Christian-Albrechts-Universität zu Kiel
Institut für Informatik
Arbeitsgruppe Software Engineering

Bachelorarbeit

Single Sign-On Authentifizierung von webbasierten Anwendungen am Beispiel von sozialen Netzwerken

Phil Johannsen(pjo@informatik.uni-kiel.de)

30. März 2012

Gutachter: Prof. Dr. Wilhelm Hasselbring
Betreuer: Jens Ehlers

Die vorliegende Bachelorarbeit befasst sich mit dem Thema der Single Sign-On Authentifizierung von webbasierten Anwendungen am Beispiel von sozialen Netzwerken. Der Prozess des Single Sign-On ist eine Variante der Authentifizierung von Nutzern bei Systemen oder Anwendungen, bei der eine bereits getätigte Authentifizierung und die Angabe der Daten des Nutzers für weitere Authentifizierungen genutzt werden können. Dieser Prozess wurde mithilfe des Open Authorization Protokolls auf webbasierte Anwendungen übertragen und bei den sozialen Netzwerken Facebook, Google Plus und LinkedIn implementiert und getestet. Dazu wurde zuerst eine Einführung und das Themengebiet gegeben und die Motivation für dieses Thema dargestellt sowie ein Überblick über die einzelnen Schritte gegeben. Nach der Einführung werden die benötigten Grundlagen zum Verständnis und zur Abgrenzung einzelner Aspekte der Arbeit gegeben. Anschließend wurde der theoretische Ablauf des Open Authorization Protokolls am Beispiel des serverseitigen Flows im Detail vorgestellt und es wurden Unterschiede zwischen den einzelnen Flows des Open Authorization Protokolls in der Version 2.0 dargestellt sowie Anwendungsmöglichkeiten der Flows aufgezeigt. Der nächste Schritt befasste sich mit dem Erstellen eines Evaluationsbeispiels bei dem der serverseitige Flow bei den genannten sozialen Netzwerken Schritt für Schritt durchgeführt wurde und Unterschiede der einzelnen Implementierungen aufgezeigt wurden. Die Implementierung bei dem sozialen Netzwerk LinkedIn basiert auf der Open Authorization Version 1.0 und wurde als Vergleich zu der aktuellen Version 2.0 herangezogen um Änderungen zwischen den Versionen aufzeigen zu können. Es wurde auf wirtschaftliche Aspekte der Nutzung von Single Sign-On Systemen eingegangen und allgemeine Vorteile bei der Benutzung von sozialen Medien aufgeführt. Abschließend wurde ein Fazit über die Erkenntnisse dieser Arbeit verfasst und ein Ausblick gegeben, wie sich das Single Sign-On zukünftig entwickeln wird. Bei der Implementierung stellte sich heraus, dass die einzelnen sozialen Netzwerke sich nicht genau an die Vorgaben der Spezifikation des Open Authorization Protokolls halten müssen und in den einzelnen Implementierungen Unterschiede bestehen. Ebenfalls stellte sich heraus, dass die Implementierung des Protokolls in der Version 1.0 schwerer und umfangreicher ist als bei der aktuellen Version. Der Einsatz von Single Sign-On Implementierungen bedeutet für den Entwickler einer Anwendung oder eines Systems einen höheren zeitlichen Aufwand, der betrieben werden muss, wenn mehrere Authentifizierungsverfahren genutzt werden sollen, dieser sich aber durch eine höhere Nutzerzahl und eine bessere Reputation auszahlt.

Inhaltsverzeichnis

1. Einleitung

In der heutigen Zeit gewinnen soziale Netzwerke wie beispielsweise Facebook und LinkedIn immer mehr an Einfluss auf die Gesellschaft. Durch eine stetig wachsende Anzahl an Nutzern werden diese Dienste immer größer und ermöglichen es dem Nutzer mit anderen Personen in Kontakt zu treten und Meinungen und Aktivitäten im Internet zu verbreiten, um sich somit Bestätigung zu suchen und neue Person kennen zu lernen.

Dabei bieten die sozialen Netzwerke dem Nutzer eine Vielzahl von Möglichkeiten ihre Meinung zu verbreiten. Der Nutzer ist in der Lage, seine Informationen auf der eigenen Profilseite zu veröffentlichen, in Gruppen und Veranstaltungen mit anderen über einen Sachverhalt zu diskutieren, Informationen von anderen Kontakten zu kommentieren oder eine eigene Anwendungen auf den Seiten der sozialen Netzwerke zu erstellen.

Die Nutzung eines sozialen Netzwerkes setzt dabei eine Registrierung des Nutzers voraus in der Kontaktdaten des Nutzers angegeben werden müssen um ein eigenes Profil auf dem sozialen Netzwerk zu erhalten. Aktivitäten die der Nutzer anschließend auf dem sozialen Netzwerk durchführt werden seinem Profil zugeordnet und mit diesem verlinkt, sodass andere Nutzer die Aktivitäten verfolgen können.

Die Datenmenge, die dabei auf den sozialen Netzwerken über den Anwender gespeichert wird, ist durch die vielen Möglichkeiten der Nutzer eigene Inhalt zu kreieren und diesen auf den sozialen Netzwerken zu veröffentlichen sehr groß und übersteigt die Auffassungsmengen von kleineren Diensten, die nicht die Möglichkeit haben, eine so große Menge an Daten selbst zu verwalten.

Die Verwendung anderer Webanwendungen kann ebenfalls eine Registrierung von Nutzern verlangen, um Daten über die Nutzer speichern zu können und Aktivitäten der Nutzer zuordnen zu können. Der Nutzer braucht bei der Anmeldung einen Benutzernamen und ein Passwort, mit denen er sich bei der Anwendung authentifizieren kann und seine Identität bestätigt.

Durch die Verwendung von Single Sign-On Mechanismen soll hier die erneute Anmeldung für den Nutzer und die Anwendung erleichtert werden, indem die Anmeldeinformationen von einer Anwendung an die nächste Anwendung weitergeleitet werden und für eine weitere Authentifizierung des Nutzer verwendet werden können.

Die Authentifizierung des Nutzers wird über den Einsatz des Open Authorization Protokolls durchgeführt und soll dem Nutzer ermöglichen sich über ein soziales Netzwerk auf

einer anderen Webanwendung anmelden zu können und der Anwendung Zugang zu den benötigten Daten für den Registrierungs- bzw. Anmeldeprozess zu verschaffen.

1.1. Zielsetzung

Das Ziel dieser Bachelorarbeit ist es durch Gegenüberstellung des theoretischen Ablaufs, der in den Open Authorization Spezifikationen festgelegt ist, und den Implementierungen des Open Authorization Protokolls bei den sozialen Netzwerken Facebook, Google Plus und LinkedIn eine qualifizierte Aussage über die jeweilige Implementierung und die Anwendbarkeit eines solchen Authentifizierungs- und Autorisierungsverfahrens treffen zu können.

Die sozialen Netzwerke Facebook und Google Plus bieten die Möglichkeit eine Webanwendung über die aktuelle Open Authorization Version 2.0 zu implementieren und anschließend auf die jeweilige API zuzugreifen, um die Daten des Nutzers auszulesen. Zwischen Facebook und Google Plus wird aufgezeigt, welche Unterschiede es bei der gleichen Version von Open Authorization geben kann und wo Gemeinsamkeiten bestehen. Der Vergleich der Version 2.0 soll sichtbar machen, in welchem Maße sich der Einsatz des Open Authorization Protokolls an die Spezifikation halten muss.

Das soziale Netzwerk LinkedIn bietet nur die Möglichkeit sich über die Open Authorization Version 1.0 zu authentifizeren und wird zum Vergleich zwischen den Versionen 2.0 und 1.0 dienen. Der Vergleich der Versionen soll Aufschluss darüber geben, was sich mit dem Wechsel von der Version 1.0 auf die Version 2.0 getan hat und welche Auswirkungen diese Unterschiede auf die Implementierung und die Anwendbarkeit des Open Authoization Protokolls haben.

Desweiteren soll ein Überblick über das Open Authorization Protokoll gegeben werden und welche Varianten durch die Spezifikation für die Implementierung des Ablaufs der Kommunikation zwischen zwei Anwendungen angeboten werden. Durch den Überblick über die verschiedenen Flows des Protokolls soll eine Eingrenzung des Anwendungsbereiches des jeweiligen Flows erfolgen und eine Beurteilung der Abläufe hinsichtlich der Übergabe persönlicher Daten des Nutzer an die Anwendung erstellt werden.

Für die Implementierung des Open Authorization Protokolls wird eine webbaiserte Anwendung kreiert, die dem Anwender die Möglichkeit bietet sich über einen der drei betrachteten sozialen Netzwerke zu authentifizieren. Die Webanwendung wird als Grundlage der Implementierung benötigt, um die einzelnen Schritte der Implementierung des Open Authorization Protokolls aufzeigen zu können.

1.2. Ablauf

Nach der Einleitung in das Thema mit zugehöriger Zielsetzung und der Vorstellung des Ablaufs werden in Kapitel 2 grundlegende Konzepte und Begrifflichkeiten, die für das Verständnis und die Eingrenzung des betrachteten Anwendungsgebietes der Bachelorarbeit notwendig sind, vorgestellt und erläutert.

Als Grundlage in das Thema wird der Begriff der Authentifizierung sowie der Single Sign-On Authentifizierung erläutert und von der Autorisierung abgegrenzt, um Missverständnisse in der Benutzung der Begriffe zu vermeiden. Ebenfalls findet eine Einführung in die sozialen Medien statt und zu welchem Teilbereich der sozialen Medien die sozialen Netzwerke zählen und was unter dem Begriff soziales Netzwerk zu verstehen ist. Als Abschluss diese Kapitels dient eine Einführung in das Open Authorization Protokoll.

Im nächsten Schritt wird der theoretische Ablauf des Authentifizierungsverfahrens dargestellt. Es erfolgt eine genau Bertachtung des theoretischen Ablaufs einer Authentifizierung über das Open Authorization Protokoll, welche auf Grundlage der Spezifikation der Version 2.0 vorgenommen wird. Im Anschluss werden die weiteren Flows der Spezifikation kurz vorgestellt und Unterschiede aufgezeigt.

In Kapitel 4 wird die Implementierung einer solchen Authentifizierung über das Open Authorization Protokoll zwischen den betrachteten sozialen Netzwerken und der erstellten Anwendung dargestellt. Grundlegende Funktionen der erstellten Anwendung werden zuerst präsentiert und anschließend werden die Implementierungen der einzelnen sozialen Netzwerke im Detail betrachtet und die Erkenntnisse aus der Implementierung hinsichtlich der Unterschiede untereinander und zur Spezifikation erläutert. Abgeschlossen wird dieses Kapitel durch eine letzte Gegenüberstellung der Implementierungen.

Das folgende Kapitel beinhaltet die Vor- und Nachteile der Nutzung von Single Sign-On Mechanismen sowie die wirtschaftlichen Aspekte, die sich durch die Nutzung von Single Sign-On Authentifizierungen in Zusammenhang mit sozialen Netzwerken ergeben.

Im letzten Kapitel wird ein Ausblick gegeben, welche Änderungen zukünftig zu erwarten sind und welche anderen Technologien sich in einem frühen Stadium der Entwicklung befinden und zum Einsatz kommen könnten. Ein abschließendes Fazit, in dem die gewonnen Erkenntnisse der Bacheloarbeit zusammengefasst werden, wird zusammen mit einem Ausblick stattfinden.

2. Grundlegende Konzepte und Begrifflichkeiten

2.1. Authentifizierung

Die Authentifizierung ist ein Prozess in dem ein Subjekt oder Objekt seine Identität nachweist, indem er eine bestimmte Eigenschaft hat über die er verifiziert werden kann. Dabei wird zwischen der Authentifizierung von Nachrichten und Nutzern unterschieden. Bei der Authentifizierung von Nachrichten wird eine digitale Signatur verwendet, die den Nutzer eindeutig identifiziert. Die digitale Signatur wird verwendet, um die Nachricht, die der Nutzer erstellt hat, an ihn zu binden. Damit wird sichergestellt das Nachrichten im digitalen Verkehr immer einem Nutzer zugeordnet werden könen und sich der Nutzer somit verpflichtend an den Inhalt der Nachricht bei beispielsweise einem Kaufvertrag halten muss. [KCL10]

Der Nachweis der Identität eines Nutzers kann auf drei verschiedene Weisen erfolgen. Die erste Methode ist sich über ein biometrisches Merkmal authentifizieren zu können und somit die Authentizität der eigenen Person zu bestätigen. Biometrische Merkmale können hierbei ein Fingerabdruck oder ähnliche Eigenschaften des Subjektkörpers sein, durch die eine eindeutige Bestimmung der physischen Identität erreicht werden kann. Die zweite Methode wäre durch den Besitz eines Gegenstandes, der eine eindeutige Signatur enthält, wie beispielsweise eine Chipkarte. Eine Chipkarte enthält die Daten eines Nutzers auf dem Chip und kann eine eindeutige Erkennung des Besitzers ermöglichen, indem ein eindeutiger Wert auf dem Chip gespeichert ist. Zusätzlich kann ein Passwort bei der Nutzung eines Besitzes die Authentifizierung ergänzen. Die Benutzung eines Passwortes zählt zur dritten Möglichkeit sich zu authentifizieren, dem Nachweis von Wissen. Dieses Wissen sollte möglichst nicht weitergegeben werden, um einen Identitätsdiebstahl zu verhindern. [UR03]

Die Authentifizierung dient dabei sowohl dem Zweck der Integration als auch der Verbindlichkeit. Die Integrität ist dadurch gewährleistet, dass durch die Authentifizierung ein unbefugter Zugriff auf die Daten eines Nutzers verhindert wird. Die Verbindlichkeit ist gesichert, weil eine Aktion von einem Nutzer durch die Authentifizierung mit dessen Signatur versehen wird. [UR03]

2.1.1. Single Sign-On Authentifizierung

Das Prinzip des Single Sign-On gibt dem Nutzer die Möglichkeit sich über einen bereits angelegten Account bei einem System, welches ein Authentifizierungsverfahren voraussetzt, bei einem weiteren Dienst authentifizieren zu können ohne dabei eine neue Registrierung mit seinen Nutzerdaten durchzuführen. Die gespeicherten Daten der ersten Registrierung dienen anschließend dem zweiten System für die eigene Authentifizierung des Nutzers. Dazu werden die Daten des ersten Systems nach Anfrage durch den Nutzer an das zweite System übergeben und dieses kann dann die benötigten Daten für die Nutzeranmeldung verwenden. Das erste System muss zu diesem Zweck einen Single Sign-On Mechanismus bereitstellen, über den die Kommunikation zum zweiten System laufen kann. [Cof11]

Die Daten des ersten Systems können auf verschiedene Weise gehandhabt werden. Die erste Variante ist, das die Daten direkt an das zweite System übergeben werden sobald der Nutzer sich auf dem zweiten System anmelden will. Die zweite Variante wäre das die Daten indirekt genutzt werden, um weitere Daten des Nutzers zu erfragen, die im ersten System gepspeichert sind. Die dritte Möglichkeit ist, dass bei der Authentifizierung beim ersten System gleichzeitig die Authentifizierung beim zweiten System erfolgt. Die letzte Möglichkeit ist, das die Daten temporär gespeichert werden und bei Bedarf übermittelt werden. [Cof11]

2.2. Autorisierung

Die Autorisierung beschreibt einen Prozess, indem eine Person, eine IT-Komponente oder eine Anwendung einem Kommunikationspartner festgelegte Rechte einräumt. In der Informationstechnologie wird unter der Autorisierung der Zugriff auf Daten und Dienste verstanden. [KCL10]

Die Autorisierung befasst sich also im Gegensatz zur Authentifizierung mit der Vergabe von Zugriffsrechten, wobei die Authentifizierung genutzt wird um Zugriff auf eigene Daten zu erlangen. Die Autorisierung folgt häufig auf die Authentifizierung um weitere Rechte bei der Benutzung von Diensten oder Daten zu erlangen.

