Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering

# Bachelorarbeit (Informatikingenieurwesen)

## Individuell Konfigurierbarer Authentifizierungs-service für Votings und Wettbewerbe

| | |
|---|---|
| **Autor** | Christian Bachmann |
| **Betreuung** | Jaime Oberle |
| **Auftraggeber** | inaffect AG |
| **Datum** | 23.12.2015 |

# Inhaltsverzeichnis

# Teil I.

# Einleitung und Abgrenzung

# 1. Einführung

## 1.1. Motivation und Fragestellung

Der Zugriff auf Services und Medien mittels mobiler Geräte steigt beständig an. So ist im Mai 2014, 60% der Zeit, die online verbracht wird, über Handy und Tablet zugegriffen worden - davon 51% mittels mobiler Applikationen. (Lipsman 2014)

# 2. Recherche

## 2.1. Fachbegriffe

Eine ausführliche Erklärung der Fachbegriffe befindet sich im Anhang unter dem Kapitel "Glossar".

## 2.2. Erläuterung der Grundlagen

In diesem Kapitel werden Funktionsweisen und Grundlage ausgeführt, die als für die Bearbeitung dieser Bachelorthesis herangezogen wurden.

### 2.2.1. Authentifizierung

Authentifizierung - beglaubigen, die Echtheit von etwas bezeugen verfolgt das Ziel (*Duden* 2014)

Eine Person oder Objekt eindeutig zu **authentifizieren** bedeute zu ermitteln ob die oder derjenige auch der ist als welcher er sich ausgibt. (Rouse 2015) Dies unterstreicht auch die Ableitung des Wortes vom Englischen Verb *authenicate*, was auf Deutsch sich als *echt erweisen, sich verbürgen, glaubwürdig sein* bedeutet. Das bekannteste Verfahren der Authenfizierung ist die Eingabe von Benutzernamen und Passwort. Weiter ist die PIN-Eingabe bei Bankautomaten oder Mobiletelefon häufig verbreitet. Die Möglichkeiten der Authentifizierung nahe zu grenzenlos. ("Http://authentifizierung.org" 2015)

### 2.2.2. Autorisierung

Autorisierung - Befugnis, Berechtigung, Erlaubnis, Genehmigung (*Duden* 2014)

Wenn die AuthenfizierungAuthentifizierung erfolgreich war erteilt das System die Autorisierung. Dabei wird der Person oder Objekt erlaubt bestimmte Aktionen/Zugriffe durchzuführen. Meist verfügen unterschiedliche Benutzer eines Systems über verschiedene Zugriffsrechte. Die korrekte Zuweisung der individuellen Rechte ist ebenfalls Bestandteil der Autorisierung.

Der Begriff Authentifizierung wird vielfach mit dem Begriff Autorisierung verwechselt. Die Authentifizierung wird vom Benutzer initiiert. Sie dient dem Nachweis, zur Ausübung bestimmter Rechte befugt zu sein. Die anschließende Autorisierung erfolgt automatisch durch das System selbst. Im Zuge der Autorisierung werden dem Benutzer seine Zugriffsrechte zugewiesen. ("Http://authentifizierung.org" 2015)

### 2.2.3. Captcha

Captcha - Test, mit dem festgestellt werden kann, ob sich ein Mensch oder ein Computer eines Programms bedient (*Duden* 2014)

Im Jahre 2000 wurde das Captcha an der Carnegie Mellon University erfunden. Captcha steht für **C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part. Luis von Ahn, Professor der Entwickler-Gruppe, erklärte die Dringlichkeit von Captcha damals so: "Anybody can write a program to sign up for millions of accounts, and the idea was to prevent that". (Burling 2012)

### Captcha Zahlen

In 2014 wurden 200 Million Captchas pro Tag eingegeben. Dabei braucht ein User durchschnittlich 10 Sekunden das entspricht 500'000 Stunden.[1]

