

# OAuth2 / Keycloak

mit ASP.NET Core (C#) und Angular

Donaueschingen, Oktober 2023

Einleitung

# Wer bin ich?

**Florian Schick**

// Unabhängiger Software Entwickler //

## Mit Fokus auf

Full-Stack mit .NET/Core, C#, Angular, Vue.js

Clean code //einfach zu lesen, einfach zu warten //



SCHICK  
SOFTWARE ENTWICKLUNG

## Kontakt

 florian.schick@schick-software.de

 +49 771 8979378

Einleitung

# Agenda

1. Erklärung der Begriffe
2. Keycloak installieren
3. Authentifizierungsabläufe
4. Authentifizierung
5. Identitätsanbieter
6. Autorisierung (rollenbasiert)
7. Autorisierung (mittels UMA)



# function

# OAuth2 / OpenID Connect

## OAuth2 (Open Authorization)

Ist ein offener Standard, um Webseiten oder Anwendungen Zugang zu den Informationen über Benutzer zu geben, ohne dass diese ihnen ihre Passwörter übermitteln müssen.

## OpenID Connect (OIDC)

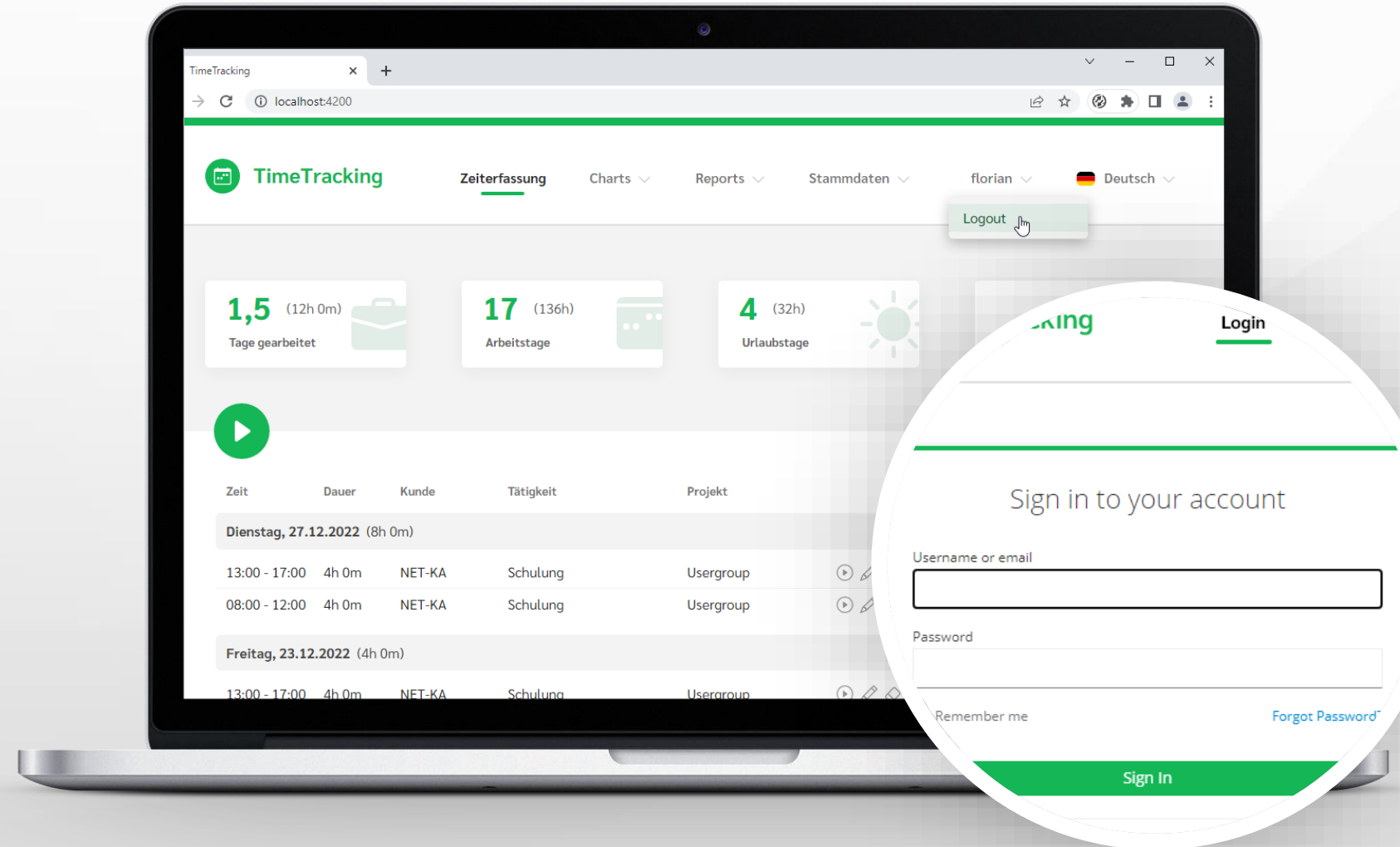
Ist eine Authentifizierungsschicht, die auf OAuth 2.0 aufbaut. Sie ermöglicht es Webseiten oder Anwendungen, die Identität eines Endbenutzers auf der Grundlage der von einem Autorisierungsserver durchgeführten Authentifizierung zu überprüfen.



