

SWAPINDO

# **Requirements Specification Service-to-service authentication**

Version 1.0

von

Benjamin Ellmer (S1910237013)

March 26, 2022

# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Einleitung</b>  | <b>2</b> |
| 1.1      | Zweck des Dokuments . . . . .                                    | 2        |
| 1.2      | Gültigkeit des Dokuments . . . . .                               | 2        |
| 1.3      | Begriffsbestimmungen und Abkürzungen . . . . .                   | 2        |
| 1.4      | Zusammenhang mit anderen Dokumenten . . . . .                    | 2        |
| <b>2</b> | <b>Allgemeine Beschreibung des Produkts</b>                      | <b>3</b> |
| 2.1      | Zusammenhang mit bereits laufenden Projekten . . . . .           | 3        |
| 2.2      | Zusammenhang mit Vorgänger- und Nachfolgeprojekten .             | 3        |
| 2.3      | Zweck des Produkts . . . . .                                     | 3        |
| 2.4      | Abgrenzung und Einbettung des Produkts . . . . .                 | 4        |
| 2.5      | Überblick über die geforderte Funktionalität . . . . .           | 4        |
| 2.6      | Allgemeine Einschränkungen . . . . .                             | 4        |
| 2.7      | Vorgaben zu Hardware und Software . . . . .                      | 4        |
| 2.8      | Benutzer des Produkts . . . . .                                  | 5        |
| <b>3</b> | <b>Detaillierte Beschreibung der geforderten Produktmerkmale</b> | <b>5</b> |
| 3.1      | Lieferumfang . . . . .   | 5        |
| 3.2      | Abläufe von Interaktionen mit der Umgebung . . . . .             | 5        |
| 3.3      | Ziele des Benutzers . . . . .                                    | 5        |
| 3.4      | Geforderte Funktionen des Produkts . . . . .                     | 6        |
| 3.4.1    | Authentifizierung-Serverseite . . . . .                          | 6        |
| 3.4.2    | Authentifizierung-Clientseite . . . . .                          | 6        |
| 3.5      | Externe Schnittstellen des Produkts . . . . .                    | 6        |
| 3.5.1    | Benutzerschnittstellen . . . . .                                 | 6        |
| <b>4</b> | <b>Vorgaben an die Projektabwicklung</b>                         | <b>6</b> |
| 4.1      | Anforderungen an die Realisierung . . . . .                      | 6        |

# **1 Einleitung**

## **1.1 Zweck des Dokuments**

Der Zweck des Dokuments ist es eine Übersicht über das Projekt "Swapindo Service-to-service authentication", welches ein Teilprojekt des Projekts "Swapindo" ist zu geben. Es werden die wichtigsten Fakten und Anforderungen in einem Dokument zusammengefasst. Darüberhinaus soll ein Überblick gegeben werden, wie diese Anforderungen umgesetzt werden.

## **1.2 Gültigkeit des Dokuments**

Dieses Dokument ist nur für das Teilprojekt "Swapindo Service-to-service authentication" gültig. In diesem Teilprojekt sollen die beliebtesten Service-zu-Service Authentifizierungsmechanismen verglichen und implementiert werden. Es ist die erste Version des Pflichtenhefts und für dieses Teilprojekt und Benjamin Ellmer ist für Änderungen in diesem Dokument verantwortlich.

## **1.3 Begriffsbestimmungen und Abkürzungen**

**TLS:** Transport Layer Security

**mTLS:** mutual TLS

**JSON:** JavaScript Object Notation

**JWT:** JSON Web Token

## **1.4 Zusammenhang mit anderen Dokumenten**

Dieses Pflichtenheft erweitert das Pflichtenheft vom Wintersemester 2020 und Sommersemester 2021.

## **2 Allgemeine Beschreibung des Produkts**

Es handelt sich um einen Teil der Realisierung des Projekts "Swapindo" der Swapindo GmbH. Im Projekt "Swapindo" gilt es grundsätzlich eine Flohmarkt-Plattform zu erschaffen. In diesem Projekt (Swapindo Service-to-Service Authentication) geht es darum die Kommunikation der Microservices des Backends vom Projekt "Swapindo" zu schützen. Es soll eine Klassenbibliothek geschrieben werden, welche die Integration der Authentifizierungsmechanismen so einfach wie möglich gestaltet.

### **2.1 Zusammenhang mit bereits laufenden Projekten**

Im Zuge des PRO Moduls werden zwei weitere Teilprojekte vom Projekt Swapindo von zwei Kollegen entwickelt. In einem der Teilprojekte geht es um die Entwicklung eines Empfehlungssystems und im zweitem Teilprojekt geht es um die Implementierung von effizienten Caching Mechanismen.

### **2.2 Zusammenhang mit Vorgänger- und Nachfolgeprojekten**

Das Projekt "Swapindo Service-to-service authentication" ist der dritte Teil des Gesamtprojekts "Swapindo" und ist der Nachfolger vom Projekt "Swapindo Frontend" aus dem Sommersemester 2021. In weiterer Folge wird das Gesamtprojekt "Swapindo" im Namen der Swapindo GmbH weiterentwickelt.