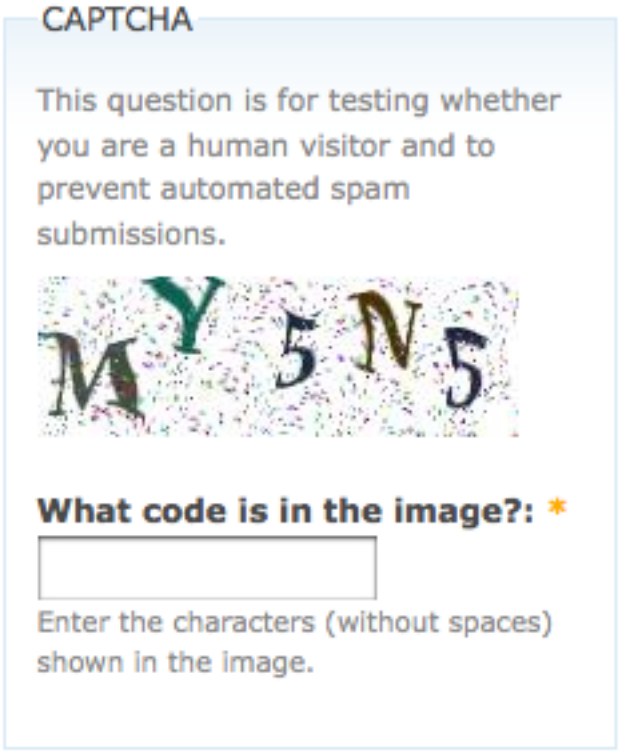


Abbildung 2.1.: Beispiele von Captchas *Quelle:drupal.org*

[1] Die statistischen Daten wurden von Google 2014 in ihrem Blog publiziert ("ReCAPTCHA Digitization Accuracy" 2014)

### 2.2.4. OAuth

OAuth ist ein Protokoll. Es erlaubt sichere API-Autorisierungen.

#### Das Bedürfnis nach OAuth

2006 implementiere Blaine Cook OpenID für Twitter und Ma.gnolia erhielt ein Dashboard welches sich durch OpenID autorisieren lies. Deshalb sich die Entwickler von Ma.gnolia und Blaine Cook um eine Möglichkeit zu finden OpenID auch für die Verwendung von APIs zu gebrauchen. Sie diskutierten ihre Implementierungen und stellten fest das es keinen offenen Standart für API-Access Delegation gab. So fingen sie an den Standart zu entwickeln. 2007 entstand daraus eine Google Group. Am 3. October 2007 war dann der OAuth Core 1.0 bereits released worden.

#### Funktionalität von OAuth

Ein Programm/API (Consumer) stellt über das OAuth-Protokoll einem Endbenutzer(User) Zugriff (Autorisierung) auf seine Daten/Funktionalitäten zur Verfügung. Dieser Zugriff wird von einem anderen Programm (Service) gemanagt. Das Konzept ist nicht generell neu. OAuth ist ähnlich zu Google AuthSub, aol OpenAuth, Yahoo BBAuth, Upcoming api, Flickr api, Amazon Web Services api. OAuth studierte die existierenden Protokolle und standardisiert und kombinierte die existierende industriellen Protokolle. Der wichtigste Unterschied zu den existierenden Protokollen ist, das OAuth sowohl offen ist und anderseits zu einem Standard geworden ist. Jeden Tag entstehen neue Webseite welche neue Funktionalitäten und Services offeriert und dabei Funktionalitäten von anderen Webseiten braucht. OAuth stellt dem Programmierer einerseits eine standartisierte Implementierung zur Verfügung. Der Endbenutzer erhält dank dieses Protokoll die Möglichkeit teile seiner Funktionalität/Daten bei einem anderen Anbieter zur Verfügung zu stellen. So kann der Endbenutzer bei der Facebook OAuth z.b. seine Posts zur Verfügung stellen nicht aber seine Freunde bekannt geben.

Dank der weiten Verbreitung gibt es nun in allen bekannten Programmiersprachen eine Implementierung sowohl von Client wie auch vom Server. Weitere Infos dazu unter oauth.net1

## 2.3. Ähnliche Produkte auf dem Markt

Dieses Unterkapitel erläutert existierenden Produkte auf dem Markt.

