Bachelorarbeit
in Medieninformatik
an der Universität zu Köln

Implementierung einer personenbezogenen Programmierschnittstelle („Personal API“) zur Abbildung des digitalen Ichs im Social Web

von Stefan Grund
http://stefangrund.de

Über diese Arbeit

Das Social Web erfreut sich bei Online-Nutzern überall auf der Welt immer größerer Beliebtheit. Im Tausch gegen komfortable Apps und Services überlassen Millionen, gar Milliarden von Menschen ihr digitales Dasein diversen Plattformbetreibern. Hierdurch und durch völlig neue, vernetzte Gerätetypen, wie Fitness-Tracker und Augmented-Reality-Brillen, die ihre erfassten Daten fast ausschließlich in den Datenbestand der Hersteller überführen, verlieren die Nutzer immer mehr die Kontrolle über *ihre eigenen* Daten.

Ziel dieser Arbeit ist es einerseits diese Entmündigung der Nutzer anhand aktueller Entwicklungen im Social Web aufzuzeigen und andererseits Lösungsansätze für eben dieses Dilemma vorzustellen. Neben dem Siegeszug der Smartphones werden dabei auch aktuelle Trends, von Wearables über Quantified Self zu sozialen Filtern, näher betrachtet. Indem hier schließlich eine personenbezogene Programmierschnittstelle – *kurz „Personal API“* – formuliert und implementiert wird, wird zudem eine eigene Lösung zur Sicherung von Nutzerdaten bereitgestellt, die das digitale Schaffen einer Person aufbewahrt, verwaltet und weiter zugänglich macht.

Die Arbeit wurde im Rahmen meines Studiums der Medienwissenschaft und Medieninformatik an der Universität zu Köln bei Prof. Dr. Manfred Thaller im Wintersemester 2013/14 angefertigt und im April 2014 vorgelegt. Getreu dem Motto *»If it's not on the Web, it doesn't exist«* habe ich die Arbeit im August 2014 nochmals für das Web, sowie als PDF und eBook aufbereitet. Online-Version, sowie die PDF- und eBook-Downloads finden sich unter http://stefangrund.de/personalapi/, wo auch die im Verlauf dieser Arbeit entworfene Software zur Verfügung gestellt wird.

***Stefan Grund**, August 2014*
Web: http://stefangrund.de

Titelblatt-Piktogramm „Cloud User“ von Jose Luis Garcia vom Noun Project

Inhaltsverzeichnis

Inhaltsverzeichnis

1. Einleitung – Das geborgte Internet

In einem viel beachteten Artikel namens ‚The Web We Lost' beschreibt der Blogger und Web-Unternehmer Anil Dash Ende 2012 die Anfangszeit des Social Webs, als Facebook und Twitter noch keine mit Milliarden von Dollars bewerteten Börsenschwergewichte waren und noch bevor Google das Konzept der Hyperlinks monetarisierte:

> In the early days of the social web, there was a broad expectation that regular people might own their own identities by having their own websites, instead of being dependent on a few big sites to host their online identity. In this vision, you would own your own domain name and have complete control over its contents, rather than having a handle tacked on to the end of a huge company's site.[1] This was a sensible reaction to the realization that big sites rise and fall in popularity, but that regular people need an identity that persists longer than those sites do.[2]

Eine eigene Webseite oder ein eigenes Weblog, auf dem Erlebnisse, Fotos und Links geteilt werden, gehören heute zur Ausnahme. Stattdessen besteht die Online-Präsenz des Großteils der Online-Nutzer[3] nur noch aus Profilseiten auf kommerziellen Plattformen. Die digitalen Erinnerungen, kreativen Leistungen und zahlreichen anderen, teils sehr persönlichen Daten befinden sich ausschließlich auf den Servern der Plattformbetreiber, wo sie untrennbar mit deren Erfolg und Ermessen verknüpft sind. Ein Kontrollverlust, den man durch den Zugewinn an Komfort ‚likebegeistert' einzugehen scheint, wie Autor und Blogger Sascha Lobo festhält und bei dem er ganz passend, wenn auch etwas ungenau[4], vom ‚geborgten Internet' spricht:

> Daten auf sozialen Netzwerken müssen unter allen Umständen so behandelt werden, als könnten sie jederzeit verloren gehen. Denn sie können jederzeit verloren gehen. Trotzdem scheint die Welt likebegeistert anders zu handeln: All ihr digitales Schaffen findet im geborgten Internet statt.[5]

Um die Kontrolle über die eigene ‚digitale Existenz' zu behalten, empfiehlt Lobo das Betreiben eines selbstkontrollierten Weblogs, da man nur dort machen könne, was man möchte.[6] Dass das mit dem Nutzungsverhalten der Mehrheit der Online-Nutzer jedoch

---

[1] Dash bezieht sich damit auf die URLs der Nutzerprofile von sozialen Netzwerken wie Facebook und Twitter, die nach dem Schema ‚dienstname.tld/nutzername' (also z.B. https://facebook.com/stefangrund, https://twitter.com/stefangrund) aufgebaut sind.

[2] Anil Dash: The Web We Lost. In: Anil Dash. A Blog About Making Culture, 13.12.2012, URL: http://dashes.com/anil/2012/12/the-web-we-lost.html (30.03.2014).

[3] Wenn im folgenden nur von Nutzern oder dem Nutzer die Rede ist, soll dies die Nutzerinnen bzw. die Nutzerin mit einschließen. Der Einfachheit halber wird hier das generische Maskulinum verwendet.

[4] Lobo benutzt hier den Begriff ‚Internet' obwohl Entwicklungen im ‚World Wide Web' gemeint sind. Im allgemeinen Sprachgebrauch wird ‚Internet' oft synonym für ‚World Wide Web' verwendet, obwohl das Internet eigentlich älter ist und nur die technische Infrastruktur des Webs darstellt.

[5] Sascha Lobo: Die Mensch-Maschine: Euer Internet ist nur geborgt. In: Spiegel Online, 17.04.2012, URL: http://www.spiegel.de/netzwelt/web/sascha-lobos-kolumne-zum-niedergang-der-blogs-in-deutschland-a-827995.html (30.03.2014).

[6] Vgl. ebd.

kaum zu vereinbaren ist[7], wird spätestens in Anbetracht der 1,2 Milliarden Nutzer deutlich, die der Branchenprimus Facebook im letzten Quartal 2013 verzeichnen konnte.[8] Die Nutzer verwenden die Angebote von Facebook, Google oder Twitter gerne und oft. Sie dienen als zentrale Anlaufstellen, auf denen alle eigenen Kontakte anzutreffen sind. Durch den Siegeszug der Smartphones wächst das ‚geborgte Internet' zudem unaufhörlich: Die Plattformen sind über Apps ständig präsent und können von den Nutzern jederzeit und überall mit weiteren Daten und Inhalten bespielt werden.

Gleichzeitig zeigen sich die Plattformbetreiber immer restriktiver, was den Umgang mit *ihren* Datenbeständen betrifft. Frei nach Tim O'Reilly[9] sollen nicht reproduzierbare, einzigartige Datensammlungen geschaffen werden, um so die Vermarktbarkeit eines Dienstes zu steigern. Eine nicht nur aus Nutzersicht, sondern auch für das gesamte Web beunruhigende Entwicklung, so der britische Physiker und Informatiker Tim Berners-Lee, der 1989 das World Wide Web begründete. Berners-Lee, der sich seit jeher für offene Standards und dezentrale Strukturen einsetzt, befürchtet, dass die Abgrenzung der sozialen Netzwerke vom restlichen Web dazu führen könnte, dass der von ihm erdachte universelle Informationsraum in ‚fragmentierte Inseln' zerfällt.[10]

Und tatsächlich sind längst nicht mehr alle Inhalte im Social Web adressierbar oder von außerhalb der kommerziellen Ökosysteme zugänglich. Während sich dies aus Nutzersicht beispielsweise darin äußert, dass vermeintlich öffentliche Beiträge erst nach Anmeldung bei einer der Plattformen angezeigt werden können, bemängeln Entwickler, dass offene Schnittstellen und Nutzungsrechte immer weiter limitiert werden. Durch das Aufkommen völlig neuer, vernetzter Gerätetypen wird die Problematik verstärkt: Neben den sozialen Netzwerken und den dazugehörigen Smartphone-Apps gibt es nun dedizierte Werkzeuge, um völlig neue Datenwelten zu erfassen – und diese gleich in den Datenbestand geschlossener Plattformen zu überführen. Ohne weiteres sind die Daten aus digitalen Schrittzählern, intelligenten Armbanduhren und Augmented-Reality-Brillen jenseits der herstellereigenen Webseiten nicht mehr zugänglich und finden nicht weiter im offenen, verknüpfbaren und archivierbaren Teil des Webs statt.

---

[7] Im Rahmen der ARD/ZDF-Onlinestudie wird regelmäßig festgestellt, dass die Verbreitung von sozialen Netzwerken in Deutschland sehr viel höher ist, als die von Weblogs; siehe Kapitel 2.1.3, S. 14.

[8] Vgl. Facebook: Quaterly Earnings Slides Q4 2013, 29.01.2014, S. 5. Abrufbar unter: http://investor.fb.com/results.cfm (30.03.2014).

[9] Tim O'Reilly prägte, wie hier später noch dargestellt wird, den Begriff des Web 2.0. In einem Artikel, in dem er den Begriff konkretisiert, empfiehlt er seinen Lesern: „For competitive advantage, seek to own a unique, hard-to-recreate source of data.", Tim O'Reilly: What is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. In: O'Reilly Media, 30.09.2005, URL: http://oreilly.com/web2/archive/what-is-web-20.html (30.03.2014), S. 5.

[10] Vgl. Tim Berners-Lee: Long Live the Web: A Call for Continued Open Standards and Neutrality. In: Scientific American 165 (2010), H. 12, S. 80-82.

Plattformeigene Exportfunktionen, sofern sie denn überhaupt existieren und genutzt werden, stellen oft nur unzureichende Kopien der nutzergenerierten Inhalte bereit, die oftmals von keinem anderen Dienst importiert werden können. Spätestens wenn Dienste eingestellt werden, gehen Daten regelmäßig verloren. Wenn ein Nutzer von einer Plattform zur nächsten wechseln möchte, werden seine bisher veröffentlichten Inhalte wegen fehlender Kompatibilität unbrauchbar, wie auch Tim Berners-Lee beobachtet:

> Once you enter your data into one of these services, you cannot easily use them on another site. Each site is a silo, walled off from the others. Yes, your site's pages are on the Web, but your data are not. You can access a Web page about a list of people you have created in one site, but you cannot send that list, or items from it, to another site.[11]

Für Nutzer, die ihre eigenen Daten auch tatsächlich besitzen möchten, ein untragbarer Zustand. Niemand weiß, ob es die Plattformen von heute in einigen Jahren noch gibt, weshalb Lösungen gefunden werden sollten, um die eigenen Daten von dort zurückzuholen und selbst zu sichern. Da die Nutzer jedoch weiterhin die großen Plattformen benutzen werden und es sich bei bestehenden Lösungen oft nur um Einzelfalllösungen handelt, bedarf es eines Werkzeugs, das in der Lage ist, die heterogenen Datensätze aus einer Vielzahl von Services zu aggregieren und zu speichern. Idealerweise ein Werkzeug, das nicht lokal ausgeführt wird, sondern Teil des Webs ist und so die kopierten Daten auch in Zukunft im Web bereitstellen kann, sollte der Dienst, von dem sie ursprünglich stammen, nicht mehr erreichbar sein.

Ein solches Werkzeug soll im Verlauf dieser Arbeit entwickelt werden. Dazu bietet sich eine Implementierung in Form einer webbasierten Programmierschnittstelle, einer Web API[12], an. Da diese auf bestehende Web-Standards setzen, ist es auf technischer Ebene möglichst einfach Daten hinzuzufügen oder abzurufen und diese einer Vielzahl von anderen Anwendungen zugänglich zu machen. Indem das Werkzeug in seiner Unterstützung von Plattformen unkompliziert erweiterbar ist, können nach und nach sämtliche vom Nutzer generierten Inhalte ergänzt werden. Ein Anwender des Tools könnte dadurch weiterhin die Dienste der diversen Anbieter nutzen, ohne fürchten zu müssen, seine Daten zu verlieren oder keinen Zugriff mehr auf diese zu haben. Seine personenbezogene Programmierschnittstelle gewährt ihm dauerhaften, zentralen Zugang auf eine Kopie seiner digitalen Daten – auf das Abbild seines *digitalen Ichs*.

---

[11] Ebd., S. 82.

[12] ‚API' steht für ‚Application Programming Interface' und wird im Deutschen ‚Programmierschnittstelle' genannt. In Kapitel 2.2 auf S. 16 wird erklärt, was genau hierunter zu verstehen ist.

Um herauszustellen, um welche Daten es in diesem Kontext geht, welche Probleme es beim Umgang mit diesen gibt und wie die technischen Voraussetzungen aussehen, ist die Arbeit in drei Teile gegliedert. Im ersten Teil, den Grundlagen, geht es zunächst um das Social Web selbst. Bevor die verschiedenen Angebote und deren Verbreitung betrachtet werden, findet eine Annäherung an die Begriffe Web 2.0, Social Web und Social Media statt. Anschließend stehen die webbasierten Programmierschnittstellen im Mittelpunkt, die zahlreichen Anwendungen des Social Webs zugrunde liegen und auch die Basis des hier entwickelten Tools darstellen. Daraufhin soll ein Überblick über aktuelle Entwicklungen gegeben werden. Hierbei wird beschrieben, wie durch Smartphones und Mobile Apps, sowie völlig neue Gerätetypen, die mit immer mehr Sensoren ausgestattet sind, eine immer größer werdende Datenmenge produziert und gleich in die Datensilos restriktiver Anbieter übertragen wird.

Ausgehend von diesen Erkenntnissen soll im zweiten Teil analysiert werden, wie bestehende, plattformeigene und von Dritten angebotene Lösungen nutzergenerierte Inhalte sichern. Danach soll anhand eines exemplarischen Nutzungsszenarios das Konzept der personenbezogenen Programmierschnittstelle konkretisiert werden. Die eigentliche Umsetzung des Tools findet schließlich im dritten Teil statt. Dabei wird ausführlich erläutert, wie die API realisiert wird, welche Funktionen es gibt und warum sich für bestimmte Praktiken und Designs entschieden wird.

Abschließend wird diskutiert werden, ob die hier entwickelte *Personal API* das ‚geborgte Internet' eines Nutzers wieder zu seinem eigenen Teil des Webs machen kann, welche Erfolgsaussichten es für eine solche Lösung gibt und wie eine Weiterentwicklung der Software aussehen könnte.

2. Grundlagen

Bevor es um den Entwurf und die Realisierung des beschriebenen Werkzeugs gehen kann, soll hier ein Überblick über den derzeitigen Zustand des Social Webs gegeben werden. Dazu soll sich zunächst dem Begriff selbst angenähert und dargestellt werden, welche Angebote es gibt und wie diese genutzt werden. Danach sollen die den Social-Web-Anwendungen zugrunde liegenden Programmierschnittstellen erläutert und anschließend aktuelle Entwicklungen aufgezeigt werden, wobei auch der Kontrollverlust der Nutzer über ihre eigenen Daten diskutiert werden wird.

2.1 Das Social Web

Tim Berners-Lee beschreibt in seinem 1999 erschienenen Buch ‚Der Web-Report'[13] seinen Traum für die zukünftige Entwicklung des Webs, den er in zwei Hälften teilt:

> In der ersten Hälfte wird das Web zu einem wesentlich leistungsfähigeren Werkzeug für die Zusammenarbeit von Menschen. Ich habe mir den Informationsraum immer als etwas vorgestellt, auf das jeder sofortigen und intuitiven Zugriff hat, und das nicht nur durchsucht, sondern in dem etwas erstellt werden kann. [...] Außerdem muß es für Gruppen aller Größen möglich sein, elektronisch leicht miteinander zu kommunizieren.
>
> In der zweiten Hälfte des Traums erweitert sich die Zusammenarbeit auf Computer. Computer werden in der Lage sein, alle Daten im Web zu analysieren – Inhalte, Links und Transaktionen zwischen Menschen und Computern.[14]

Fünfzehn Jahre später deutet einiges darauf hin, dass sich dieser Traum zu beiden Teilen im so genannten *Web 2.0* oder *Social Web* erfüllt hat: Es bietet ideale Bedingungen zum kollaborativen Arbeiten, zum Austausch und zum Erstellen von Inhalten durch Millionen von Nutzern, deren Daten zu jeder Zeit algorithmisch geordnet, untersucht und ausgewertet werden können – und werden.

Die Social-Web-Angebote werden heute millionenfach genutzt, um Informationen aller Art miteinander zu teilen, sich kreativ auszudrücken oder Kontakte zu knüpfen und zu pflegen. Bevor hier dargestellt werden kann, welche Plattformen es gibt und wie diese genutzt werden, soll sich zunächst den oft synonym verwendeten Termini *Web 2.0*, *Social Web*, *Social Media* oder *Social Network*, sowie deren deutschsprachige Pendants angenähert werden.

---

[13] Tim Berners-Lee: Der Web-Report. Der Schöpfer des World Wide Web über das grenzenlose Potential des Internets. München 1999.

[14] Ebd., S. 229-230.

2.1.1 Begriffsklärung

Während *Social Web* und *Web 2.0* gemeinhin dasselbe Phänomen beschreiben, ist Web 2.0 das deutlich prominentere Schlagwort. Web 2.0 kennzeichnet dabei keineswegs eine neue technische Version des World Wide Webs, sondern bezeichnet eine Veränderung des Webs weg vom, bis kurz nach der Jahrtausendwende vorherrschenden Prinzip des kommerziellen Inhalte-Monopols durch Unternehmen, hin zu dem, was hierzulande oftmals als ‚Mitmach-Web' bezeichnet wird. Im Web 2.0, so der Gedanke, konsumiert der Nutzer nämlich nicht mehr nur professionell erstellte Inhalte, sondern beteiligt sich aktiv an der Gestaltung des Webs und kreiert eigene Inhalte. Der Begriff wurde maßgeblich von Tim O'Reilly, dem Gründer des auf informationstechnologische Fachbücher spezialisierten O'Reilly Verlags, geprägt, der 2004 mit der ‚Web 2.0 Conference' erstmals eine Konferenz zum Thema veranstaltete. Der Web-2.0-Begriff verbreitete sich rasch „und wurde schnell zum Oberbegriff für sämtliche Erneuerungen im Web"[15], so dass O'Reilly ihn 2005 präzisierte.[16]

Dabei bezieht O'Reilly neben der Idee des ‚Mitmach-Webs' mit ‚user-generated content' auch technische, ökonomische und rechtliche Aspekte mit ein und beschreibt, dass Web-2.0-Angebote sich unter anderem dadurch auszeichnen, dass sie das Web und nicht mehr klassische Desktop-Betriebssysteme als Plattform nutzen („The Web as Platform"[17]), wodurch die Software auf vielen unterschiedlichen Endgeräten benutzbar wird. Auch stellt er explizit heraus, dass die „Daten, die von den Nutzern permanent generiert werden"[18] im Mittelpunkt der Anwendungen stehen. Weiterhin spricht er von ‚leichtgewichtigen' Programmiermodellen, die in Web-2.0-Angeboten verwendet werden sollten, um so leicht zugängliche Schnittstellen (APIs) zu schaffen, die eine weiterführende Verarbeitung der Daten ermöglichen.[19] Im vorliegenden Kontext ist dieser Punkt besonders interessant, da im Verlauf dieser Arbeit schließlich eine solche Programmierschnittstelle entworfen werden soll, die wiederum auf den entsprechenden Schnittstellen der einzelnen Angebote aufsetzt.

Da es sich bei Web 2.0 um einen weit gefassten, unscharfen Sammelbegriff handelt, der, wie bereits erwähnt wurde, gerne für sämtliche Neuerungen im Web gewählt wird, bildete sich in den vergangenen Jahren zudem der Begriff des *Social Web* heraus.

---

[15] Anja Ebersbach, Markus Glaser, Richard Heigl: Social Web. Konstanz 2008, S. 23.

[16] Vgl. Tim O'Reilly: What is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. In: O'Reilly Media, 30.09.2005, URL: http://oreilly.com/web2/archive/what-is-web-20.html (30.03.2014).

[17] Vgl. ebd., S. 1.

[18] Ebersbach, Glaser, Heigl 2008, S. 25.

[19] Vgl. O'Reilly 2005; Ebersbach, Glaser, Heigl 2008, S. 23-29.