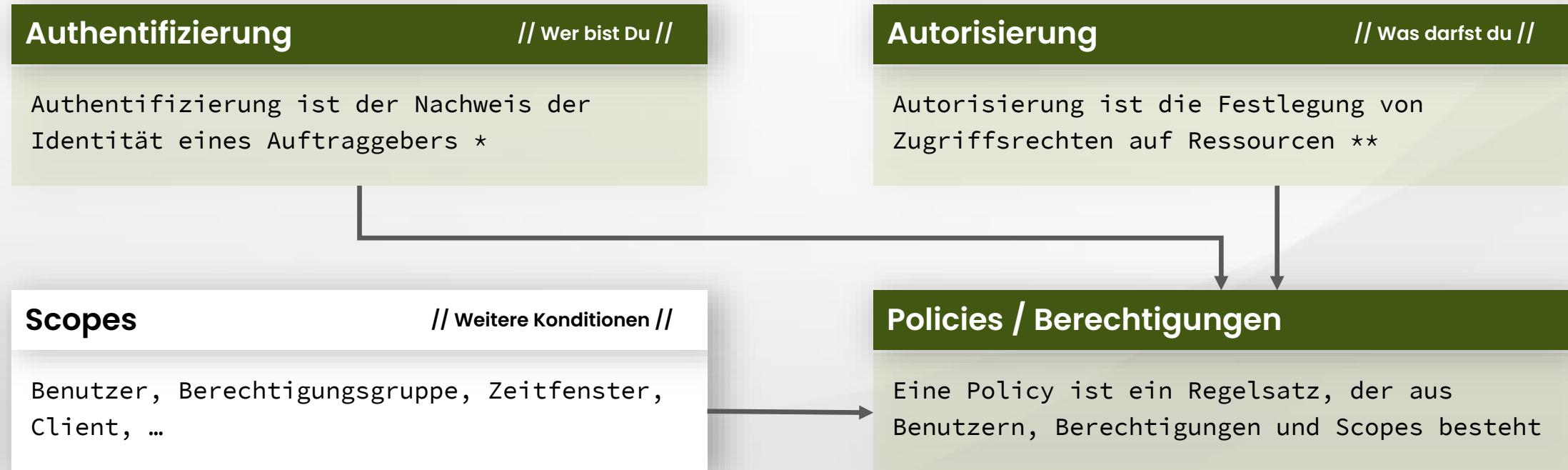
Definition von Begriffen

# Keycloak

Keycloak ist eine Open-Source Software, welche **Single Sign-On mit Identitäts- und Zugriffsmanagement** für moderne Anwendungen und Dienste ermöglicht



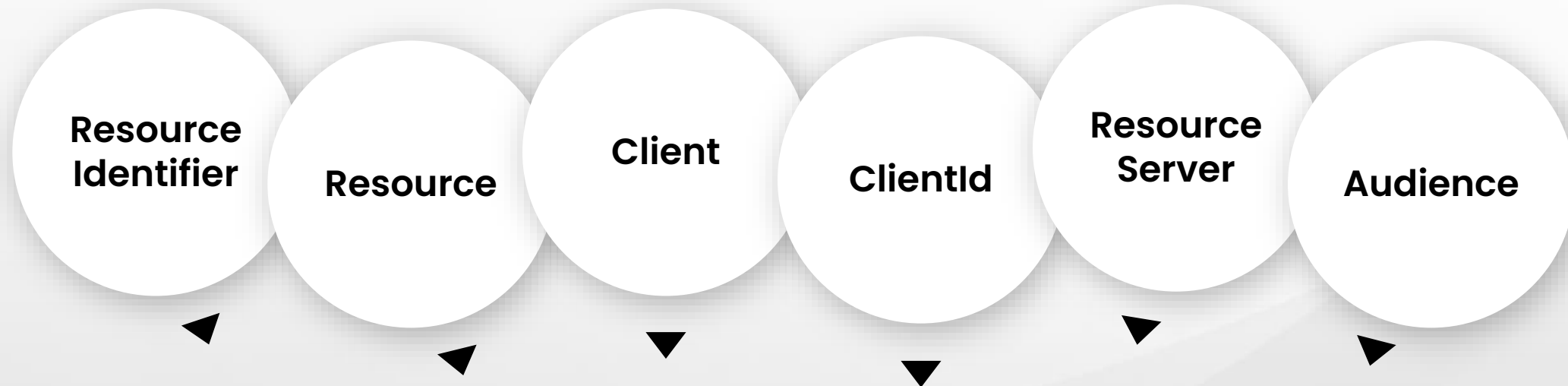
# Authentifizierung ≠ Autorisierung



\* Ein Auftraggeber ist die Identität eines Benutzers, Computersystems, Geräts, ...

