Requirements

**Zu implementierende Dienste**:

* OwnCloud
* Dropbox
* Github
* Google Docs
* Bitbucket
* Slack

**OwnCloud:**

* Datei in ein angegebenes Verzeichnis hochladen **OA1**
* Verzeichnis und-Dateistruktur laden **OA2**
* Datei aus angegebenen Verzeichnis herunterladen **OA3**

**Dropbox:**

* Datei in ein angegebenes Verzeichnis hochladen **DA1**
* Verzeichnis und-Dateistruktur laden **DA2**
* Datei aus angegebenen Verzeichnis herunterladen **DA3**

**Github:**

* Repositories als Liste laden **GiA1**
* Verzeichnis und-Dateistruktur innerhalb eines Repositories laden **GiA2**

**Google Docs/Drive:**

* Datei in ein angegebenes Verzeichnis hochladen **GoA1**
* Dateistruktur laden **GoA2**
* Datei aus angegebenen Verzeichnis herunterladen **GoA3**

**Bitbucket:**

* Repositories als Liste laden **BA1**
* Verzeichnis und-Dateistruktur innerhalb eines Repositories laden **BA2**

**Slack:**

* Channels als Liste laden **SA1**
* Nachricht an Channel senden **SA3**

**Servicecomposition REST Api:**

Funktionale Anforderungen:

* Zugang zu funktionalen Anforderungen der einzelnen Services **A1**
* Caching von Anfragen **A2**

Nicht-Funktionale Anforderungen:

* Cross-Plattform accessibility **A3**
* Fehlersemantik **A4**

Entitäten

Dienst

Eigenschaften

Name

Nutzer

Eigenschaften

Name

Passwort

Authentifizierungs-Daten (für jeden Dienst)

Team

Eigenschaften

Name einzigartig(ID)

Passwort

Dienste die im Team genutzt werden

**UserService**:

* Create User
* Delete User
* Read User
* Update User
* IsLoginCorrect

**TeamService**:

* Create Team
* Delete Team
* Read Team
* Update Team

Bounded Context für Modell des Users. Der Teamservice muss nur den Namen der User kennen, aber nicht sein Passwort. Also ha jeder Service sein eigenes User Modell.