Hiermit ist der Teilbereich des Web 2.0 gemeint, bei dem es „nicht um Formate und Programmierarchitekturen, sondern um die Unterstützung sozialer Strukturen und Interaktionen über das Netz geht“[20]. Im Gegensatz zu Web 2.0 meint Social Web also weder juristische Herausforderungen, noch Geschäftsmodelle oder spezielle Techniken, wie z.B. Ajax[21], sondern bezieht sich ausschließlich auf webbasierte Anwendungen zum Informationsaustausch und zur Interaktion zwischen den Nutzern, sowie die dabei entstehenden Daten. Social Web ist somit oft deckungsgleich mit dem Begriff der *Social Software*[22], wobei hierunter auch Instant-Messaging-Dienste, wie z.B. ICQ, oder internetbasierte Computerspiele subsumiert werden können, die nicht auf dem Web, sondern anderen Protokollen und Techniken aufsetzen und somit nicht dem Social Web zugeordnet werden sollten.[23]

Social-Web-Anwendungen zeichnen sich dadurch aus, dass das Individuum oder eine Gruppe im Mittelpunkt stehen. Die Dienste machen die verschiedenen Aktionen einzelner oder mehrerer Nutzer nachvollziehbar und unterscheiden sich damit von „Programmen oder herkömmlichen Webseiten, die quasi anonym genutzt werden“[24]. Die so sichtbar gemachten Personen, Beziehungen und Inhalte werden dabei in Relation zueinander gesetzt, wodurch für Nutzer und Unternehmen ein (vermarktbarer) Mehrwert entsteht.[25]

Neben Web 2.0, Social Web und Social Software tritt mit *Social Media* zudem ein weiterer populärer Ausdruck, dessen deutschsprachiges Pendant *soziale Medien* sich längst im allgemeinen und auch akademischen Sprachgebrauch etabliert hat. Wie die vorausgehenden Begriffe beschreiben soziale Medien zunächst die neuen Möglichkeiten zur Veröffentlichung von Inhalten und zum Austausch mit anderen, betrachten diese aber oftmals auch aus einer medienzentrierten Perspektive: Neben den eigentlichen Plattformen und Werkzeugen, wie Facebook und Twitter oder Weblogs und Wikis, meint Social Media „auch ‚klassische‘ Medien, also journalistisch-redaktionell erstellte Angebote“[26] und deren Auseinandersetzung und Zusammenspiel mit den neuen Diensten und Möglichkeiten. Bei *Social Networks* bzw. *sozialen Netzwerken* handelt es sich schließlich, wie bei Weblogs oder Wikis, um eine Gattung innerhalb von Web 2.0,

---

[20] Ebersbach, Glaser, Heigl 2008, S. 29.

[21] Ajax (englisches Apronym für ‚Asynchronous JavaScript and XML‘) ist eine Technik zur asynchronen Datenübertragung zwischen Browser und Server, die in zahlreichen Web-2.0-Diensten angewandt wird.

[22] Vgl. Hajo Hippner: Bedeutung, Anwendungen und Einsatzpotenziale von Socal Software. In: HMD. Praxis der Wirtschaftsinformatik 43 (2006), H. 252, S. 6-16.

[23] Vgl. Ebersbach, Glaser, Heigl 2008, S. 29-32.

[24] Ebd., S. 31.

[25] Vgl. ebd., S. 29-32.

[26] Jan-Hinrik Schmidt: Social Media. Wiesbaden 2013, S. 8.

Social Web oder Social Media. Das Adjektiv ‚social' oder ‚sozial' ist dabei stets problematisch, da die Angebote „selbstverständlich auch für a-soziale Zwecke wie Mobbing oder Datenmissbrauch verwendet werden [können]"[27] und so im Umkehrschluss der Eindruck entstehen könnte, es gäbe auch nicht-soziale Medien, wobei „Medien als Kommunikationsmittel immer an den Austausch zwischen Menschen gebunden [sind]"[28].

Wenngleich die genannten Begriffe im Allgemeinen synonym verwendet werden und häufig dieselben Entwicklungen, Ideen und Angebote umschreiben, wurde für den Titel dieser Arbeit bewusst der am engsten definierte Begriff *Social Web* gewählt, da die von der Programmierschnittstelle zu verarbeitenden Nutzerdaten aus dem Social Web stammen und die API, als ausschließlich webbasierte, den Informationsaustausch unterstützende Anwendung, auch selbst ein Teil des Social Webs ist.

2.1.2 Die wichtigsten Formen

Das Social Web besteht aus einer Vielzahl von unterschiedlichen Anwendungen, die in verschiedene Formen und Gattungen zusammengefasst werden können. Zwar macht die Schnelllebigkeit des Feldes eine Klassifikation schwierig, dennoch haben sich in der Literatur in den letzten Jahren einige Kriterien herausgebildet, in die sich die diversen Angebote einteilen lassen:

- ***Weblogs***, verkürzt auch *Blogs* genannt, sind persönlich gefärbte und/oder thematisch spezialisierte Webseiten, die meist von Einzelpersonen geführt werden und durch ihre auf Softwareseite einfache Handhabung vom Autor, dem so genannten *Blogger*, schnell aktualisiert und verändert werden können. Die aus der Vernetzung der einzelnen Blogs entstehende Gemeinschaft wird als *Blogosphäre* bezeichnet.

- ***Wikis*** sind Web-Anwendungen zur kollaborativen Erstellung und Bearbeitung von Texten. Der Fokus liegt dabei auf dem Thema selbst, der einzelne Autor ist kaum erkennbar. Wiki-Systeme, wie z.B. MediaWiki, erfordern keinerlei Programmierkenntnisse, sondern lediglich das Aneignen einer simplen Syntax. Das bekannteste Beispiel für ein Wiki ist die Online-Enzyklopädie Wikipedia.

- ***Social Networks*** oder *soziale Netzwerke*, auch *Online-Communities* genannt, zeichnen sich dadurch aus, dass „der Mensch und seine Beziehungen [...] im Mittelpunkt stehen,

---

[27] Ebd., S. 15.

[28] Ebd.

quasi Mittel und Inhalt der Anwendung sind“[29]. Die Plattformen ermöglichen es, Kontakte zu knüpfen und zu pflegen, das Gestalten einer eigenen Profilseite und den Austausch über interessenbasierte Gruppen. Die Dienste können erst nach Registrierung genutzt werden. Dabei gibt der Nutzer Informationen über sich, seine Interessen und/oder seine beruflichen Kompetenzen preis, wodurch er dann von anderen Nutzern des Netzwerks gefunden und kontaktiert werden kann. Neben allgemein zugänglichen, privaten Netzwerken, gibt es spezialisierte Netzwerke für Studenten, Geschäftskontakte oder bestimmte Regionen. Das bekannteste und größte Social Network ist Facebook, das ursprünglich als Plattform für Studierende gestartet war, mittlerweile aber für jeden zugänglich ist.

- ***Social-Sharing-Dienste***, auch als *objektzentrierte Software* oder *Multimediaplattformen* bekannt, sind Anwendungen, „die sich mit dem Bereitstellen und dem Tausch von digitalen Inhalten beschäftigen“[30]. Hier steht das Teilen (seltener das Bearbeiten) von z.B. Bookmarks, Dokumenten, Fotos oder Videos im Mittelpunkt. Die oft, aber nicht zwingend, vom Nutzer selbst produzierten Inhalte können von anderen Nutzern angesehen, bewertet und kommentiert werden. Ein zentrales Element ist hierbei das ‚Tagging‘, bei dem Inhalte mit passenden Schlagworten (Tags) versehen werden, um sie so für andere auffindbar zu machen. Beispiele für objekt-zentrierte Plattformen sind Dienste wie Flickr und Instagram für Fotos, Scribd für Dokumente, Soundcloud für Musikstücke oder YouTube für Videos.

Die Einteilungen sind dabei alles andere als starr. Bei vielen Angeboten handelt es sich um Hybride der gerade genannten Formen. Während etwa Facebook gemeinhin als Social Network aufgefasst wird, haben längst Funktionalitäten von Weblogs und Social-Sharing-Webseiten Einzug in den Dienst gefunden. So ist Facebook – ohne den zugekauften Foto-Dienst Instagram einzubeziehen – mit 350 Millionen hochgeladenen Fotos pro Tag mittlerweile der größte Foto-Sharing-Service der Welt[31], etwas das bislang dedizierten Services vorbehalten zu sein schien. Derweil integrieren die Social-Sharing-Plattformen wiederum Funktionen, die ursprünglich bei sozialen Netzwerken verortet waren.[32] So ermöglicht Twitter, ein Microblogging-Dienst, der mit seinen auf 140 Zeichen beschränkten Statusmitteilungen, den so genannten ‚Tweets‘, zunächst eine

---

[29] Ebersbach, Glaser, Heigl 2008, S. 79.

[30] Ebd., S. 33.

[31] Vgl. A Focus on Efficieny. A whitepaper from Facebook, Ericsson and Qualcomm. In: Internet.org, 16.09.2013, URL: http://internet.org/efficiencypaper (30.03.2014), S. 33.

[32] Vgl. Ebersbach, Glaser, Heigl 2008, S. 33 ff.; Schmidt 2013, S. 11-14.

Spielart des Bloggens bedient, seinen Nutzern auch das Erstellen von Profilseiten, das Befreunden mit anderen Nutzern und die öffentliche und private Kommunikation mit diesen. Das Einbinden von Bildern, Links und Videos in die Tweets, sowie eine Verschlagwortung über so genannte ‚Hashtags' ist ebenfalls möglich.

2.1.3 Verbreitung der Angebote

Die verschiedenen Angebotsformen stoßen auf unterschiedlich großen Anklang bei den Online-Nutzern. Aufgrund der Unübersichtlichkeit und Dynamik des Social Webs ist es jedoch schwierig, verlässliche Aussagen über die Nutzung und Verbreitung der diversen Angebote zu treffen und wegen der unterschiedlichen Märkte, in denen die diversen Unternehmen agieren, und etwaigen Sprachbarrieren unmöglich sämtliche Dienste zu erfassen. Im Fokus der medialen Berichterstattung und des wissenschaftlichen Diskurses stehen jedoch regelmäßig die englischsprachigen, meist US-amerikanischen, international agierenden Dienste, die faktisch auch zu den größten zählen.

Die weltweit mit Abstand größte Plattform ist Facebook, wo im letzten Quartal 2013 rund 1,228 Milliarden aktive Nutzer pro Monat verzeichnet werden konnten, von denen 757 Millionen die Facebook-Services sogar täglich nutzten.[33] Basierend auf einem Bericht mit Nutzungsdaten von Dezember 2013 gehören nach Facebook diese fünf US-amerikanischen Angebote zu den meist genutzten im Social Web[34]:

Plattform	Zugehörigkeit	Form	aktive Nutzer/Monat
YouTube	Google	Social Sharing (Video)	1 Milliarde
Google Plus	Google	Social Network	327 Millionen
Tumblr	Yahoo	Blogging	300 Millionen
Twitter	-	Microblogging	240 Millionen
LinkedIn	-	Social Network	184 Millionen

Tabelle 1: Die nach Facebook meist genutzten US-amerikanischen Social-Web-Angebote nach einem Bericht des Business Insider von Dezember 2013.[35]

---

[33] Vgl. 4. Quartalsbericht 2013 von Facebook: Quaterly Earnings Slides Q4 2013, 29.01.2014, S. 5. Abrufbar unter: http://investor.fb.com/results.cfm (30.03.2014).

[34] Neben asiatischen Angeboten, die wegen unmöglicher Vergleichbarkeit durch den Autor hier keine weitere Berücksichtigung finden, listet der Bericht auch Messaging-Dienste wie WhatsApp, Line und WeChat auf. Diese sind der Argumentation in Kapitel 2.1 folgend aber nicht zum Social Web zu zählen, weshalb sie hier ebenfalls unberücksichtigt bleiben.

[35] Vgl. Marcelo Ballve: List Of The World's Largest Social Networks. In: Business Insider, 17.12.2013, URL: http://www.business insider.com/the-worlds-largest-social-networks-2013-12 (30.03.2014). Die angegebenen Werte wurden durch eigene Recherche überprüft.

Zur Verbreitung der verschiedenen Web-Anwendungen unter deutschen Online-Nutzern liefert die seit 1997 jährlich durchgeführte ARD/ZDF-Onlinestudie gute Erkenntnisse. Danach haben vor allem soziale Netzwerke in den letzten Jahren deutlich an Attraktivität gewonnen: Private Netzwerke wie Facebook werden mittlerweile von 46 Prozent aller deutschsprachigen Online-Nutzer ab 14 Jahren zumindest gelegentlich genutzt, berufliche Netzwerke wie LinkedIn oder Xing von 10 Prozent. Für 75 Prozent der 14 bis 29 jährigen, 48 Prozent der 30 bis 49 jährigen und 38 Prozent der Nutzer ab 50 Jahren gehören private Netzwerke bereits zum Medienalltag und werden täglich genutzt. Die Netzwerke werden dabei allem voran dazu verwendet, um mit den eigenen Kontakten über private Nachrichten, Beiträgen auf den Profilseiten oder per Chat zu kommunizieren. Doch auch das Betrachten von Fotos und Videos, sowie das Teilen von Webinhalten sind beliebte Tätigkeiten innerhalb der Netzwerke.[36]

Aufgrund der teils exzessiven Nutzung kommen die Medienforscherinnen Birgit van Eimeren und Beate Frees zu der Schlussfolgerung, dass „Mark Zuckerbergs Vision, dass Facebook für seine 1,3 Milliarden Nutzer weltweit das ‚Betriebssystem des Internets' sein soll, heute schon für einen Teil seiner Nutzer Realität sein [mag]"[37], indem Facebook es seinen Nutzern „mit immer neuen Anwendungen wie Spielen, Nachrichtentickern, Videos etc. [ermöglicht], die Plattform nicht mehr verlassen zu müssen"[38]. Eine Beobachtung, die die hier bereits erwähnte Hybridisierung der Social-Web-Formen unterstreicht.

Noch populärer als die sozialen Netzwerke sind nur Videoportale wie z.B. YouTube, die von 60 Prozent der Online-Nutzer genutzt werden, und die Wikipedia, die sogar von 74 Prozent der deutschen Nutzer regelmäßig verwendet wird. Inzwischen ist mit 16 bzw. 7 Prozent jedoch auch ein wachsendes Interesse an Weblogs und Microblogging, genauer gesagt Twitter, zu beobachten. Diese Formen des Social Webs wurden in Deutschland trotz ihrer medialen Omnipräsenz bisher nur zurückhaltend genutzt. Auch wenn die ARD/ZDF-Onlinestudie nur die Nutzung eines Bruchteils der zahlreichen Social-Web-Anwendungen abfragt und ermittelt, ist daran doch zu erkennen, dass die Angebote, allen voran die alle Formen assimilierenden sozialen Netzwerke, angenommen werden und eine immer größere Nutzerschaft finden.

---

[36] Vgl. Social Media. In: Webseite der ARD/ZDF-Onlinestudie, 2013, URL: http://www.ard-zdf-onlinestudie.de/index.php?id=397, http://www.ard-zdf-onlinestudie.de/index.php?id=434 (30.03.2014).

[37] Birgit van Eimeren, Beate Frees: Ergebnisse der ARD/ZDF-Onlinestudie 2013. Rasanter Anstieg des Internetkonsums – Onliner fast drei Stunden täglich im Netz. In: Media Perspektiven (17) 2013, H. 7-8, S. 362.

[38] Ebd.

2.2 Programmierschnittstellen im Social Web

Wie anhand der zahlreichen, erwähnten Social-Web-Anwendungen zu sehen ist, hat sich Tim O'Reillys 2004 formulierte These, dass sich das Web (2.0) zur einer Softwareplattform entwickeln wird, bestätigt. Nun kommunizieren nicht weiter nur Menschen mit Menschen und Menschen mit Maschinen, sondern auch Maschinen untereinander. Damit die Maschinen bzw. die Programme miteinander interagieren können, bedarf es bestimmter Schnittstellen und Konventionen. Eine solche Schnittstelle wird *Application Programming Interface*, kurz *API,* oder im Deutschen *Programmierschnittstelle* genannt und ist praktisch überall in der Softwareentwicklung zu finden. Betriebssysteme und Programmiersprachen verfügen ebenso über APIs, wie Desktop-, Mobile- oder Webanwendungen.

2.2.1 API

APIs ermöglichen und regeln die Kommunikation zwischen zwei Systemen, „[s]ie dienen zum Austausch und der Weiterverarbeitung von Daten und Inhalten zwischen verschiedenen Websites, Programmen und Anbietern, und ermöglichen so Dritten den Zugang zu vorher verschlossenen Datenpools und Benutzerkreisen“[39]. Eine API kann somit als das maschinenlesbare Äquivalent zu einem *(Graphical) User Interface*, kurz *GUI*, wie es von einem menschlichen Nutzer verwendet wird, verstanden werden. Wie die folgende Darstellung zeigt, ermöglichen beide, die Programmier- und die grafische Benutzerschnittstelle, als das so genannte Front End, den Zugriff auf die Softwarelogik im Back End, wo die eigentliche Datenverarbeitung stattfindet.

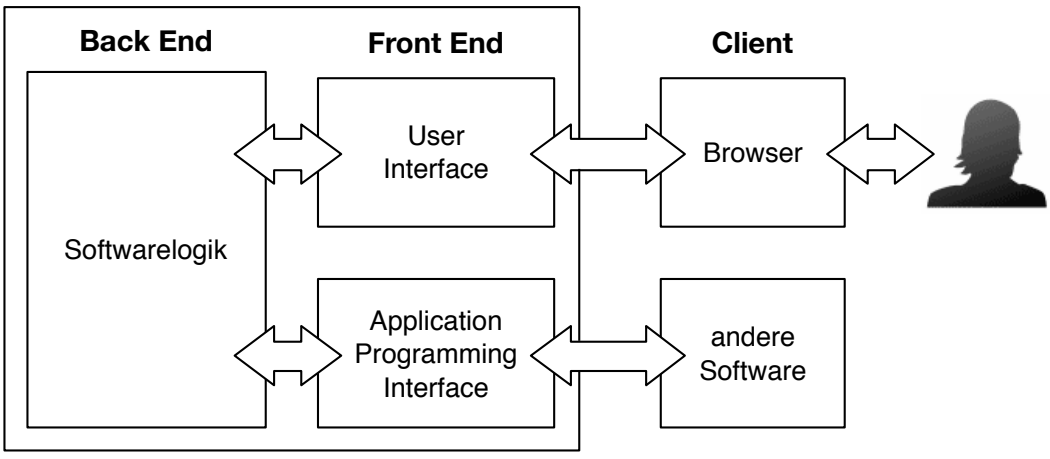

Abbildung 1: Der Stellenwert der API innerhalb einer Webanwendung.[40]

[39] Web APIs. Ein nicht-technischer Erklärungsversuch. In: Gründerszene.de, 09.11.2009, URL: http://www.gruenderszene.de/allgemein/web-apis-ein-nicht-technischer-erklarungsversuch (30.03.2014).

[40] Vgl. ebd.

Implementierungen, die den Austausch von Systemen über das Web ermöglichen, werden gemeinhin *Web Services* genannt, wobei der Begriff dadurch verwässert wird, dass in den Medien häufig sämtliche Webanwendungen als ‚Webservices' bezeichnet werden. Zudem meint Web Services im ursprünglichen Sinne Softwareanwendungen, die mit den so genannten, vom W3C[41] spezifizierten WS*-Standards realisiert wurden. Neben zahlreichen Klassen gehören zu diesen die Web Services Description Language (WSDL) und das Netzwerkprotokoll SOAP[42], bei denen Informationen über komplexe XML-Nachrichten ausgetauscht werden. Im und durch das Social Web hat sich in den letzten Jahren mit *REST* jedoch ein anderer Softwarearchitekturstil für APIs durchgesetzt und Standards wie SOAP und WSDL in den Enterprise-Bereich verdrängt.

2.2.2 REST

REST steht für *Representational State Transfer* und zeichnet sich durch eine vergleichsweise einfache Implementierung auf Basis der bestehenden Web-Standards aus.[43] Der REST-Begriff wurde vom Informatiker Roy Fielding geprägt, der zuvor bereits an der Spezifikation des Hypertext Transfer Protocols (HTTP) mitgewirkt hat, das für die Übertragung der Inhalte im World Wide Web zuständig ist. Im Rahmen seiner im Jahr 2000 angefertigten Dissertation ‚Architectural Styles and the Design of Network-based Software Architectures'[44] beschreibt Fielding die Merkmale von REST, aus denen sich vor allem die folgenden Grundprinzipien für REST-konforme Dienste ableiten:

- **Adressierbarkeit:** Alle Ressourcen, das sind jegliche Inhalte, Daten und Funktionen, müssen über eine eindeutige Adresse, die so genannte URI[45], verfügen, um eindeutig angesprochen werden zu können.

- **Zustandslosigkeit:** Es werden keinerlei Zustandsinformationen gespeichert, weshalb jede Anfrage komplett in sich geschlossen sein muss und jede Antwort alle benötigten Informationen beinhaltet.

---

[41] W3C steht für World Wide Web Consortium, das 1994 von Tim Berners-Lee gegründete Gremium zur Standardisierung von Web-Technologien, wie z.B. HTML oder XML.