### 2.3.1. OAuth-Provider

Die grössten OAuth-Provider wie Google, Facebook und Twitter erziehlen eine weiter Verbreitung weltweit:

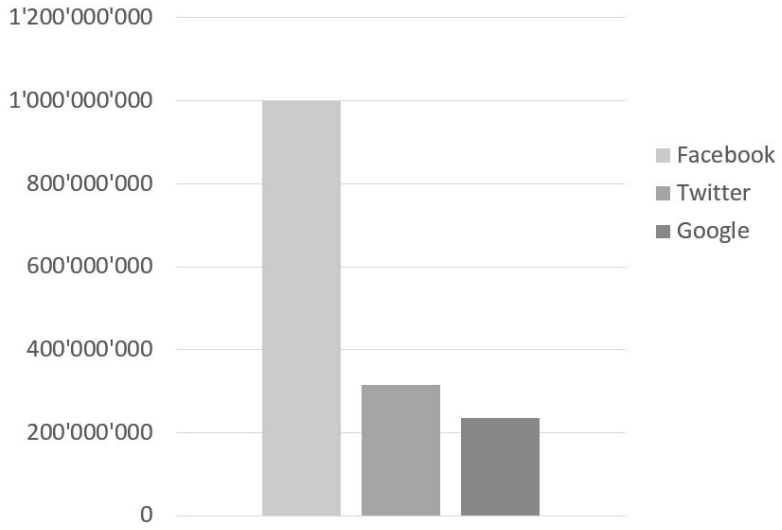


Abbildung 2.2.: Aktive Nutzer Weltweit[2]

Ganze *78%* (Interactive 2015) der Schweizer Bevölkerung nutzten SocialMedia und besitzen dadurch einen OAuth-Account:

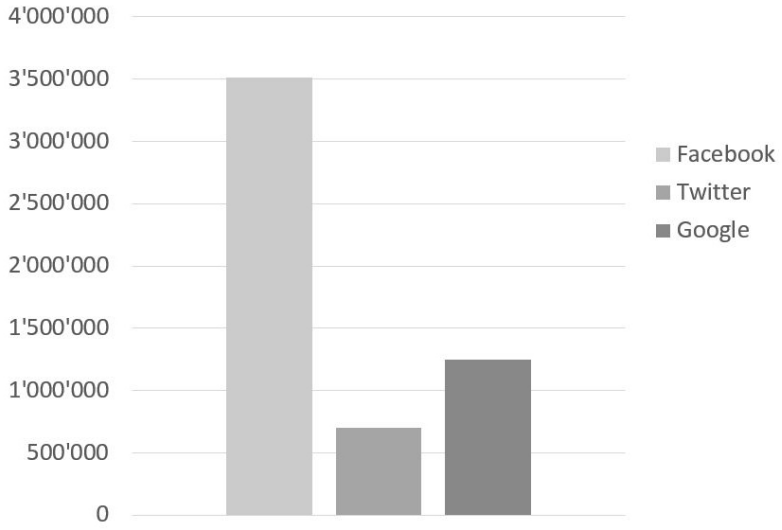


Abbildung 2.3.: Anzahl Schweizer Nutzer[3]

[2]Das Statistik wurde basierend auf den Daten von socialmedia-institute ("SMI (SocialMedia Institute)" 2015)erstellt. Facebook und Twitter Daten sind am 5. November 2015 und die Google Daten sind im 2014 erhoben worden.

[3]Das Statistik wurde basierend auf den Daten von Goldbach Interactive (Interactive 2015) generiert. Die Daten sind im März 2015 erhoben.

#### Vorteile

78% der Schweizer Bevölkerung besitzt bereits einen OAuth Account. Das Protokoll ist ein etablierter Standard.

#### Nachteile

Mehrfachregistrierungen sind möglich. Jenach OAuth-Provider werden verschiedene Daten zur Verfügung gestellt. Pro OAuth Provider kann man sich registrieren einen Abgleich der verschiedenen OAuth Provider wird vom OAuth-Protokoll nicht zur Verfügung gestellt. 22% der Bevölkerung müsste sich vor Nutzung noch registrieren. Die Implementierung ist trotz vielen Libaries nicht ohne tiefere Programmierkenntnisse möglich.

### 2.3.2. playbuzz.com

Youtube von Google ist im Jahr 2015 mit Abstand die meist verbreiteste ("Statistik Plattform" 2015) Videopublishing-Plattform. Medienhäuser nutzen Youtube um einfach Ihren Artikel mit einem Video zu ergänzen. Neben der einfachen Integration profitieren die Medienhäuser von der zusätzlichen Verbreitung über youtube.com und der einfachen viralen Verbreitungsmöglichkeiten von youtube. PlayBuzz möchte das Youtube für Votings, Quiz und ähnlicher Embeded Content zu werden. Neben MTV, Focus, Time oder Bild verwendet seit Herbst 2015 auch ein grosses Medienhaus der Schweiz die Plattform. Tamedia erfasst neu auf 20minuten Votings und Umfragen mit PlayBuzz.

