SCHLÜMPFE
Ekaterina Anisimova, Tobias Frank, Konstantin Hungs, Steffen Scheller
Software Engineering 2 Thomas Ziemer

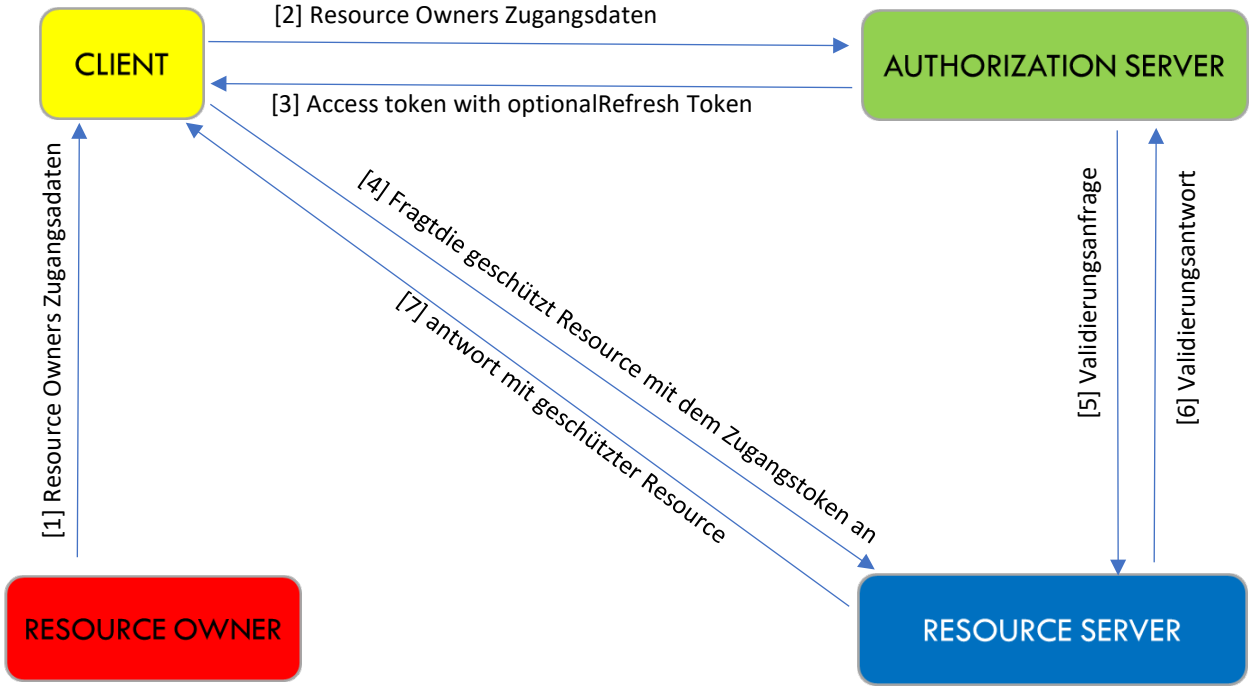
OAUTH2 Handout

Es handelt sich bei Oauth um ein Prinzip, welches als Standard von der IETF im Jahr 2010 in der Version 1.0a veröffentlicht wurde. Dieses setzte sich aber nie besonders durch, da die Entwickler dazu gezwungen wurden, sich mit Kryptografie auseinanderzusetzen. Im Jahr 2012 wurde dann die Version 2.0 veröffentlicht, die das Oauth verfahren so beschreibt, wie wir es heute kennen und nutzen. Die beiden Versionen sind miteinander nicht kompatibel und die Version 1.0a wird nicht mehr empfohlen, daher gehen wir im Folgenden nur auf die Version 2.0 ein.

Das grundlegende Prinzip ist, dass ein Nutzer sich (z.B. über einen Webbrowser) bei einem Dienst einloggt. Jedoch wird die Authentifizierung von einem anderen (externen) Server, bei dem er bereits einen Account besitzt, durchgeführt. Dies hat den Vorteil, dass der Nutzer seine Accountdaten direkt bei dem neuen Dienst nutzen kann, um auf dessen Resourcen zugreifen zu können.
Um zu vermeiden, dass der Username und das Passwort zu dem neuen Dienst übertragen werden müssen, wird ein Autorisierungstoken verwendet, welches bei jedem Request mitgeschickt wird.
Dieses Token wird nach der Authentifizierung am Autorisierungsserver generiert und muss je nach Anwendung nach einer bestimmten Zeit erneuert werden.

Bei Oauth gibt es vier verschiedene Rollen, die eingenommen werden können:

- Den Eigentümer der Resource (Meist der Nutzer)
- Den Server, auf dem Resource liegt
- Die Anwendung auf Client-Seite
- Den Autorisierungsserver



1. Der Besitzer der Resource loggt sich über die Anwendung (Client) mit seinem Benutzernamen und Passwort ein
2. Die Anwendung überträgt die Daten an den Autorisierungsserver
3. Der Autorisierungsserver stellt ein Access-Token aus (optional auch ein Refresh Token)
4. Mit dem Access-Token fragt die Anwendung dann die geschützten Nutzerresourcen an
5. Der Resourcenserver validiert den Token über eine Schnittstelle am Autorisierungsserver
6. Der Autorisierungsserver antwortet, dass das Token valide und nicht abgelaufen ist
7. Der Resourcenserver gibt die Resource frei