[42] SOAP war ursprünglich ein Akronym für ‚Simple Object Access Protocol', was mittlerweile aber verworfen wurde, weil SOAP subjektiv alles andere als einfach (simple) einzusetzen ist.

[43] Zwar sind theoretisch auch Implementierungen auf Basis anderer Standards denkbar, faktisch meint REST aber immer Web-Standards wie HTTP, URI, XML usw.

[44] Vgl. Roy Fielding: Architectural Styles and the Design of Network-based Software Architectures. Doktorarbeit, University of California, 2000.

[45] URI steht für Uniform Resource Identifier und meint den eindeutigen Namen und die eindeutige Adresse einer Ressource. Eine Unterart der URI ist die von Web-Adressen bekannte URL (Uniform Resource Locator), bei der es sich im Prinzip um eine URI handelt, deren Zugriffsmechanismus auf HTTP oder FTP festgelegt ist. Da zunächst nur diese Protokolle verwendet wurden, werden URI und URL oft synonym verwendet.

- **Unterschiedliche Repräsentation:** Die Ressourcen sollen von ihren Repräsentationen entkoppelt werden. Das heißt, dass es für ein und dieselbe Ressource, die unter einer URI erreichbar ist, unterschiedliche Darstellungsformen, z.B. im JSON- oder XML-Format[46], geben soll, die formale Repräsentation also nicht mit den eigentlichen Inhalten verknüpft ist.

- **Einheitliche Schnittstelle:** Es soll ein kleiner, vorbestimmter Satz an Operationen verfügbar sein, der auf die einzelnen Ressourcen anwendbar ist. Beim Hypertext Transfer Protocol sind das die so genannten Request-Methoden[47], mit denen sich u.a. Ressourcen lesen (GET-Methode), erzeugen (POST-Methode), aktualisieren (PUT-Methode) und löschen (DELETE-Methode) lassen.

Wie diese Prinzipien praktisch umgesetzt werden, ist nicht weiter definiert und bleibt den Entwicklern überlassen. Implementierungen, die den Prinzipien entsprechen, sind *REST-konform* oder *RESTful* und werden dementsprechend *RESTful Web Services* oder schlicht *REST APIs* oder *Web APIs* genannt.[48]

2.2.3 Web APIs

Obwohl das Konzept der APIs, wie eben herausgestellt wurde, bereits lange verbreitet ist, erfahren sie im Web gerade eine neue Popularität. Dabei werden sie für die unterschiedlichsten Zwecke eingesetzt: Um Inhalte einer Plattform auf anderen Webseiten anzuzeigen, um verschiedenste Angebote als Mashup[49] miteinander zu verknüpfen oder um innerhalb von Mobile Apps auf das Back End von Webanwendungen zuzugreifen. Die Webseite ProgrammableWeb, die einen Überblick über das ‚Web als Plattform' geben will, zählt aktuell über 11.100 APIs, von denen mehr als 7.100 REST-konform entworfen wurden.[50] Vor allem sind es jedoch die APIs großer Firmen, wie Facebook, Google und Twitter, die das Feld dominieren und in zahlreichen anderen Anwendungen aufgegriffen werden. Mittlerweile gibt es sogar Unternehmen, deren Geschäftsmodell ausschließlich auf der Arbeit mit einer dieser APIs basiert.

---

[46] Mit JSON- (JavaScript Object Notation) und XML (Extensible Markup Language) lassen sich Daten hierarchisch strukturieren und so auch maschinell auswerten. Wobei JSON deutlich kompakter ist.

[47] Die Request-Methoden finden sich in der HTTP-Spezifikation des W3C. Vgl. Method Definitions. In: Roy Fielding et al.: Hypertext Transfer Protocol – HTTP/1.1, Juni 1999, URL: http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html (30.03.2014).

[48] Vgl. Leonard Richardson, Sam Ruby: RESTful Web Services. Sebastopol 2007, S. 79 ff.

[49] Mashup bezeichnet „die Verbindung zweier Produkte zu einem neuen". Im Bezug auf Webanwendungen meint Mashup „die Neukombination von Inhalten und Techniken unterschiedlicher Anbieter" über deren API. Dirk von Gehlen: Mashup. Lob der Kopie. Berlin 2011, S. 206.

[50] Stand: 18.03.2014, siehe: http://www.programmableweb.com/. Zum Vergleich: Im April 2008 listete die Webseite *nur* 740 APIs auf. Vgl. E. Michael Maximilien, Ajith Ranabahu, Karthik Gomadam: An Online Platform for Web APIs and Service Mashups. In: IEEE Internet Computing 12 (2008), H. 5, S. 32.

Am Beispiel von Twitter wird der Erfolg der Web APIs besonders deutlich. Es gibt zahlreiche Desktop-, Mobile- und Web-Apps, die es dem Nutzer nach erfolgreicher Autorisierung ermöglichen, über die Twitter API Tweets zu lesen, zu schreiben und zu favorisieren, sich mit anderen Nutzern zu be- oder entfreunden und Nachrichten mit diesen auszutauschen. Sämtliche Interaktionen mit der Plattform werden sowohl bei den Twitter-eigenen Apps, als auch bei Drittanbieter-Applikationen über die umfassende REST API des Unternehmens getätigt, weshalb sie zur wahrscheinlich wichtigsten Komponente des Services geworden ist.[51] Die API selbst ist ausführlich dokumentiert[52], um Entwicklern einen problemlosen Einstieg zu ermöglichen.

Zur Veranschaulichung der oben beschriebenen REST-Prinzipien und der Funktionsweisen von Web APIs soll hier kurz eine Anfrage an die Twitter API durchexerziert werden. Um z.B. die letzten Statusmitteilungen eines Nutzers abzufragen, muss der Client einen HTTP-GET-Request (Prinzip der **einheitlichen Schnittstelle**) an die folgende URI senden[53] (**Adressierbarkeit**), die sich aus dem Protokoll (`https://`), der Subdomain unter der die API erreichbar ist (`api.`), der Twitter-Domain (`twitter.com`), der Version der API (`1.1`), der Ressource (`statuses/user_timeline`), der Repräsentation der Ressource (`.json`) und dem Parameter (`user_id`) zusammensetzt, der bestimmt wessen letzte Tweets abgerufen werden sollen (hier die von User `1234`):

```
https://api.twitter.com/1.1/statuses/user_timeline.json?user_id=1234
```

Gemäß dem gewünschten Ausgabeformat antwortet der API-Server hierauf im JSON-Format (**unterschiedliche Repräsentationen**) und listet in rückwärts sortierter Reihenfolge die letzten Statusmitteilungen des angegebenen Nutzers auf. Das JSON-Objekt jeder einzelnen Mitteilung, die maximal ja nur 140 Zeichen beinhalten können, umfasst dabei rund hundert Zeilen und fast ebenso viele, teils optionale Eigenschaften (**Zustandslosigkeit**), wie den Zeitpunkt der Veröffentlichung des Tweets, die Geokoordinaten, an denen er verfasst wurde, verwendete Hashtags und Hyperlinks und selbst das Profilbild des Autors und Hintergrundbild von dessen Profilseite.

---

[51] Biz Stone, einer der Gründer von Twitter, sagte bereits 2007, einem Jahr nach dem Start des Dienstes: „The API has been arguably the most important, or maybe even inarguably, the most important thing we‘ve done with Twitter. It has allowed us, first of all, to keep the service very simple and create a simple API so that developers can build on top of our infrastructure and come up with ideas that are way better than our ideas.“ Vgl. Sean Ammirati: Twitter's Open Platform Advantage. In: ReadWrite, 05.09.2007, URL: http://readwrite.com/2007/09/05/twitter_open_platform_advantage (30.03.2014).

[52] Siehe Twitter: REST API v1.1 Resources, URL: https://dev.twitter.com/docs/api/1.1 (30.03.2014).

[53] Twitter verlangt in der Praxis noch eine Authentifizierung durch Anwendung und Nutzer, was für die Veranschaulichung der Funktionsweise hier aber unbedeutsam ist.

Das Design und die Funktionsweise der Web APIs variieren jedoch von Plattform zu Plattform. Während Twitter zum Beispiel eine API-Versionierung per Pfadangabe (`/1.1`) gewählt hat, verzichten andere Anbieter möglicherweise auf eine Versionsnummer oder setzen die Versionierung anders um. Bei Foursquare, einem Dienst zum Teilen des eigenen Aufenthaltsorts, wird z.B. ein Parameter mit dem aktuellen Datum übergeben, wodurch der Server mit der an diesem Tag gültigen API antwortet. Mit der Zeit haben sich so verschiedene *Best Practices* etabliert, nach denen eine REST API idealerweise entworfen werden sollte, die aber keinesfalls verbindlich sind.[54] Web APIs werden somit (fast) universell einsetzbar, wann immer zwei Systeme über das Web miteinander kommunizieren. Durch die Verwendung bestehender Web-Standards sind sie plattformunabhängig, das heißt, dass die durch Server und Client verwendeten Betriebssysteme und Programmiersprachen unbedeutend sind, da alle modernen Sprachen den HTTP-Standard und auch die Verarbeitung von JSON- und/oder XML-Dateien beherrschen.

Web APIs sind so zu einem wesentlichen Teil des Social Webs geworden, indem sie die Kommunikation zwischen den diversen Diensten und mit deren Mobile Apps ermöglichen. Auch bei den im nächsten Kapitel beschriebenen Entwicklungen, den neuen Gerätetypen und der aus diesen resultierenden Quantified-Self-Bewegung spielen Web APIs eine zentrale Rolle.[55] Jedoch sind auch bei den vermeintlich offenen APIs in zunehmendem Maße Nutzungs- und funktionale Beschränkungen zu beobachten, wie die Fachzeitschrift ‚Computer' des IEEE[56] im August 2013 festhält:

> Some companies are becoming less willing to share the data that other companies currently access via APIs, making APIs less useful [...]. These companies consider the data to be valuable and prefer to keep it in-house [...].[57]

So beschränkte Twitter, das in vielerlei Hinsicht als Prototyp eines Social-Web-Dienstes angesehen werden darf, Mitte 2012 seine bis dahin gänzlich offene Web API, indem Funktionen privatisiert und Abfragen für externe Entwickler limitiert wurden. Den Entwicklern, deren Anwendungen den Dienst Jahre zuvor populär gemacht haben, wurde empfohlen nicht weiter an klassischen Twitter-Clients zu arbeiten, da derartigen Apps nicht weiter die erforderlichen Nutzungsrechte erteilt würden. Eine Abkehr vom offenen Web, wie sie auch bei anderen Anbietern zu beobachten ist.

---

[54] Auf technische Details wird hier später bei der Implementierung einer eigenen API eingegangen.

[55] Vgl. Lee Garber: The Lowly API Is Ready to Step Front and Center. In: Computer 44 (2013), H. 8., S. 14-17.

[56] Institute of Electrical and Electronics Engineers, der weltweit größte technische Berufsverband.

[57] Garber 2013, S. 17.

2.3 Aktuelle Entwicklungen

Wie bereits herausgestellt wurde, sind Anwendungen des Social Webs für viele Nutzer längst alltäglich und durch mobile Endgeräte allgegenwärtig. Es scheint, als sei – trotz datenschutzrechtlicher Bedenken – jeder auf einer der Plattformen vertreten, um seine Freunde, Fremde und Follower[58] an seinem Leben oder Schaffen teilhaben zu lassen oder an deren teilzuhaben. Tim Berners-Lee zeigt sich indes zwiegespalten, was die Zukunft seiner Erfindung anbelangt. Auf der einen Seite preist er die Möglichkeiten und Vorteile, die sich durch das World Wide Web für Menschen überall auf der Welt ergeben haben. Auf der anderen Seite sieht er „unter Regierungen, großen Organisationen und Konzernen einen gefährlichen Trend, das Web kontrollieren zu wollen“[59]. Neben dem Missbrauch des Webs zur flächendeckenden Überwachung der Bevölkerung und der drohenden Abschaffung der Netzneutralität durch die Zugangsprovider[60], verletzen auch die Betreiber von sozialen Netzwerken und anderen Social-Web-Anwendungen die Prinzipien des Webs, das seit jeher universell, gebührenfrei, offen und dezentralisiert sei, indem sie ihre Plattformen aus ökonomischen Gründen immer mehr vom restlichen, offenen Web abgrenzen.[61]

Das Social Web besteht dabei nicht mehr nur aus Webseiten, die vom heimischen Desktop-Rechner aus aufgerufen werden, sondern ist durch Smartphones und deren größeren Pendants, den Tablet-PCs, mobil und omnipräsent geworden. Für viele Nutzer hat sich das Smartphone als Inbegriff der inhaltlichen und technischen Medienkonvergenz längst zu der zentralen Stelle ihres Medienalltags entwickelt. Doch auch völlig neue Gerätetypen, wie Fitness-Tracker und Smartwatches, finden eine immer größere Nutzerschaft und generieren immer mehr Daten. Im folgenden soll ein Überblick über diese Entwicklungen gegeben werden und abermals aufgezeigt werden, wie immer mehr Nutzerdaten ausschließlich im ‚geborgten Internet‘ existieren.

---

[58] ‚Follower‘ werden die Abonnenten/Leser eines Nutzers auf Twitter genannt.

[59] Marcel Rosenbach: Tim Berners-Lee über das Web: "Nichts ist perfekt". In: Spiegel Online, 12.03.2014, URL: http://www.spiegel.de/netzwelt/web/tim-berners-lee-ueber-das-internet-und-25-jahre-www-a-957978.html (30.03.2014).

[60] Das Web zeichnet sich unter anderem durch das Fehlen von (technisch-bedingten) Hierarchien aus. Jede Information hat demnach grundsätzlich die gleiche Chance zur Teilnahme und Übertragung. Netzneutralität bezeichnet diese wertneutrale Übertagung von Daten. Telekommunikationsunternehmen überall auf der Welt bedrohen dieses Prinzip, indem sie z.B. den Zugang zu datenintensiven Diensten, wie Video-Streaming-Angeboten, beschränken und kostenpflichtig machen wollen. Die Telekommunikationsunternehmen rechtfertigen das mit der Deckung der hohen Kosten des Netzausbaus. In einigen Staaten ist die Netzneutralität derweil bereits gesetzlich festgelegt.

[61] Vgl. Berners-Lee 2010, S. 80-84.

2.3.1 Smartphones und Tablets

Der Siegeszug der Smartphones, den rechenstarken und internetfähigen Mobiltelefonen mit meist berührungsempfindlichen Bildschirmen, der 2007 mit der Markteinführung des Apple iPhone begann, hat die mobile Internetnutzung zu einem Massenphänomen gemacht, wie der Medienforscher Thorsten Müller festhält:

> Betrug der Anteil der Mobilnutzung im Jahr 1999 nur 5 Prozent, so ist die Zahl der mobilen Internetnutzer in den vergangenen vier Jahren sprunghaft angestiegen: von 13 Prozent im Jahr 2010, über 20 Prozent im Jahr 2011 und 23 Prozent im Jahr 2012 auf nunmehr 41 Prozent im Jahr 2013. Auch wenn der Marktanteil der Apple-Produkte in Deutschland bei den Smartphones aktuell nur noch bei unter 20 Prozent liegt, so ist die Bedeutung des iPhones für die Verbreitung der Smartphone-Technologie bzw. des iPads bei den Tablet-PCs als Pionierleistung unbestritten.[62]

Bereits heute verfügen 56 Prozent der deutschen Online-Haushalte über Smartphones. Tablet-PCs, die tragbaren, flachen, ebenfalls über den Bildschirm gesteuerten Computer, sind mittlerweile in 19 Prozent der Haushalte vertreten, während sie 2012 in nur 8 Prozent der Online-Haushalte vorzufinden waren. Damit verbreiten sie sich schneller als jede andere Kategorie von mobilen Endgeräten.[63] Auch international ist eine hohe Verbreitung zu verzeichnen.[64] Da diese Marktdurchdringung in nur sechs bzw. drei Jahren nach der Vorstellung der richtungsweisenden Apple-Produkte iPhone und iPad erreicht werden konnte, ist auch in den kommenden Jahren ein Anstieg der Verbreitung von Smartphones und Tablets, sowie der mobilen Internetnutzung zu erwarten. Infolgedessen werden immer mehr Online-Nutzer *always on*, also ständig und überall mit dem Internet verbunden sein.

Dass Facebook durch 945 Millionen seiner 1,228 Milliarden Nutzer bereits jetzt zum Großteil auf mobilem Wege genutzt wird, ist danach nicht weiter überraschend.[65] Auch Twitter wurde von 184 Millionen seiner 241 Millionen aktiven Nutzer im letzten Quartal 2013 über mobile Endgeräte genutzt[66], wobei die Benutzung der Plattformen zumeist nicht mehr über die (eventuell für Mobiltelefone angepasste) Webseite im Browser des Smartphones erfolgt, sondern über die jeweiligen, plattformeigenen Apps.

---

[62] Thorsten Müller: Habitualisierte Mobilnutzung – Smartphones und Tablets gehören zum Medienalltag. In: Media Perspektiven (17) 2013, H. 9, S. 410.

[63] Vgl. Birgit van Eimeren: „Always on“ – Smartphone, Tablet & Co. als neue Taktgeber im Netz. In: Media Perspektiven (17) 2013, H. 7-8, S. 386-387.

[64] Vgl. The Nielsen Company: The Mobile Consumer. A Global Snapshot. In: Nielsen.com, Februar 2013, URL: http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2013%20Reports/Mobile-Consumer-Report-2013.pdf (30.03.2014).

[65] Vgl. Facebook 2013, S. 5-6.

[66] Twitter: Twitter Reports Fourth Quarter and Fiscal Year 2013 Results. In: Twitter Investor Relations, 05.02.2014, URL: https://investor.twitterinc.com/releasedetail.cfm?releaseid=823321 (30.03.2014).

2.3.2 Native Apps

Programme für Mobilgeräte, so genannte Apps, haben den Vorteil, dass sie *nativ* sind, also speziell für ein mobiles Betriebssystem entwickelt wurden. So können sie die Funktionen und Rechenleistung der modernen Mobiltelefone und Tablet-PCs besser ausnutzen als eine Webseite, obwohl sie im Hintergrund auf denselben Netzwerkprotokollen aufsetzen wie ein Web-Browser.[67] Da über die Web APIs der Dienste jedoch nur die eigentlichen Inhalte und keine Designelemente oder Programmbestandteile mehr geladen werden müssen, bieten sie eine bessere Zugriffsgeschwindigkeit. Über die in die Betriebsysteme integrierten *App Stores* können die Apps auf unkomplizierte Art und Weise von den Nutzern heruntergeladen und kinderleicht installiert werden. Laut ARD/ZDF-Onlinestudie 2013 benutzen bereits 44 Prozent der deutschen Online-Nutzer Apps auf ihren Smartphones und Tablets. Dabei sind Instant-Messaging-Apps, wie etwa WhatsApp[68], und die Apps der sozialen Netzwerke die am häufigsten genutzten Programme auf Smartphones. Die Nutzer von Tablet-PCs verwenden hingegen Nachrichten-Apps und ebenfalls Social-Networking-Apps am häufigsten.[69]

Die Apps stellen dabei nicht nur eine neue Zugangsmöglichkeit zum Social Web dar, sondern sind teilweise sogar deren einzige Repräsentation. Dienste wie Instagram und Vine, mit denen per Smartphone aufgenommene Fotos bzw. kurze Videos geteilt werden können, oder Foursquare, das es ermöglicht, in Lokalitäten ‚einzuchecken' und anderen so seinen Aufenthaltsort mitzuteilen, lassen sich nur durch die jeweiligen Apps nutzen (selbst die Registrierung ist nur über die Apps möglich) oder durch diese erst im vollem Umfang verwenden. Die dazugehörigen Webseiten dienen oft nur dazu, die nutzergenerierten Inhalte – und die Plattform selbst – auch im Web sichtbar zu machen. Die Apps fungieren damit als attraktive User Interfaces, die einerseits die Möglichkeiten der modernen Geräte gekonnt ausnutzen, andererseits aber auf der bestehenden Infrastruktur des World Wide Web aufbauen, indem sie über Web APIs mit den Servern ihrer Plattformen kommunizieren.

Viele Unternehmen, die ihre Inhalte vormals exklusiv im Web realisierten, setzen mittlerweile auf die Produktion von nativen Apps, da sich diese über die Zugänglichkeit

---

[67] Hier sind natürlich nur die vernetzten Apps, wie eben bei sozialen Netzwerken, gemeint. Eine simple Taschenrechner-App verwendet in der Regel gar keine Netzwerkprotokolle.