2012 wurde Playbuzz von Shaul Olmert (Sohn des Premie Minster von Israel Ehuad Olmert) und Tom Pachys ins Leben gerufen. Der offizielle Launch war im Dezember 2013. Im Juni 2014 wurde Playbuzz bereits das 1. Mal unter den Top 10 Facebook Shared Publishers aufgelistet. Im Juni 2014 konnte Playbuzz bereits 70 millionen unique views aufweisen. Im September 2014 kamen 7 von den 10 Top Shares auf Facebook laut forbes.com von Playbuzz. Playbuzz setzt nach eigenen Angaben auf Content wie Votes und Quizes welcher gerne Viral geteilt wird und ermöglicht Endnutzer und Redaketeueren einfache Verwendung. ("Interview Mit Shaul Olmert" 2015) ("PlayBuzz" 2015)

#### Vorteile

Playbuzz ist kostenlos und lässt sich einfach integrieren. Durch Verwendung von Playbuzz kann die Verbreitung des eigenen Inhalts stark gesteigert werden. Die Verwaltungsoberfläche und Reports sind übersichtlich und einfach zu bedienen.

#### Nachteile

Der Verweis auf Playbuzz ist ersichtlich. Auch beim Posten auf den SocialMedia-Kanälen ist die Herkunft von Playbuzz offensichtlich. Die Möglichkeiten in Funktionalität und Design haben Grenzen. Individuelle Erweiterungen sind nicht einfach möglich.

## 2.4. Grundlegende Sicherheitsprinzipien

In diesem Unterkapitel werden die Grundlagen der Sicherheitsprinzipien vermittelt auf denen danach eine Authentifizierungssoftware aufgebaut werden kann.

### 2.4.1. KISS

**K**eep **I**t **S**tupid *and* **S**imple

Ein verbreitetes Problem unter Software Engineers und Programmier heute ist, dass sie dazu tendieren Probleme zu kompliziert und verschachtelt zu lösen. 8-9 von 10 Entwickeln machen den Fehler, Probleme zu wenig auseinander zu brechen und alles in einem grossen Programm zu lösen. Anstatt es in kleinen Paketen verständlich zu programmieren.(Hanik 2015)

Software Entwickler profitieren von Kiss:

- You will be able to solve more problems, faster.
- You will be able to produce code to solve complex problems in fewer lines of code
- You will be able to produce higher quality code
- You will be able to build larger systems, easier to maintain
- You're code base will be more flexible, easier to extend, modify or refactor when new requirements arrive
- You will be able to achieve more than you ever imagined
- You will be able to work in large development groups and large projects since all the code is stupid simple

#### KISS fördert die Sicherheit

Die Begründung warum KISS die Sicherheit fördert liefer Saltzer und Schroeder: Ungewollte Zugriffspfade können nur durch zeilenweise Codeinspektion entdeckt werden und die wiederum setzt voraus, dass Designs einfach und klein sein sind.Designs müssen so beschaffen sein, dass sie abgeschlossene Bereiche enthalten, über die konkrete und sichere Aussagen über Zugriffsmöglichkeiten und Effekte getroffen werden können. (Kriha and Schmitz 2009, 93)

### 2.4.2. Default-is-deny

Ob eine Person oder Programm Zugriff auf Daten/Funktionen haben, sollte nicht durch Verbote sondern durch explizite Erlaubnis geregelt werden. Dies bedeutet solange keine explizite Erlaubnis gesetzt ist, kann das Programm oder die Person nicht auf die Daten oder Funktionen zugreifen. You *deny* it. So simpel und logisch diese Idee klingt, umso verwunderlich ist wie viele Organisationen und Entwicklungsfirma nicht dieses vorgehen verwenden. z.B. Filesysteme setzten auf Verbote anstatt auf explizite Erlaubnise. (Rothman 2015) , (Kriha and Schmitz 2009, 94)

