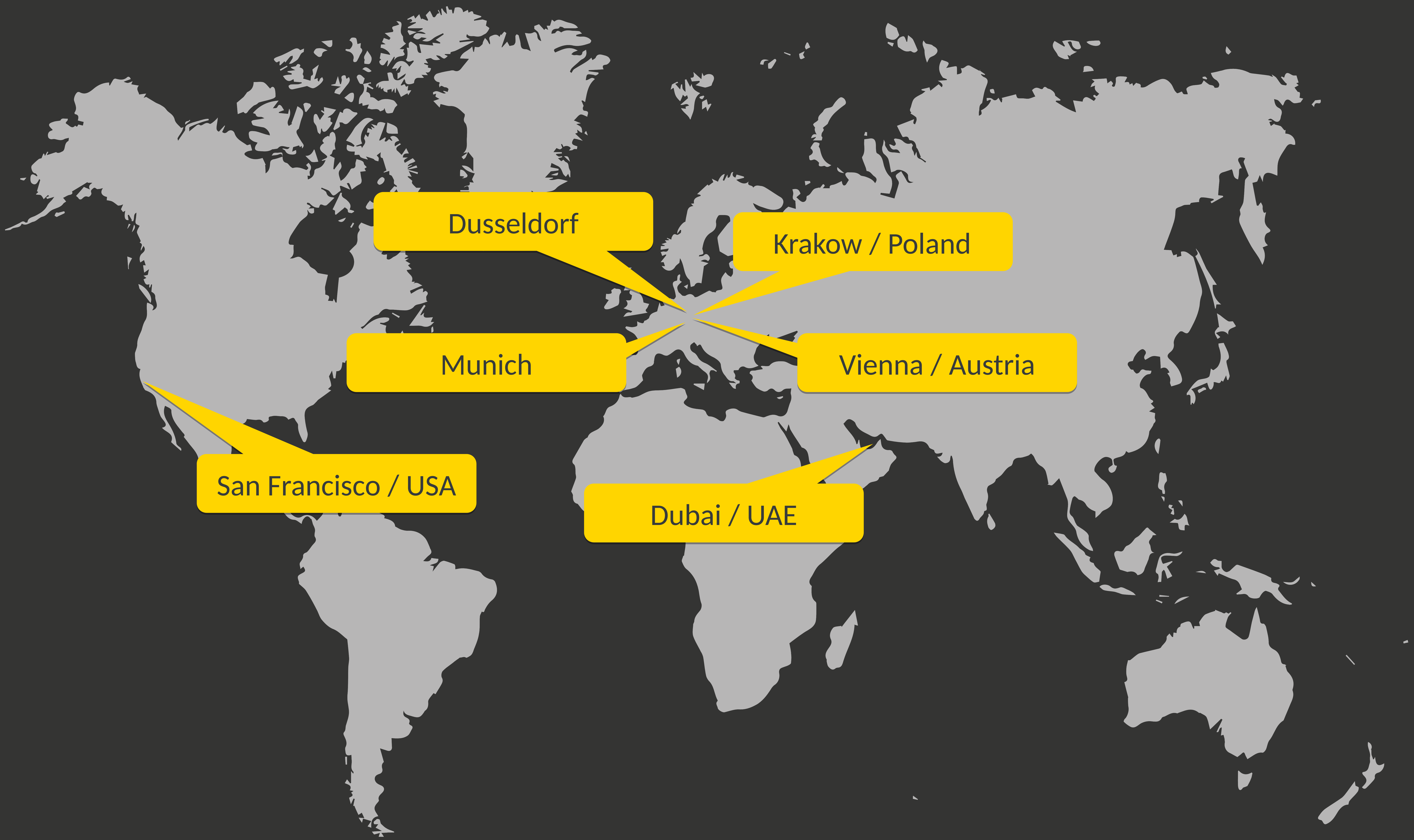
Software Engineer

consol
Enterprising IT

Application Security with OAuth2 and Keycloak

Dr. Marco Bungart
Software Engineer

Who is ConSol?


Dusseldorf
Krakow / Poland
Munich
Vienna / Austria
San Francisco / USA
Dubai / UAE

Portfolio

IT Consulting

No matter whether coaching, consulting in a special field or complete management of your IT project: With more than 30 years of competence, our consulting services are above all tailored to you and realizable.