[68] Zwar wurde bereits herausgestellt, dass es sich bei Instant-Messaging-Diensten wie WhatsApp nicht um Anwendungen des Social Web handelt, gerade WhatsApp ist jedoch dahingehend interessant, dass die App im Februar 2014 für 19 Milliarden US-Dollar von Facebook gekauft wurde. Offensichtlich erkannte Facebook, dass das eigene Geschäftsmodell durch Dienste, die nicht mehr im Web, sondern ausschließlich auf Smartphones stattfinden, nachhaltig bedroht werden könnte und sah Handlungsbedarf. Es wird interessant sein zu sehen, ob und inwieweit WhatsApp in die Facebook-Services und somit ins Social Web integriert werden wird.

[69] Vgl. van Eimeren 2013, S. 388-389.

der integrierten App Stores schnell und einfach monetarisieren lassen. Tim Berners-Lee erachtet diesen Schritt, weg von allgemein zugänglichen, webbasierten Apps, hin zu proprietären, nativen Apps, als problematisch, da sich die damit erstellten Inhalte nicht weiter im Web befinden und folglich ebenso wenig verknüpft und geteilt werden können.[70] In dem Konkurrenzkampf zwischen nativen Apps und dem offenen Web sehen andere überdies sogar den ‚Battle of the Decade'[71], in dem sich die Ausrichtung der Softwareprogrammierung für die nächsten Jahre entscheiden wird – und, wenn es nach den Technikjournalisten Chris Anderson und Michael Wolff geht, auch die Zukunft des Webs, das sie wegen des Aufstiegs der Apps unlängst für tot erklärt haben.[72]

Abbildung 2, von links nach rechts: Der digitale Schrittzähler Fitbit One, die erste Version der Augmented-Reality-Brille Google Glass und die Smartwatch Pebble.

2.3.3 Wearables, Sensoren und soziale Filter

Darüber hinaus treten zu den Smartphones und Tablets in zunehmendem Maße neue mobile Geräte, aus teils völlig neuen und unterschiedlichen Produktkategorien. Die neuen Gerätetypen werden durch den Nutzer meist am Körper getragen und erfassen Daten, die den Smartphones bisher verborgen blieben.[73] Neben digitalen Schrittzählern und Fitness-Armbändern, die die körperlichen Aktivitäten ihres Trägers aufzeichnen, gibt es so genannte Smartwatches, intelligente Armbanduhren, die die Funktionen von Smartphones ans schnell erreichbare Handgelenk bringen sollen, und Augmented-Reality-Brillen, die Informationen gleich ins Sichtfeld ihrer Träger einblenden.[74] Sogar Textilien und Pflaster werden mit Sensoren ausgestattet und somit zu weiteren Daten-

---

[70] Vgl. Berners-Lee 2010, S. 80-83.

[71] Vgl. Tommi Mikkonen, Antero Taivalsaari: Apps vs. Open Web: The Battle of the Decade. In: Proceedings of the 2nd Workshop on Software Engineering for Mobile Application Development, 27.10.2011, URL: http://livelykernel.cs.tut.fi/publications/BattleOfTheDecade-Mikkonen-Taivalsaari.pdf (30.03.2014).

[72] Vgl. Chris Anderson, Michael Wolff: The Web Is Dead. Long Live the Internet. In: Wired, 17.08.2010, URL: http://www.wired.com/magazine/2010/08/ff_webrip/ (30.03.2014).

[73] Wobei auch hier die Grenzen verschwimmen: Das iPhone 5S verfügt z.B. über einen Bewegungssensor, der ähnlich einem dedizierten Fitness-Tracker die Schritte eines Nutzers erfassen kann.

[74] Siehe Abb. 2.

lieferanten. Teils ergänzen die Geräte so das Smartphone, werden durch dieses gesteuert und stellen darüber eine Internetverbindung her, teils sind sie selbst vollständig vernetzt und unabhängig von den Mobiltelefonen benutzbar.[75]

Die *Appccessoires* oder *Wearables* genannten Geräte ermöglichen das Erfassen völlig neuer Datenwelten und erweitern das Social Web sukzessive um diese. So gibt es Online-Communities, in denen sich Nutzer anhand der Daten ihrer Fitness-Tracker im gegenseitigen, sportlichen Wettstreit befinden, oder Angebote, auf denen das von der Augmented-Reality-Brille aufgezeichnete Sichtfeld als Video eingestellt oder gleich live übertragen werden kann. Auch gibt es Plattformen, auf denen die Bilder von ständig am Körper getragenen Kameras gesammelt und aufbereitet werden, wodurch für den Nutzer ein digitales fotografisches Gedächtnis entsteht. Dass all das auch in den etablierten Social Networks geteilt und mitgeteilt werden kann, ist selbstverständlich.

Auch jenseits von *Wearable Computing*-Systemen[76] nimmt die Anzahl der Sensoren beständig zu. Zahlreiche Alltagsgegenstände werden vernetzt und mit Bild-, Ton- und Bewegungssensoren ausgestattet. Unter dem Stichwort *Smarthome* schreitet so z.B. die Automatisierung der eigenen vier Wände voran. Dabei werden durch die Zunahme von Sensoren immer mehr Daten produziert, die im und durch das Internet verarbeitet werden. Diese „schiere, immer monströser werdende Datenmenge erfordert viele neue und effiziente Filtermechanismen“[77], die „sich individuell auf jede Person in jeder Situation einstellen“[78].

Sascha Lobo sieht diese Instanz in den sozialen Medien, die in Zukunft eine Filterfunktion für sämtliche unserer Daten übernehmen werden. Insbesondere am Beispiel des sozialen Netzwerks Google Plus, das immer mehr mit Googles Kerngeschäft, der Suchmaschine, verzahnt wird, sei diese Entwicklung bereits heute zu beobachten. Unterschiedliche Google-Plus-Nutzer bekommen danach basierend auf ihren im Netzwerk hinterlegten Eigenschaften, Interessen und Kontakten unterschiedliche Suchergebnisse von Google präsentiert. Aufgrund der durch unzählige Sensoren produzierten Datenmenge entwickeln sich die sozialen Medien so zu sozialen

---

[75] Vgl. Melanie Swan: Sensor Mania! The Internet of Things, Wearable Computing, Objective Metrics, and the Quantified Self 2.0. In: Journal of Sensor and Actuator Networks (1) 2012, H. 3, S. 217-227.

[76] Wearable Computing bezeichnet das (Forschungs-)Feld, das sich mit tragbaren Computersystemen beschäftigt, die am Körper der Nutzer befestigt sind. Die Wearable Computers unterscheiden sich von anderen, mobilen Computern dadurch, dass die hauptsächliche Tätigkeit des Nutzers nicht die Benutzung des Computers selbst, sondern eine durch den Computer unterstützte Tätigkeit in der realen Welt ist. Ein Alltagsbeispiel wären z.B. Hörgeräte.

[77] Sascha Lobo: Die Aufgabe der sozialen Medien. In: Anda, Béla; Endrös, Stefan; Kalka, Jochen; Lobo, Sascha (Hrsg.): SignsBook – Zeichen setzen in der Kommunikation. Wiesbaden 2012, S. 244.

[78] Ebd.

Filtern.[79] So platziert Facebook schon lange die Beiträge von Kontakten, mit denen häufiger interagiert wird, an prominenterer Stelle als die von Kontakten, mit denen sich selten ausgetauscht wird. Die Empfehlungsalgorithmen des Online-Versandhändlers Amazon oder der Online-Videothek Netflix sind weitere Beispiele für derartige Filter.

Kritiker bemängeln, dass durch die zunehmende Personalisierung der Angebote eine *Filter Bubble* oder *Filterblase* entsteht, in der ein Nutzer nur noch mit Informationen behelligt wird, die er bereits kennt und die seiner Weltvorstellung entsprechen. Da die Plattformen die Filteralgorithmen nicht offenlegen und nicht ersichtlich ist, welche (Nutzer-)Daten erfasst und einbezogen werden, könne zudem nicht nachvollzogen werden, wie die präsentierten Ergebnisse zustande kommen.

2.3.4 Quantified Self – Die Vermessung des Selbst

In Verbindung mit den neuen Gerätetypen hat sich unter dem Schlagwort *Quantified Self* eine Bewegung entwickelt, deren Vertreter die neuen Techniken einsetzen, um sich selbst und gewisse Aspekte ihres Lebens zu vermessen. Die so genannten Quantified Selfer, Self-Tracker oder Lifelogger wollen durch die so gewonnen Daten andernfalls verborgen gebliebene Erkenntnisse über sich selbst erlangen und Optimierungsbedarf erkennen. Der Sammelbegriff Quantified Self wurde 2007 von den Technikjournalisten Kevin Kelly und Gary Wolf geprägt, die damals erkannten, dass sich immer mehr Methoden und Geräte zur Quantifizierung des eigenen Körpers und eigener Verhaltensweisen entwickelten. Als Anlaufstelle richteten sie die Webseite quantifiedself.com ein, auf der sich seitdem unter dem Motto ‚self knowledge through numbers' über Trends und Entwicklungen der Selbstvermessung ausgetauscht und informiert werden kann und über die Treffen für Interessierte überall auf der Welt organisiert werden.[80]

In den weltweiten ‚Meetups' tauschen die Selbstvermesser ihre Erfahrungen aus und vernetzen sich mit den Entwicklern und Anbietern von Self-Tracking-Produkten, von denen es indes, sowohl auf Software-, als auch auf Hardware-Seite, immer mehr gibt. Da sich viele Produkte an den Massenmarkt richten, sind diese oftmals „primär als Motivationsinstrumente konzipiert"[81] und setzen auf Gamification-Konzepte[82], indem

---

[79] Vgl. ebd., S. 242-246.

[80] Vgl. Christian Grasse, Ariane Greiner: Mein digitales Ich. Wie die Vermessung des Selbst unser Leben verändert und was wir darüber wissen müssen. Berlin 2013, S. 20-28; Alexandra Carmichael: Kevin Kelly on The History and Future of QS. In: Quantified Self.com, 14.10.2012, URL: http://quantifiedself.com/2012/10/kevin-kelly-on-the-history-and-future-of-qs/ (30.03.2014).

[81] Rudi Klausnitzer: Das Ende des Zufalls. Wie Big Data uns und unser Leben vorhersagbar macht. Salzburg 2013, S. 50.

[82] Gamification oder Gamifizierung bezeichnet die Verwendung spieltypischer Elemente in einem spielfremden Kontext, um Anwender so auf eine spielerische Art und Weise zu etwas zu motivieren.

sie ihre Nutzer miteinander in Wettbewerb treten lassen, deren Aktivitäten mit Punkten versehen und besondere Nutzerleistungen durch digitale Abzeichen aufwerten. In den Apps und auf den Webseiten der Plattformen werden die erfassten Werte ansprechend in Diagrammen, Graphen und Tag Clouds visualisiert und erleichtern dem Nutzer so die Interpretation der Daten.[83]

Allerdings meint Quantified Self nicht nur die Erfassung des eigenen Datenkörpers, sondern auch andere Faktoren im Leben des Self-Trackers: Gelesene Bücher und gesehene Filme können ebenso gezählt werden, wie Tastaturanschläge oder der eigene Stromverbrauch. Demnach prophezeit Kevin Kelly, dass dieses Jahrhundert „ein Marsch in Richtung Quantifizierbarkeit aller Aspekte – der externen wie der internen – unseres persönlichen Lebens“[84] werden wird. „In diesem Umfeld werden auch die auf den ersten Blick vielleicht trivial wirkenden Apps Teil eines größeren Bildes“[85], so Kelly. Was kurzfristig nur der Erfassung einzelner, zunächst belanglos erscheinender Daten dient, könnte langfristig zu den aufschlussreichen Anwendungen führen, wie sie derzeit unter dem Schlagwort *Big Data*[86] erdacht werden.[87] Zwar mag der quantifizierte Fitnesszustand eines einzelnen Self-Trackers zunächst nur für diesen selbst interessant sein, wenn diese Daten jedoch in einer großen Menge von möglichst vielen Menschen vorliegen, werden sie bzw. die Erkenntnisse, die sich daraus ziehen lassen, z.B. auch für das Gesundheitswesen attraktiv.[88] Nach Meinung der Quantified-Self-Vertreter sollte jedoch „immer nur das Individuum, von dem die Daten stammen“[89], Entscheidungen darüber treffen dürfen, wie die Daten verwendet werden. Dies gestaltet sich regelmäßig als problematisch: Wie bei vielen Social-Web-Anwendungen ist es auch im Bereich der Self-Tracking-Produkte bisweilen sehr schwierig, die Kontrolle über die eigenen Daten zu behalten. Die durch die Geräte erfassten Daten werden in der Regel nur auf die Plattform des Herstellers übertragen, wo sie sich der Nutzerkontrolle entziehen. Häufig wird keine Exportfunktion angeboten oder ist erst nach Abschluss einer kostenpflichtigen Premium-Mitgliedschaft zugänglich. Zwar sind die Daten meist über die bereitgestellte Web API abrufbar, für den durchschnittlichen Anwender ist dies jedoch keine Option.

---

[83] Vgl. Grasse, Greiner 2013, S. 20-28; Klausnitzer 2013, S. 55-56.

[84] Klausnitzer 2013, S. 47.

[85] Ebd.

[86] Big Data beschreibt eine neue Generation der Datenverarbeitung. Sie beschäftigt sich mit derart großen Datenmengen, die konventionelle Datenbanksysteme an ihre Grenzen stoßen lassen und von deren bislang unmöglicher Verarbeitung man sich neue Erkenntnisse erhofft. Vgl. ebd., S. 88-89.

[87] Vgl. Swan 2012, S. 247-248.

[88] In Großbritannien gibt es tatsächlich bereits Bestrebungen, Selbstvermessungs-Apps einzusetzen, um so das Gesundheitssystem zu revolutionieren. Vgl. Klausnitzer 2013, S. 47-48.

[89] Vgl. ebd.

3. Analyse und Konzeption

Nachdem herausgestellt wurde, welche Datenfülle durch Smartphones, Tablets und Tracking-Devices entsteht und von den diversen Plattformen gespeichert wird, soll nun ein Mittel gesucht werden, mit dem ein Nutzer die Kontrolle über seine Daten zurückerlangen kann. Zunächst soll dazu definiert werden, was genau das *digitale Ich* eines Nutzers ausmacht und welche Nutzerdaten bewahrt werden sollen. Danach wird zur Veranschaulichung ein Nutzungsszenario entworfen, in dem dargelegt wird, wie eine typische Verwendung verschiedener Social-Web-Dienste aussehen könnte. Ausgehend davon werden bereits bestehende Lösungen betrachtet, mit denen ein Nutzer die gewünschten Daten aus den diversen Diensten exportieren kann. Inwieweit diese sich für das eingangs formulierte Ziel eignen, wird anschließend diskutiert, wobei die eigene Zielsetzung konkretisiert wird.

3.1 Das digitale Ich – Was soll gespeichert werden?

Wie bereits zu Beginn angeführt, ist es das Ziel dieser Arbeit, das digitale Ich in Form einer auf ihn bezogenen Programmierschnittstelle abzubilden. Dazu stellt sich nun die Frage, welche der Nutzerdaten für den vorliegenden Kontext von Belang sind und was hier folglich als digitales Ich zu verstehen ist.

3.1.1 Begriffsklärung

Beim digitalen Ich handelt es sich nicht um einen feststehenden und definierten, jedoch um einen vielfach verwendeten Begriff: Das digitale Ich ist Namensgeber von Fernseh-diskussionsrunden[90], Reportagen[91], Büchern[92], Kapiteln[93], Artikeln[94] und Broschüren[95]. Hierbei ist zu sehen, dass man als digitales Ich im weitesten Sinne die Repräsentation einer Person im Digitalen meint. Der Ausdruck taucht 2007/08 erstmals in deutsch-sprachigen Online-Medien auf und bezieht sich zunächst auf die Netzidentität eines

---

[90] Vgl. BECKMANN, D 1999-2014: DAS DIGITALE ICH - (ÜBER)LEBEN IM DATENDSCHUNGEL, R: Thomas Sohns, Erstausstrahlung: 12.09.2013.

[91] Vgl. DAS DIGITALE ICH - COMPUTER, MENSCHEN, EMOTIONEN, A 2013, R: Hannes M. Schalle.

[92] Vgl. Christian Grasse, Ariane Greiner: Mein digitales Ich. Wie die Vermessung des Selbst unser Leben verändert und was wir darüber wissen müssen. Berlin 2013.

[93] Vgl. Selbstverwirklichung im Netz: Das digitale Ich. In: Peter Kemper, Alf Mentzer, Julika Tillmanns: Wirklichkeit 2.0. Medienkultur im digitalen Zeitalter. Stuttgart 2012.

[94] Vgl. Christian Stöcker: Identität im Netz: Das digitale Ich liegt in Scherben. In: Spiegel Online, 29.07.2008, URL: http://www.spiegel.de/netzwelt/web/identitaet-im-netz-das-digitale-ich-liegt-in-scherben-a- 567899.html (30.03.2014)

[95] Vgl. Christina Quast: Im Blickpunkt: Das digitale Ich (Broschüre des Grimme Instituts). Oktober 2012, URL: http://www.grimme-institut.de/imblickpunkt/pdf/IB-Das-digitale-Ich.pdf (30.03.2014)

Online-Nutzers, also die Darstellung und Wahrnehmung seiner Person im Web.[96] Indes wird dabei stets herausgestellt, dass sich dieses selbst konstruierte digitale Ich eines Online-Nutzers von dessen tatsächlicher Persönlichkeit unterscheiden kann, um sich so z.B. möglichst positiv oder interessant zu präsentieren.[97] In wissenschaftlichen Disziplinen wie der Pädagogik, der Psychologie und der Soziologie, die sich auch mit der Identitätsbildung im Internet beschäftigen, scheint das Schlagwort des digitalen Ichs jedoch nur selten bis gar nicht verwendet zu werden.[98]

Eine weitere Bedeutung des Begriffs bildete sich erst später heraus und ist vor allem bei Diskussionen um die Privatsphäre und den Datenschutz von Nutzern gegenüber kommerziellen Anbietern wie Google und Facebook oder neuerdings Geheimdiensten allgegenwärtig. Hier versteht man als digitales Ich die Gesamtheit der Daten, die jemand bewusst oder unbewusst im Web erzeugt und durch die ein digital gespeichertes Abbild von dessen Äußerungen, Interessen und Kontakten entsteht. Im Zusammenhang mit Quantified Self meint digitales Ich schließlich die Ansammlung der gemessenen (Körper-)Daten einer Person.[99]

3.1.2 Abgrenzung

In einer Broschüre des Grimme Instituts über das digitale Ich heißt es, dass „[Social Media-Profile] das Herz des digitalen Ichs sind“[100]. Diese sollen auch im Rahmen dieser Arbeit im Mittelpunkt stehen. Wann immer im folgenden vom digitalen Ich die Rede ist, wird sich demnach auf die Gesamtheit der Daten und Inhalte bezogen, die ein Einzelner bewusst und absichtlich im Web erzeugt hat. Dies können im weitesten Sinne kreative Leistungen, wie z.B. auf Facebook veröffentlichte Beiträge und Statusmitteilungen von Twitter, oder absichtlich preisgegebene personenbezogene Primärdaten, wie der eigene Aufenthaltsort aus Foursquare oder Quantified-Self-Daten, sein. Also all das, was man als das ‚digitale Schaffen‘ einer Person bezeichnen könnte.[101]

---

[96] Eigene Recherche. Während sich die erste genuine Nennung durch eine zeitlich eingegrenzte Google-Suche bereits 2003 in einer Pressemitteilung findet, taucht der Begriff erst ab 2007 und 2008 regelmäßig in deutschen Online-Medien auf, vgl. Stöcker, 2008; Red Dot: Design Team of the Year 2003: Nokia Design Team. In: Red Dot Design Award, 2003, URL: http://red-dot.de/pd/celebration/design-team-of-the-year-2003-nokia-design-team/ (30.03.2014). Die Suche nach ‚digitales ich‘ und ‚digitale ich‘ bei Google Trends bestätigt ein Aufkommen des Begriffs in 2007/2008, da erst seitdem entsprechende Suchanfragen an die Google-Suche gestellt werden; vgl. Google Trends, URL: http://www.google.de/trends/explore#q=digitale%20ich%2C%20digitales%20ich (30.03.2014).

[97] Vgl. Quast, 2012, S. 1.

[98] Eigene Recherche, basierend auf Buchtiteln in den genannten Disziplinen.

[99] Vgl. Grasse, Greiner 2013, S. 20-24.

[100] Quast 2012, S. 1.

[101] Vgl. Lobo: Die Mensch-Maschine: Euer Internet ist nur geborgt, 2012.

Bloße Metadaten und nicht öffentliche Nutzerdaten, z.B. aus privaten Mitteilungen, werden nicht berücksichtigt. Auch qualitative Maßstäbe sollen an dieser Stelle keine Rolle spielen. Ob und welche nutzergenerierten Inhalte besonders bedeutend sind und daher bewahrt werden sollen, kann und soll in dieser Arbeit nicht entschieden werden. Vielmehr wird in diesem Kontext vorausgesetzt, dass es sich beim digitalen Ich einer Person generell um etwas handelt, das unabhängig von kommerziellen Interessen erhalten werden soll.

