Software-Architektur: Praktikumsaufgabe

Authentication Prozess bei ShareIt

Tobias Huber, Christian Keller, Michael Fischer, Andreas Neumeier

---

Hochschule München, SS17
Software-Architektur
Problem Based Learning

Dozent: Prof. Dr. Axel Böttcher
Abgabedatum: 15. Mai 2017

1. Ausgangsbasis:

In diesem Abschnitt soll eine Projektidee, zur Erweiterung des ShareIt Projektes aufgezeigt werden. Dafür wird zuerst die Ausgangslage beschrieben und anschließend der vorgeschlagene Lösungsweg.

1.1. Problemstellung:

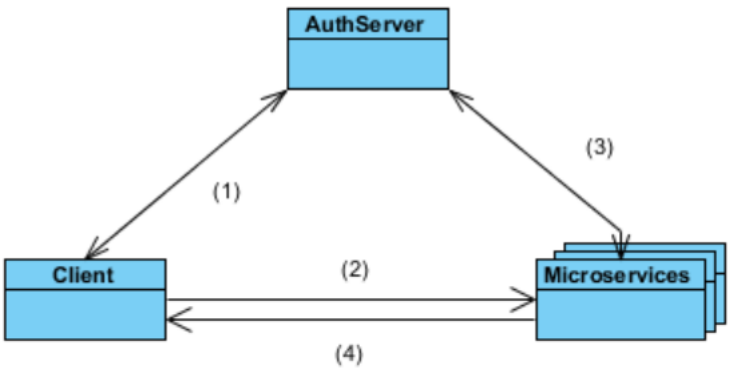
In dem vorangegangenen Projekt „ShareIt Teil I“ wurde ein System zur Verwaltung von Medien entwickelt, über welches Studenten Ihre Medien (z.B. Bücher, Discs) austauschen können. Diese Plattform ist derzeit von allen Personen mit allen Rechten Nutzbar. Aus diesem Grund können die Daten der Seite von jeder beliebigen Person ausgelesen und manipuliert werden.

1.2. Systemidee Benutzerverwaltung & Skizzierung der Anforderungen:

Um dieses Problem zu lösen soll nun das System um eine Authentifizierung inklusive Rechtesystem erweitert werden.
Dafür sollen sich Studierende, die ShareIt nutzen wollen, vorab als Benutzer am System registrieren können. Geplant sind zwei Arten von registrierten Benutzern, nämlich Administratoren und normale Benutzer. Ein registrierter Benutzer kann sich am System an- und abmelden. Ist ein Benutzer angemeldet, so kann er selbst die von ihm gespeicherten Stammdaten bearbeiten. Außerdem kann der Benutzer sich die von ihm zur Verfügung gestellten Leihgaben sowie die von ihm aktuell entliehenen Medien anzeigen lassen.
Zusätzlich zu den Funktionalitäten für normale Benutzer können Administratoren einen neu registrierten Benutzer freigeben. Benutzer, die mehrfach gegen die selbstdefinierten Regeln der ShareIt Community verstoßen haben, können von einem Administrator von der weiteren Nutzung des Systems ausgeschlossen werden.

1.3. Lösungsidee:

Erstellen Sie einen Service, der die Authentifizierung der Benutzer übernimmt. Dieser Service soll nach dem Prinzip des OAuth funktionieren. Ein Benutzer wird durch seinen Namen und ein Passwort identifiziert und erhält ein Token. Bei einer Anfrage an die Microservices ist das Token mit enthalten. Der Microservice kann dieses dann am AuthServer validieren. Bei der Validierung des Tokens findet zusätzlich eine Rechteprüfung statt. Ein Microservice schickt zusammen mit dem Token eine RechteID an den AuthServer. Dieser überprüft dann, ob die geforderte ID dem Benutzer zugewiesen ist.