IT Solutions

We develop IT solutions with a high utility value and pursue ideas beyond our current business. The results? For example our software ConSol CM, our open source offer for test automation or Red Hat Solutions.

Software Engineering

Whether with modern software architecture, cloud-native software, IT integration or customized CI/CD pipelines – we understand our technological know-how as an offer to master the challenges of digitalization.

IT Operations

Managed services, IT operations or support – at ConSol you choose exactly the services that an efficient IT requires. For any project size, whether on premises or in the cloud. Individual SLAs ensure calculable costs at all times.

Your Host

Dr. Marco Bungart

- 2008 – 2013: Studied Bioinformatics/Computer Science in Jena
- 2013 – 2018: Ph.D. student in Kassel
- Since 2018: Software Engineer at ConSol
- Twitter, github, bitbucket, stackoverflow, … : turing85
- Interests: Keycloak, GraalVM, Quarkus

OAuth2 & Keycloak

The “good old days”

- Monolithic applications: server-side Authentication/Authorization
  - Authentication/Authorization managed through session-state
  - Scaling was done vertically, not horizontally
- Moving to the cloud & multiple services: How to sync session states?
  - There are ways, but they introduce complexity
- What about social logins (Google, Facebook, …)?

→ Solution: Do not hold the session state on the server-side!

What is OAuth2?

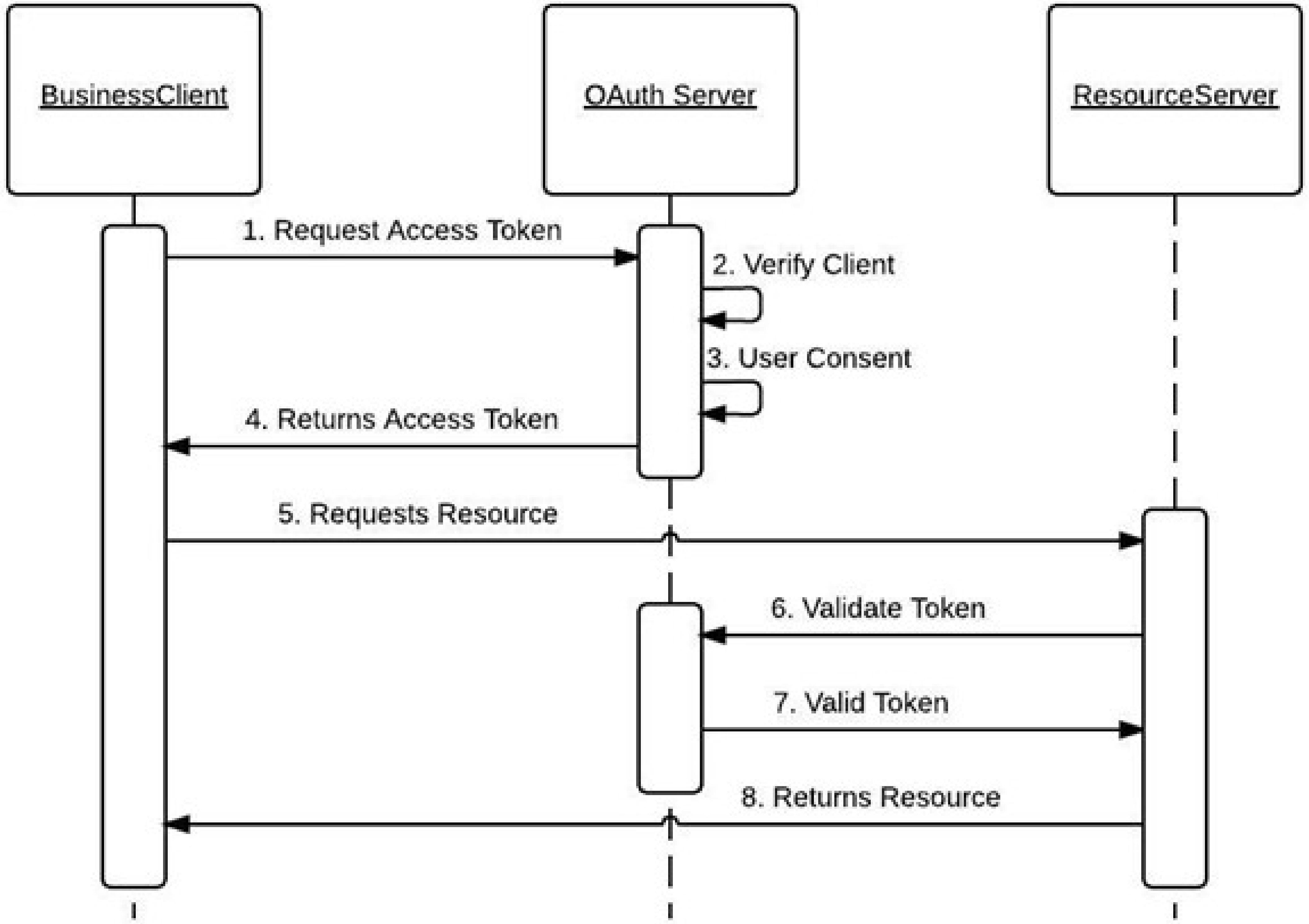
- Protocol for authentication/authorization
- Standard – RFC 6794
- Base for OpenID Connect (OIDC, “social login”) – RFC 8414
- Allows “Single Sign-On” for ecosystem