2.3. Soziale Medien

Die Sozialen Medien unterscheiden sich von den Massenmedien durch den ausschließlichen Vertrieb über digitale Kommunikationskanäle und Anwendungen. Die Konsumenten nehmen bei den sozialen Medien eine interaktive Rolle ein und werden somit zu einem wichtigen Bestandteil. Die Interaktion der Konsumenten wird durch Text, Bild, Video oder Audio erreicht und im Gegensatz zu den Massenmedien gibt es nur geringe Ein-

trittsbarrieren, um an den Medien teilzuhaben. Die sozialen Medien umfassen dabei zwei Dimensionen. Die erste Dimension ist der soziale Austausch, also die Kommunikation von Konsumenten. Sie ermöglichen es dem Konsumenten sich selbst Darzustellen und Informationen über die eigene Person preiszugeben, um sich selbst anderen Personen zu offenbaren. [Hae10]

Die zweite Dimension umfasst die Kooperation, um auf Grundlage der Kommunikation einen vom Konsumenten bestimmten Inhalt zu schaffen, dem User Generated Content. Andreas M. Kaplan und Michael Haenlein klassifizieren die sozialen Medien in sechs Kategorien auf Grundlage von vier Medientheorien. Die Dimension der Kommunikation umfasst die Theorien des Impression Management und die Self-Disclosure Theorie. Die Dimension der Kooperation wird durch die Medienreichhaltigkeitstheorie und die Social Presence Theorie abgebildet. (siehe Anhang A) [Hae10]

Kollektivprojekte, wie Wikipedia, und Blogs, wie Twitter, nehmen in Ihrer Klassifizierung den niedrigsten Stellenwert ein, weil sie textbasiert sind und meist nur geringe Möglichkeiten für weitere Medien bereitstellen. Ebenfalls ist die Interaktion der Konsumenten begrenzt möglich, sodass die Dimension der Kooperation zum Erschaffen neuer Inhalte nur gering gegeben ist. Auf der zweiten Ebene sind die Content Communities, wie Youtube, und die sozialen Netzwerke, wie Facebook, klassifiziert. Dem Konsumenten sind hier die Möglichkeiten geboten, weitere Medien wie Bilder und Videos zu nutzen und diese in Kooperation mit weiteren Konsumenten zu erschaffen und zu bewerten. Auf der höchsten Ebene sind nach Kaplan und Haenlein die virtuellen Spiele, wie World of Warcraft, und virtuelle Welten, wie Second Life, eingeteilt, die dem Konsumenten die größte Vielfalt an Kommunikationsmitteln erlauben und den direkten Kontakt von Konsumenten erlauben und somit die Kooperation unterstützen. [Hae10]

2.3.1. Soziale Netzwerke

Der Begriff der sozialen Netzwerke beschreibt in seiner grundlegenden Form die Beziehung von Menschen zueinander. Soziale Netzwerke treten überall auf, wo Menschen mit anderen Menschen in Kontakt treten, zum Beispiel innerhalb einer Familie, eines Freundeskreises oder in einer Firma. In Bezug auf die sozialen Medien ist der Begriff der sozialen Netzwerke durch den Namen Soziale Netzwerk Dienste(kurz SND) oder Online Soziale Netzwerke(kurz OSN) von der ursprünglichen Bedeutung abgegrenzt. Die SNDs zeichnen sich besonders durch den Fokus auf die Konsumenten und ihre Beziehungen untereinander aus. [KCL10]

Zu den typischen Merkmalen der sozialen Netzwerke gehören die Registrierung des Konsumenten, die zu Beginn der Nutzung einmalig durchgeführt werden muss und grundlegende Informationen der Konsumenten wie Name, Emailadresse, Anschrift und Geburtstag beinhaltet. Ein weiteres Merkmal ist die Profilseite auf der die bereitgestellten Daten des

Konsumenten für andere Nutzer des sozialen Netzwerks in strukturierter Form dargestellt werden und die Beziehung zu anderen Nutzern offengelegt werden.

Die SNDs bieten dem Konsumenten eine Vermittlungsfunktion, die er in seinem natürlichen Umfeld nicht besitzt und ermöglichen es ihm neue Kontakte zu knüpfen, die sowohl für das Finden eines Berufes, neuer Freunde oder Lebenspartner geeignet sind. Ältere Vermittlungsdienste wie Klein-Anzeigen, Stellenmärkte und Partnervermittlungen werden in sozialen Netzwerken für den Konsumenten erweitert und können diese ersetzen. [Stu11]

2.4. Open Authorization

Open Authorization(kurz OAuth) ist ein offenes Protokoll, das im Oktober 2007 veröffentlicht wurde. Mit der zunehmenden Dezentralisierung und dem Cloud Computing tritt der Prozess der Authentifizierung und Autorisierung vermehrt auf und das OAuth Protokoll bietet dem Konsumenten eine einfach Alternative seine bereits verwendeten Nutzerdaten von einer Anwendung an eine Andere zu übergeben ohne diese manuell eintragen zu müssen. [Ham10]

OAuth wurde entworfen, um eine standardisierte API-Delegation zu erstellen, mit deren Hilfe von einer höhergeordneten Instanz Zugriffsrechte an untergeordnete Instanzen gegeben werden können. Die Umsetzung dieses Standards wurde zuerst nur von Blaine Cook, der vor dem Entwurf von OAuth mit einer Implementierung von OpenID für Twitter beschäftigt war, und Chris Messina von Ma.gnolia initiiert. [Ham10]

Iim November 2008 wurde der Einrichtung einer OAuth Arbeitsgruppe in der Internet Engineering Task Force(kurz IETF) zugestimmt und die weitere Entwicklung von OAuth von dieser übernommen. Die IETF Organisation beschäftigt sich mit der technischen Weiterentwicklung des Internets, um dessen Funktionsweise unter zuhilfenahme von standardisierten Internetprotokollen zu verbessern. Die IETF Organisation wird dabei von bekannten Firmen wie dem Google Unternehmen, Facebook, Twitter und weiteren namhaften Unternehmen unterstützt. [Ham10]

OAuth nimmt in dem traditionellen Client-Server Authentifizierungsmodell eine neue Rolle ein, die des Ressourcenbesitzers. Der Client erfragt Daten, die vom Ressourcenbesitzer kontrolliert werden, aber auf einem anderen Server abgelegt sind. Der Server erhält dabei die Möglichkeit sowohl die Autorisierung des Ressourcenbesitzers als auch die Identität des Clients zu überprüfen. Der Ressourcenbesitzer kann hierbei ein anderen Client sein oder ein Endnutzer. Die typische Nutzung des OAuth Protokolls umfasst dabei die Authentifizierung des Resourcenbesitzers, die Autorisierung der gewählten Anwendung durch den Resourcenbesitzer sowie die Authentifizierung der Anwendung beim Server. [DR12]

3. Theoretischer Ablauf des OAuth Authentifizierungsmechanismus

In diesem Kapitel geht es um den Theoretischen Ablauf des OAuth Protokolls, der durch die Spezifikation der jeweiligen Version festgelegt ist. Zunächst einmal werden die Voraussetzungen betrachtet, die vor Beginn einer Implementierung beachtet werden müssen, wenn über das OAuth Protokoll zwischen der entwickelten Anwendung und dem Autorisierungsserver kommuniziert werden soll. Die entwickelte Anwendung bezeichnet hier die Anwendung, auf der sich Nutzer über den Autorisierungsserver authentifizieren wollen, im weiteren als Client bezeichnet. Der Autorisierungsserver ist der Server, der die Daten des Nutzers verwahrt und wo der Nutzer ein bestehendes Profil besitzt über das er es sich anmelden kann. Anschließend werden die einzelnen Schritte am Beispiel des Authorization Code Grant erläutert, weiterhin serverseitiger Flow genannt, die vom Entwickler der Anwendung durchzuführen sind, damit sich Nutzer über den Autorisierungsserver anmelden können. Eine grafische Darstellung des serverseitigen Flows ist im Anhang B enthalten. Es folgt eine Übersicht über die weiteren Abläufe, die in den OAuth Spezifikationen aufgelistet sind und es wird eine Eingrenzung des jeweiligen Anwendungsgebiets vorgenommen. Im letzten Schritt dieses Kapitels werden die im Evaluationsbeispiel genutzten APIs der Autorisierungsserver kurz vorgestellt.

3.1. Voraussetzungen

Bevor mit der Implementierung des Programmiercodes angefangen werden kann, wird für den jeweiligen Autorisierungsserver, mit der der Client verbunden werden soll, eine API benötigt. Die API bietet die benötigten Schnittstellen und Interaktionsmöglichkeiten um mit dem Autorisierungsserver zu kommunizieren und den weiteren Kommunikationsverlauf zwischen diesem und dem Client zu gewährleisten. Die APIs sollten dabei von dem Autorisierungsserver bereitgestellt werden, um die berechtigten Zugriffsmöglichkeiten und Funktionen im Rahmen der normalen Verwendung bereitzustellen.

Der Entwickler eines Clients, der sich mit einer anderen Seite authentifizieren will, muss sich ebenfalls über die eigene Entwicklungsumgebung Gedanken machen. Diese kann es ihm erleichtern die verschiedenen Schritte während der Implementierung und das Ein-

binden der API durchzuführen, indem es auf die Entwicklungsumgebung spezialisierte Bibliotheken geben könnte. Die Bibliotheken stellen dem Entwickler meist Klassen und Methoden bereit mit deren Hilfe einzelne Schritte der Implementierung erleichtert werden, weil die Schritte nicht selbst implementiert werden müssen, sondern funktionierende Komponenten genutzt werden können.

Ebenfalls muss festgelegt werden, um welche Art von Client es sich handeln soll, also ob es eine einfache Anwendung für ein Smartphone, eine Anwendung auf dem internen System oder eine Webanwendung handeln soll, die auf einem Server ausgeführt wird. Je nach Art der Anwendung unterscheidet sich der einzusetzende Ablauf des OAuth Protokolls.

3.2. Schritte des serverseitigen Flows des OAuth Protokolls

3.2.1. Client Registrierung

Die Client Registrierung ist der erst Schritt bei der Benutzung des OAuth Protokolls, der eine Interaktion zwischen Autorisierungsserver und Entwickler des Clients benötigt. Der Entwickler benötigt für die Registrierung seines Clients meist ein Benutzerkonto bei dem Autorisierungsserver, mit dessen Hilfe er seinen Client über ein Anmeldeformular registrieren kann. In dem Anmeldeformular muss der Entwickler den Namen des Clients angeben sowie die URL, an die der Autorisierungsserver Anfragen und Antworten auf Fragen senden soll, im weiteren Redirection URL genannt. Das Anmeldeformular umfasst meist auch eine Beschreibung des Clients sowie ein Feld, in dem das Logo des Clients eingefügt werden kann. Ebenfalls müssen die geltenden Benutzungsrichtlinien des Autorisierungsservers zugestimmt werden, damit einem unsachgemäßen Umgang mit den Daten des Autorisierungsservers rechtlich vorgesorgt ist.

Die Registrierung wird benötigt, um zwei Schlüssel-Wert Paare zu erhalten. Das erste Paar ist die Client ID, die benötigt wird, damit der Client eindeutig zugeordnet werden kann. Das zweite Paar ist ein Client Secret, über den der Client sich bei dem Autorisierungsserver authentifizieren kann. Dies ist ebenfalls eine Sicherheitsmaßnahme, um eine unberechtigte Nutzung zu verhindern. Sowohl die Client ID als auch das Client Secret sind einzigartige Zeichenfolgen, die der Autorisierungsserver nur einmal vergeben sollte.

Das Client Secret wird nur verwendet, wenn es sich um einen vertraulichen Client handelt, bzw. wenn Daten des Autorisierungsservers als vertraulich eingestuft sind. Handelt es sich lediglich um öffentliche Daten auf die jeder Zugang hat, wird kein Client Secret benötigt.

Als nächstes wird der Autoriserungsendpunkt aufgerufen und der nächste Schritt des Authentifizierungsprozesses beginnt.

3.2.2. Nutzer Authentifizierung

Der Autorisierungsendpunkt ist eine URL Adresse zu dem der Client weitergeleitet wird, wenn sich ein Nutzer mit dem Client authentifizieren will. Das Erlangen dieser Adresse ist je nach Autorisierungsserver unterschiedlich und muss vom Entwickler in Erfahrung gebracht werden. Üblicherweise findet sich diese Adresse in der Dokumentation des Autorisierungsserves oder muss auf andere Art vom Autorisierungsserver in Erfahrung gebracht werden wie beispielsweise über telefonischen Kontakt mit einem bereitgestellten Kundenservice.