### 2.4.3. Open Design

Abgeleitet von der Kryprotografie: Nicht das Design der Software sollte die Sicherheit sein, sondern der verwendete Schlüssel. Dieses Konzept gilt es in der Softwareentwicklung und Systemtechnik nur bedingt einzuhalten. Die Software soll nach dem Ansatz entworfen werden. Mindestens intern soll das Software-Design durch einen Design-Review Prozess analysiert werden. In manchen Fällen macht es jedoch das Softwaredesign geheimzuhalten um einem Angreifer nicht zu viele Informationen zur Verfügung zu stellen. (Kriha and Schmitz 2009, 95)

### 2.4.4. Zusammenfassung der Sicherheitsprinzipien

Die wichtigsten Sicherheitsprinzipien zusammengefasst: * Software muss aus kleinen, isolierten Einheiten aufgebaut werden, deren externe Beziehungen am Interface deutlich werden. Damit wird sowohl praktische Schadensreduzierung durch Isolation als auch eine schnelle und einfache Sicherheitsanalyse möglich. * Zugriffsentscheidungen dürfen nur auf der Basis expliziter, minimaler und keinesfalls durch immer und global verfügbare Permissions fallen. * Das Softwaredesign von Applikationen sollte wenn möglich öffentlich sein. Zumindest sollte ein interner Review-Prozess stattfinden, in dessen Verlauf eine Sicherheitsanalyse durch nicht an der Entwicklung Beteiligte erstellt wird.

# 3. Glossar

**ORM** ORM steht für object-relational mapping und ist eine Technik mit der Objekte einer Anwendung in einem relationalen Datenbanksystem abgelegt werden kann.
**Node** Node oder Node.js ist eine Plattform welche es erlaubt JavaScript serverseitig auszuführen. `https://nodejs.org/`
**RPC** Remote Procedure Call ist eine Technologie um Funktionsbausteine in einem anderen Prozess aufzurufen.
**Jasmine** Jasmine ist ein verhaltensbasiertes Testframework für JavaScript. `http://jasmine.github.io`
**Karma** Karma ist ein Testrunner-Framework zur kontinuierlichen Ausführung von UnitTests. `http://karma-runner.github.io`
**Mocks** Mocks sind Code-Attrappen, die es ermöglichen, noch nicht vorhandene oder nicht verfügbare, Funktionalitäten und Objekte zu simulieren. `http://de.wikipedia.org/wiki/Mock-Objekt`
**Github** Github ist ein Cloud basierter SourceCode Verwaltungsdienst für Git. `https://github.com`

# 4. Verzeichnisse

Neues Verzeichnisse

## 4.1. Abbildungsverzeichnis

## 4.2. Quellenverzeichnis

## 4.3. Tabellenverzeichnis


Burling, Stacey. 2012. "CAPTCHA: The Story Behind Those Squiggly Computer Letters." http://phys.org/news/2012-06-captcha-story-squiggly-letters.html.

*Duden*. 2014. Vol. 26. Dudenredaktion.

Hanik, Filip. 2015. "Kiss." https://people.apache.org/~fhanik/kiss.html.

"Http://authentifizierung.org." 2015. http://authentifizierung.org/.

Interactive, Goldbach. 2015. "Nutzerzahlen Der Wichtigsten Plattformen." https://twitter.com/revogt/.

"Interview Mit Shaul Olmert." 2015. https://www.youtube.com/watch?v=X_fQ1uG9rFY.

Kriha, Walter, and Roland Schmitz. 2009. *Sichere Systeme*. Xpert.press. Berlin, Heidelberg: Springer Berlin Heidelberg.

Lipsman, Andrew. 2014. "Major Mobile Milestones in May: Apps Now Drive Half of All Time Spent on Digital." http://www.comscore.com/Insights/Blog/Major-Mobile-Milestones-in-May-Apps-Now-Drive-Half-o

"PlayBuzz." 2015. http://www.playbuzz.com.

"ReCAPTCHA Digitization Accuracy." 2014. http://www.google.com/recaptcha/digitizing.

Rothman, Mike. 2015. "Default Deny." https://securosis.com/blog/network-security-fundamentals-default-de

Rouse, Margaret. 2015. "Authentifizierung - Definition." http://www.searchsecurity.de/definition/Authentifizierung.

"SMI (SocialMedia Institute)." 2015. http://socialmedia-institute.com/.

"Statistik Plattform." 2015. http://de.statista.com/.