\*\* Ressource ist eine Information wie z.B. Kunden, Dokumente, ...

# Namen wie Schall und Rauch



Identifizieren das, was Keycloak unter einem „Client“ versteht

# Live-Beispiel

## Code eines Projektes auf GitHub

```
{  
  ...  
  "KeycloakResourceUrl": "http://keycloak:8080/...",  
  "JwtBearer": {  
    "Authority": "http://keycloak:8080/auth/realms/webinar",  
    "Audience": "api"  
  },  
  "ClientCredentialsTokenRequest": {  
    "Address": "http://keycloak:8080/auth/realms/webinar/...",  
    "ClientId": "api",  
    "ClientSecret": "21199147-846d-44b7-bc3e-1f5145877514"  
  }  
}
```



# OAuth2 / Keycloak

Keycloak installieren

Authentifizierungsabläufe

Authentifizierung

Identitätsanbieter

Autorisierung (rollenbasiert)

Autorisierung (mittels UMA)

## Keycloak installieren

### Installation

- OpenJDK installieren
- Keycloak Zip-Deployment herunterladen
- `bin/kc.sh (start-dev|start)ausführen`

### Modi

```
start-dev: HTTP, HTTPS  
start : nur HTTPS
```

# OAuth2 / Keycloak

## Keycloak installieren und konfigurieren

Authentifizierungsabläufe

Authentifizierung

Identitätsanbieter

Autorisierung (rollenbasiert)

Autorisierung (mittels UMA)

## Keycloak konfigurieren

### Realm

- Ein Realm verwaltet Benutzer, Anmeldedaten, Rollen und Gruppen
- Ein Benutzer gehört zu genau einem Realm und meldet sich dort an
- Realms sind von einander isoliert

### Client

- Clients sind Webseiten oder Anwendungen, für welche die Authentifizierung eines Benutzers angefordert werden kann
- Clients können als Ressourcenserver agieren