Die Weiterleitung zum Autorisierungsendpunkt erfolgt durch eine HTTP GET Anfrage, in der im Header vom Client ein Schlüssel-Wert Paar Client ID sowie ein weiteres Paar für die Wahl des zu verwendenden OAuth Flows, der als Antworttyp bezeichnet wird. Optional können die Redirection URL sowie ein Schlüssel-Wert Paar für die Reichweite der erfragten Daten vom Client angegeben werden, weiterhin scope genannt. Ebenfalls wird die Angabe eines eindeutigen Schlüssel-Wert Paares mit der Bezeichnung state empfohlen, um das Abfangen der versendeten Anfragen und Antworten zu erschweren.

Die Nutzer Authentifizierung wird genutzt, um die Identität des Nutzers zu überprüfen und den Zugang vor unautorisierten Personen zu verhindern. Die Authentifizierung wird üblicherweise durch die Abfrage eines Nutzernamens und eines Passwortes durchgeführt. Es bietet sich ebenfalls die Möglichkeit die Authentifizierung über bereits aktive Cookies oder andere Mechanismen zur Überprüfung der Identität ausführen zu lassen, wie zum Beispiel OpenID.

Bei der Nutzer Authentifizierung muss der Autorisierungsserver Sicherheitsmaßnahmen in seinem Authentifizierungsprozess einbauen, damit sichergestellt ist, das bei der Übermittlung des Nutzernamens und des Passwortes keine Informationen an Dritte übergeben werden. Welche Sicherheitsmaßnahmen ergriffen werden, ist je nach Autorisierungsserver unterschiedlich und es sind keine festen Richtlinien durch das OAuth Protokoll vorgegeben.

Ist die Authentifizierung erfolgreich durchgeführt wird der Nutzer zur Client Autorisierung weitergeleitet.

3.2.3. Client Autorisierung

Der Nutzer wird in diesem Schritt aufgefordert den Client zu autorisieren, damit Daten, die an den Client übermittelt werden sollen, vom Autorisierungsserver erhalten werden können. Die zu übermittelnden Daten werden ihm in seinem Webbrowser angezeigt und es ist eine Bestätigung durch den Nutzer erforderlich, um diesen Schritt abzuschließen.

Die Client Autorisierung ist ein zwingender Prozess, der bei jedem Nutzer mindestens einmal ausgeführt werden muss. Das Schlüssel-Wert Paar scope, eingeführt in Kapitel

3.2.2, dient hier zum Anzeigen der erfragten Daten.

Der User wird anschließend per HTTP Antwort zur Redirection URL des Clients weitergeleitet. Die HTTP Anfrage enthält neben der Redirection URL noch ein Schlüssel-Wert Paar im Header, in der ein Autorisierungscode angegeben ist und ein unverändertes Schlüssel-Wert Paar state, sofern zuvor eins übertragen wurde.

3.2.4. Access Token Anfrage

Die Access Token Anfrage wird genutzt, um ein Access Token vom Tokenendpunkt des Autorisierungsservers zu erhalten. Das Access Token wird im nächsten Schritt zur Übermittlung der Nutzerdaten vom Autorisierungsserver zum Client benötigt.

Nach dem Erhalt des Autorisierungscodes kann der Client eine HTTP POST Anfrage an den Tokenendpunkt des Autorisierungsservers senden, damit das Access Token übertragen werden kann. Der Unterschied zwischen einer HTTP GET Anfrage und einer HTTP POST Anfrage besteht darin, dass bei einer GET Anfrage die Schlüssel-Wert Paare im Header der URL aufgelistet sind und in der POST Anfrage im Körper der Anfrage. Die POST Anfrage ist daher bei der Übermittlung von sicherheitskritischen Anfragen zu bevorzugen, weil die Daten nicht direkt aus der URL ausgelesen werden können.

Zusätzlich zum Autorisierungscode werden in dem HTTP POST Körper die Schlüssel-Wert Paare Grant Type, für den verwendeten Flow, und die Redirection URL, sofern sie zurvor angegeben würde, benötigt.

Handelt es sich um einen vertraulichen Client, so muss zusätzlich zu den bereits genannten Schlüssel-Wert Paaren die Client ID und das Client Secret angegeben werden, damit der Client beim Autorisierungsserver authentifiziert werden kann.

Wenn die Schlüssel-Wert Paare der Anfrage gültig waren, wird ein Access Token an den Client übergeben. Die Übergabe der Daten erfolgt dabei durch das Einfügen der Daten in den Körper der HTTP Antwort des Autorisierungsservers. In ihm werden immer das Access Token als Zeichenfolge und der Tokentyp aufgelistet. Die Angabe eines Schlüssel-Wert Paares für die Ablaufzeit des Access Tokens wird empfohlen. Optional können Schlüssel-Wert Paare für ein Refresh Token und das scope Paar enthalten sein. Das Refresh Token wird für eine spätere Anfrage nach einem Access Token verwendet, wenn das Access Token abgelaufen ist und der Nutzer den OAuth Ablauf nicht erneut durchlaufen hat.

3.2.5. Zugriff auf die Nutzerdaten

Nach Erhalt des Access Tokens vom Autorisierungsserver, kann der Client eine Anfrage nach den in der Reichweite enthaltenen Daten beim Resourcenserver machen. An dieser Stelle wird auf die jeweilige API des Resourcenservers zugegriffen. Dazu muss das Access Token in der Anfrage in Form einer HTTP GET Anfrage mitgeliefert werden und

vom Resourcenserver validiert werden. Die Validierung enthält meist eine Koordination zwischen Autorisierungs- und Resourcenserver, kann aber auch auf anderen Wegen erfolgen. Während der Koordination wird die Vergabe des Access Tokens sowie die Ablaufzeit überprüft.

Die Antwort auf die HTTP Anfrage kann anschließend zur Erstellung eines JSON Objekts genutzt werden, aus dem die erfragten Daten durch den Client ausgelesen werden können.

3.2.6. Refresh Token

Der Client, kann sofern er ein Refresh Token vom Autorisierungsserver erhalten hat, ein weiteres Access Token erhalten. Das Refresh Token wir dafür an den Tokenendpunkt in einer HTTP POST Anfrage übergeben und enthält zusätzlich ein Schlüssel-Wert Paar für den Grant Typ sowie optional das scope Paar. Die Reichweite darf dabei nicht über die vom Endnutzer autorisierte hinausgehen. Das Schlüssel-Wert Paar Grant Typ muss bei dieser Anfrage auf den Wert "refresh_token"gesetzt werden, damit der Autorisierungsserver die Anfrage richtig erkennt. Wenn der Client Typ vertraulich war muss zusätzlich der Client erneut authentifiziert werden.

Nach der Validierung des Refresh Tokens wird, wie in Kapitel 3.2.6 beschrieben, ein Access Token an den Client übermittelt. Die Antwort des Autorisierungsservers kann dabei ebenfalls ein neues Refresh Token enthalten, welches das Genutzte ersetzt und daher vom Client gespeichert werden sollte.

3.3. Unterschiede der Oauth Flows

Durch die Spezifikation des OAuth Protokolls werden drei weitere Flows definiert. Der Implicit Grant, auch clientseitiger Flow genannt, der Ressource Owner Password Credentials Grant und der Client Credentials Grant.

3.3.1. Clientseitiger Flow

Der clientseitige Flow wird genutzt, um ein Access Token zu erhalten, bietet allerdings nicht die Möglichkeit ein Refresh Token zu erhalten, weil er den Tokenendpunkt nicht verwendet, sondern direkt ein Access Token nach der Autorisierung durch den Nutzer erhält. Wie beim serverseitigen Flow handelt es sich um einen weiterleitenden Flow und es muss daher die Möglichkeit gegeben sein mit dem Webbrowser zu interagieren und eingehende Anfragen zu verwerten. Anders als bei dem serverseitigen Flow wird der Access Token nicht separat nach der Autorisierung erfragt, sondern als Ergebnis der Autorisierung übermittelt. Im Gegensatz zum serverseitigen Flow bietet sich der clientseitige Flow bei

Webanwendungen an, die nicht wie der serverseitige Flow in der Lage sind ein Geheimnis zu bewahren. Für eine grafische Darstellung siehe Anhang C.

Der erste Schritt des clientseitigen Flows ist wie beim serverseitigen Flow. Der Client initialisiert den Flow durch Weiterleitung des Webbrowsers des Nutzers zum Autorisierungsserver. Es werden die gleichen Daten mitgeliefert wie in Kapitel 3.2.2 beschrieben.

Der zweite und dritte Schritt ist ebenfalls identisch und es findet die Authentifizierung und Autorisierung durch den Nutzer statt.

Der nächste Schritt unterscheidet sich vom serverseitiger Flow. Sofern der Nutzer den Client autorisiert hat, wird der Webbrowser zurück zum Client geleitet und enthält dabei das Access Token in einem URL Fragment.

Der Webbrowser des Nutzers folgt nun den Weiterleitungsinstruktionen, indem eine Anfrage an die auf einem Webserver gespeicherten Client Resourcen gesendet wird. Die Anfrage enthält dabei nicht mehr das Access Token Fragment, sondern das Fragment wird lokal vom Webbrowser gelagert.

Auf einem Webserver gelagerte Resourcen des Clients geben nun eine Webseite wieder, die typischerweise aus einem HTML Dokument mit einem eingebetteten Skript besteht, das in der Lage ist auf die komplette URL, die vom Webbrowser bewahrt wurde, zuzugreifen und anschließend das Access Token und weitere Parameter aus dem Fragment zu extrahiert.

Im letzten Schritt des clientseitigen Flows führt der Webbrowser des Nutzers das Skript aus, validiert das Access Token beim Autorisierungsserver und gibt es an den Client weiter.

3.3.2. Resource Owner Password Credentials Grant

Der Resource Owner Password Credentials Grant bietet sich bei Anwendungen an, wo zwischen dem Nutzer und dem Client ein hohes Maß an Vertrauen gilt zum Beispiel wenn der Client das Betriebssystem des Nutzers ist und Daten von einem anderen Server benötigt werden. Der Autorisierungsserver sollte hier besondere Vorkehrungen treffen und diesen Flow nur zugänglich machen, falls kein anderer Flow verfügbar ist. Der Client muss ebenfalls in der Lage sein den Nutzernamen und das Passwort des Nutzers zu erhalten. Eine grafische Darstellung ist in Anhang D enthalten.

Dieser Flow unterscheidet sich stark von den bereits erwähnten Flows. Im ersten Schritt liefert der Endnutzer dem Client seine Authentifizierungsdaten.

Im nächsten Schritt erfragt der Client ein Access Token vom Tokenendpunkt des Autorisierungsserver, indem die Authentifizierungsdaten des Nutzers in der Anfrage mitgeliefert werden. Während der Anfrage wird ebenfalls der Client authentifiziert.

Im nächsten Schritt folgt folgt nun die eigentliche Authentifizierung durch den Autorisierungsserver sowie das Validieren der Authentifizierungsdaten des Nutzers. Wenn beides korrekt ist, wird ein Access Token vergeben.

3.3.3. Client Credentials Grant

Dieser Flow bietet sich hauptsächlich an, wenn ein Nutzer auf seine eigenen Daten zugreifen will. Dies kann vorkommen, wenn der Nutzer von einem System, das nur über geringe Speichermöglichkeiten verfügt, wie beispielsweise eine Fotokamera, Daten zu einem anderen System mit einem größeren Speicher übertragen will. Ein grafische Übersicht ist in Anhang E enthalten.

Dieser Flow unterscheidet sich ebenfalls stark von den bereits genannten. Der Client authentifiziert sich hier lediglich mit seinen eigenen Authentifizierungsdaten mit dem Autorisierungsserver und erfragt ein Access Token vom Tokenendpunkt.

Wenn die Authentifizierung gelungen ist, wird ein Access Token vom Autorisierungsserver an den Client zurückgegeben.

3.4. Eingesetzte APIs

3.4.1. Facebook Graph API

Einführung Facebook

Facebook ist ein sozialer Netzwerk Dienst, der im Jahr 2004 unter der Domain www.facebook.com gelauncht wurde. Zunächst nur für Studenten der Harvard University bereitgestellt , wurde der Zugriff für Studenten anderer Universitäten erweitert und steht seit 2006 für jeden Anwender zur Verfügung. Der Name „Facebook“ hat seinen Ursprung in der amerikanischen Tradition ein Bilderbuch über die Studenten an Colleges auszugeben. [Stu11]

Im Mai 2007 wurde die Facebook Plattform in Facebook integriert und bot von da an den Nutzern die Möglichkeit eigene Anwendungen zu erstellen, die mit den Facebook Features interagieren, wodurch neuer Content auf Facebook von den Nutzer erstellt werden kann.

Heutzutage bietet die Facebook Platform sogenannte Social Plugins und Authentifizierungsverfahren an, wodurch der Nutzer nicht nur Anwendungen auf der Facebookseite erstellen kann sondern auch eigen erstelle Webseiten mit der Facebook Platform verbinden und somit auf Funktionen und Daten des Netzwerkes zugreifen kann. [Fac]

Besonderheiten der Graph API

Die Graph API von Facebook bietet den Entwicklern die Möglichkeit auf Daten der Nutzer zuzugreifen. Per Anfrage an die Graph API können Daten erfragt und als JSON Objekt ausgegeben werden. Öffentliche Daten können ohne weitere Autorisierung abgefragt werden, indem die ID des Nutzers bekannt ist oder der Nuzername als ID verwendet wird.

So können alle öffentlichen Daten einer Person oder einer Seite auf Facebook abgerufen werden. [Fac]

Für den Zugang zu weiteren persönlichen Daten des Nutzers muss der Entwickler ein Access Token von Facebook erhalten. Die Authentifizierung und Autorisierung durch den Nutzer erfolgt dabei über das OAuth 2.0 Protokoll. Für diesen Vorgang muss der Entwickler eine Implementierung des Ablaufes vornehmen.

Facebook bietet dabei neben dem Auslesen der Daten durch die API auch die Möglichkeit Nachrichten an den Nutzer zu schicken.

3.4.2. Google API

Einführung Google Plus

Google Plus ist ein sozialer Netzwerk Dienst, der seit Juni 2011 unter der Domain plus.google.com erreichbar ist. Zuerst nur auf Einladung von bereits registrierten Nutzern verfügbar, ist es seit September 2011 frei zugänglich. Google Plus gehört zur Google Inc. und ersetzt das zuvor vertriebene soziale Netzwerk Orkut der Google Gruppe. [Goo]

Google Plus setzt dabei wie Facebook auf eine Online Community, die ihre freizeitlichen Interessen und Aktivitäten auf Google Plus veröffentlichen können.