3.2 Nutzungsszenario

Da es unmöglich ist, sämtliche der hier bereits genannten (und nicht genannten) Social-Web-Anwendungen in der Konzeption und Realisierung eines eigenen Werkzeugs zu berücksichtigen, sollen nun die Plattformen eingegrenzt werden, die für den weiteren Verlauf dieser Arbeit relevant sein werden. Um nicht ständig alle Eventualitäten aufgreifen zu müssen, soll hier stellvertretend mithilfe eines anschaulichen Szenarios beschrieben werden, wie die Nutzung ausgewählter Angebote durch einen fiktiven Nutzer aussehen könnte. Die Wahl der Dienste soll dabei möglichst viele der zuvor beschriebenen Formen (Weblogs, soziale Netzwerke und Social-Sharing-Dienste) und Entwicklungen (‚klassische' Netzwerke, ortsbasierte Dienste, ausschließlich per App nutzbare Plattformen und Quantified-Self-Anwendungen) einbeziehen. Die primär genutzten Dienste sind hervorgehoben.

Der Nutzer N ist 30 Jahre alt. Da er technisch interessiert ist und sich gerne mit anderen austauscht, betreibt er ein eigenes, durch WordPress betriebenes ***Weblog****. Nach Lektüre der ‚Famous 5-Minute Install'*[102] *war die Installation von WordPress auf seinem eigenen, im billigsten Tarif angemieteten Webspace kein Problem mehr. Obwohl N bei* ***Facebook*** *angemeldet ist und dort vor allem mit Freunden und Bekannten aus seinem Ort und seiner Schulzeit befreundet ist, nutzt N das weltgrößte Social Network nur ungern, da er um seine Privatssphäre fürchtet und die Plattform zudem unübersichtlich findet. Stattdessen verwendet N lieber andere Social-Web-Dienste, allen voran den Microblogging-Dienst* ***Twitter****. Dort teilt er gerne sehens- und lesenswerte Links mit seinen persönlich oftmals unbekannten Followern. Mit seinem Smartphone aufgenommene Schnappschüsse stellt N ab und zu über die Foto-Sharing-App* ***Instagram*** *online. Über* ***Foursquare*** *teilt er seinen Freunden zudem regelmäßig seinen Auf-*

---

[102] Siehe Wordpress' Famous 5-Minute Install, URL: https://codex.wordpress.org/Installing_WordPress#Famous_5-Minute_Install (30.03.2014).

enthaltsort mit, um sich schnell mit diesen treffen zu können, sollten diese in der Nähe sein. Zudem hat N sich über Amazon einen digitalen Schrittzähler von ***Fitbit*** *gekauft, da er in einem Büro tätig ist und so einen Überblick darüber erhalten möchte, ob er sich in der Arbeitszeit ausreichend bewegt hat. Darüber hinaus ist N leidenschaftlicher Filmfan und hält die von ihm gesehenen Filme mit Sichtungsdatum und Bewertung in einer Excel-Liste fest.*

3.3 Bestehende Lösungen zur Sicherung der Nutzerdaten

Um tatsächlich im Besitz der eigenen, bei den diversen Diensten hinterlegten Daten zu bleiben, ist es ratsam, eine Kopie dieser anzulegen. Unabhängig von den Strategien und Techniken zur Datensicherung, mit denen sich die Informatik als Forschungsgegenstand beschäftigt, können bei der Sicherung von Nutzerdaten aus dem Social Web generell zwei Lösungsansätze beobachtet werden:

1. Die **lokale Datensicherung**, bei der ein Nutzer aktiv ein Archiv mit seinen Daten herunterlädt und dieses lokal auf seinem Rechner speichert.
2. Die **webbasierte Datensicherung**, bei der die Nutzerdaten über die API einer Plattform abgerufen werden und fortlaufend, ohne Zutun des Nutzers, auf einem Server im Web gespeichert werden.

Die lokale Datensicherung, bei der ein Nutzer über eine entsprechende Webseite einfach per Mausklick an eine Kopie seiner Daten gelangt, ist zwar auch für durchschnittliche Nutzer nachzuvollziehen, jedoch dahingehend problematisch, dass sie aktiv ausgeführt werden muss. Wenn Sicherheitskopien bestehen, sind sie so nur selten aktuell, weil der Nutzer den Dienst seit dem Herunterladen seines Datenarchivs wahrscheinlich weiterbenutzt hat. Die webbasierte Datensicherung hat den Vorteil, dass sie in Form eines entsprechenden Tools nur einmal eingerichtet wird und fortan die Daten eines Nutzers ohne dessen weitere Beteiligung automatisiert sichert. Zudem befinden sich die Daten nicht mehr nur auf dem vermeintlich unsicheren Rechner des Nutzers, sondern auf einem entfernten, wahrscheinlich sicheren Webserver. Allerdings handelt es sich bei derartigen Tools meist um Einzelfalllösungen für ausgewählte Plattformen, die selbst konfiguriert werden müssen und somit für die meisten Anwender kaum benutzbar sind.

Ausgehend von dem eben beschriebenen Nutzungsszenario sollen nun die Lösungen zur Datensicherung bei den von N verwendeten Angeboten betrachtet werden. Neben plattformeigenen, sollen dabei auch unterschiedliche Lösungen von Drittanbietern und alternative Ansätze vorgestellt werden.

3.3.1 Plattformeigene Lösungen

Die meisten Social-Web-Dienste bieten, sofern sie denn überhaupt eine Exportfunktion bereitstellen, ihren Nutzern das Herunterladen ihrer Daten in einem ZIP-Archiv[103] an, das diese auf ihrem Rechner lokal speichern und verwahren können. So auch die beiden größten Plattformen aus dem Nutzungsszenario, Facebook und Twitter, bei denen der Nutzer in den Diensteinstellungen ein Archiv seiner Daten anfordern kann und dann per Email darüber informiert wird, wann seine Inhalte heruntergeladen werden können. Die ZIP-Archive enthalten die Nutzerinhalte dabei in unterschiedlichen Dateiformaten. Die ausgegebenen Kopien sind jedoch aufgrund eher technischer Formate kaum zugänglich (JSON) oder nur schwer weiterverwertbar (HTML) und oft mit keinem anderen Angebot kompatibel. Facebook setzt ausschließlich auf HTML-Dateien, die sich zwar lokal öffnen und betrachten, aber nur schwer weiterverarbeiten lassen. Bei Twitter liegen die kopierten Tweets im JSON- und CSV-Format[104] vor, wobei sich die JSON-Dateien über eine mitgelieferte HTML-Seite lokal ansehen und sogar durchsuchen lassen. Bei Twitter eingestellte Bilder fehlen hingegen vollkommen. Eine Import-funktion, mit der sich die eigene Datensicherung wieder einspielen lässt oder Daten aus anderen Plattformen importiert werden können, sucht man bei den genannten Angeboten vergebens. Dies und die Tatsache, dass die Exportfunktionen beider Dienste erst mehrere Jahre nach deren Start hinzugefügt wurden[105], zeigt, dass es den Betreibern vorrangig um die werbewirksame Beruhigung ihrer Nutzerschaft geht und Datenexport und -sicherung keine allzu hohe Priorität genießen.

Die weiterhin von N genutzten Social-Sharing-Dienste Foursquare und Instagram stellen überhaupt keine Exportfunktion zur Verfügung. Sicherheitskopien der eigenen Check-Ins und Schnappschüsse lassen sich hier nur über Lösungen von Drittanbietern erstellen. Bei Fitbit, der Plattform der gleichnamigen Fitness-Tracking-Produkte, wird zunächst ebenfalls keine Exportfunktion bereitgestellt. Um die vom Fitbit-Schrittzähler gemessenen Schritte und Distanzen, sowie per Smartphone-App zusätzlich angegebene Daten, wie Gewicht, Kalorienzunahme oder Schlafenszeit, exportieren zu können, wird der Nutzer aufgefordert, eine kostenpflichtige Premium-Mitgliedschaft für 44,99 Euro pro Jahr abzuschliessen. Erst danach ist ein Download der eigenen Daten im CSV- und Excel-Format möglich.[106]

---

[103] ZIP ist ein Format zur Datenkompression, das zudem als Containerdatei fungiert, die mehrere Dateien speicherplatzsparend in einem so genannten ZIP-Archiv zusammenfassen kann.

[104] CSV (‚Comma-separated values') ist ein simples Textformat zur Darstellung von Tabellen.

[105] Bei Facebook wurde die Exportfunktion erst Mitte 2012, also acht Jahre nach dem Start des Dienstes, hinzugefügt. Bei Twitter wurde sie Ende 2012, also sechs Jahre nach dem Start, hinzugefügt.

[106] Siehe Abb. 3 auf S. 33.

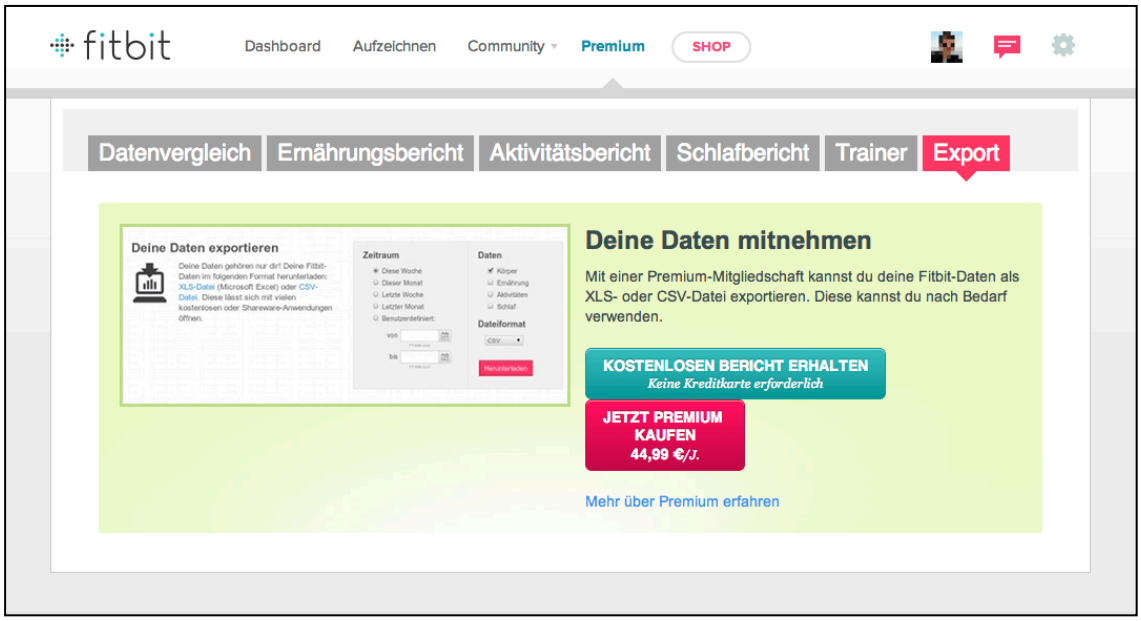

Abbildung 3: Screenshot der Fitbit-Webseite, auf der Nutzer ihre bei Fitbit hinterlegten Daten als Excel- oder CSV-Datei herunterladen können – sofern sie eine kostenpflichtige Premium-Mitgliedschaft abschliessen; URL: https://www.fitbit.com/premium/export (30.03.2014).

3.3.2 Lösungen von Drittanbietern

Für zahlreiche Social-Web-Dienste, die keine eigene Exportfunktionalität anbieten oder lange keine angeboten haben, gibt es Lösungen von Drittanbietern, die durch Desktop-, Mobile- oder Web-Anwendungen das Sichern der Nutzerdaten ermöglichen. Sämtliche dieser Lösungen bauen auf den Web APIs der einzelnen Dienste auf und nutzen diese, um die gewünschten Daten zu extrahieren. Oft können die Daten, wie bei den plattform-eigenen Exportlösungen heruntergeladen und lokal gespeichert werden, oftmals sehen sie aber auch eine webbasierte Datensicherung vor. Die Drittanbieterlösungen reichen dabei von Open-Source-Projekten einzelner Entwickler bis hin zu umfassenden Dienst-leistungen eigenständiger Unternehmen.

Ein Beispiel für die Implementierung eines einzelnen Software-Entwicklers ist etwa Instaport[107], das nach Authentifizierung des Anwenders durch Instagram, den Download der beim Foto-Sharing-Dienst veröffentlichten Bilder ermöglicht. Während es sich hierbei um ein Webseite handelt, die eine lokale Datensicherung ermöglicht, erlaubt TweetNest[108], das ebenfalls von einem einzelnen Programmierer entwickelt wurde, die webbasierte Datensicherung. Das Tool muss dazu vom Anwender auf einem eigenen Webserver installiert werden und ruft anschließend in einem festgelegten Zeitintervall dessen Statusmitteilungen von Twitter ab und speichert sie in einer eigenen Datenbank.

---

[107] Instaport von Florian Brandel, URL: http://instaport.me/ (30.03.2014).

[108] TweetNest von Andy Graulund, URL: http://pongsocket.com/tweetnest/ (30.03.2014).

Da vor allem Lösungen wie TweetNest für gewöhnliche Social-Web-Nutzer keine Option darstellen, haben sich mittlerweile Unternehmen am Markt positioniert, die für eine bestimmte Gebühr die Nutzerdaten diverser Plattformen sichern. Die Firma Social Safe[109] bietet so z.B. eine Desktop-Anwendung für Mac OS X und Windows an, mit der Nutzer ihre Daten aus diversen Social Networks auf ihren Rechner kopieren können. Der jährlich zu zahlende Beitrag berechnet sich dabei nach der Anzahl der verwendeten Accounts. Nicht lokal, sondern auf den firmeneigenen Servern speichert der ebenfalls gebührenpflichtige Anbieter Frostbox[110] die Daten *seiner* Nutzer, die unter anderem aus Facebook, Twitter, Instagram und Foursquare importiert werden können. Für N ließen sich so zwar die Daten seiner Social-Media-Profile (teils kostenpflichtig) sichern, aber gerade neuere und vor allem Quantified-Self-Dienste, wie eben Fitbit, werden von den Drittanbietern nur selten oder gar nicht unterstützt.

3.3.3 Alternative Lösungsansätze

Da die bisher beschriebenen Lösungen zur Datensicherung oft unzureichend, umständlich oder sogar mit Kosten verbunden sind, haben sich mittlerweile, vor allem aus der so genannten ‚Netzgemeinde'[111] heraus, einige andere Lösungsansätze herausgebildet, von denen im folgenden drei vorgestellt werden sollen.

3.3.3.1 IndieWeb

Das *IndieWeb* stellt einen Gegenentwurf zu den kommerziellen Social-Web-Plattformen dar. Seinen Ursprung fand das IndieWeb 2011 im IndieWebCamp, einer von den US-amerikanischen Informatikern Tantek Çelik und Aaron Parecki, sowie der Cyborg-Anthropologin Amber Case veranstalteten Konferenz, deren Ziel es war, Online-Nutzer dazu zu ermutigen, ihre Inhalte grundsätzlich auf ihren eigenen Webseiten zu veröffentlichen, anstatt sie ausschließlich in den geschlossenen Ökosystemen der diversen Dienste einzustellen. Mittlerweile umfasst die IndieWeb-Bewegung bereits mehrere hundert Entwickler weltweit, die eigene Standards definieren und meist quelloffene Software herausbringen. Über Webmentions, einer Fortentwicklung der von Weblogs

---

[109] Social Safe, URL: http://www.socialsafe.net/ (30.03.2014).

[110] Frostbox, URL: http://www.frostbox.com/ (30.03.2014).

[111] ‚Netzgemeinde' meint zum einen die Gesamtheit der Online-Nutzer, zum anderen diejenigen, die sich aktiv mit Netzthemen auseinandersetzen, z.B. in ihren Weblogs oder Podcasts. Sascha Lobo spricht bei letzterem von einer „Hobby-Lobby für das freie und offene Internet", die in Deutschland aus „vielleicht dreißigtausend Leute[n]" besteht. Sascha Lobo: Abschied von der Utopie: Die digitale Kränkung des Menschen. In: FAZ.NET, 11.01.2014, URL: http://www.faz.net/aktuell/feuilleton/debatten/abschied-von-der-utopie-die-digitale-kraenkung-des-menschen-12747258.html, S. 3 (30.03.2014).

bekannten Ping- und Trackbacks, sollen Online-Nutzer etwa auch jenseits der geschlossenen Systeme von Facebook oder Twitter darüber informiert werden, wenn sie von anderen Nutzern auf deren Webseiten adressiert werden. Durch die Verwendung von Mikroformaten[112] sollen Webinhalte einfach ausgelesen und auf anderen Webseiten eingebunden, also problemlos geteilt werden können.

Die IndieWeb-Systeme, die bisher nur als einzelne Komponenten und nicht als umfassendes Softwarepaket zur Verfügung stehen, arbeiten dabei so, dass Nutzer ihre Inhalte zunächst über die auf ihrem eigenen Server laufende IndieWeb-Anwendung auf ihrer persönlichen Webseite veröffentlichen und diese anschließend zu den großen Plattformen syndiziert werden. Dieses als *‚Publish (on your) Own Site, Syndicate Elsewhere‘* oder kurz *POSSE* beschriebene Veröffentlichungsmodell ermöglicht es den IndieWeb-Vertretern einerseits selbst in Besitz und Kontrolle ihrer Daten zu bleiben, andererseits aber auch die Vorteile der Social-Web-Plattformen zu nutzen, indem die dortigen Freunde und Follower erreicht werden können. Die Kommentare auf die syndizierten Inhalte werden wiederum von vielen IndieWeb-Implementierungen über die Web APIs der Plattformen abgefragt, abgespeichert und auf der eigenen Webseite angezeigt.[113]

3.3.3.2 Reclaim Social Media

Reclaim Social Media, ein Projekt, das Sacha Lobo zusammen mit dem Blogger Felix Schwenzel auf der Web-2.0-Konferenz Re:publica im Februar 2012 vorstellte, verfolgt einen etwas anderen Ansatz als die IndieWeb-Systeme. Hier sollen die Inhalte weiterhin auf den großen Plattformen produziert, danach aber automatisiert auf den eigenen Server kopiert werden. Dies wird analog zum POSSE-Modell als *‚Publish Elsewhere, Syndicate (to your) Own Site‘*, kurz *PESOS*, bezeichnet.[114]

Technisch umgesetzt wird Reclaim Social Media als Plugin für die freie Weblog-Software WordPress. Aktuell ermöglicht das Plugin unter anderem das Importieren von Daten aus Facebook, Twitter, Instagram und Foursquare. Zudem können Daten aus Moves, einer Smartphone-App importiert werden, die als Schrittzähler fungiert. Andere Quantified-Self-Dienste werden bislang nicht unterstützt, auch ist es bisher nicht möglich selbst Dienste hinzuzufügen. Darüber hinaus ist die Umsetzung als WordPress-

---

[112] Durch Mikroformate können Informationseinheiten in HTML-Dokumenten semantisch ausgezeichnet werden und sind so für Mensch und Maschine lesbar. Siehe auch http://microformats.org/ (30.03.2014).

[113] Vgl. Ben Werdmuller: The IndieWeb as a minimally viable social web ecosystem. In: W3C Open Social, 08.08.2013, URL: http://www.w3.org/2013/socialweb/papers/MVP.html (30.03.2014).

[114] Siehe auch Reclaim Social Media, URL: http://reclaim.fm/ (30.03.2014).

Plugin in soweit problematisch, dass die gesicherten Nutzerdaten außerhalb der eigenen WordPress-Installation ohne weiteres nicht nutzbar und weiterzuverarbeiten sind, wie Schwenzel auch selbst in seinem Weblog festhält: „Reclaim ist eine Einbahnstraße: aus dem Silo raus, auf die eigene Website. Fertig.“[115]

3.3.3.3 Personal API von Naveen Selvadurai

Naveen Selvadurai, der 2009 den Dienst Foursquare mitbegründete, beschreibt im Mai 2013 auf seiner privaten Webseite einen Ansatz, der ähnlich dem PESOS-Modell Inhalte von diversen Plattformen aggregiert, die Daten jedoch nicht wie Reclaim Social Media nur für die eigene Implementierung bereithält, sondern in Form einer Web API für zahlreiche Anwendungsfälle zugänglich macht. Selvadurai, der sich als interessierter Self-Tracker ausschließlich auf Quantified-Self-Daten bezieht, über seine Motivation:

> So far, I’ve used various tools and hacks over the years to collect this data. But I’ve long wanted it all in one place – or, at least, something to give me the illusion of ‘one place’. A dataset that is a single repository and view of my body as opposed to various silos of data scattered across different services and devices.[116]

Ziel dieses Ansatzes soll es zum einen sein, sämtliche erfassten (Körper-)Daten an einer Stelle, die unabhängig von den jeweiligen Plattformbetreibern und Produkteherstellern ist, zu sammeln bzw. zu sichern und eine zentrale Anlaufstelle zur Verarbeitung dieser Daten zu schaffen. Besonderen Reiz sieht Selvadurai darin, dass die Daten so potentiell von anderen Nutzern verarbeitet werden können, die daraus vielleicht neue Erkenntnisse ziehen können oder sogar Anwendungen auf Basis seiner Person entwickeln.