### Benutzer

Benutzer sind alle Benutzer des aktuellen Realm

# OAuth2 / Keycloak

Keycloak installieren und konfigurieren

Authentifizierungsabläufe

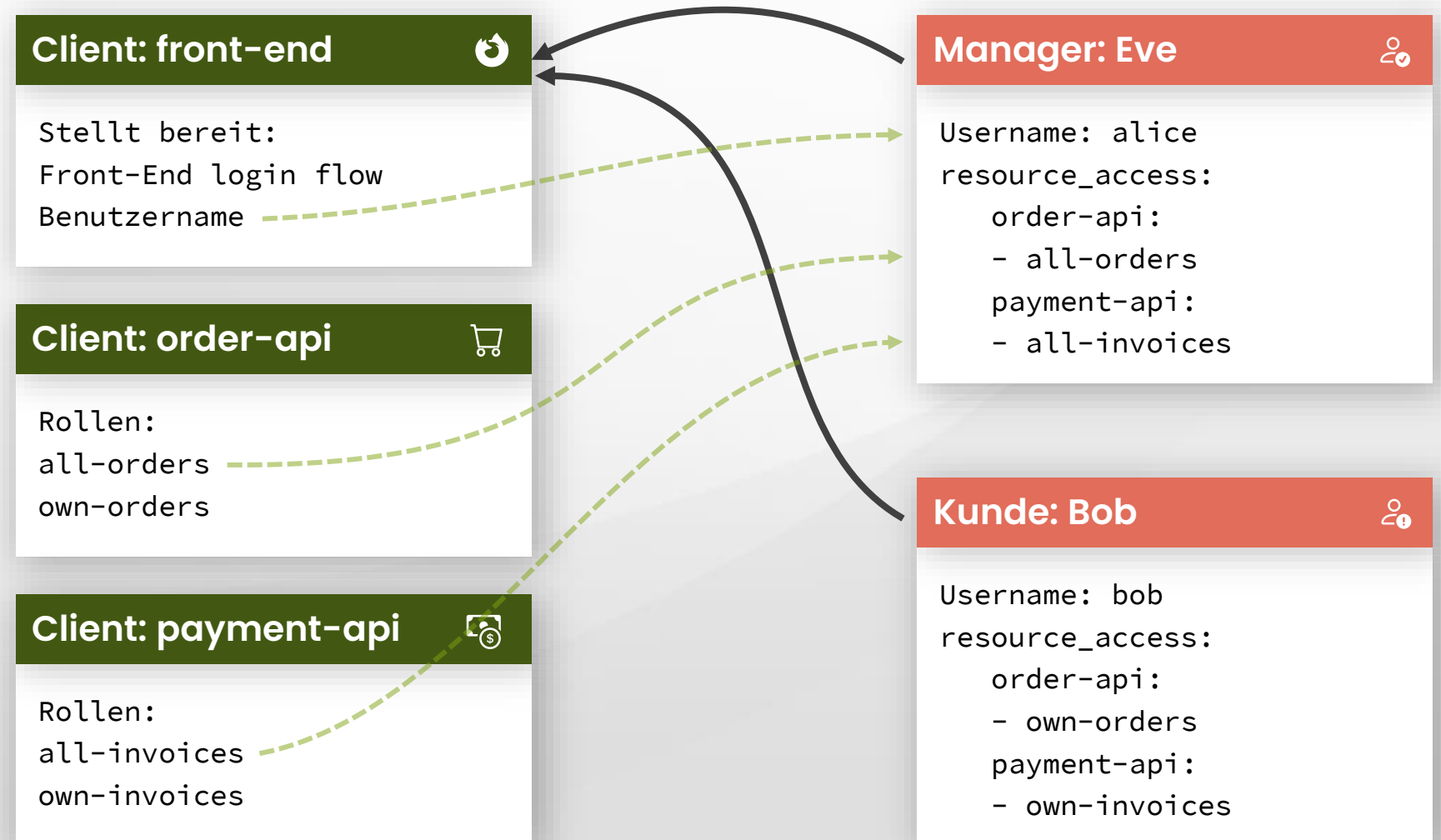
Authentifizierung

Identitätsanbieter

Autorisierung (rollenbasiert)

Autorisierung (mittels UMA)

## Beziehung zwischen Clients und Benutzern



# OAuth2 / Keycloak

Keycloak installieren und konfigurieren

## Authentifizierungsabläufe

Authentifizierung

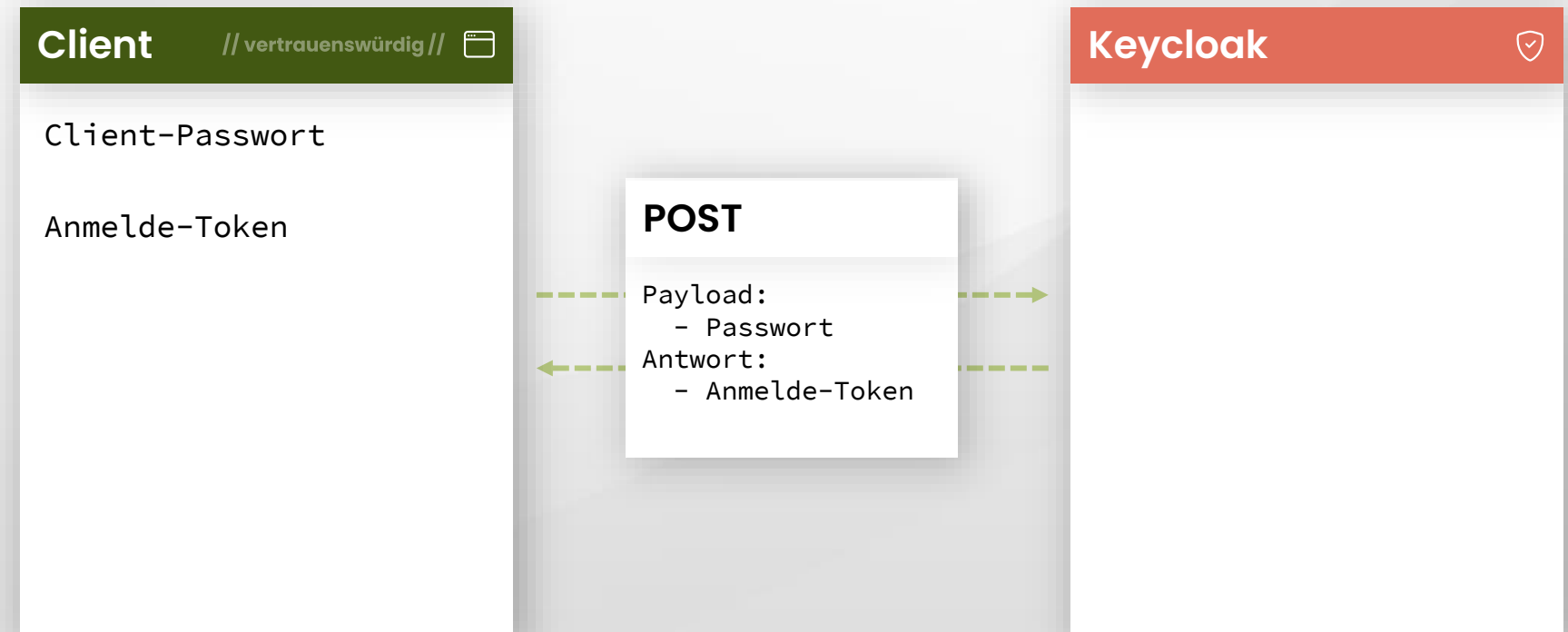
Identitätsanbieter

Autorisierung (rollenbasiert)