→ Good User Experience

Why OAuth2?

- Battle-hardened
- Different “flows” for different use cases
- Authentication/Authorization without server-state
- OAuth server grants token on successful authentication
- Authentication is stored in, e.g., the Frontend
- Backend system(s) evaluate token to grant Authorization (not part of this talk)
- Separation of concerns: OAuth server authenticates (“who?”), backend service authorizes (“what?”)
- Good client library support

How does OAuth2 work (schema)?



Source: https://docs.oracle.com/cd/E82085_01/160027/JOS%20Implementation%20Guide/Output/oauth.htm

Why OAuth2?

- Battle-hardened
- Different “flows” for different use cases (not part of this talk)
- Authentication/Authorization without server-state
- OAuth server grants token on successful authentication
- Authentication is stored in, e.g., the Frontend
- Backend system(s) evaluate token to grant Authorization (not part of this talk)
- Separation of concerns: OAuth server authenticates (“who?”), backend service authorizes (“what?”)
- Good client library support

Plenty of Fish, why Keycloak?

- Many OAuth products available, e.g. Auth0, Okta, Keycloak
- Keycloak is one of the few verified, open-source, free-to-use OAuth2 & OIDC servers (See here for an exhaustive list)
  - Developed by RedHat JBoss since 2014
  - On-Prem hosting & Configuration Flexibility
    - Cloud- & Cluster-compatible
  - User Federation (ldap, kerberos, …, plugable!)
  - In active development
  - Enterprise support available through Red Hat SSO

A little bit of Nomenclature

- **Resource Owner**: owns a resource, i.e. the User
- **Authorization Server**: Keycloak
  - Provides Authentication through tokens
- **Tokens** are signed & stored in, e.g., the frontend
- **Resource Server**: a (backend-)Server, providing resources
  - Makes Authorization decision based on token
  - May invoke the Authentication Server
- **Client**: makes request against Authorization server to, e.g., create or validate tokens, can be a Resource Server

A word of advice

If you use OAuth as Authentication/Authorization to your cluster, do **not** deploy the OAuth provider within the cluster.

If something in your cluster goes haywire, the OAuth provider service within the cluster may be affected, and you may have locked yourself out.

Alternatively, deploy to dedicated nodes (e.g. infrastructure nodes).

Live Demo

Live Demo

Source: https://makeameme.org/meme/lets-pray-to-9000c9f697

Resources

Resources

- Official OAuth2 Website
- Official OpenID Website
- Jwt.io - Website to decode JWT Tokens
- Keycloak Getting Started Guide
- Keycloak Docker Image
- https://github.com/ConSol/keycloak-webinar

Recap

Recap

- OAuth as Authentication/Authorization without server-state
  - Standard
  - Battle-hardened
- Keycloak as open-source, free-to-use & verified provider
  - Configurable
  - User Federation
  - Enterprise Support
- Keycloak configuration
  - Clients, Users, Roles
  - Mappers

Thank you!

consol
Enterprising IT

ConSol
Consulting & Solutions Software
GmbH

St.-Cajetan-Straße 43
D-81669 Munich
Germany
Tel.: +49-89-45841-100
info@consol.de
www.consol.com
Twitter: @consol_de