Besonderheiten der Google API

Die Google API bietet den Entwicklern die Möglichkeit auf die Daten der verschiedenen Google Dienste zuzugreifen, darunter auch Google Plus. Dafür bieten sie eine Reihe von Autorisierungs- und Authentifizierungsmechanismen, um mit dem Nutzer und den Google Servern zu kommunizieren. [Goo]

Google bietet einerseits die Möglichkeit sich über das OAuth 2.0 Protokoll zu authentifizieren und zu autorisieren, sowie über die Vorgängerversion OAuth 1.0. Bei Nutzung des OAuth 2.0 Protokolls sind verschiedene Flows für die jeweilige Anwendung verfügbar.

Ebenfalls bietet Google die Möglichkeit sich über OpenID und eine hybridform aus OpenID und OAuth zu authentifizieren und zu autorisieren.

Die Wahl der angesprochenen API wird in den Google OAuth Flows durch die Festlegung der zu erfragenden Reichweite durch das in Kapitel 3.2.2 vorgestellte scope Schlüssel-Wert Paar und der folgenden Autorisierung durch den Nutzer sowie bei der späteren Anfrage an die Google API mit dem Access Token des OAuth Dialogs bestimmt.

Bei Google Plus bietet Google nur die Möglichkeit Daten auszulesen, aber nicht Nachrichten an den Nutzer zu schicken.

3.4.3. LinkedIn API

Einführung LinkedIn

LinkedIn ist ein sozialer Netzwerk Dienst, der sich mit dem beruflichen Umfeld der Nutzer befasst, sowie mit dem Knüpfen neuer Geschäftsbeziehungen. Es wurde 2003 gegründet und ist somit das älteste betrachtete Netzwerk. [Stu11] LinkedIn wurde unter der Domain www.linkedin.com veröffentlicht.

Im Gegensatz zu den bereits vorgestellten sozialen Netzwerk Diensten ist LinkedIn nicht im privaten Sektor angesiedelt, sondern befasst sich mit den beruflichen Aspekten der Nutzer. Nutzer haben die Möglichkeit Lebensläufe zu veröffentlichen und auf Stellenangebote von anderen Nutzern oder Unternehmen zu reagieren bzw. von Unternehmen angeworben zu werden.

Für die generelle Nutzung von LinkedIn fallen für den Nutzer keine Kosten an, jedoch kann auch ein gebührenpflichtiges Profil erstellt werden, mit dem weitere Funktionen möglich sind.

Besonderheiten der LinkedIn API

Die API von LinkedIn ermöglicht es Entwicklern eigene Clients über das OAuth Protokoll in der Version 1.0 mit LinkedIn zu verknüpfen. Weil LinkedIn im beruflichen Sektor der Nutzer aktiv ist, ist die Nutzung des OAuth 1.0 Protokolls vorteilhaft, weil die Reichweite der erfragten Daten nicht vorab festgelegt werden muss und eine Vielzahl von Daten über die Unternehmen und Qualifikationen der Nutzern zugänglich gemacht wird, sofern diese vom Nutzer bereitgestellt werden. [Lin]

Für eine einfache Authentifizierung mit dem LinkedIn Profil wird ebenfalls der clientseitige Flow der OAuth Version 2.0 angeboten, über den ein Nutzer sich auf einer anderen Anwendung authentifizieren kann. Der Reichweite der Daten, die vom Client abgerufen werden können, sind hier allerdings Grenzen gesetzt, sodass diese Variante lediglich zur einfachen Authentifizierung genutzt wird und keine richtige Autorisierung auf das gesamte Profil des Nutzers vorgesehen ist.

4. Evaluationsbeispiel

4.1. Börsenspiel

Als Evaluationsbeispiel wurde ein Börsenspiel kreiert, welches zusammen mit dem Institut für Agrarökonomie an der Christian Albrechts Universität zu Kiel geplant wurde. In dem Börsenspiel geht es darum, dem Anwender den Börsenmarkt spielerisch darzustellen und den Ablauf und die Möglichkeiten des Börsenmarktes abzubilden. Als Grundlage für die Funktionen des Börsenspiel wurde ein bereits laufendes System als Vorbild gewählt, welches von dem Institut vertrieben wird. Das laufende Börsenspiel ist seit einigen Jahren in Betrieb und wird jeweils im Wintersemestern der Universität Kiel für Interessenten zu einem derzeitigen Preis von 20 Euro angeboten. Durch die Einnahmen des Börsenspiels werden einerseits die Kosten des Servers für das Börsenspiel getragen und andererseits Geldgewinne für die Spieler als zusätzlichen Anreiz ausgegeben. Durch die langjährige Laufzeit und den unzureichend dokumentierten Entwicklungsprozess und Optimierungsprozess des Systems ist die Weiterentwicklung des alten Systems als ineffizient erklärt worden und es wurde der Entwicklung eines neuen Systems zugestimmt.

Das neu kreierte Börsenspiel bieten dem Anwender die Funktion Terminkontrakte zu kaufen, die einen Kaufvertrag zwischen dem Börsenmarkt und dem Spieler repräsentieren, im weiteren Verlauf Futures genannt. Ein Future zeichnet sich dabei dadurch aus, dass ein Käufer zu einem festgelegten Zeitpunkt eine festgelegt Menge von Waren vom Verkäufer verbindlich abzunehmen hat. Der Spieler kann dabei sowohl die Rolle des Käufers, durch den Kauf am Börsenmarkt, und die Rolle des Verkäufers an den Börsenmarkt einnehmen. Die Preise und zur Verfügung stehenden Mengen werden dabei in einer Datenbank verwaltet und orientieren sich an den aktuellen Börsenkursen für die jeweiligen Güter. Die Börsenkurse wurden im alten System zu diesem Zweck aus einer Liste auf einer Internetdomain in eine Microsoft Excel Datei übertragen und anschließend in Javacode übertragen und im eigenen System eingebunden. Durch die unzureichend dokumentierte Implementierung des alten Systems und die mangelnden Fachkenntnisse zur Wartung des Systems von den Angestellten des Instituts sollte hier während der Erstellung des neuen Systems eine effizientere und verständlichere Variante implementiert werden, die Daten als PDF Dateien erhält und anschließend das PDF Dokument sofort in Java ausliest und in die Datenbank überträgt.

Der Spieler hat während der laufenden Runde weiterhin die Möglichkeit seine Daten einzusehen, sowie gekaufte Güter zu verwalten. Ebenfalls bietet die Vorlage die Option Aktienkurse grafisch darzustellen, um somit Preisentwicklungen transparenter zu machen. Durch die grafische Darstellung der Preisentwicklung wird dem Spieler ein Überblick über vergangene Kurse gegeben, auf dessen Grundlage der Spieler eine Prognose über den weiteren Verlauf aufstellen kann und eine Orientierung des Kursverlaufes an der Börse erhält.

4.1.1. Entwicklungsumgebung

Die Implementierung des neuen Systems als dynamisches Webprojekt in Eclipse ist auf Grundlage des Model-View-Controller Architekturmusters(kurz MVC-Muster) entworfen. [Bal05] Das Modell ist in diesem Muster für die Verwaltung der Daten in der Datenbank zuständig und bietet die notwendigen Funktionen neue Objekte der einzelnen Datenklassen zu erstellen und zu bearbeiten. Der Controller verfügt über die notwendigen Methoden, um auf die einzelnen Objekte des Modells zuzugreifen und Änderung an ihnen zu veranlassen. Die Veränderungen werden dann in dem Modell vorgenommen und die View wird nach Bedarf angepasst. Die View ist dafür zuständig die Daten des Modells anzuzeigen und die Präsentation anzupassen.

In dem Börsenspiel ist das Modell durch die Klassen in dem Paket futures.entity dargestellt in dem Entities die Objekte erzeugen. Die Controller Klassen befinden sich in dem Paket futures.control und enthalten Methoden für die Steuerung der Futures, der Spieler und für Informationen. Die Darstellung der View wird im Paket futures.presentation durch einzelne Java Beans veranlasst.

4.1.2. Authentifizierung von Spielern

Um die Spieler zu verwalten wurden mehrere Authentifizierungsmechanismen implementiert, die es den Anwendern erlauben sich über mehrere Varianten mit dem Börsenspiel zu authentifizieren und die Autorisierung zur Anwendung des Systems zu erhalten. Der grundlegende Authentifizierungsmechanismus umfasst die Registrierung des Spielers bei erstmaliger Benutzung des Systems und ein Anmeldeverfahren für die erneute Verwendung.

Die simpelste Authentifizierung findet auf den Seiten des Börsenspiels selbst statt. Der Spieler kann per Betätigung eines Links auf der Startseite die Registrierung starten. Die Klasse session im Paket futures.beans wird hierbei eingesetzt und verwaltet die Darstellung des Webbrowsers in den nachfolgenden Schritten. Durch die Betätigung des Links wir der Nutzer durch die session auf eine neue HTML Seite weitergeleitet. Auf dieser wird der Benutzer gebeten den Nutzerbedingungen zuzustimmen, um anschließend die für die

Registrierung notwendigen Daten angeben zu können.

Der Nutzer hat nun die Möglichkeit seine Daten auf der HTML Seite in HTML Felder einzutragen. Die eingetragenen Daten werden nach Eingabe in der Session Bean als Player Entität für den weiteren Verlauf der Session verknüpft. Wenn alle erforderlichen Daten eingetragen wurden, kann per Betätigung des Login Button die Registrierung abgeschlossen werden, indem die Methode PlayerBean.save() ausgeführt wird. Die Methode überprüft zuerst ob die erstellte Entität bereits in der Datenbank vorhanden ist. Dazu wird eine Anfrage an die derzeitig instanzierte Steuerung gestellt und eine SQL Anfrage auf der Datenbank durchgeführt, die überprüft ob die verwendete Emailadresse bereits in der Datenbank vorhanden ist. Ist die Emailadresse nicht vergeben wird der Nutzer in die Datenbank eingetragen und als angemeldeter Spieler durch die Player Bean auf die Startseite für Spieler weitergeleitet und kann neue Funktionen des Börsenspiels in Anspruch nehmen und seine Profildaten verwalten.

Wenn der Spieler erneut auf das Börsenspiel zugreift kann er sich über den Login Link authentifizieren. Die Authentifizierung erfordert die Eingabe der Emailadresse und des Passwortes des Spielers und eine weitere Betätigung eines implementierten Buttons. Die Session Bean stellt eine Anfrage an die Steuerung, die dann eine SQL Anfrage nach der Emailadresse auf der Datenbank vornimmt. Ist die Emailadresse vergeben, wird das Passwort des Spielers aus der Entität erfragt und mit der eingetragenen verglichen. Stimmt das Passwort ebenfalls überein, hat der Spieler den Authentifizierungsprozess erfolgreich ausgeführt und wird je nach Berechtigung an die Administrator oder Spieler Startseite weitergeleitet durch die Session Bean.

Zusätzlich bietet das Börsenspiel die Option sich über einen bestehenden Account auf einem sozialen Netzwerk Dienst per Single Sign-On zu authentifizieren. Dazu wurden die drei sozialen Netzwerk Dienste Facebook, Google Plus sowie LinkedIn betrachtet und durch den serverseitigen OAuth Flow in das Börsenspiel integriert. Die Authentifizierungsmechanismen werden in den nachfolgenden Unterkapiteln genauer betrachtet.

4.2. Facebook OAuth Implementierung

4.2.1. Weiterleitung um Facebook OAuth Dialog

Der erste Schritt bei der Facebook Implementierung umfasst die Registrierung des Clients. Dazu werden auf einer vorgegebenen Seite auf der Facebook Platform, nach der Anmeldung mit einem gültigen Profi,l der Name des Börsenspiels sowie die Redirection URL angegeben.

Der nächste Schritt besteht darin auf der Seite des Börsenspiels eine Funktion einzubauen, um den Nutzer vom Client zur Adresse des OAuth Dialoges von Facebook zu

leiten. In diesem Fall wird auf der Startseite des Börsenspiels ein Hyperlink per HTML Referenzierung sowie ein Button implementiert, der den User nach Betätigung auf die Authentifizierungs- und Autorisierungsseite von Facebook weiterleitet. Die Implementierung einer HTML Referenzierung sowie eines Buttons sind durch den geringen Codeaufwand eine einfache Methode die Weiterleitung des Webbrowsers des Nutzers zu veranlassen.

Die HTTP GET Anfrage besteht aus der URL des Autorisierungsservers und den Schlüssel-Wert Paaren Client ID und Redirection URL sowie optional aus den Paaren scope und state. Scope wird eingesetzt um die angeforderten Daten manuell über die Basisinformationen, die immer übermittelt werden können, hinausgehen zu lassen. Zu den Basisinformationen zählen Vorname, Nachname und ID des Nutzers von Facebook . Das Paar state kann hier genutzt werden, um die Korrektheit des nach erfolgreicher Authentifizierung und Autorisierung des Nutzers erhaltenen Autorisierungscode Schlüssel-Wert Paares zu validieren. Weil für die Authentifizierung am Client die Emailadresse benötigt wird, wird im Schlüssel-Wert Paar scope der Wert ëmailëingetragen.

Die Klasse FBURLCreator implementiert die Methode getFBOAuthDialogURL(), die die notwendigen Paare im Header der HTTP GET Anfrage einfügt und die URL zurückgibt. Die Werte der einzelnen Paare werden als finale statische Stringobjekte vom Entwickler in der Klasse abgelegt.

Das Paar für die Redirection URL muss hier entgegen der Spezifikation angegeben werden. Dies kann als zusätzliche Sicherheitsmaßnahme angesehen werden. Da die Client ID aus der URL ausgelesen werden kann, könnten diese durch Änderung der Redirection URL die Weiterleitung des Nutzers auf eine andere Seite veranlassen. Während der Authentifizierung und Autorisierung würde der Nutzer dies jedoch nicht merken, wenn die Client ID übereinstimmt, sodass der Nutzer anschließend auf eine unbekannte Seite weitergeleitet werden kann. Die Angabe der Redirection URL ist eine sinnvolle zusätzliche Maßnahme, die dem Entwickler des Clients nur geringen zusätzlichen Aufwand bereitet.