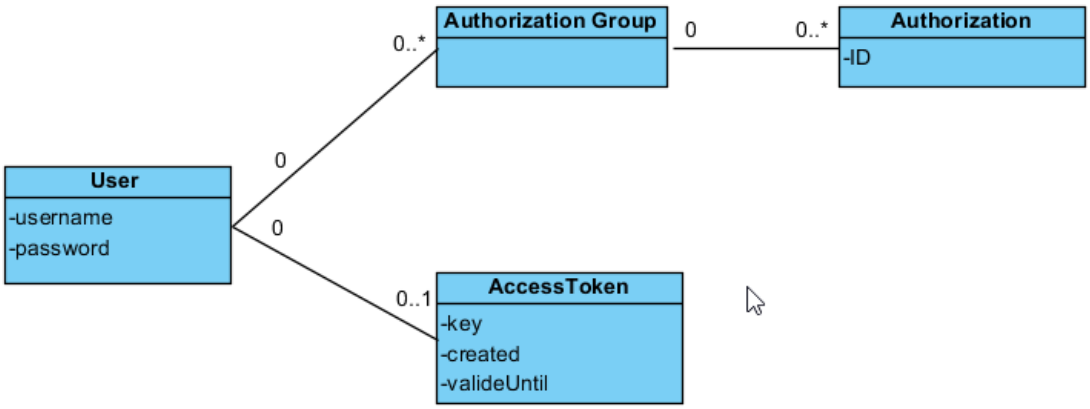
Vereinfachung:

- Der Benutzer muss vor der Verwendung der Microservices authentifiziert werden, z.B. beim Login. Es erfolgt keine automatische Weiterleitung von den Microservices zum AuthServer.
- Passwörter müssen nicht verschlüsselt werden.

Aufbau:

Datenhaltung:

Der Service kennt Benutzer, Autorisierungsgruppen, Autorisierungen und Tokens. Die Benutzer bestehen aus Benutzernamen und Passwort. Die Autorisierungsgruppen haben eine Liste von Rechten, die deren Benutzer erhalten sollen. Autorisierungen besitzen nur eine ID (z.B. media.book.create). Ein Benutzer kann mehreren Autorisierungsgruppen zugewiesen werden und Autorisierungsgruppen können mehrere Autorisierungen zugeteilt sein.
Die Tokens die der Service ausgeben soll enthalten einen Schlüssel, einen Zeitstempel und ein Ablaufzeitpunkt.



REST Schnittstelle:

Der Service benötigt zwei REST Schnittstellen. Eine externe Schnittstelle ermöglicht die Authentifizierung eines Benutzers, eine interne Schnittstelle ermöglicht es den Microservices, ein Token zu validieren.

Der AuthService überprüft den Benutzernamen und das Passwort. Falls beide korrekt sind, wird ein neues Token angelegt und zurückgesendet. Bei der Validierung übergibt ein Microservice sowohl das Token als auch die geforderte Autorisierung (das benötige Recht). Der OAuth-Service überprüft, ob das Token vorhanden und gültig ist. Danach wird die Autorisierung überprüft. So wird sichergestellt, dass das Token gültig ist und der Benutzer die nötigen Rechte besitzt. Der Service antwortet mit einem ValidationResult. Es transportiert das Ergebnis der Validierung und gegebenenfalls Statusinformationen warum die Validierung fehlgeschlagen ist. Orientieren Sie sich dazu am MediaResult aus der Aufgabe ShareIt.

2. Umzusetzende Anforderungen:

2.1. Ziele:

In dieser Praktikumsaufgabe soll ein funktionierendes Authentifizierungssystem für den ShareIt Service erstellt werden. Dabei soll auf den in der Lösungsidee vorgeschlagenen Lösungsweg zurückgegriffen werden.

2.2. Zwingend umzusetzende Anforderungen:

- Rest basierter AuthService
- Mock Objekte welche an den Authentifizierungsservice angehängt werden, um Benutzer zu simulieren, da derzeit keine Datenbankanbindung vorhanden ist.
- Rechtegruppen, welchen die Benutzer zugewiesen werden können. (zuweisen der Gruppen an die Mock Objekte)
- Media Webservice bezüglich Autorisierung überarbeiten.

2.3. Optional umzusetzende Anforderungen:

- Erstellen einer Weboberfläche zum Registrieren von Benutzern