### **2.3 Zweck des Produkts**

Der Zweck des Produkts ist die Absicherung des Backends vor ungewollten Zugriffen. Da es mehrere Mechanismen hierfür gibt, werden die bekanntesten Mechanismen mTLS und self-signed JWT beide implementiert und Verglichen. Am Ende dieses Projekts soll es nicht mehr möglich sein ohne Authentifizierung auf die Services des Backends von Swapindo zuzugreifen.

## **2.4 Abgrenzung und Einbettung des Produkts**

Da andere Entwicklungsprozesse nicht unterbrochen werden sollen, sollen die Authentifizierungsmechanismen noch nicht in die aktiven Services eingebaut werden. Es soll lediglich die Grundlage geschaffen werden, um die Services mit den implementierten Authentifizierungsmechanismen auszustatten. Das Endprodukt des Projekts soll eine Klassenbibliothek sein, die mit wenig Zeilen Code die Einbettung der Authentifizierungsmechanismen in ASP.Net Services erlaubt.

## **2.5 Überblick über die geforderte Funktionalität**

- Das Produkt soll ermöglichen, dass von Kommunikationspartner eine Authentifizierung gefordert wird (Serverseite)
- Das Produkt soll die Authentifizierung gegenüber einem Kommunikationspartner ermöglichen (Clientseite)

## **2.6 Allgemeine Einschränkungen**

Das Teilprojekt beschäftigt sich hauptsächlich mit der Authentifizierung zwischen den Microservices. Um das Backend vollständig zu schützen wird auch Authorisierung benötigt. Außerdem wird auch End-User Authentifizierung benötigt, welche hauptsächlich im Vorgängerprojekt implementiert wurde und in diesem Projekt in das Backend eingebettet wurde.

## **2.7 Vorgaben zu Hardware und Software**

Zur Versionierung soll GIT verwendet werden. Das Produkt soll mit so wenig Aufwand wie möglich in die bereits bestehenden Microservices des Backends vom Projekt "Swapindo" eingebettet werden können. Diese Microservices sind in der Programmiersprache C# geschrieben und wurden mit dem ASP.Net Core 5.0 Framework entwickelt.

## **2.8 Benutzer des Produkts**

Die Benutzer des Produkts sind die Backend-Entwickler von Swapindo. Diese sind nach Fertigstellung des Projekts dafür zuständig die Mechanismen in die Services des Backends einzubetten.

# **3 Detaillierte Beschreibung der geforderten Produktmerkmale**

Gefordert:

- Klassenbibliothek zu einfachen Integration von Service-to-service authentication mit mTLS
- Klassenbibliothek zu einfachen Integration von Service-to-service authentication mit self-signed JWTs

## **3.1 Lieferumfang**

Der Lieferumfang beträgt den Source Code der Klassenbibliothek, sowie zwei Beispielservices, an denen die Verwendung der Klassenbibliothek gezeigt wird.

## **3.2 Abläufe von Interaktionen mit der Umgebung**

Die Backend Entwickler müssen die Bibliothek als referenziertes Projekt einbinden. Wenn diese eingebunden ist, können sie mit wenigem Befehlen die geforderten Authentifizierungsmechanismen integrieren.

## **3.3 Ziele des Benutzers**

Die Bibliothek soll mit wenigen Zeilen Code eingebunden werden können. Außerdem sollen die eingebundenen Mechanismen eine fehlerfreie und effiziente Authentifizierung implementieren. Das Ziel des Benutzers ist die Absicherung seines Backends von unauthentifzierten Zugriffen.

## **3.4 Geforderte Funktionen des Produkts**

### **3.4.1 Authentifizierung-Serverseite**

Auf der Serverseite soll eine Authentifizierung des Clients verlangt werden. Wenn die Authentifizierung fehlschlägt, kann der Client auf keine Ressourcen zugreifen. Es soll möglich sein, zwischen self-signed JWTs und mTLS auszuwählen.

### **3.4.2 Authentifizierung-Clientseite**

Es soll eine Möglichkeit für die Clients geboten werden, sich gegenüber einem Server zu authentifizieren. Hierbei soll die Möglichkeit bestehen, zwischen self-sigend JWTs und mTLS auszuwählen.

## **3.5 Externe Schnittstellen des Produkts**

### **3.5.1 Benutzerschnittstellen**

Die Benutzer können die Klassenbibliothek in ihre Projekte einbinden. Da es sich um eine reine Ansammlung von Code handelt gibt es keine grafischen Schnittstellen.

# **4 Vorgaben an die Projektabwicklung**

## **4.1 Anforderungen an die Realisierung**

Software:

- Die Bilbliothek soll für das ASP.Net Framework geschrieben werden
- GIT soll für die Versionsverwaltung verwendet werden