Eine solche personenbezogene Programmierschnittstelle bezeichnet Selvadurai als *Personal API*, einen Terminus, dessen Erfindung er seinem Foursquare-Kollegen Eric Friedman zuschreibt. Selvadurais Personal API ist unter der Subdomain api.naveen.com erreichbar, wo sich zunächst eine simple Dokumentation der verfügbaren Funktionen und Ressourcen findet. Danach umfasst seine API bisher seinen quantifizierten Schlaf, seine georeferenzierten Aufenthaltsorte, seine Schritte, sein Gewicht, sowie einen aus all diesen Werten berechneten Aktivitätsindex. Die Daten können nach Tagen sortiert abgerufen werden und stammen von Fitbit, Foursquare, der vernetzten Waage Withings, sowie dem Fitness-Tracking-Armband Nike Fuelband.

---

[115] Felix Schwenzel: Reclaim Social Media – FAQ. In: Wirres.net, 25.02.2014, URL: http://wirres.net/article/articleview/7082/1/6/ (30.03.2014).

[116] Naveen Selvadurai: A Personal API. In: Naveen Selvadurais Weblog, 28.05.2013, URL: http://x.naveen.com/post/51808692792/a-personal-api (30.03.2014).

Um z.B. die täglich gelaufenen Schritte abzufragen, wird die folgende URI aufgerufen:

```
http://api.naveen.com/v0/steps
```

Die Anfrage wird vom Server im JSON-Format beantwortet, wobei jedes Objekt neben der Anzahl der Schritte (`value`), das dazugehörige Datum (`date`) und eine einzigartige Kennung (`id`) enthält.[117] Die maschinenlesbare Repräsentation ermöglicht es nun, die Daten algorithmisch auszuwerten oder grafisch ansprechend, z.B. als Diagramm, aufzubereiten.

```
01 [
02     {
03         "date": "2013-04-23T04:00:00",
04         "id": "5180340036ee3f000c088027",
05         "value": 3726
06     },
07     {
08         "date": "2013-04-22T04:00:00",
09         "id": "5180b68e36ee3f0014088027",
10         "value": 13819
11     },
12     {
13         "date": "2013-04-21T04:00:00",
14         "id": "5180b69a36ee3f0019088027",
15         "value": 5834
16     },
17     ...
18 ]
```

Beispiel 1: JSON-Repräsentation der Schritte von Naveen Selvadurai.

Für Selvadurais Idee und Implementierung einer Personal API gab von der ‚Netzgemeinde' und der Fachpresse sehr viel positiven Zuspruch. In der Folge setzten einige andere Entwickler eigene, personenbezogene APIs um und in Beiträgen zum Thema wird schnell deutlich, dass eine derartige Programmierschnittstelle auch für eine breitere Nutzerschaft interessant wäre. „Let's get a version of this available for everyone"[118] heißt es etwa in einem Artikel über Selvadurais API. Jedoch hat bislang weder er selbst, noch ein anderer Entwickler eine umfassende, frei zugängliche Lösung veröffentlicht, mit der sich eine Personal API ohne Entwicklungsaufwand umsetzen ließe.[119]

---

[117] Siehe Beispiel 1.

[118] Andrew Cross: The Personal API. In: Medium, 13.06.2013, URL: https://medium.com/better-humans/f9fba15cdcb2 (30.03.2014).

[119] Siehe auch http://api.naveen.com/ (30.03.2014).

3.4 Zielsetzungen einer eigenen Lösung

Während Naveen Selvadurais Personal API deutlich zeigt, dass sich Web APIs hervorragend dazu eignen, unterschiedliche Arten von Nutzerdaten – über den simplen Aufruf einer URI – zugänglich zu machen, wurde in Kapitel 2.2 bereits aufgezeigt, dass das Hinzufügen von Daten durch Web APIs ebenso einfach ist: Über den Aufruf einer URI und die Übergabe bestimmter Parameter lassen sich bei einem REST-konformen System ohne großen Aufwand neue Nutzerdaten hinzufügen.

Die Umsetzung einer Lösung zur Sicherung von Nutzerdaten in Form einer Web API ist daher naheliegend. Dazu soll Selvadurais Konzept einer Personal API dahingehend erweitert werden, dass sie nicht nur die Quantified-Self-Daten des Nutzers umfasst, sondern sämtliche Daten, die dieser im Social Web veröffentlicht hat. Die auf den Nutzer bezogene Programmierschnittstelle soll so ein Abbild von dessen digitalem Ich darstellen. Die Personal API aggregiert dazu die Nutzerdaten aus mehreren externen APIs, normalisiert deren heterogene Datensätze und stellt sie über eine einheitliche Schnittstelle wieder zur Verfügung. Anstatt also wie bisher verschiedenartige APIs mit unterschiedlichen Authentifizierungsmechanismen einzeln abfragen zu müssen, können sämtliche Anfragen zur Abfrage von Nutzerdaten nun an eine einzige API gerichtet werden, wie auch die folgenden Grafiken (Abb. 4 und 5) verdeutlichen:

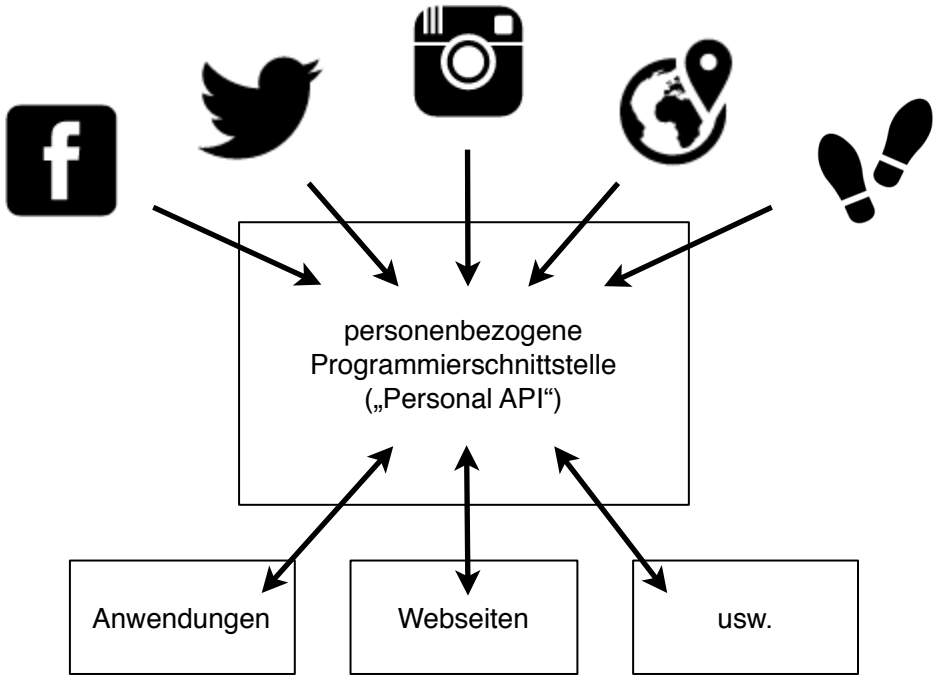

Abbildung 4: Position einer API, die Daten aus anderen APIs aggregiert. Die normalisierten Daten können anschließend über die aggregierende API angesprochen und verarbeitet werden. Ein Austausch mit den APIs, die die Daten ursprünglich bereitstellten, ist hier nicht vorgesehen.

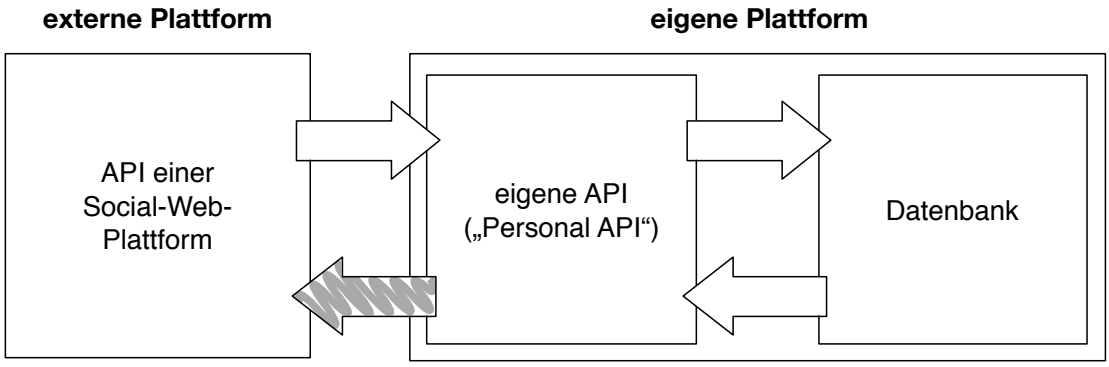

Abbildung 5: Aufgabe der eigenen API im Verhältnis zwischen einer externen Plattform und der eigenen Plattform. Der grau markierte Pfeil deutet darauf hin, dass es prinzipiell möglich wäre, dass durch die eigene API auch die externe Plattform mit Inhalten befüllt wird, dies ist hier zunächst jedoch nicht vorgesehen.

Zwar handelt es sich bei einer API um eine vergleichsweise abstrakte Lösung zur Sicherung und Verbreitung von Nutzerdaten, aufgrund der hohen Kompatibilität zu anderen Anwendungen ist sie jedoch ein besonders mächtiges Werkzeug. Andere Dienste und Programme können auf den bereitgestellten Schnittstellen aufbauen und so etwa automatisiert Daten einspielen oder auslesen. Diagrammatische Auswertungen lassen sich damit ebenso einfach umsetzen, wie GUI-Repräsentationen sämtlicher Daten und Funktionen der personenbezogenen Programmierschnittstelle.

3.4.1 Technische und ideelle Prinzipien

Damit die hier realisierte Personal API möglichst viele Nutzer anspricht, sollten bestimmte technische und ideelle Prinzipien berücksichtigt werden. Die Installation und Konfiguration des Tools soll so etwa auch für technisch weniger versierte Anwender möglich sein. Das Eingreifen im Programmcode soll in der Regel nur beim ersten Einrichten nötig sein, ansonsten soll jede Interaktion über das Front End stattfinden. Zudem soll das Tool problemlos auf einer Vielzahl von Webserver-Konfigurationen laufen, um möglichst überall eingesetzt werden zu können.

Die Personal API soll verschiedenste Social-Web-Plattformen und Anwendungen unterstützten. Daher ist ein modularer Aufbau immens wichtig. Die Kommunikation mit den externen Diensten soll über die Module in funktionale Einheiten zerlegt werden, die unabhängig voneinander funktionieren. Ein Modul soll dabei zum einen dafür zuständig sein, dass das Hauptprogramm weiß, wie es einen bestimmten Datentyp verwaltet und andererseits die Interaktion mit dem externen Dienst regeln. Interessierte Anwender sollen in der Lage sein, selbstständig eigene Module zu erstellen. Die Module sollen dabei beliebig zur API hinzugefügt und entfernt werden können. Die durch die Module

definierten Datentypen sollten dabei nach ihrem eigentlichen Inhalt, nicht nach der Plattform, auf der sie ursprünglich gespeichert wurden, oder dem Gerät, durch das sie zuvor ermittelt wurden, benannt werden. Wenn ein Modul etwa den Umgang mit Statusmitteilungen aus Twitter verwaltet, sollte der Name des Datentyps, der gleichzeitig auch der Name der API-Ressource ist, nicht ‚twitter' oder ‚tweets' sondern ‚statuses' sein. So hat der Nutzer die Möglichkeit mithilfe weiterer Module auch Statusmitteilungen von anderen Diensten unter ‚statuses' zu vereinen und hat sich tatsächlich von den großen Plattformbetreibern losgesagt. Aus Aufenthaltsorten von Foursquare werden so ‚places', aus den per Fitbit gemessenen Schritten ‚steps'. Wenn der Nutzer sich dann von Fitbit abwendet und seinen digitalen Schrittzähler wechselt, kann die Sicherung unter ‚steps' – nach Hinzufügen eines entsprechenden neuen Moduls – nahtlos fortgeführt werden.

Um mit den anderen Diensten kommunizieren zu können, speichert die Software natürlich die erforderlichen API-Schlüssel der jeweiligen Plattformen. Zudem enthält sie als webbasierte Datensicherung des Nutzers zahlreiche persönliche Daten. Um diese sensiblen Informationen zu schützen, muss die Personal API natürlich möglichst sicher gestaltet werden, um Unbefugten den Zugriff zu verwehren. Damit die gesicherten Daten über die API nicht von jedermann modifiziert werden können, bedarf es zudem einer Form der Authentifizierung. Diese soll bewusst möglichst simpel umgesetzt werden, da Nutzer bei Verwendung der Web APIs der kommerziellen Plattformen regelmäßig bereits daran scheitern, dass diese komplexe Sicherheitsmechanismen implementiert haben und die Inhalte nicht mehr über den simplen Aufruf einer URI abgefragt werden können.

3.4.2 Die Personal API im Nutzungsszenario

Auch für den fiktiven Nutzer N aus dem zuvor dargestellten Nutzungsszenario würde sich die soeben beschriebene Personal API zur Sicherung seiner Daten und Inhalte aus den verschiedenen, von ihm verwendeten Social-Web-Plattformen anbieten. N verfügt bereits über Speicherplatz auf einem Webserver, auf dem er sein Weblog per WordPress installiert hat, und es wäre ein einfaches auch die Personal API dort einzurichten. Da die Software niedrige Systemanforderungen hat und die Konfiguration ebenso einfach wie bei der berühmten Fünf-Minuten-Installation von WordPress ist, dürfte N damit keine Probleme haben. Nachdem er seine Datenbankinformationen in die Personal API eingetragen hat und sämtliche Dateien per FTP[120] auf den Webserver übertragen hat, wird er im Browser Schritt für Schritt durch die Einrichtung der API geführt.

---

[120] FTP (File Transfer Protocol) ist analog zu HTTP das Protokoll zur Übertragung von Dateien.

Anschließend muss er die einzelnen Module konfigurieren, die mit seinen Social-Web-Diensten kommunizieren. Durch die Anleitungen, die bei jedem Modul angezeigt werden, kann N schnell die benötigten API-Schlüssel besorgen und eintragen. Danach kann er die Personal API – immer noch im Browser – auffordern, seine aktuellen Daten von den Services abzuholen und in der eigenen Datenbank zu speichern. Über einen so genannten Cronjob, mit dem sich wiederkehrende Aufgaben auf einem Webserver automatisieren lassen, kann N die Aggregation seiner Daten für einen bestimmten Zeitintervall festlegen, so dass die Personal API bspw. zwei Mal pro Tag seine neuesten Inhalte abfragt und sichert.

Da alle von N verwendeten Dienste, namentlich Facebook, Twitter, Instagram, Foursquare und Fitbit, über offene Web APIs verfügen, hat N fortan eine automatische, webbasierte Datensicherung seiner Daten. Sollte N sich nun überlegen, dass er auch eine Kopie seiner Weblogeinträge in der Personal API speichern möchte, ließe sich dies über ein einfaches Modul erledigen, das die API nach Statusmitteilungen (aus Facebook und Twitter), Fotos (aus Instagram), Aufenthaltsorten (aus Foursquare) und Schritten (aus Fitbit) um einen zusätzlichen Datentyp für Weblogeinträge, z.B. ‚entries‘, erweitert. Auch ließen sich seine Amazon-Einkäufe oder die gesehenen und bewerteten Filme, die N bisher in einer Excel-Liste verwaltet hat, über Module problemlos zum Teil der API machen. Durch die Modularität der Software ist N auch für die nächsten Jahre Internetnutzung gewappnet, da sich auch neue, heute nichtmal erdachte Plattformen über ihre Web APIs stets in seine Personal API integrieren lassen.

4. Realisierung der Personal API

Bitte beachten Sie, dass es sich bei der vorliegenden Implementierung um einen Prototypen handelt. Teilweise sind die im folgenden beschriebenen Komponenten noch ziemlich *hacky*, was vor allem der kurzen Bearbeitungszeit der Bachelorarbeit geschuldet ist. Der hier beschriebene Prototyp – die v0.1, wenn Sie so wollen – sollte in dieser Form keinesfalls in einem Produktivsystem eingesetzt werden.

Die Implementierung der hier entworfenen Personal API wird im folgenden ausführlich dargestellt. Dabei werden ausgehend von den eben beschriebenen Konzepten jeweils die einzelnen Komponenten der Software betrachtet, wobei der Fokus schließlich auf der Web API selbst liegen wird.[121] Währenddessen wird immer wieder auf den Programmcode verwiesen, der selbst ausführlich kommentiert wurde und unter http://stefangrund.de/personalapi/code/ zum herunterladen und einsehen bereit steht. Anhand der laufenden Personal API des Autors werden zudem einzelne Funktionen der API erläutert. Diese findet sich unter http://api.stefangrund.de. Zur sinnvollen Ausführung des Programmcodes im Rahmen einer eigenen Personal API werden natürlich Accounts (und Inhalte) bei verschiedenen Social-Web-Plattformen benötigt. Hier werden

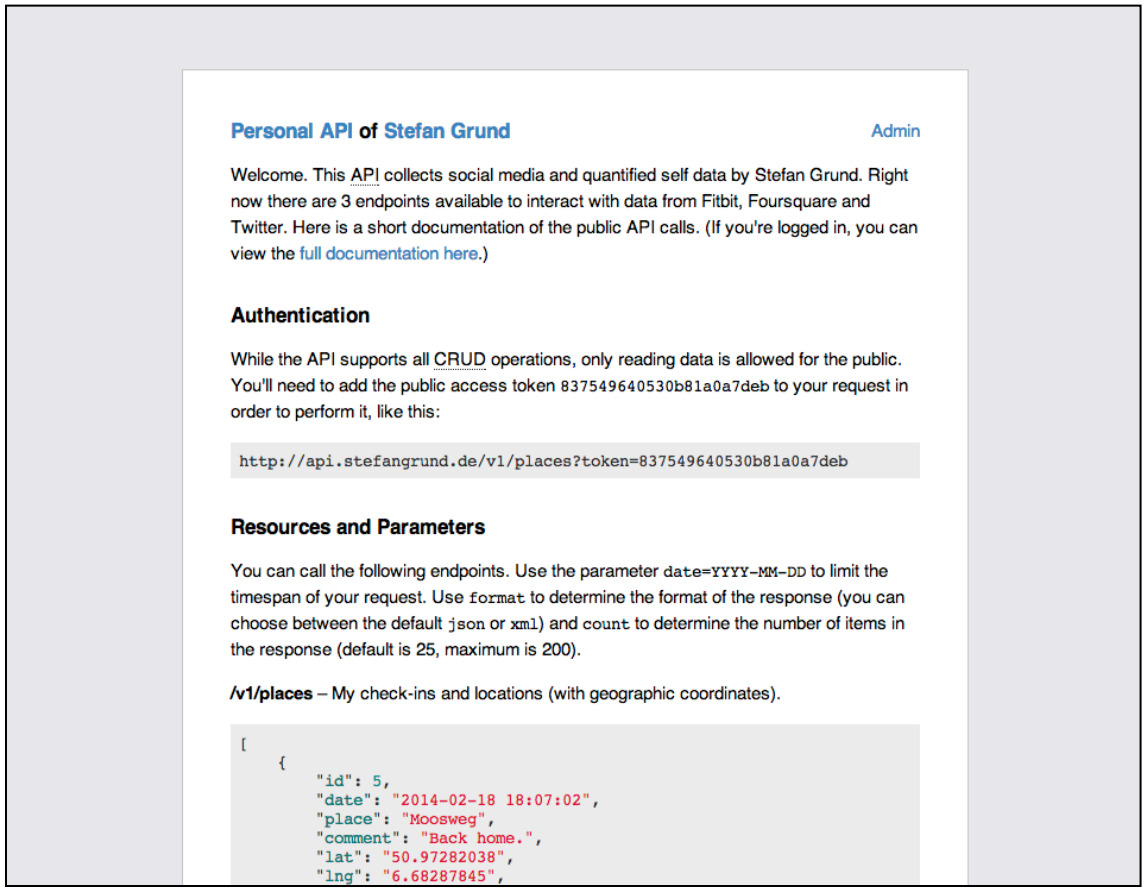

Abbildung 6: Screenshot der Personal API von Stefan Grund, URL: http://api.stefangrund.de/.

[121] Da die Personal API unter http://stefangrund.de/personalapi/code/ fortwährend verbessert und weiterentwickelt wird, können sich hier beschriebene Funktionen und Komponenten von der aktuellen Version des Programms unterscheiden.

dementsprechend die verschiedenen, je nach Plattform mehr oder weniger öffentlichen Profile des Autors verwendet.