Das Schlüssel-Wert Paar Antworttyp muss nicht angegeben werden, obwohl es durch die Spezifikation gefordert war. Theoretisch hätte dieses Paar den Wert "code"haben müssen, damit die Übermittlung eines Autorisierungscodes vom Client beantragt wird. Die Zuweisung, um welche Art von Client es sich handel,t übernimmt dementsprechend der Autorisierungsserver von Facebook eigenständig. Das Weglassen des Antworttyps bringt dem Entwickler des Clients an dieser Stelle nur eine geringe Zeitersparnis und das Einfügen dieses Paares wäre keine Schwierigkeit für den Entwickler, der bereits drei andere Paare angeben musste.

4.2.2. Facebook OAuth Dialog

Sofern die Weiterleitung erfolgreich war, wird der Nutzer zum OAuth Dialog von Facebook weitergeleitet. Hat der Nutzer sich bereits bei Facebook angemeldet und noch eine gültige Sitzung geöffnet, muss die Authentifizierung des Nutzers bei Facebook nicht vorgenommen werden, sondern durch die Speicherung der Daten in Cookies wird dieser Schritt übersprungen.

Anschließend muss der Nutzer das Börsenspiel einmalig autorisieren. Dazu bekommt er im OAuth Dialog eine Übersicht über die zusätzlich erfragten Daten, die über die Basisinformationen hinausgehen. Ist der Nutzer einverstanden und betätigt den bereitgestellten Button, um dies zu symbolisieren, wird er unter Zuhilfenahme der angegebenen Redirection URL, die mit der Angabe bei der Registrierung der Anwendung bei Facebook übereinstimmen muss, per HTTP Antwort zum Client zurückgeleitet.

Das die Basisinformationen im OAuth Dialog von Facebook nicht angezeigt werden ist als kritisch anzusehen, weil der Nutzer im ersten Augenblick nur davon ausgeht, dass seine Emailadresse übermittelt wird. Die Übermittlung der Emailadresse ist in den meisten Fällen unerheblich, weil es sich nicht um eine berufliche genutzte Emailadresse handeln muss, sondern meist eine private Emailadresse bei sozialen Netzwerken wie Facebook genutzt wird. Das der Name des Nutzers übermittelt wird ohne das ihm dies angezeigt wird, könnte vom Nutzer nicht gewollt sein.

4.2.3. Access Token Anfrage

Zurück auf der Seite des Börsenspiels sollte die URL zusätzlich zur Redirection URL im Header ein Schlüssel-Wert Paar Code haben. Die dargestellte Seite muss dabei keinen Inhalt anzeigen, sondern dient in erster Linie zur Ausführung eines Filters oder Servlets, um den folgenden Datenverkehr zwischen der Facebook Plattform und dem Börsenspiel zu gewährleisten.

Um den weiteren Datenverkehr zu verwalten, wurde im Börsenspiel ein HTTPServlet implementiert, der nur bei dem Aufruf der Redirection URL ausgeführt wird. Ein HTTPServlet ist eine abstrakte Klasse, der bei der Kommunikation zwischen zwei Systemen via HTTP genutzt wird. Das Interface Servlet liefert die notwendigen Methoden, um das HTTPServlet zu initialisieren und bei Bedarf auszuführen. Die vom HTTPServlet erbende Klasse FBServlet stellt die benötigten Methoden für die weitere Kommunikation bereit. Bevor das Servlet ausgeführt werden kann, muss es im Deployment Descriptor (hier web.xml) des Projektes festgelegt werden. Der Deployment Descriptor ist eine Konfigurationsdatei, die bei Aufruf einer Webseite zusätzliche Funktionen aufruft, wenn sie an der bestimmten Stelle zum Einsatz kommen sollen.

Bei Aufruf der Redirection URL kann das HTTPServlet jetzt ausgeführt werden und die

implementierte Methode der Klasse FBServlet doGet() wird durchgeführt. Als Parameter werden dieser Methode ein Instanz der HTTPServletRequest Klasse übergeben, der die Weiterleitung durch die HTTP Antwort repräsentiert, und eine Instanz der HTTPServletResponse Klasse, zu der der Nutzer nach dem Durchlauf des Servlets weitergeleitet wird. Das HTTPServletRequest liefert durch die bereitgestellte Methode getParameter() mit einem Stringobjekt mit dem Wert "codeäls Parameter den Wert für das Schlüssel-Wert Paar Code zurück. Das Paar Code entspricht dem Paar Autorisierungscode in der Spezifikation.

Das Schlüssel-Wert Paar Code wird dem Börsenspiel als einziges übertagen und wird zur Erstellung einer neuen HTTP Anfrage benötigt. Die Klasse URL bietet dafür eine mögliche Implementierung, um den Request durch die Methode openStream() auszuführen. Dafür wird der Klasse URL bei der Erstellung ein Stringobjekt übergeben, welcher die anzufragende URL angibt. Die URL enthält zusätzlich die Schlüssel-Wert Paare Code, Client ID, Client Secret und Redirection URL.

Die Methode openStream() stellt über die Klasse URLConnection eine Verbindung mit dem Facebbookserver her und liefert den Input Stream zurück, der auf die Anfrage generiert wird. Der Input Stream ist dabei die HTTP Antwort auf die Anfrage und enthält die Schlüssel-Wert Paare Access Token und Ablaufzeit.

Die Anfrage ist entgegen der Spezifikation eine HTTP GET Anfrage, sodass die Daten nicht im Körper der Anfrage mitgeliefert werden, sondern im Header. Durch die Übermittlung der Paare im Header ist das Auslesen des Client Secret aus der URL einfacher, weil die URL diese direkt enthält. Bei der HTTP POST Anfrage, wie gefordert, wären die Paare im Körper der Anfrage "versteckt". Durch die Nutzung der GET Anfrage kann das Client Secret also leichter von Dritten entdeckt werden und für ungewollte Zwecke genutzt werden. Zusammen mit dem Paar für die Client ID könnten Dritte sich beim Autorsierungsserver authentifizierne. Es schützt nur noch die Redirection URL davor, dass kein Dritter Daten an eine andere Seite verschickt.

4.2.4. Datenzugriff und Authentifizierung des Nutzers beim Client

Die Klasse future.facebookconnect.FBUserService implementiert in der Methode loginFacebook() mit einem Stringobjekten für das Access Token als Parameter, eine erneute Anfrage an den Server von Facebook, indem auf die Graph API zugegriffen wird. Die Klasse URL wird erneut genutzt, um die HTTP GET Anfrage durchzuführen. Es wird nur das Access Token benötigt für den Datenzugriff benötigt.

Der Input Stream zu einem String konvertiert kann zur Instanzierung einer JSONObject Klasse genutzt werden, aus dem die Daten des Nutzers ausgelesen werden können. Zur Konvertierung des Input Streams wird die Klasse Helper im Paket futures.helper verwendet, die in der Methode urlToString() die einzelnen Bytes der Antwort zu einem

String zusammensetzt und diesen anschließend wiedergibt.

Die Instanz der JSONObject Klasse enthält die Daten, die zu Beginn des Authentifizierungsprozesses durch das Schlüssel-Wert Paar scope übergeben wurden, sowie die Basisinformationen des Nutzers. Ebenfalls wird durch diese Klasse die Methode getString() bereitgestellt mit einem Stringobjekt als Parameter, der den Namen der erfragten Information repräsentiert, mit der die Daten einzelnd ausgelesen werden können.

Anschließend wird eine neue Instanz eines Spielers erstellt, mit den Nutzerdaten aus der JSONObject Instanz als Parameter. Mithilfe einer bedingten Abfrage wird festgestellt, ob der Player bereits in der Datenbank vorhanden ist. Die Steuerung gibt uns hier ein Modell zurück, welches entweder mit dem neuen Player übereinstimmt oder einen neuen Spieler symbolisiert. Ist der Spieler bereits registriert, wird er mit seinem bestehenden Account angemeldet. Sollte er sich nicht registriert haben, wird er in der Datenbank angelegt und anschließend angemeldet.

Der Nutzer wird nun auf die Startseite für angemeldete Nutzer geleitet, indem die Methode sendRedirect der HTTPServletResponse Klasse des FBServlets aufgerufen wird.

4.3. Google Plus Implementierung

4.3.1. Weiterleitung zum Google OAuth Dialog

Der erste Schritt bei Google ist ebenfalls die Registrierung des Clients auf einer bereitgestellten Seite von Google. Es gibt keine separate Registrierung, um den Client nur bei Google Plus zu registrieren. Auch hier werden der Name des Clients und die Redirection URL benötigt, damit die Werte für die Schlüssel Client ID und Client Secret erhalten werden können

Der erste Schritt der Implementierung beschäftigt sich wieder mit dem Erstellen eines Hyperlinks oder eines Buttons, über den der Google OAuth Dialog aufgerufen werden kann. Die beiden Varianten sind ebenfalls auf der Startseite des Börsenspiels zu finden. Durch Anklicken eines der Varianten wird die Klasse GPURLCreator aufgerufen und die getGPOAuthDialogURL() ausgeführt, die eine HTTP GET Anfrage an den Google Autorisierungsserver sendet. Die GET Anfrage muss die Schlüssel-Wert Paare Client ID, Redirection URL, Scope und response_type enthalten.

Das Scope Paar muss angegeben werden, weil bei den Google Diensten keine Informationen frei zugänglich sind. Würde der Parameter fehlen würden keine Nutzerdaten vom Client erfragt werden. Das Paar response_type muss den Wert "code"haben, wie in den Spezifikationen angegeben.

Optional kann wieder das Paar state angegeben werden, sowie die Paare approval_propmpt und access_type. Das Paar approval_prompt ist standardmäßig auf den Wert äuto"gesetzt

und erfüllt mit diesem Wert keine zusätzliche Funktion. Wird der Wert "forceängegeben, muss die Autorisierung des Clients bei jedem Authentifizierungsprozess erneut vom Nutzer durchgeführt werden. Das Paar access_type ist standardmäßig auf den Wert önline"gesetzt und führt nur bei dem Wert öfflineëine zusätzliche Funktionalität aus, der Anfrage nach einem Refresh Token.

Die Nutzung der letztgenannten Paare bietet dem Client erweiterte Funktionalitäten an, die für das Börsenspiel allerdings nicht benötigt werden. Der Autorisierungsprozess soll nicht erneut durchgeführt werden, um dem Nutzer des Clients die Authentifizierung so schnell wie möglichen durchführen zu lassen. Ebenfalls brauchen wir kein Refresh Token, weil die Daten zur Anmeldung bei jeder Authentifizierung erneut abgefragt werden und ein Zugriff auf Daten, wenn der Nutzer nicht anwesend ist, keinen Zweck erfüllt, weil nur ein read-only Zugang zum Profil auf Google Plus von Google umgesetzt ist.

Die Weiterleitung zum Google OAuth Dialog hält sich genau an die Spezifikationen und alle benötigten Paare müssen angegeben werden. Zusätzlich bietet Google noch weitere Funktionalitäten für den Client an, die genutzt werden können.

4.3.2. Google OAuth Dialog

Die Weiterleitung zum OAuth Dialog von Google wird jetzt durchgeführt. Hat der Nutzer sich bereits mit seinem Google Account bei Google authentifiziert, wird dieser Schritt wie bei Facebook übersprungen.

Die Autorisierung durch den Nutzer folgt. Dazu bekommt er im OAuth Dialog eine Übersicht über die erfragten Daten, die auch die Basisinformationen mit einbeziehen. Ist der Nutzer einverstanden und betätigt den bereitgestellten Button, um dies zu symbolisieren, wird er unter Zuhilfenahme der angegebenen Redirection URL, die mit der Angabe bei der Registrierung der Anwendung bei Google übereinstimmen muss, per HTTP Antwort zum Client geleitet.

4.3.3. Access Token Anfrage

Die Antwort des Autorisierungsservers sollte zusätzlich zur Redirection URL das Schlüssel-Wert Paar code haben. Die Seite liefert ebenfalls keinen Inhalt zurück, sondern wird zum Aufruf eines Servlets verwendet. Die Klasse GPServlet implementiert wieder die Methode doGet(), um auf die HTTP Antwort des Google Autorisierungsservers zu reagieren. Der Wert für den Schlüssel code wird durch die Methode getParameter() ausgelesen und eine HTTP POST Anfrage muss erstellt werden, um ein Access Token zu erhalten. Im Gegensatz zu Facebook muss hier eine POST Anfrage erstellt werden und keine GET Anfrage, was zur Folge hat, dass die bei Facebook verwendete URL.openStream() Methode nicht funktioniert.

Die Google Java API liefert hier bereits eine implementierte Methode zur Erstellung und Übermittlung des Request, im Börsenspiel wurde aber zusätzliche in der Klasse Helper die Methode sendPostRequest() implementiert, die eine POST Anfrage erstellt und die Antwort empfangen kann. Dazu werden der Methode zwei Stringobjekte als Parameter übergeben. Der erste Paramater beinhaltet die Adresse des Google Tokenendpunkts, an den die Anfrage gesendet werden soll. Der zweite Parameter repräsentiert den Körper der POST Anfrage. Die Anfrage Methode POST wird durch Aufruf der setMethode() Methode aus der Klasse HTTPURLConnection auf POST gesetzt. Dazu wird die URLConnection bei Aufruf der Methode zu einer HTTPURLConnection gecastet.

Im Körper der Anfrage müssen die Schlüssel-Wert Paare Client ID, Client Secret, Redirection URL, code sowie ein weiteres Paar grant_type angegeben werden. Das Paar grant_type muss beim serverseitigen Flow den Wert äuthorization_code"haben. Die Client ID und das Client Secret werden wie bei Facebook zur Authentifizierung des Clients verwendet. Auch hier hält sich Google genau an die Spezifikationen des OAuth Protokolls und es müssen alle vorgegebenen Paare angegeben werden.

Damit die POST Anfrage gesendet werden kann, muss eine Instanz der Klasse URLConnection in der Methode sendPOSTRequest() angelegt werden, welche implementierte Methoden zur Übermittlung der Anfrage und zum Empfang der Antwort bereitstellt. Während bei der Facebook Implementierung die Methode openStream() der Klasse URL verwendet wurde, die eine Instanz der Klasse URLConnection verwendet, damit die Anfrage und die Anwort empfangen werden können, muss bei der POST Anfrage diese Klasse manuell instanziert werden und die Methode openConnection() der URL Klasse zur Erzeugung eingesetzt wird. Beide Methoden geben die Antwort über die Methode getInputStream() der URLConnection zurück, wobei openStream() eine verkürzte Version darstellt.