Autorisierung (mittels UMA)

## Authentication flows // vereinfacht //

### Client Credentials Grant / Service accounts roles



Keycloak installieren und konfigurieren

## Authentifizierungsabläufe

Authentifizierung

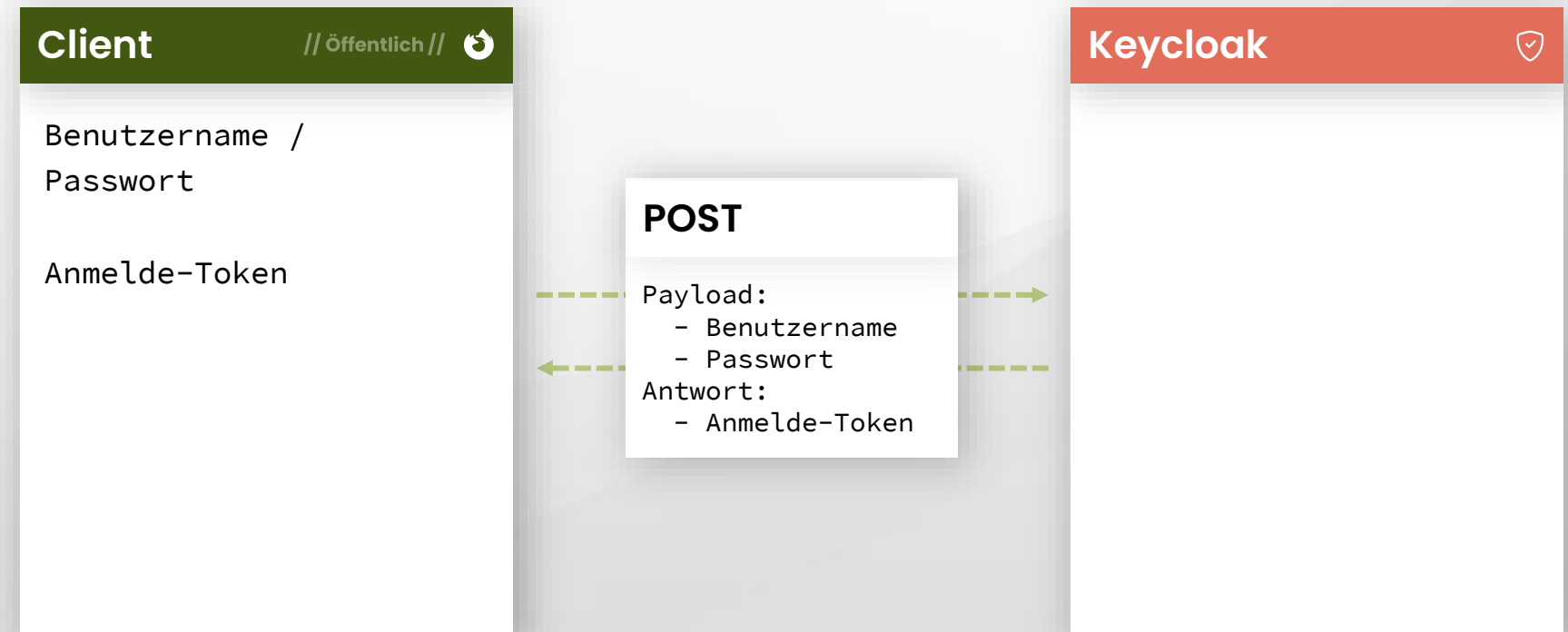
Identitätsanbieter

Autorisierung (rollenbasiert)

Autorisierung (mittels UMA)