4.1 Systemanforderungen

Da es ein erklärtes Ziel ist, dass die personenbezogene Programmierschnittstelle auch von technisch weniger versierten Anwendern eingesetzt werden kann und auf einer Vielzahl von Webserver-Konfigurationen laufen sollte, wurde die Personal API mit PHP umgesetzt, einer auf dynamische Webseiten spezialisierten, weit verbreiteten Skriptsprache, deren Skripte auf praktisch jedem Webserver laufen.[122] Zudem zeichnet sich PHP durch eine breite Unterstützung von Datenbanken aus, darunter auch MySQL, das hier ebenfalls aufgrund seiner hohen Verbreitung verwendet werden soll.

4.2 Installation

Auch der Installationsprozess ist möglichst einfach gehalten und orientiert sich an der ‚Famous 5-Minute Install' von WordPress, einem Content-Management-System für Weblogs, das ebenfalls auf PHP und MySQL setzt und unter anderem aufgrund der einfachen Installation mittlerweile von Millionen von Webseiten eingesetzt wird und so zum meist genutzten CMS geworden ist. Wie bei WordPress ist es auch bei der Personal API zunächst erforderlich, dass der Anwender seine Datenbankdaten und das Stammverzeichnis seiner API in einer Konfigurationsdatei (`/inc/config.php`) angibt. Anschließend lässt sich die Personal API über den Browser aufrufen, wo der Anwender durch die restlichen Schritte der Installation durchgeführt wird.

4.2.1 Installationsprozess

Die Installation gestaltet sich dabei folgendermaßen: Die Software überprüft zunächst – wie immer –, ob eine Verbindung zur angegebenen Datenbank hergestellt werden kann. Ist dies nicht möglich, wird der Anwender gebeten, seine Datenbankdaten in `/inc/config.php` nochmals zu überprüfen. Kann eine Verbindung hergestellt werden, wird überprüft, ob es bereits eine zum Projekt gehörende Datenbanktabelle in der Datenbank gibt. Ist dies der Fall, wird das User Interface im Browser dargestellt. Findet sich keine dazugehörige Tabelle, wird der Installationsprozess eingeleitet (`/inc/install_process.php`). Dabei werden die für das Programm erforderlichen Datenbanktabellen erstellt, ein Benutzer angelegt und die ersten Authentifizierungsschlüssel, die *Tokens*, generiert. Um Konflikten vorzubeugen, für den Fall, dass die zur Verfügung gestellte

---

[122] Vgl. Lorna Jane Mitchell: PHP Web Services. Sebastopol 2013, S. VII.

Datenbank auch von anderen Anwendungen verwendet wird, haben alle Datenbanktabellen der Personal API das Präfix `papi_`.[123]

4.2.2 Benutzererstellung

Neben den Datenbanktabellen für das Ereignisprotokoll, die Tokens und die Moduleinstellungen wird zu Beginn auch die Tabelle `papi_user` angelegt. In dieser wird der Benutzername und das Passwort des Anwenders, sowie ein öffentlich angezeigter Name und eine öffentlich angezeigte URL gespeichert. Während des Installationsprozesses wird der Anwender zur Eingabe dieser Daten aufgefordert. Das Passwort wird dabei aus Sicherheitsgründen lediglich als Hashwert in der Datenbank gespeichert. Dieser wird durch eine zufällige Zahlenfolge und die kryptologische Hashfunktion `bcrypt` generiert. Beim Einloggen wird versucht, diesen Hashwert mit dem eingegebenen Passwort zu entschlüsseln. Funktioniert dies, wird der Nutzer in den Administrationsbereich weitergeleitet, andernfalls wird er nicht eingeloggt. Da der Hashwert nur durch das Passwort selbst entschlüsselt werden kann, ist er für Angreifer wertlos und die Personal API ist selbst bei Verlust noch ausreichend geschützt.[124] Die Verschlüsselungsfunktion `encrypt()` findet sich in `/inc/global_functions.php`, in der sich die im ganzen Projekt immer wieder verwendeten Funktionen befinden.

Nachdem der Anwender seine Benutzerdaten eingegeben hat, ist die Installation abgeschlossen. Er wird nun aufgefordert, die Module zu konfigurieren, über die sich die Software mit den APIs der externen Plattformen austauscht. Wenn er dies bestätigt, wird er auf die entsprechende Seite im Administrationsbereich weitergeleitet und kann die Module entsprechend der dort angezeigten Instruktionen einrichten.

4.2.3 User Interface

Für die im Browser ablaufende Installation und den Administrationsbereich wurde ein einheitliches, minimalistisches Design gewählt, damit die vermittelten Informationen im Fokus stehen.[125] Dazu wurde ein simples Template-System eingerichtet, durch welches die Design bestimmenden Skripte `/inc/header.php` und `/inc/footer.php` in die einzelnen Seiten eingebunden werden. Die Webseiten sind *responsiv*, damit sich das User Interface der Personal API an die verschiedenen Bildschirmgrößen anpasst und

---

[123] Siehe http://stefangrund.de/personalapi/#video:1, das verschiedene Funktionen der Personal API, darunter auch die Installation, demonstriert.

[124] Vgl. Elbert F.: How to store passwords safely with PHP and MySQL. In: Alias.io, 25.04.2013, URL: http://alias.io/2010/01/store-passwords-safely-with-php-and-mysql/ (30.03.2014).

[125] Siehe Abb. 6 auf S. 42 und Abb. 7 auf S. 45.

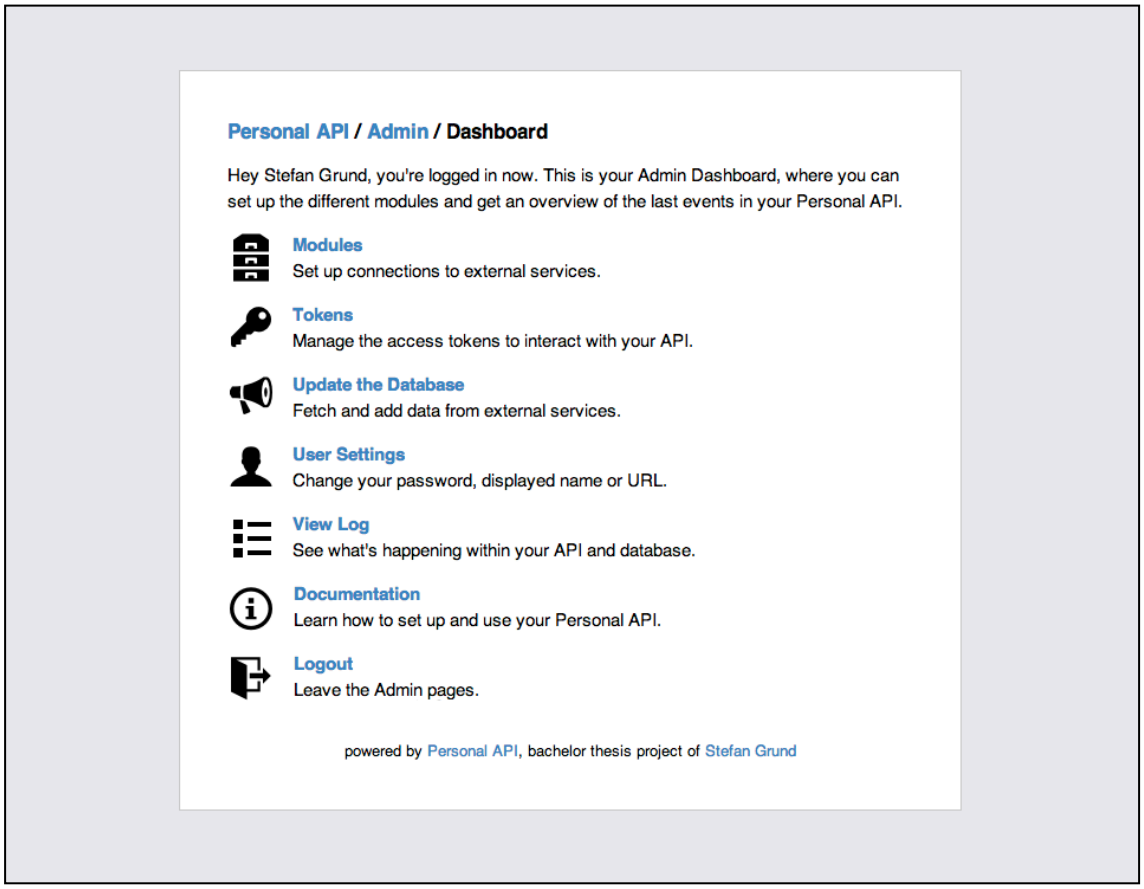

Abbildung 7: Screenshot des Administrationsbereichs, URL: http://api.stefangrund.de/admin/.

auch auf mobilen Endgeräten problemlos nutzbar ist.[126] Zur Gestaltung wurden valides HTML 5, valides CSS 3 und ein eigener Icon-Font verwendet. Die Animationen werden durch die freie JavaScript-Bilbiothek jQuery (in Version 1.11.0) realisiert, das Syntax-Highlighting durch die ebenfreie Bibliothek highlight.js.

4.3 Administration

Der Administrationsbereich ist neben der Startseite die einzige grafische Repräsentation der Personal API. Hier werden Module konfiguriert, Tokens verwaltet und die manuelle Aktualisierung der Datenbank angestoßen. Zudem können die während der Installation angegebenen Benutzerdaten geändert und ein Ereignisprotokoll eingesehen werden. Da die API an sich sehr abstrakt arbeitet, werden unter anderem durch dieses Ereignisprotokoll verschiedene Aktionen sichtbar gemacht. So wird z.B. jeder erfolgreiche und fehlgeschlagene Login-Versuch mit Usernamen und IP-Adresse oder jeder Aktualisierungsvorgang protokolliert und angezeigt.

[126] Siehe Abb. 8 auf S. 46.

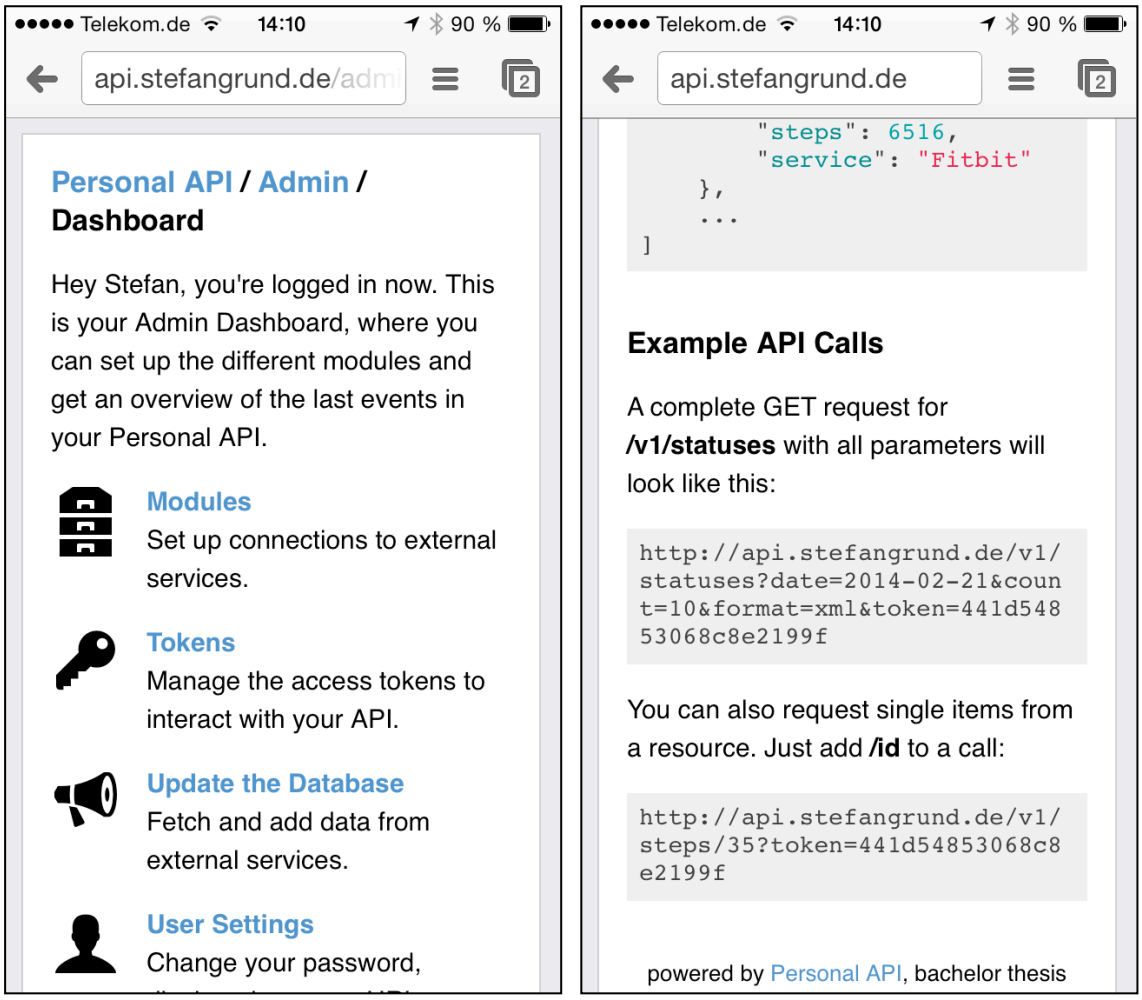

Abbildung 8: Beispiele des Responsive Webdesigns der Personal API auf dem iPhone.

Darüber hinaus beinhaltet der Administrationsbereich eine ausführliche Dokumentation sämtlicher Funktionen der Personal API und nicht nur die auf der öffentlich einsehbaren Startseite dokumentierten GET-Requests. Die Dokumentation wird dabei dynamisch generiert, so dass die angezeigten Tokens den tatsächlich erstellten Tokens entsprechen. Um Missbrauch abzuwehren, werden die Seiten im Administrationsbereich erst nach erfolgreichem Login und dann nur für den Verlauf einer Session angezeigt. Diese wird nach zehn Minuten Inaktivität automatisch beendet und erfordert danach eine erneute Anmeldung des Benutzers.

4.3.1 Module

Den Modulen werden zwei Aufgaben zuteil: Zum einen verwalten sie die unterschiedlichen Datentypen (Aufenthaltsorte, Statusmitteilungen, Schritte, ...), zum anderen regeln sie die Kommunikation mit den Social-Web-Diensten, die diese Daten bereitstellen. Ein einzelnes Modul ist damit in der Lage, die Personal API um die Unterstützung eines neuen Datentyps bzw. Dienstes zu erweitern. Die Idee ist, dass Anwender

bei Bedarf die Möglichkeit haben, eigene Module zu entwickeln und diese mit anderen Personal-API-Nutzern austauschen zu können.

4.3.1.1 Erkennen von Modulen

Die Module der Personal API befinden sich im Ordner `/modules`. Durch die Bezeichnung werden sowohl der Name des Datentyps als auch der Name der API-Ressource bestimmt. Der Ordner eines Moduls, das etwa die gemessenen Schritte von Fitbit verwalten soll, hieße nach den vorangegangenen Überlegungen[127] idealerweise ‚steps'. Ein Modul-Ordner muss ein PHP-Skript namens `datentyp_def.php` enthalten, in dem die Funktionen des Moduls definiert werden. Im Fall der Fitbit-Schritte wäre der Name also `steps_def.php`. Zudem sollte der Modul-Ordner alle Bibliotheken beinhalten, die benötigt werden, um mit der Web API eines externen Dienstes zu kommunizieren (meist sind das Bibliotheken, welche die Arbeit mit den Authentifizierungsmechanismen der Dienste erleichtern). Auch eine Beispieldatei (`datentyp_sample.json`), die zeigt, wie die JSON-Repräsentation des Datentyps aussieht, sollte enthalten sein (diese wird zur Veranschaulichung auch in die öffentliche Dokumentation eingebunden).[128]

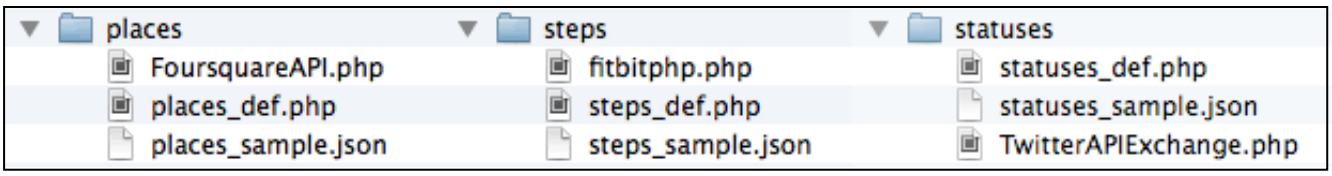

Abbildung 9: Modul-Ordner für verschiedene Datentypen. ‚places' verwaltet Aufenthaltsorte aus Foursquare, ‚steps' Schritte aus Fitbit und ‚statuses' Statusmitteilungen von Twitter. Die jeweiligen Modul-Ordner enthalten neben den Definitonen der Modul-Funktionen, Bibliotheken für den Austausch mit den externen Diensten (z.B. FoursquareAPI.php) und die Beispieldateien.

Wenn das Hauptprogramm wissen muss, welche Module installiert sind, wird `/modules` nach dem Muster `datentyp/datentyp_def.php` durchsucht und ein Array mit den gefundenen Modulen angelegt. In der Regel wird `/modules` jedoch nur dann durchsucht, wenn eine Aufzählung der vorhandenen Module benötigt wird, um etwa zu überprüfen, ob für diese bereits eine Datenbanktabelle erstellt wurde.[129] Bei Abfragen der API oder ähnlichem wird nicht jedes Mal überprüft, welche Module vorhanden sind, da nur die fertig konfigurierten Module, also die mit Datentypen, bei denen auch tatsächlich Daten vorliegen, bei der Ausgabe berücksichtigt werden.

---

[127] Siehe Kapitel 3.4.1, S. 39.

[128] Siehe Abb. 9.

[129] Die Erkennung der Module wird durch die `glob()`-Funktion innerhalb der `Modules`-Klasse realisiert.

4.3.1.2 Aufbau eines Moduls

Wenngleich es für das Funktionieren des Hauptprogramms nicht von Belang ist, ob ein Modul vorhanden ist oder wie das Modul selbst arbeitet, gibt es drei obligatorische Funktionen, die das Modul in `datentyp_def.php` bereitstellen muss, die ebenfalls jeweils mit dem Präfix `datentyp_` beginnen müssen (wobei datentyp_ weiterhin als Platzhalter für `places_`, `statuses_`, `steps_` usw. dient):

- **`datentyp_table()`** gibt ein Array mit der Definition des Schemas der Datenbanktabelle des Datentyps zurück, die bei der Konfiguration des Moduls angelegt wird.
- **`datentyp_settings()`** gibt ein Array mit den im Administrationsbereich benötigten Hilfetexten und Feldern für die Konfiguration des Moduls zurück und legt somit auch gleich fest, wie viele Variablen (API-Schlüssel, Benutzernamen, usw.) bei dem Modul gespeichert werden müssen.
- **`datentyp_saveData()`** ist für das Speichern neuer Nutzerdaten aus den externen Social-Web-Diensten zuständig und wird vom Hauptprogramm immer dann aufgerufen, wenn per Cronjob oder manuell ein Aktualisierungsvorgang gestartet wird. Wie das Modul die Daten dabei erhält, ist für das Hauptprogramm unbedeutend.

Damit innerhalb der Module der Austausch mit der API komfortabel möglich ist, ohne dass MySQL-Befehle geschrieben werden müssen, stehen zwei Funktionen bereit:

- **`doGetRequest($url)`** ermöglicht das Durchführen eines GET-Requests an eine beliebige, als einziger Funktionsparameter angegebene URI per cURL[130]. Der Rückgabewert der Funktion ist die JSON-Response des Servers als Array.
- **`doPostRequest($url, $parameters)`** führt analog dazu einen POST-Request an eine beliebige URI durch, wobei die POST-Parameter als zweiter Funktionsparameter angegeben werden.

Der angehängte Programmcode der Personal API enthält bereits drei Module, nämlich `places` für Aufenthaltsorte aus Foursquare, `statuses` für Statusmitteilungen aus Twitter und `steps` für Schritte aus Fitbit. Die drei Datentypen verfügen dabei über verschiedene Eigenschaften. Während es bei `places` um georeferenzierte Orte samt Breiten- und Längenangabe geht, stehen bei `statuses` kurze Textnachrichten und bei `steps` Zahlenwerte im Mittelpunkt. Zudem unterscheiden sich alle drei Dienste in ihren verwendeten Authentifizierungsmechanismen und folglich auch in der Konfiguration der Personal-API-Module.

---

[130] cURL (Client for URLs) ist eine Bibliothek zum Übertragen von Dateien per HTTP und Bestandteil vieler Betriebssysteme und Programmiersprachen.