Die Anfrage wird an Google übermittelt und das Access Token kann aus dem erhaltenen Input Stream extrahiert werden. Im Gegensatz zu Facebook kann die Antwort des Tokenendpunkts in eine Instanz der Klasse JSONObject konvertiert werden und das Access Token zusammen mit dem Ablaufdatum durch die vorgestellte Methode getString() abgerufen werden

4.3.4. Datenzugriff und Authentifizierung des Nutzers beim Client

Die Klasse future.googleplusconnect.GPUserService implementiert in der Methode loginGooglePlus() mit einem Stringobjekt als Parameter, der das Access Token repräsentiert, die notwendigen Funktionen, um das Access Token gegen die Nutzerinformationen auszutauschen. Dafür wird eine HTTP Anfrage, die das Access Token als Schlüssel-Wert Paar erhält, an Google gesendet, äquivalent zur Implementierung von Facebook. Hier handelt es sich wieder um eine HTTP GET Anfrage.

Die Antwort auf die GET Anfrage wird wieder zur Instanzierung der Klasse JSONObject verwendet. Die Instanz des JSONObject erhält die zu Beginn im Parameter scope festgelegten Daten und keine weiteren Informationen. Als letzten Schritt wird eine neue Spieler Instanz erstellt und es erfolgt eine Überprüfung, ob der Nutzer bereits mit der angegebenen Emailadresse in der Datenbank eingetragen ist oder ob ein neuer Spieler in die Datenbank übernommen wird. Der Nutzer wird nun als angemeldeter Spieler auf die Nutzerstartseite geleitet.

4.4. LinkedIn Implementierung

4.4.1. Weiterleitung zum LinkedIn OAuth Dialog

Durch den Einsatz des OAuth Protokolls in der Version 1.0 gibt es bei LinkedIn keine Wahlmöglichkeit zwischen verschiedenen Flows, sondern es kann nur der serverseitige Flow gewählt werden, der über die Spezifikation festgelegt ist.

Auch bei LinkedIn ist die Registrierung des Börsenspiels der erste Schritt, der durchgeführt werden muss. Bei der Registrierung gibt es bereits einige Unterschiede, die sich im späteren Verlauf der Implementierung bemerkbar machen werden. Die Redirection URL wird in der OAuth Spezifikation noch callback_url genannt und dient zur Weiterleitung von LinkedIn zum Client. Die callback_url muss im späteren Verlauf des Flows nicht mehr zwingend angegeben werden, wie bei den bereits betrachteten Implementierungen von Facebook und Google Plus, sondern es kann auf die angegebene Adresse der Registrierung zurückgegriffen werden, sofern der Parameter später nicht mehr angeführt ist. Der Entwickler des Clients erhält bei der Registrierung zusätzlich ein customer_key und eine customer_secret ähnlich der Client ID und dem Client Secret des OAuth 2.0 Protokolls.

Die erste benötigte Implementierung im Börsenspiel beschäftigt sich wieder mit der Erstellung eines Hyperlinks oder Buttons, der zur Weiterleitung zu LinkedIn genutzt wird. Im Gegensatz zu den anderen Implementierungen wird hier auf die callback_url per HTTP GET Anfrage weitergeleitet, damit die erste HTTP POST Anfrage verwendet werden kann, welche die Schlüssel-Wert Paare consumer_key, nonce, signature_methode, timestamp und version im Körper der Anfrage beinhalten muss und zusätzlich könnte eine weitere callback_url angegeben werden, sofern die bei der Registrierung angegebene nicht genutzt werden soll. Die bereits verwendete Methode sendPOSTRequest() der Klasse Helper wird zur Übertragung der POST Anfrage eingesetzt..

Der Parameter nonce ist gleichwertig zum state Parameter, signatur_method wird zur Verschlüsselung des signature Wertes verwendet und hat in dieser Implementierung den Wert "PLAINTEXT", wodurch kein signature Paar angegeben werden muss. Das Schlüssel-Wert Paar timestamp gibt von einem bestimmten Zeitpunkt an berechnet, die

bis zum Aufruf verstrichene Zeit wieder. Unter dem Parameter version wird die verwendete Version des OAuth Protokolls angegeben. Die genannten Paare sind in der Spezifikation der Version 1.0 als notwendig angegeben und werden auch von LinkedIn benötigt.

Die vom HTTPServlet erbende Klasse LIServlet wird eingesetzt, um die Post Anfrage senden zu können und die Antwort zu erhalten, damit der Nutzer anschließend zum OAuth Dialog von LinkedIn weitergeleitet werden kann. Die POST Anfrage wird wie bei den anderen Implementierungen in der doGet() Methode ausgeführt.

Zur Erstellung des Post Requests wird wieder die Klasse Helper verwendet, die als Parameter die genannten Paare als Stringobjekt erhält sowie die anzuwählende Adresse von LinkedIn. Die Anfrage Methode POST wird durch Aufruf der setMethode() Methode auf POST gesetzt und anschließend über das URLConnection Objekt zum LinkedIn Autorisierungsserver weitergeleitet. Als Antwort auf die POST Anfrage wird ein Schlüssel-Wert Paar oauth_token und ein Paar token_secret zurückgegeben, wobei nur das oauth_token Paar verwendet wird und der Wert des Paares zurückgegeben wird.

Das Schlüssel-Wert Paar oauth_token_secret wird verwendet, wenn das Paar signature in den weiteren Anfragen verwendet wird. Weil die signature_methode allerdings den Wert "PLAINTEXT"hat und die Verbindung zum LinkedIn Autorisierungsserver das Sicherheitsprotokoll SSL verwendet, muss dieses Paar nicht genutzt werden. Der Einsatz des Paares signature ist eine Sicherheitsmaßnahme, die ergriffen werden kann, damit eine zusätzliche Validierung der Anfrage eingestetzt werden kann. Ebenfalls muss das Paar eingesetzt werden, wenn die Verbindung nicht über SSL geschützt ist.

Im ersten Abschnitt der Implementierung von LinkedIn gibt es folglich einige Unterschiede. Der Client leitet den Webbrowser des Nutzers zurück zum Client, wo das LIServlet zum ersten mal ausgeführt wird und eine HTTP POST Anfrage versendet wird, damit zwei Schlüssel-Wert Paare erhalten werden können. Nach Erhalt der beiden Paare wird der Nutzer zum LinkedIn Autorisierungsserver weitergeleitet.

Nach Erhalt der beiden Paare kann die vom HTTPServlet implementierte Methode resp.sendRedirect() mit der Adresse des Autorisierungsserver und dem Schlüssel-Wert Paar oauth_token im Header genutzt werden, um eine HTTP GET Anfrage an den Autorisierungsserver zu schicken und den Nutzer zum LinkedIn OAuth Dialog weiterzuleiten.

4.4.2. LinkedIn OAuth Dialog

Der Webbrowser des Nutzer öffnet nun den OAuth Dialog, in dem der Nutzer sich zuerst bei LinkedIn authentifizieren muss, sofern dies noch nicht getan wurde. Anschließend folgt die Autorisierung des Clients. Durch den fehlenden Parameter scope, der beim OAuth 2.0 Protokoll die Reichweite der erfragten Daten angibt, können bei LinkedIn alle Daten erfragt werden, die in den Einstellungen des Nutzers als öffentlich festgelegt wurden. Die vom Client genutzten Daten sind für den Nutzer folglich nicht transparent.

Nach der Autorisierung des Clients durch den Nutzer, wird der Webbrowser per HTTP GET Anfrage zurück zum Client geleitet.

4.4.3. Access Token Anfrage

Der Webbrowser des Nutzers sollte nach einem erfolgreich durchgeführten OAuth Dialog zu der callback_url des Client zurückgelangen. Erneut wird das Servlet LIServlet ausgeführt und die Methode doGet() aufgerufen. Diese Methode liefert diesmal die Parameter oauth_token und oauth_verifier aus der HTTP Antwort zurück. Das oauth_token Paar entspricht dem zuvor erhaltenen. Der oauth_verifier symbolisiert bei Anfrage nach einem Access Token, das der Nutzer das Börsenspiel erfolgreich autorisiert hat.

Eine neue HTTP Post Anfrage kann mit dem neu erhaltenen Paar oauth_verifier durchgeführt werden, um ein Access Token von LinkedIn zu beantragen. Dazu wird wieder die Methode sendPOSTRequest() der Klasse Helper ausgeführt. Im Körper der POST Anfrage stehen die fünf Schlüssel-Wert Paare, die bereits bei der ersten Anfrage übergeben wurden, sowie die Paare oauth_token und der oauth_verifier. Im Header der POST Anfrage steht die URL des Tokenendpunktes. Ein Stringobjekt wird von der Methode zurückgegeben, die den Wert des erhaltenen Schlüssel-Wert Paares oauth_token repräsentiert, und als Access Token genutzt werden kann.

Bei der Implementierung des OAuth Protokolls in der Version 1.0 wird das Schlüssel-Wert Paar oauth_token für zwei verschiedene Tokens verwendet, dem Request Token im ersten Schritt der Implementierung und dem Access Token in diesem Schritt. Die doppelte Verwendung eines Paares kann hier für Verwirrung sorgen und der Entwickler des Clients muss über die Benutzung der jeweiligen Funktion des Tokens informiert sein, um das Paar gezielt einzusetzen. Die Einführung eines weiteren Paares mit einer anderen Bezeichnung wäre vorteilhaft.

Die Anfrage nach einem Access Token ist bis auf die Verwendung von mehreren Schlüssel-Wert Paaren fast identisch zur Implementierung der Google OAuth Version 2.0, allerdings wird die Antwort nicht zur Instanzierung der Klasse JSONObject verwendet.

4.4.4. Datenzugriff und Authentifizierung des Nutzers beim Client

Die Klasse future.linkedinconnect.LIUserService implementiert in der Methode loginLinkedIn() mit einem Stringobjekt als Parameter, die das Access Token repräsentiert, eine erneute Anfrage an den Server von LinkedIn, indem auf die LinkedIn API zugegriffen wird. Die Antwort auf die HTTP GET Anfrage beinhaltet je nach Wahl des field Schlüssel-Wert Paares in der URL die zurückgegebenen Daten, sofern der Nutzer sie als öffentlich akzeptiert hat. Die URL als String ausgelesen kann wie bei den anderen Implementierungen zur Instanzierung einer JSONObjekt Klasse verwendet werden und es erfolgt die bereits bei

den anderen Implementierungen vorgestellte Anmeldung des Nutzers mithilfe der Emailadresse und der LinkedIn ID des Nutzers.

Der Datenzugriff über die LinkedIn API unterscheidet sich wie die Access Token Anfrage ebenfalls hauptsächlich über die Verwendung mehrerer Schlüssel-Wert Paare. Zusätzlich wird das Paar field benutzt, welches an dieser Stelle das fehlende scope Paar bei der ersten Anfrage ersetzt.

5. Wirtschaftliche Aspekte der Nutzung von Single Sign-On Authentifizierungen bei sozialen Medien

5.1. Allgemeine Aspekte der sozialen Medien

Die Nutzung von Single Sign-On Authentifizierungen mit einem sozialen Netzwerk bieten der erstellten Anwendung eine Reihe von Vorteilen. Durch die Einbeziehung eines sozialen Netzwerkes, welches selbst Social Media Marketing betreibt, wird die erstellte Anwendung ebenfalls zu einem Social Media Marketing Teilnehmer. Unter dem Begriff Social Media Marketing wird eine Marketingstrategie verstanden, die durch Einsatz von digitalen Medien einen Bezug zu Gemeinschaften herstellt und diese aktiv in die Marktforschung mit einbezieht. [Lem11] Bei diesem Marketing geht es also darum, dass die Nutzer eines sozialen Mediums durch dessen Einsatz bewusst oder unbewusst das Marketing von Unternehmen beeinflussen und für diese zur Marktforschung genutzt werden können.

Die Nutzer beeinflussen die Marktforschung durch die kooperative Beurteilung eines Produktes eines Unternehmens. [Lem11] Die Beurteilung kann aus einer geschriebenen Nachricht bestehen, die auf einem sozialen Medium veröffentlicht wurde, sowie durch den Einsatz von Funktion wie beispielsweise dem "Like"Button von Facebook. Der Einsatz dieser beiden Funktionen kann für den Social Media Marketing Betreiber eine aktuelle Beurteilung eines Produktes liefern, die sowohl Auskunft über das Produkt und dessen geforderte Eigenschaften gibt, sowie Auskunft über Nutzergruppen, die an einem Produkt Interesse finden. Das Börsenspiel könnte diese Funkion durch die Beurteilung von Nutzern nutzen, die sich über ein soziales Netzwerk angemeldet haben, indem die Beiträge der Nutzer abgerufen werden.

Der Faktor der Aktualität ist dabei ein wesentlicher Vorteil gegenüber den Massenmedien. [Lem11]] Bei Massenmedien müssen Studien und Umfragen bei den Nutzern geführt werden, die zeitaufwendig und mit hohen Kosten verbunden sind. Bei den sozialen Medien werden Meinungen über ein Produkt über digitale Kommunikationskanäle verbreitet,

was eine vereinfachte Kommunikation darstellt. Das Unternehmen muss nicht aktiv nach neuen Studienteilnehmern suchen, sondern die Studienteilnehmer nehmen meist aus eigenem Interesse an den Bewertungen teil, weil sie ihre Meinung anderen Nutzers mitteilen wollen. Das Unternehmen muss die abgegebenen Beurteilungen dann nur noch aus einer Datenbank abrufen und auswerten. Die Aktualität ist ebenfalls ein wichtiger Faktor für das Börsenspiel, weil dieses nur zu einem bestimmten Zeitpunkt, dem Wintersemestern an der Universität Kiel, in Betrieb genommen wird. Durch die schnelle Übertragung von Nachrichten auf vor allem sozialen Netzwerken kann das Börsenspiel zeitnah für Aufmerksamkeit sorgen, bevor das Wintersemester vorbei ist oder ein Einstieg für den Nutzer für zu spät erscheint.