# Authentication flows // vereinfacht //

## Resource Owner Password Credentials Grant / Direct Access Grant



# OAuth2 / Keycloak

Keycloak installieren und konfigurieren

## Authentifizierungsabläufe

Authentifizierung

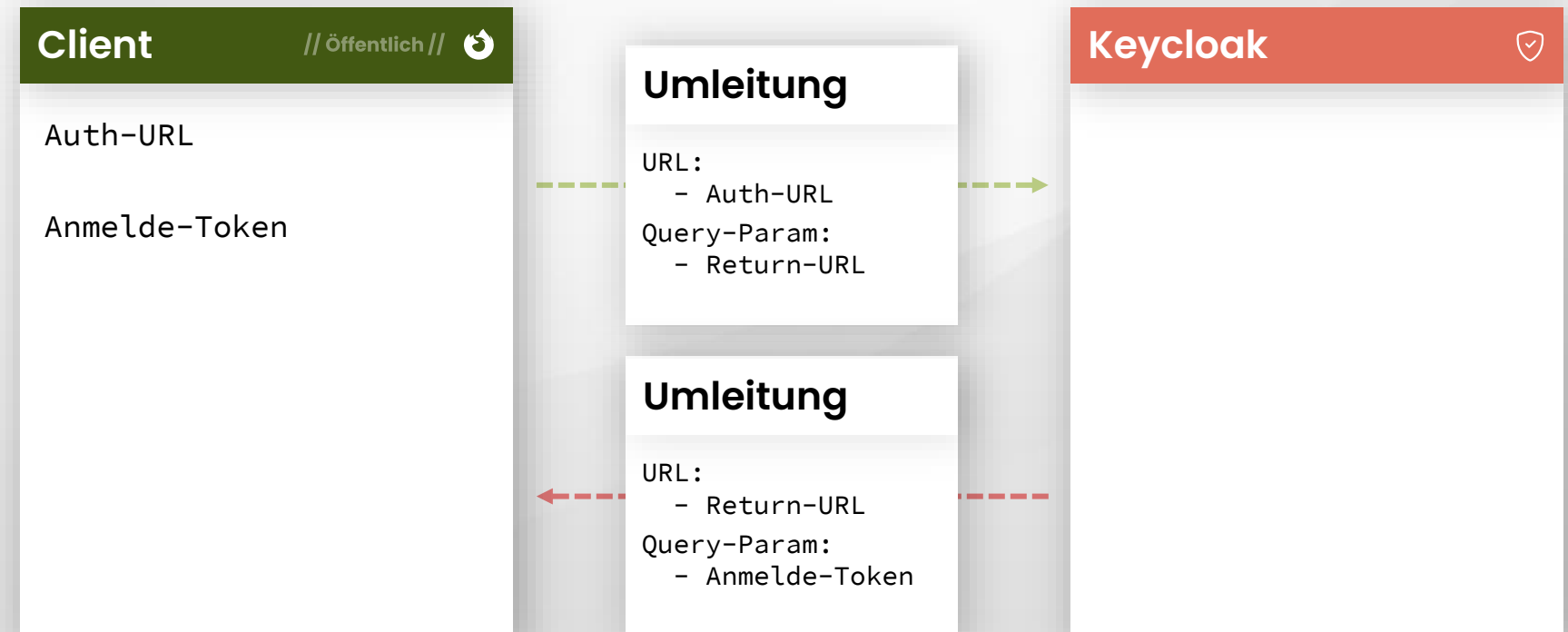
Identitätsanbieter

Autorisierung (rollenbasiert)

Autorisierung (mittels UMA)