Ein weiterer wichtiger Aspekt ist die Reichweite der Personen, die über soziale Medien angesprochen werden, und eine daraus resultierende höhere Nutzerzahl. [Stu11] Weil soziale Medien nicht ortsgebunden sind, sondern von nahezu überall erreichbar sind, können mehr Nutzer angesprochen werden, die sonst nicht auf das Produkt eines Unternehmens aufmerksam geworden wären. Dies trifft auch auf das Börsenspiel zu. In den vergangenen Jahren wurde das Börsenspiel hauptsächlich von Studenten der Universität Kiel genutzt, weil nur dort für das Börsenspiel geworben wurde. Mit dem Einsatz von sozialen Netzwerken als zusätzliche Vertriebsplattform wird eine weitaus höhere Personenzahl angesprochen, die über "Like"Buttons oder eingetragene Anwendungen bei einem bekannten Nutzer, auf das Börsenspiel aufmerksam gemacht werden und Interesse an einem solchen Spiel zeigen könnten.

Das Ansehen eines Unternehmens kann zusätzlich gefördert werden, in dem gute Bewertungen über die Produkte abgeben wurden oder das Unternehmen durch die eigene Präsenz auf sozialen Medien den eigenen Ruf fördern kann. [Stu11] Ein besserer Ruf ist für das Unternehmen in sofern vorteilhaft, weil das Unternehmen unter den Nutzern eine höhere Wertschätzung einnimmt und gegenüber anderen Produkten bevorzugt werden kann. Ein schlechter Ruf übt sich folglich negativ auf die Nutzungsbereitschaft aus. Das Börsenspiel könnte in sofern von einem hohen Ansehen profitieren, weil Nutzer sich für dieses Produkt entscheiden könnten anstatt eines anderen. Die gebührenpflichtige Nutzung des Börsenspiels könnte Nutzer abschrecken es zu nutzen und ihr Geld anderweitig auszugeben. Bei einem hohen Ansehen könnte der Nutzer sich dennoch entscheiden sein Geld in dieses Produkt zu investieren.

5.2. Single Sign-On Aspekte

Das Single Sign-On bietet dazu noch einige weitere Vorteile. Der Nutzer einer Single Sign-On Authentifizierung muss sich nur einen Benutzernamen und ein Passwort für den Zugriff auf mehrere Anwendungen merken. Damit wird ihm einerseits der Zugriff auf

mehrere Dienste erleichtert, andererseits kann er bei dem übergeordneten System, über welches die Authentifizierung läuft, eine komplexere Kombination aus Nutzername und Passwort wählen, wodurch eine höhere Sicherheit der Anmeldung gewährleistet ist. [Cof11] Der Nutzer hat folglich sowohl eine zeitlichen Vorteil, als auch einen sicherheitstechnischen Vorteil. Beide Vorteile treffen auf das Börsenspiel zu.

Des weiteren wird durch eine Single Sign-On Authentifizierung und der Autorisierung des Nutzers auf Daten für die untergeordneten Systeme Zeit gespart, Daten eigenständig sammeln zu müssen. Berechtigte Daten können von dem übergeordneten System abgerufen werden und für weitere Zwecke genutzt werden. Dabei sind bei sozialen Medien, speziell den sozialen Netzwerken, eine große Menge von Daten verfügbar, die genutzt werden können. Das Börsenspiel nutzt diesen Vorteil nur bedingt, weil es hauptsächlich Daten abfragt, die für die Registrierung von Nöten sind und keine weiteren Daten des Nutzers abruft. Ein eigenes Authentifizierungsverfahren würde die gleichen Daten sammeln und würde folglich den selben Nutzen erfüllen.

Weil die Anmeldung nur auf einem System erfolgt und weitere Authentifizierungen über das gleiche System durchgeführt werden, ist der Nutzer eher bereit sich bei unbekannten Anwendungen zu registrieren, da er sein übliches Passwort nicht übertragen muss. Der Nutzer bekommt zusätzliches Vertrauen zu einer unbekannten Anwendung durch die Registrierung der Anwendung bei bereits bekannten Systemen, die als vertrauensvoll eingestuft werden. Das Börsenspiel profitiert vor allem durch die Steigerung des Vertrauens, weil es bei sozialen Netzwerken als Anwendung eines vielleicht bekannten Nutzers für Aufmerksamkeit sorgen kann.

Das Single Sign-On hat allerdings auch Nachteile. Sollte das Passwort eines Nutzers durch andere Anwender gestohlen werden, ist der Zugriff auf mehrere Anwendungen möglich. Wenn ein Nutzer sich also auf einem System registriert und dieses für weitere Authentifizierungen nutzt, kann das gestohlene Passwort für mehrere Anwendungen genutzt werden auf denen ungewollte Aktionen durchgeführt werden könnten. Für das Börsenspiel stellt dies eine Bedrohung dar, weil Passwörter für soziale Netzwerke nicht unbedingt komplex sind. Sollte ein Passwort bekannt geworden sein, können Aktivitäten des Nutzers am Börsenspiel erheblich gestört werden, was durch die gebührenpflichtige Nutzung des Börsenspiels einen zusätzlichen Nachteil bildet, da Kosten für die Nutzung angefallen sind.

Ebenfalls bringt es einer Anwendung, die ausschließlich über ein Single Sign-On System authentifiziert wird, nichts, wenn Nutzer das Single Sign-On System nicht nutzen. Handelt es sich um eine ausschließliche Nutzung und es ist kein eigenes Authentifizierungsverfahren auf der Anwendung implementiert, hat der Nutzer nur die Möglichkeit sich erst bei dem Single Sign-On System zu registireren, bevor er die eigentlich gewollte Anwendung nutzen kann. Das Börsenspiel wird von diesem Nachteil nicht beeinflusst, weil es über einen separaten Authentifizierungsprozess verfügt.

6. Fazit

Der Einsatz des OAuth Protokolls bietet eine gute Möglichkeit einen zusätzlichen Authentifizierungsmechanismus in eine Webanwendung einzubauen. Durch die hohen Nutzerzahlen von sozialen Netzwerken kann durch die Verknüpfung der Webanwendung eine größere Anzahl von Personen erreicht werden, die die Webanwendung sonst wahrscheinlich nicht ohne weitere Werbemaßnahmen gefunden hätten. Soziale Netzwerke sind daher ein guter Anlaufpunkt, wenn für eine Webanwendung weitere Nutzer gesucht werden. Ebenfalls ist der Authentifizierungs- und Autorisierungsprozess über das OAuth Protokoll für den Nutzer einfach und schnell zu vollziehen ohne das er erneut seine Nutzerdaten in einem weiteren Anmeldeformular eintragen muss.

Für die Einbindung eines OAuth Protokolls bieten sich sowohl die Version 1.0 als auch die Version 2.0 an. Wenn der Entwickler einer Webanwendung es bevorzugt die komplette Implementierung seines Codes selbst zu erstellen bietet sich eher die Version 2.0 an, weil sich die Implementierung dieses Ablaufs als wesentlich einfacher und überschaubarer erweist. Der Nutzer wird zum OAuth Dialog geleitet, ein Code wird zurückgegeben und in einem weiteren Schritt kann ein Access Token erfragt werden mit dem auch schon die Daten des Nutzers erfasst werden können.

Die Version 1.0 hingegen erfordert bei der eigenständigen Implementierung mehr Zeit, weil mehrere Schritte von Nöten sind und eine größere Anzahl von Schlüssel-Wert Paaren zum Einsatz kommen müssen. Viele der Schlüssel-Wert Paare sind dabei nicht unbedingt notwendig, wie zum Beispiel das nonce Paar oder das timestamp Paar. Während bei der Version 2.0 das Paar state nur empfohlen wird anzugeben und bei den beiden betrachteten Implementierungen von Facebook und Google auch nicht notwendig ist, wird es in der Version 1.0 noch als Musskriterium angesehen. Damit dieses Paar angegeben werden kann muss erst einmal ein Algorithmus entworfen werden, der einen Wert für dieses Paar erstellen kann, welcher so komplex sein sollte, dass er nicht zweimal den gleichen Wert wiedergibt. Ein bereits verwendetet Wert wird vom Autorisierungsserver nicht akzeptiert, was wiederum zu ungewollten Fehlern während des Einsatzes des OAuth Protokolls führen kann.

Die Erstellung einer eigenen Signatur in der Version 1.0 stellt eines der größten Herausforderungen dar, sodass entweder gute Kenntnisse mit dem Umgang von Signaturen und Signaturmethoden gefragt sind oder der Entwickler einen hohen Zeitaufwand betreiben

muss, bis er die Signaturen richtig erstellen kann. Der Einsatz der Signatur im Evaluationsbeispiel wäre sicher ein interessanter Zusatz gewesen, allerdings ist der zu betreibende Zeitaufwand dafür ohne ausreichende Kenntnisse zu hoch.

Ebenfalls sind die Daten die vom Nutzer an den Client übergeben werden bei der OAuth Version 2.0 sehr transparent, sodass der Nutzer genau weiss, welche Daten er an den Client übersendet. Bei der Version 1.0 ist diese Transparenz nicht gegeben, weil die Daten nicht angezeigt werden, die der Client abruft. Bei Facebook ist die Transparenz des Datenflusses nicht ganz so genau, allerdings hängt das damit zusammen das generelle Informationen über den Nutzer von vornherein zugänglich sind ohne eine Autorisierung durch den Nutzer. Das hängt allerdings mit der Einstellung des Unternehmens zusammen, die an sich für eine freien Zugang von Daten im Internet sind und die Privatsphäre nicht so hoch angesehen ist.

Der Prozess der Single Sign-On Authentifizierung an sich ist eine schnelle und bequeme Lösung für den Nutzer seine bereits verwendeten Daten an andere Anwendungen zu übertragen, wodurch am Ende der Entwickler den größten Vorteil gewinnt, auch wenn er mehr Zeit für die Implementierung benötigt. Dadurch das Nutzer sich über vertraute Anwendungen mit einer weiteren Anwendung authentifizieren können, ist die Hemmschwelle einer weiteren Registrierung bei einer weniger bekannten Anwendung deutlich geringen als bei Anwendungen wo Daten direkt angegeben werden müssen.

Besonders lohnenswert ist das Single Sign-On bei sozialen Netzwerken für Unternehmen, die eine große Datenmenge verwalten bzw. auswerten wollen. Durch die vielen Möglichkeiten Daten abzufragen und die Fülle an verfügbaren Daten bei sozialen Netzwerken sind diese sicher die richtige Anlaufstelle für weitere Datenverarbeitungen. Für kleine Anwendungen wie das Börsenspiel ist nur die Verbreitung des Börsenspiels und der Gewinn neuer Nutzer vorteilhaft. Die Implementierung eines eigenen Single Sign-On Mechanismus, sodass Nutzer sich über das Börsenspiel authentifizieren können, ist ebenfalls nur bedingt vorteilhaft, weil keine ausreichenden Nutzerzahlen vorhanden sind, um den zeitlichen Aufwand der Implementierung zu rechtfertigen. Die Einführung der Single Sign-On Authentifizierung hingegen ist für kleine Anwendungen mit wenigen Nutzern bei Systemen mit hohen Nutzerzahlen, über die die Authentifizierung laufen kann, eher als vorteilhaft anzusehen.

Das Evaluationsbeispiel ist in der derzeitigen Form nicht einsatzfähig. Es verfügt zwar über mehrere Authentifizierungsverfahren und stellt bereits erste Funktionen bereit, aber die Funktionen sind dennoch sehr beschränkt. Die Implementierung der Authentifizierungsverfahren ist ebenfalls noch sehr fehleranfällig, weil auf einzelne Fehler des Authentifizierungsprozesses im jetzigen Stand nur bedingt eingangen wird. Vorallem das Abfangen von Fehlermeldungen muss optimiert werden, um einen reibungslosen Ablauf der Authentifizierungen zu gewährleisten.

6.1. Aublick

Die Single Sign-On Authentifizierung ist noch eine junge Form der Authentifizierung von Nutzern, die noch nicht weit verbreitet ist. Dies wird sich in den nächsten Jahren jedoch vermehrt ändern. OAuth als Form des Single Sign-On findet mehr Akzeptanz unter den Vertreibern von Webanwendungen und wird in den nächsten Jahren sicherlich weitere Anhänger finden, die das Protokoll für eine Authentifizierung verwenden. OAuth in der Version 2.0 ist seit 2010 verfügbar und hat bereits jetzt viele OAuth 1.0 Implementierungen abgelöst.

Allerdings wird die Version 1.0 wohl noch einige Zeit in Betrieb bleiben, wie man am Beispiel von Google Plus sehen kann. Obwohl es letztes Jahr auf den Markt gekommen ist, sind beide Versionen von OAuth vertreten, was hier sicherlich auch mit der Bereitstellung weiterer Google Dienste, auf die schon länger zugegriffen werden kann, zusammenhängt. LinkedIn hat bereits einen clientseitigen Ablauf auf Basis des OAuth 2.0 Version implementiert und hat vor ihren serverseitigen Ablauf im Laufe der nächsten Jahre ebenfalls anzupassen.

Xing, ein weiteres soziales Netzwerk für die berufliche Orientierung ähnlich zu LinkedIn, hat begonnen eine erste OAuth API in der Betaphase für eine begrenzte Anzahl von Nutzern zu öffnen, die die API testen können. Hier wird es in in naher Zukunft eine Möglichkeit geben sich über den OAuth 2.0 Ablauf mit dem System zu verbinden und Xing als Authentifizierungssystem zu benutzen.

Eine weitere Single Sign-On Authentifizieren und Autorisierung wird derweilen von OpenID entworfen. Mit dieser soll es möglich sein über OpenID Provider wie Google Accounts ein Authentifizierungsverfahren für eine sichere Anmeldung bei Webanwendungen zu gewährleisten. In einer erhältlichen frühen Version, die bereits über die Google Dienste genutzt werden kann, verläuft der OpenID Authentifizierungsprozess noch zum Teil ähnlich zum OAuth Protokoll. Dabei wird die Autorisierung durch den Nutzer und die spätere Authentifizierung der Anwendung und der Erhalt von Daten nahezu identisch gehandhabt. Nur die Authentifizierung des Nutzer unterscheidet sich, durch die Authentifizierung über einen OpenID Provider und nicht über einen festen Provider wie beispielsweise Facebook. Der Grund dafür liegt in der Wahlmöglichkeit eines der OpenID Provider. Weil es mehrere Provider gibt, muss zu Beginn des Authentifizierungsverfahrens ein Provider gewählt werden, über den die Authentifizierung statt finden soll.