## Authentication flows // vereinfacht //

### Implicit Flow // veraltet //



Keycloak installieren und konfigurieren

## Authentifizierungsabläufe

Authentifizierung

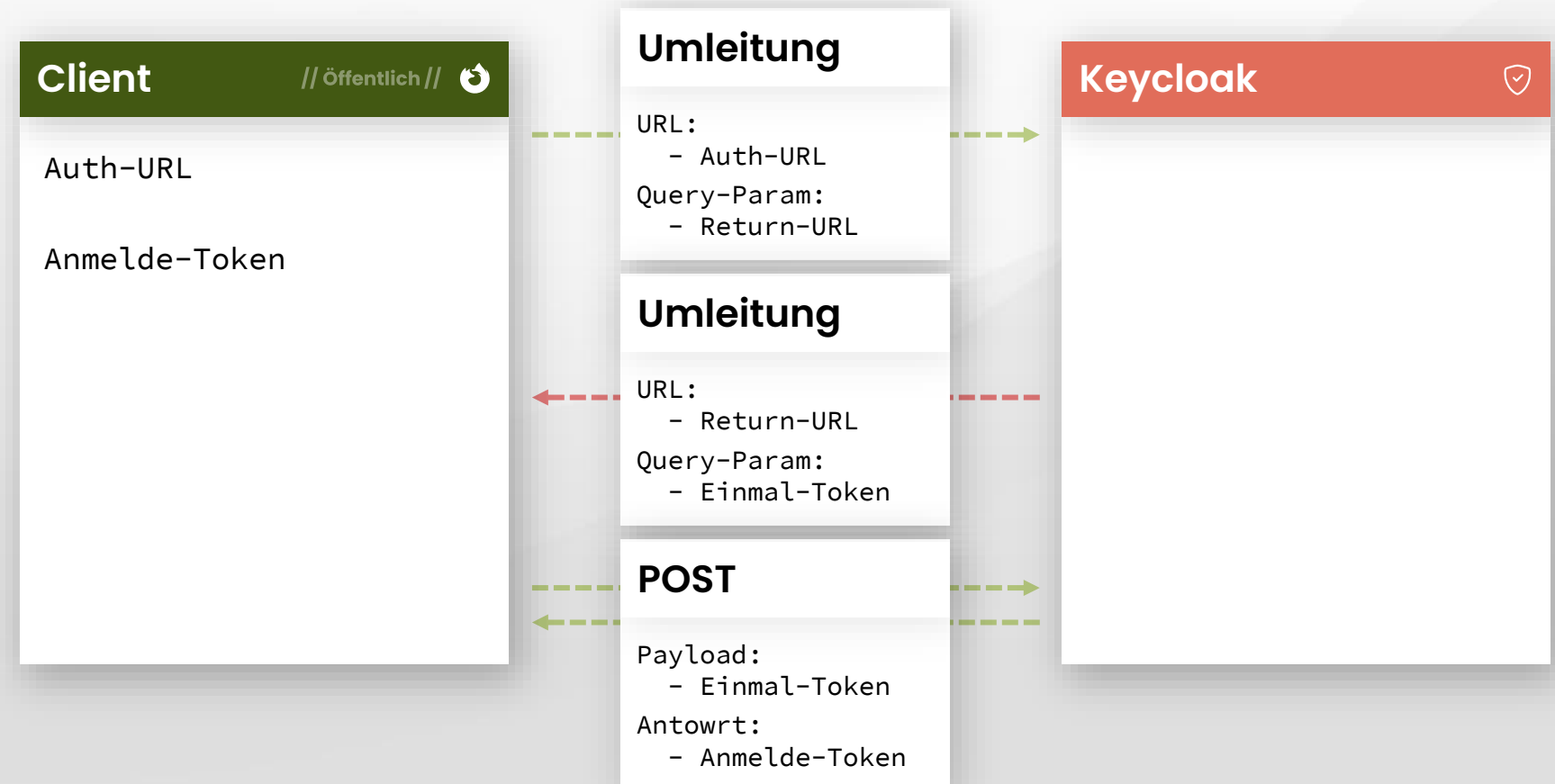
Identitätsanbieter

Autorisierung (rollenbasiert)

Autorisierung (mittels UMA)

# Authentication flows // vereinfacht //

## Authorization Code Flow / Standard flow





Keycloak installieren und konfigurieren

Authentifizierungsabläufe

**Authentifizierung**

Identitätsanbieter

Autorisierung (rollenbasiert)

Autorisierung (mittels UMA)

## Authentifizierung / ASP.NET

### Authentifizierung für Controller aktivieren

```
[Authorize], [AllowAnonymous]
```

### Authentifizierung für Minimal-API aktivieren

```
webApplication.MapGet(...).RequireAuthorization(...)  
webApplication.MapGet(...).AllowAnonymous()
```

Keycloak installieren und konfigurieren

Authentifizierungsabläufe

**Authentifizierung**

Identitätsanbieter

Autorisierung (rollenbasiert)

Autorisierung (mittels UMA)

## Authentifizierung / ASP.NET

### Dienste registrieren

```
serviceCollection.AddAuthentication()  
serviceCollection.AddJwtBearer(options => ...)
```

### Applikation konfigurieren

```
webApplication.UseAuthentication();  
webApplication.UseAuthorization();
```

Keycloak installieren und konfigurieren

Authentifizierungsabläufe

**Authentifizierung**

Identitätsanbieter

Autorisierung (rollenbasiert)

Autorisierung (mittels UMA)

## Authentifizierung / OpenAPI

### Dienste registrieren

```
swaggerGenOptions.AddGenericAuthorization();  
swaggerGenOptions.AddAuthorizationCodeFlow();  
  
swaggerUIOptions.OAuthClientId("api");  
swaggerUIOptions.OAuthUsePkce();  
swaggerUIOptions.EnablePersistAuthorization();
```

Keycloak installieren und konfigurieren

Authentifizierungsabläufe

**Authentifizierung**

Identitätsanbieter

Autorisierung (rollenbasiert)

Autorisierung (mittels UMA)

## Authentifizierung / Angular

### Keycloak beim Start initialisieren

```
provide: APP_INITIALIZER,  
useFactory: () => () => new Keycloak(<options>).init(...)
```

### HTTP-Interceptor registrieren

```
provide: HTTP_INTERCEPTORS,  
useClass: AuthenticationInterceptor,
```

# OAuth2 / Keycloak

Keycloak installieren und konfigurieren

Authentifizierungsabläufe

Authentifizierung

Identitätsanbieter




Autorisierung (rollenbasiert)

Autorisierung (mittels UMA)













## Identitätsanbieter


### Demo: Hinzufügen eines Identitätsanbieters

User-defined:

 Keycloak OpenID Connect	 OpenID Connect v1.0	 SAML v2.0
---	---	---

Social:

 BitBucket	 Facebook	 GitHub
 GitLab	 Google	 Instagram
 LinkedIn	 Microsoft	 OpenShift v3
 OpenShift v4	 PayPal	 StackOverflow

 TimeTracking [Login](#)

Sign in to your account


Username or email


Password

☐ Remember me [Forgot Password?](#)

[Sign In](#)

Or sign in with

 GitHub

 Google

New user? [Register](#)

Keycloak installieren und konfigurieren

Authentifizierungsabläufe

Authentifizierung

Identitätsanbieter

**Autorisierung (rollenbasiert)**

Autorisierung (mittels UMA)

## Autorisierung / ASP.NET

### Autorisierung für Controller aktivieren

```
[Authorize(Roles = "Manager")]
```

### Autorisierung für Minimal-API

```
webApplication.MapGet(...).RequireAuthorization(...)
```

### Dienste registrieren

```
serviceCollection.AddAuthorization()
```

# OAuth2 / Keycloak

Keycloak installieren und konfigurieren

Authentifizierungsabläufe

Authentifizierung

Identitätsanbieter

Autorisierung (rollenbasiert)

Autorisierung (mittels UMA)

## Keycloak Rollentransformation



JWT

```
"resource_access": {  
  "api": {  
    "roles": [ "Manager" ]  
  }  
}
```

C# // schematisch //

```
var clientRoles = principal.GetChildren("resource_access.api.roles")  
foreach (var role in clientRoles)  
  identity.AddClaim(new Claim(ClaimsIdentity.DefaultRoleClaimType, role));
```

Keycloak installieren und konfigurieren

Authentifizierungsabläufe

Authentifizierung

Identitätsanbieter

Autorisierung (rollenbasiert)

**Autorisierung (mittels UMA)**

## UMA // User Managed Access protocol //

### UMA

Als OAuth-basierter Protokollstandard für die Zugangsverwaltung verlagert UMA den Prozess der Erstellung und Handhabung von Autorisierungsregeln von der Anwendung auf einen Identitätsserver.



Keycloak installieren und konfigurieren

Authentifizierungsabläufe

Authentifizierung

Identitätsanbieter

Autorisierung (rollenbasiert)

**Autorisierung (mittels UMA)**

## UMA Keycloak

### UMA aktivieren

“Client authentication” und “Authorization” in dem Client, welcher als Ressourcen-Server agiert, aktivieren

### Anpassen der Standard-Policy

Die Standard-Policy basiert auf Javascript, welches nicht mehr in Java enthalten ist (Java > 15.x)

- Entweder die ‘Nashorn’ JS engine der lokalen JAVA-Installation hinzufügen
- Die Standard-Policy ersetzen
- Das Docker-Image von Keycloak verwenden

Keycloak installieren und konfigurieren

Authentifizierungsabläufe

Authentifizierung

Identitätsanbieter

Autorisierung (rollenbasiert)

Autorisierung (mittels UMA)

## UMA Keycloak

### Autorisierungs-Ressourcen ertsellen

**Resources:** Zu schützende Daten wie Kunden, Artikel, ...

**Scopes:** Auszuführende Aktionen wie Lesen, Erstellen, Aktualisieren, Löschen, ...

**Policies:** Bedingungen wie "innerhalb von Geschäftszeiten", "ist in der Gruppe", ...

**Permissions:** Welche Richtlinie darf welche Aktion für eine Ressource durchführen?

### Client für das Front-End erstellen

Der Client "api" wurde zu einem vertraulichen Client und benötigt damit ein Client-Secret. Damit die Autorisierung mit Angular funktioniert, brauchen wir einen neuen Client "Frontend".

# Ressourcen

## Demo Applikation

<https://github.com/fschick/Keycloak.ASPNet.Angular>

## Real-World Applikation mit Keycloak

<https://github.com/fschick/TimeTracking>

## Keycloak REST API client

<https://github.com/fschick/Keycloak.RestApiClient>

## ASP.NET One-Time Token Authentication

<https://github.com/fschick/Authentication.OneTimeToken>

# Ressourcen

## An Illustrated Guide to OAuth and OpenID Connect

<https://developer.okta.com/blog/2019/10/21/illustrated-guide-to-oauth-and-oidc>

## Background information to OAuth and OpenID Connect

[LDAP Wiki OAuth 2.0](#)  
[LDAP Wiki OpenID Connect](#)

## Use Keycloak as Identity Provider in ASP.NET Core 6

<https://nikiforovall.github.io/aspnetcore/dotnet/2022/08/24/dotnet-keycloak-auth.html>





# Vielen Dank

[www.schick-software.de](http://www.schick-software.de)