Literaturverzeichnis

[Bal05] Helmut Balzert. *Grundlagen der Informatik-Konzepte und Notationen in UML 2, Java 5, C und C++, Algorithmik und Software-Technik, Anwendungen.* Number 978-3-8274-1410-6. Spektrum Akademischer Verlag, Heidelberg, 2 edition, 2005.

[Cof11] David Coffin. Single sign-on. In *Expert Oracle and Java Security*, pages 149–176. Apress, 2011. 10.1007/978-1-4302-3832-48.

[DR12] Dick Hardt David Recordon. *The OAuth 2.0 Authorization Protocol draft-ietf-oauth-v2-23.* Internet Engineering Task Force, 23 edition, 2012.

[Fac] Facebook developers. http://developers.facebook.com/docs/.

[Goo] Google developers. https://developers.google.com/accounts/docs/GettingStarted?hl=de.

[Hae10] Andreas M. Kaplan & Michael Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business Horizon*, 53(1):59–68, 2010.

[Ham10] Eran Hammer. http://hueniverse.com/2010/05/introducing-oauth-2-0/, 2010.

[KCL10] Detlef Schoder Kenneth C. Laudon, Jane P. Laudon. *Wirtschaftsinformatik - Eine Einfuehrung.* Number 978-3827373489. Pearson Studium, Muenchen, 2 edition, 2010.

[Lem11] Gerald Lembke. *Social Media Marketing.* Number 978-3-589-23908-5. Cornelsen Verlag Scriptor, Berlin, 1 edition, 2011.

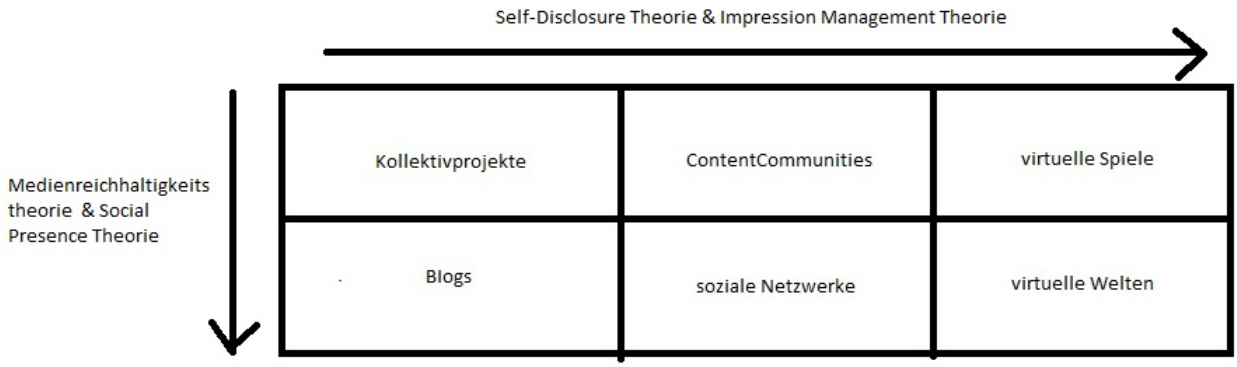
[Lin] Linkedin developers. http://developer.linkedin.com/documents/authentication.

[Stu11] Reto Stuber. *Erfolgreiches Social Media Marketing mit Facebook, Twitter, Xing & Co.* Number 978-3-8158-3063-5. Data Becker, Duesseldorf, 4 edition, 2011.

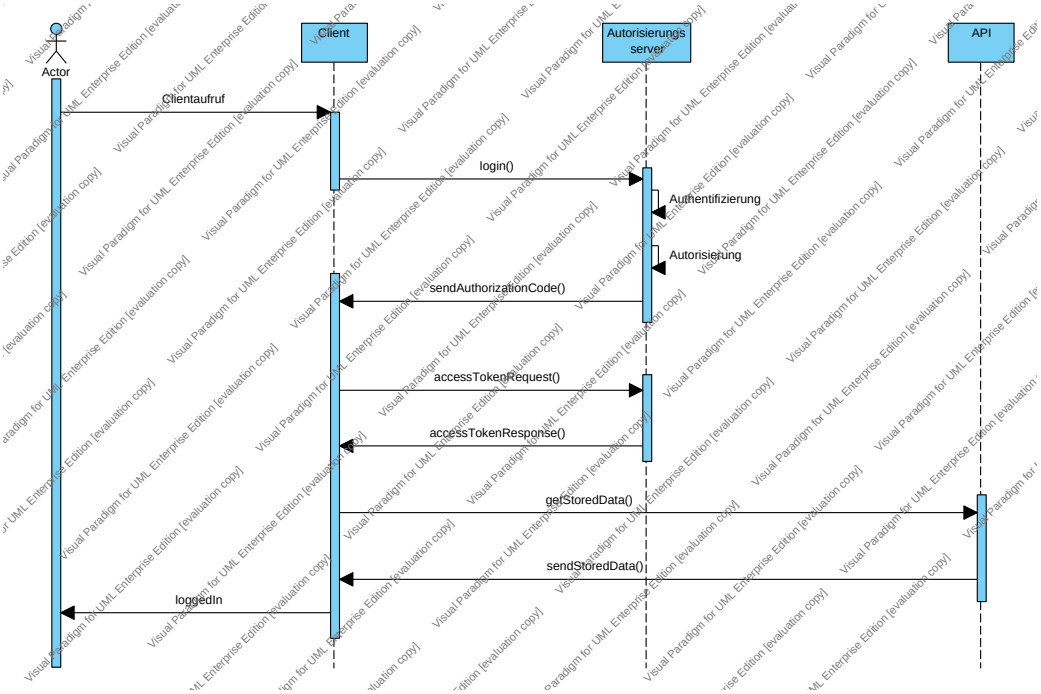
[UR03] Paul Levi Ulrich Rembold. *Einfuehrung in die Informatik fuer Naturwissenschaftler und Ingenieure.* Number 9783446219328. Hanser Fachbuchverlag, 4 edition, 2003.

A Anhang

A. Kategorisierung von sozialen Medien

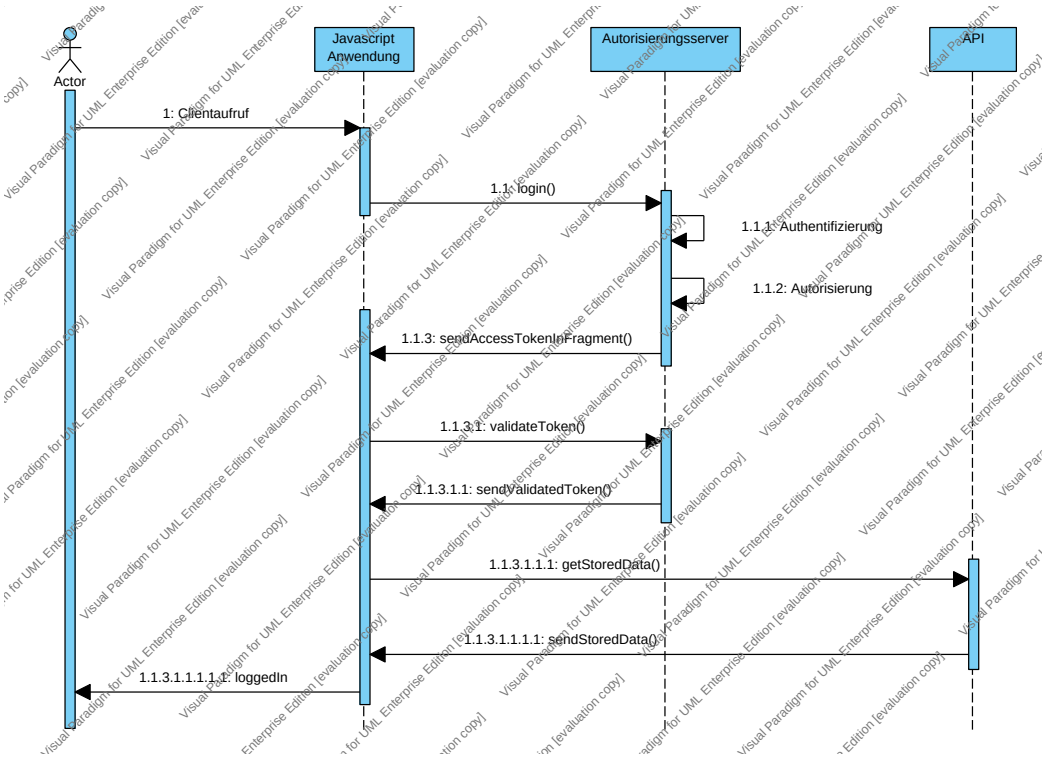


B. Serverseitiger Flow



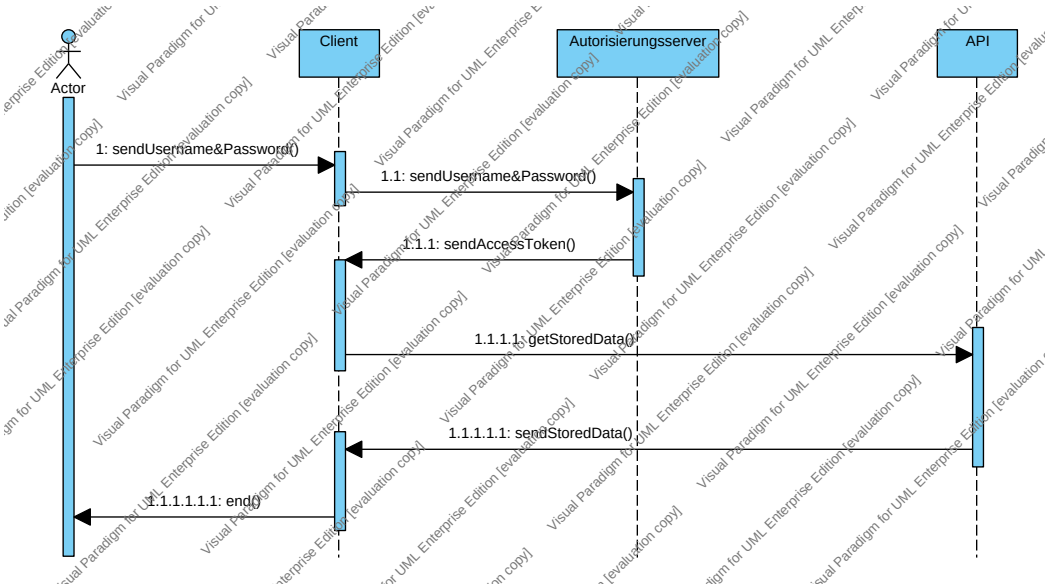
Flow.pdf

C. Clientseitiger Flow



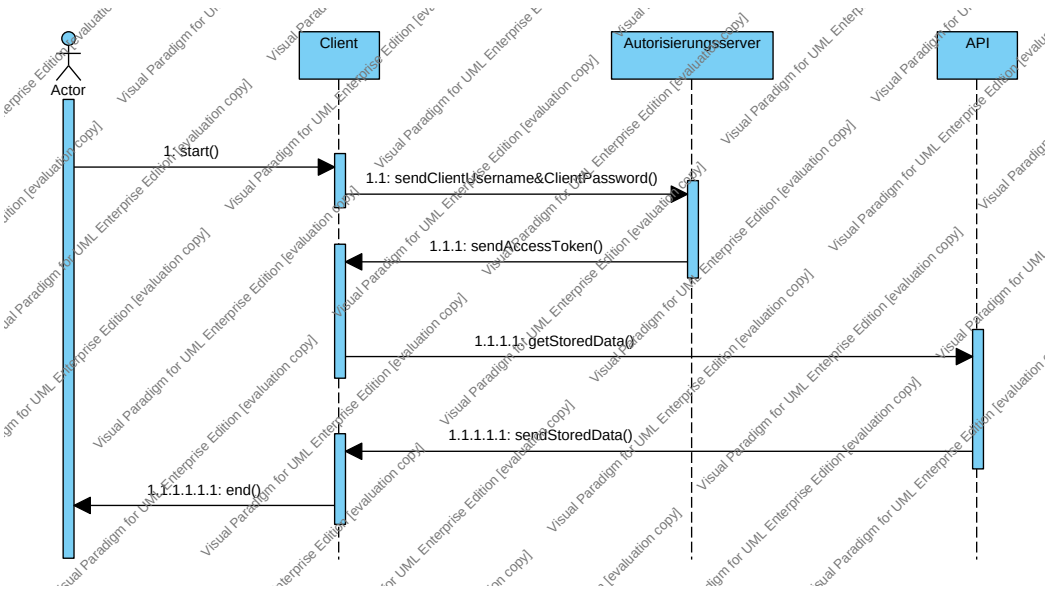
Flow.pdf

D. ResourceOwnerPasswordCredentialsGrant



Grant.pdf

E. ClientCredentialsGrant



Grant.pdf

B Glossar

Medienreichhaltigkeitstheorie:

Die Medienreichhaltigkeitstheorie ist eine Kommunikationstheorie die das Verhältnis zwischen den kommunizierten Inhalten und dem Kommunikationsmedium als proportional angibt. Je vielseitiger der Kommunikationsinhalt ist, desto reichhaltiger muss das Medium nach dieser Theorie sein.

Impression Manegement Theorie:

Die Impression Management Theorie besagt, dass Personen versuchen den Eindruck, den andere Personen über sie haben, zu kontrollieren und bewusst zu steuern.

Open ID:

Open ID ist ein dezentrales Authentifizierungssystem, das Nutzern erlaubt sich bei webbasierten Anwendungen über ein bei einem Open ID Provider angelegtes Profil zu authentifizieren.

Self-Disclosure Theorie:

Die Self-Disclosure Theorie besagt, dass der Eindruck über eine Person sowohl bewusst als auch unbewusst durch dessen Auftreten beeinflusst wird.

Self-Presence Theorie: Die Self-Presence Theorie besagt, dass ein Kommunikationsmedium am effektivsten ist, wenn das Medium über die notwendige soziale Präsenz verfügt, die für die Kommunikation zwischen Personen benötigt wird.

C Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Die eingereichte schriftliche Fassung der Arbeit entspricht der auf dem elektronischen Speichermedium.

Weiterhin versichere ich, dass diese Arbeit noch nicht als Abschlussarbeit an anderer Stelle vorgelegen hat.

Datum, Unterschrift