4.3.1.3 Konfiguration eines Moduls

Damit es sich bei den Modulen nicht um Einzelfalllösungen einzelner Nutzer handelt und sie prinzipiell mit jeder Personal API zusammenarbeiten können, dürfen die für die externen Plattformen benötgiten API-Schlüssel etc. keinesfalls *hardgecoded*, also direkt in den Programmcode geschrieben werden, sondern müssen als Variablen behandelt werden, die von der jeweiligen Personal API verwaltet werden. Die Moduleinstellungen werden in der Tabelle `papi_modules` gespeichert[131] und können im Administrationsbereich unter ‚Modules' verwaltet werden.

4.3.2 Aktualisierung der Datenbank

Um die aktuellen Nutzerdaten aus den diversen Social-Web-Plattformen in der Personal API abbilden zu können, muss die Datenbank regelmäßig um diese neue Daten ergänzt werden. Der Aktualisierungsvorgang läuft dabei wie folgt ab: Das Skript `update.php` im Stammverzeichnis startet die Aktualisierung, indem es die Methode `runSaveJobs` der Klasse `Modules`, die für jeglichen Austausch mit den Modulen zuständig ist, aufruft. (1.) `runSaveJobs` durchläuft alle vorhandenen Module und ruft deren `saveData`-Funktionen auf. (2.) Die einzelnen `saveData`-Funktionen rufen unter Verwendung externer Bibliotheken über die Web APIs der externen Plattformen die aktuellen Nutzerdaten ab und überprüfen, ob sich die abgefragten Nutzerdaten bereits in der eigenen Datenbank befinden. Sollten sie bereits hinzugefügt worden sein, wird nichts gespeichert und `runSaveJobs` geht zum nächsten Modul über (zurück zu 1.). Befinden sich die Nutzerdaten noch nicht in der Datenbank, dann werden sie (3.) über die eigene API zum eigenen Datenbestand hinzugefügt. Daraufhin werden die Nutzerdaten der nächsten Plattform abgefragt (zurück zu 1.).[132]

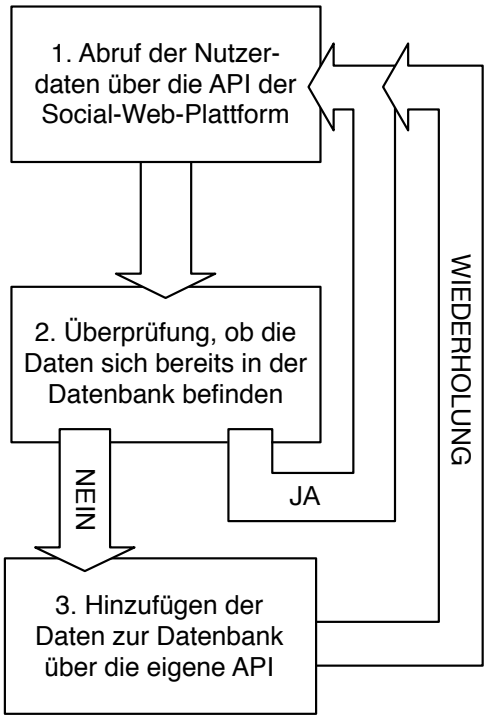

Abb. 10: Aktualisierungsvorgang.

[131] Da es von Modul zu Modul unterschiedlich viele Variablen sein können, werden die Modul-Variablen in der Datenbank als ein JSON-String gespeichert, der bei Bedarf wieder aufgelöst wird.

[132] Siehe Abb. 10.

Der Aktualisierungsvorgang kann auf zwei Wegen gestartet werden: Manuell über den Punkt ‚Update the Database' im Administrationsbereich und automatisiert durch einen Cronjob des Webservers. Dazu wird in beiden Fällen das Skript `/update.php` aufgerufen. Während der Skriptaufruf über den Administrationsbereich durch den Login geschützt ist, lässt es sich *von außerhalb* nur mit einem Authentifizierungsschlüssel ausführen, damit nicht jeder in der Lage ist, die Datenbank zu aktualisieren.

4.3.3 Authentifizierung

Über die öffentlich bereitgestellte Web API lassen sich, wie im nächsten Kapitel noch genauer beschrieben wird, nicht nur gesicherte Nutzerdaten auslesen, sondern auch erzeugen, verändern und löschen. Damit nicht ein jeder die Datensicherung bearbeiten oder, wie gerade beschrieben, den Aktualisierungsvorgang starten kann, bedarf es eines Authentifizierungsmechanismus, der sicherstellt, dass nur berechtigte Nutzer und Anwendungen die erforderlichen Schreibrechte haben. Während die großen Plattformen auf für gewöhnliche Nutzer komplizierte Verfahren setzen, ist hier bewusst[133] eine einfach zu handhabende Authentifizierung implementiert worden.

Anhand von *Tokens*, zufallsgenerierten Authentifizierungsschlüsseln, die als Parameter an jeden API-Request angehangen werden müssen, wird überprüft, ob die für eine Aktion erforderlichen Berechtigungen bestehen. Bei der Installation werden bereits zwei dieser Tokens, ein öffentlicher *Public Token*, der nur Leserechte besitzt, und ein geheimer *Master Token*, der Berechtigungen für alle CRUD-Operationen hat[134], erstellt. Über den Administrationsbereich können weitere Tokens mit unterschiedlichen Rechten erstellt werden. Zudem wird dort auch angegeben, wie oft ein Token verwendet wurde, um ungewöhnliches Nutzungsverhalten schnell ausfindig machen zu können.

Token	Permissions	Author (Usage)
2468679515337 50e69dbe1	Create, Read	Stefan's app (11)
127542604153346df5d16cb	Read	public (187)
5227648853374120db462	Create, Read, Update, Delete	master (235)

Abbildung 11: Übersicht der Tokens und ihrer Berechtigungen im Administrationsbereich.

[133] Siehe Kapitel 3.4.1, S. 40.

[134] CRUD steht für Create, Read, Update, Delete und beschreibt elementare Operationen der Datenverarbeitung, also das Erzeugen neuer Datensätze, die Leseoperationen, das Modifizieren bestehender Datensätze und das Löschen von Datensätzen.

4.4 Web API

Die Web API ist die wichtigste Komponente der Personal API – wie bereits der Name dieser Arbeit suggeriert. Durch sie können Nutzerdaten möglichst einfach zur eigenen Datenbank hinzugefügt oder editiert werden und die gesicherten Nutzerdaten öffentlich wieder bereitgestellt werden. Da die grundlegenden Funktionsweisen einer Web API bereits zu Beginn erläutert wurden[135], soll an dieser Stelle aufgezeigt werden, wie die API hier genau entworfen wurde und welche Entscheidungen getroffen worden sind.

4.4.1 URI-Design

Wie bereits erwähnt wurde, ist die Einrichtung der Personal API unter einer Subdomain ratsam. Ist die Domain zudem der Name des Anwenders, wie bei `api.naveen.com` und `api.stefangrund.de`, wird besonders deutlich, dass es sich um eine *personenbezogene* Programmierschnittstelle handelt. Der nicht näher bestimmte Aufruf der API (also nur `api.stefangrund.de`) führt dabei zu einer öffentlichen Dokumentation, in der der Aufbau der API, die Ressourcen und die Parameter vorgestellt werden und in der erklärt wird, wie die Schnittstelle ausgelesen werden kann.[136]

Damit auch in Zukunft Änderungen am URI-Design durchgeführt werden können, ohne dass die alten, bereits verwendeten URIs ungültig werden, sind sämtliche URIs der Personal API versioniert (`http://api.stefangrund.de/v1/...`). Zudem wird durch die Versionsnummer ersichtlich, dass nicht nur die Dokumentation aufgerufen werden soll, sondern tatsächlich eine Anfrage an die API gestellt wird. Praktisch wird der Webserver dabei über die `.htaccess`-Konfigurationsdatei instruiert, sämtliche Anfragen, die mit `/v1` beginnen, an das PHP-Skript `/api.php` weiterzuleiten, das anschließend die weitere Verarbeitung der Anfrage übernimmt (Bsp. 2). Sobald es weitere Versionen der API gibt, müssten diese natürlich in `.htaccess` berücksichtigt werden.

```
01 RewriteEngine On
02 RewriteBase /
03 RewriteRule ^v1/([^\.]+).(xml|json|atom)$ api.php?url=$1&format=
   $2 [QSA,L]
04 RewriteRule ^v1/([^\.]+)$ api.php?url=$1 [QSA,L]
```

Beispiel 2: Die .htaccess-Konfigurationdatei, die Anfragen an `/v1` an `/api.php` weiterleitet.

Das Skript `/api.php` bestimmt anschließend die verwendete HTTP-Request-Methode, löst die angefragte URI auf und identifiziert eventuell übergebene Parameter.

---

[135] Siehe Kapitel 2.2, S. 16.

[136] Vgl. Mark Massé: Rest API Design Rulebook. Sabastopol 2012, S. 14.

4.4.1.1 Ressourcen

Als *Ressourcen* werden sämtliche Daten und Funktionen bezeichnet, die durch die API angesteuert werden können. `/v1/steps` ist ebenso eine Ressource wie `/v1/places/18` oder `/v1/statuses?count=10`. Jeder Slash definiert dabei eine Hierarchieebene. Die Datentypen, wie hier z.B. `steps`, werden als *Collection* bezeichnet, die untergeordneten Elemente als *Documents.*[137] Ein simpler Request der Ressource `/v1/steps/35` sieht demnach folgendermaßen aus:

```
http://api.stefangrund.de/v1/steps/35
```

Wie in `.htaccess` definiert, wird der Teil hinter /`v1/` an das Skript `/api.php` übergeben und dort nach Collection und Document aufgelöst. Bei `/v1/steps/35` ist also klar, dass es sich um das Document `35` *in* der Collection `steps` handelt. Die `35` ist dabei die ID des Datensatzes, also der 35. Eintrag in der Datenbank. Die nachfolgende Anfrage wird von der API hingegen ausdrücklich nicht unterstützt, da ein angehängter Slash eine neue Hierarchiebene impliziert, die schlichtweg nicht vorhanden ist:[138]

```
http://api.stefangrund.de/v1/steps/35/
```

Zudem wird hier bei den Ressourcen bewusst auf Verben, also Operationen, verzichtet. Es gibt keine Ressourcen wie `/v1/getSteps` oder `/savePlaces`, sondern ausschließlich Ressourcen, die Inhalte abbilden, also `/v1/places`, `/v1/statuses` und `/v1/steps`. Die Operationen werden über die HTTP-Request-Methoden auf die Ressourcen angewandt, wie gleich noch näher beschrieben wird.[139]

4.4.1.2 Parameter

An die Ressourcen können bestimmte Parameter angehängt werden, die zum Teil die Antwort des Webservers beeinflussen und teils obligatorisch sind. Der Token-Parameter muss etwa bei jeder Anfrage vorkommen, da ansonsten eine Fehlermeldung ausgegeben wird. Der obige Request muss tatsächlich so aussehen:

```
http://api.stefangrund.de/v1/steps/35?token=127542604153346df5d16cb
```

Durch den Token (hier: `127542604153346df5d16cb`) wird überprüft, ob die erforderlichen Rechte für die gewünschte Anfrage bestehen. Hat der angegebene Token nur

---

[137] Vgl. Massé 2012, S. 15-16.
[138] Vgl. ebd, S. 12.
[139] Vgl. Brian Mulloy: Web API Design. Crafting Interfaces that Developers Love. San Jose 2012, S. 4-6.

Leserechte können mit ihm keine Datensätze erzeugt, bearbeitet oder gelöscht werden. Die Tokens und ihre Berechtigungen werden im Administrationsbereich unter ‚Tokens' erstellt und verwaltet.[140]

Daneben stehen drei weitere Parameter zur Verfügung. Mit `format` kann das Format der Ausgabe zwischen JSON und XML gewählt werden. Wird kein Format angegeben ist JSON das Standardausgabeformat. Über den Paramter `count` kann die Anzahl der ausgegeben Elemente bestimmt werden. Das Maximum ist 200, der Standardwert ist 25. Mit `date` kann ein Datum im Format `YYYY-MM-DD` angegeben werden, wodurch nur Elemente von diesem Tag angezeigt werden. Da `count` und `date` die Anzahl der ausgegebenen Elemente einschränken, können sie nur bei Collections verwendet werden. Der Parameter `format` *kann* bei jeder Ressource, also auch Documents, angegeben werden, `token` *muss* überall – unabhängig von der Request-Methode – angehängt werden. Die übrigen Parameter sind nur bei GET-Requests sinnvoll, weshalb sie in deren Funktion `requestGET()` in `/inc/api_functions.php` verarbeitet werden.

Ein vollständiger GET-Request mit allen Parametern sieht z.B. wie folgt aus:

```
http://api.stefangrund.de/v1/statuses?date=2014-03-30&count=3&
format=xml&token=127542604153346df5d16cb
```

4.4.2 Interaktionen und Operationen

Wie bereits erwähnt gibt es keine Ressourcen zur Bearbeitung der gesicherten Nutzerdaten. Vielmehr sind alle CRUD-Operationen über HTTP-Request-Methoden verfügbar.[141] Dabei werden die Methoden POST, GET, PUT und DELETE entsprechend der folgenden Tabelle auf die Ressourcen angewandt:

Ressource	POST (Create)	GET (Read)	PUT (Update)	DELETE (Delete)
/statuses	Erstellt neues Element in ‚statuses'	Listet die Elemente in ‚statuses' auf	Nicht möglich	Nicht möglich
/statuses/123	Nicht möglich	Zeigt das Element mit der ID 123 an	Aktualisiert das Element mit der ID 123	Löscht das Element mit der ID 123

Tabelle 2: Anwendung der CRUD-Operationen auf Collections und Documents.[142]

[140] Siehe http://api.stefangrund.de/admin/tokens.php, sowie Kapitel 4.3.3, S. 50.
[141] Siehe Kapitel 2.2.2, S. 18.
[142] Vgl. Mulloy 2012, S. 7.

Wie in der Tabelle ersichtlich wird, lassen sich innerhalb von Elementen (`/statuses/123`) keine neuen Elemente erzeugen. Zudem wurde aus Sicherheitsgründen darauf verzichtet, komplette Collections (`/statuses`) editieren oder löschen zu können.

Um die gewünschten Operationen durchzuführen, wird bei jeder Anfrage an die API zunächst die Request-Methode bestimmt (in `/api.php`). Ausgehend davon wird eine der Funktionen aus `/inc/api_functions.php` aufgerufen, also `requestPOST()`, `requestGET()`, `requestPUT()` oder `requestDELETE()`, und die Anfrage entsprechend verarbeitet.[143] Da POST und GET bei der Personal API am häufigsten genutzt werden (auch intern[144]), sollen diese zur Veranschaulichung kurz erläutert werden.

4.4.2.1 GET-Request

Die in der Personal API gesicherten Nutzerdaten können über einen GET-Request abgerufen und anschließend weiterverarbeitet oder angezeigt werden.

```
GET http://api.stefangrund.de/v1/steps?token=127542604153346df5d16cb
```

Der obige GET-Request an die Ressource `/v1/steps` wird vom Webserver mit der folgenden, hier ab Zeile 14 gekürzten Antwort bedacht, wobei die Einträge rückwärts chronologisch, mit dem aktuellsten Eintrag oben, sortiert sind. Da kein Format spezifiziert wurde, wird im JSON-Format geantwortet.

```
01 [
02     {
03         "id": 35,
04         "date": "2014-02-04 00:00:00",
05         "steps": 12040,
06         "service": "Fitbit"
07     },
08     {
09         "id": 34,
10         "date": "2014-02-03 00:00:00",
11         "steps": 7076,
12         "service": "Fitbit"
13     },
14     ...
15 ]
```

---

[143] Die Verarbeitung der POST- und GET-Requests gestaltet sich dabei ohne Schwierigkeiten, da PHP so genannte Superglobals, vordefinierte Variablen, für beide Request-Methoden bereitstellt. PUT-Requests werden jedoch zunächst nicht von PHP unterstützt, weshalb in der Personal API eine eigene Pseudo-Superglobal eingesetzt wird (siehe `requestPUT()` in `/inc/api_functions.php`). Bei DELETE handelt es sich lediglich um einen GET-Request mit der Anweisung zum Löschen, weshalb er auch problemlos verarbeitet werden kann.

[144] Siehe Kapitel 4.3.1.2, S. 48.

Die Eigenschaften der ausgegebenen JSON-Objekte bzw. der XML-Elemente variieren dabei je nach Datentyp, so wie diese im Schema der Datenbanktabelle des jeweiligen Moduls definiert wurden. So hat ein Document aus `statuses` keine Eigenschaft `steps`, sondern `status` (Bsp. 3). Bei places wiederum gibt es neben der Eigenschaft `place` auch `comment` für einen optionalen Kommentar zum Aufenthaltsort, sowie `lat` und `lng` für den Breiten- und Längengrad des Ortes (Bsp. 4, im XML-Format).

```
01 [
02     {
03         "id": 836,
04         "date": "2014-03-26 17:36:34",
05         "status": "1:0 für den FC im Kampf #effzeh gegen
            #Bachelorarbeit.",
06         "org_service": "Twitter",
07         "org_id": "448861129549578240"
08     }
09 ]
```

Beispiel 3: Document aus `statuses` im JSON-Format.

```
01 <?xml version="1.0" encoding="UTF-8"?>
02 <response>
03     <node>
04         <id>6</id>
05         <date>2014-01-26 17:52:54</date>
06         <place>Cinenova</place>
07          <comment>12 Years a Slave.</comment>
08         <lat>50.95134309</lat>
09         <lng>6.91153123</lng>
10         <org_service>Foursquare</org_service>
11         <org_id>52e53d6611d28d5cd6501967</org_id>
12     </node>
13 </response>
```

B**eispiel 4:** Document aus `places` im XML-Format.

4.4.2.2 POST, PUT- und DELETE-Requests

Das Erzeugen, Aktualisieren oder Löschen von Datensätzen über POST-, PUT- und DELETE-Requests ist entsprechend Tabelle 2 nicht mit jeder Ressource möglich. POST kann nur auf Collections (z.B. `/v1/statuses`), PUT und DELETE nur auf einzelne Documents (z.B. `/v1/statuses/123`) angewendet werden. Zudem können bei POST oder PUT tatsächlich nur die Eigenschaften angegeben werden, die der jeweilige Datentyp besitzt. Das Erzeugen eines Datensatzes in `statuses` mit der Eigenschaft

`steps` führt folglich zu einer Fehlermeldung durch die API. Generell beziehen sich die Antworten des Webservers bei den genannten Methoden nicht mehr auf die Inhalte selbst, sondern auf den Ausgang der Operation, also ob sie erfolgreich war oder nicht.

Soll in `statuses` etwa eine neue Statusmitteilung mit dem Inhalt ‚Erzeugt durch einen POST-Request!' hinzugefügt werden (Bsp. 5), antwortet der Webserver mit einer in `code`, `message` und `description` gegliederten JSON- oder XML-Datei (Bsp. 6).

```
POST http://api.stefangrund.de/v1/statuses?token=12345
Content-Type: application/x-www-form-urlencoded
Content-Length: 71

status=Erzeugt+durch+einen+POST-Request%21&date=2014-03-30+
00%3A08%3A15
```

Beispiel 5: POST-Request durch den ein neues Element in `statuses` erstellt werden soll.

```
01 {
02     "code": 201,
03     "message": "Created",
04     "description": "Element 'statuses/1234' successfully
        created."
05 }
```

Beispiel 6: Antwort des Webservers, dass ein neues Element in `statuses` erstellt wurde.

4.4.3 Ausnahmebehandlung

Sämtliche Anfragen an die API werden entsprechend der HTTP-1.1-Spezifikation[145] mit den gängigen Statuscodes beantwortet. Auf Operationen, die nicht durchgeführt werden können, nicht möglich sind oder zu denen ein Token keine Berechtigung hat, fügt die API zudem auch noch eine Antwort im JSON- oder XML-Format hinzu.[146] Diese benennt (`message`) den Statuscode (`code`) und liefert zudem eine Beschreibung des Problems (`description`). So ist für den Nutzer der API schnell ersichtlich, warum eine Anfrage vielleicht nicht durchgeführt werden konnte (Bsp. 7).

```
01 {
02     "code": 401,
03     "message": "Unauthorized",
04     "description": "Your token is missing or not valid."
05 }
```

Beispiel 7: Antwort des Webservers bei Verwendung eines fehlenden oder falschen Tokens.

---

[145] Vgl. Status Code Definitions. In: Roy Fielding et al.: Hypertext Transfer Protocol – HTTP/1.1, Juni 1999, URL: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html (30.03.2014).

[146] Vgl. Mulloy 2012, S. 10-12.

4.5 Beispielanwendung

Die Personal API lässt sich über verschiedene Desktop- und Web-Applikationen testen.[147] Das Potential der personenbezogenen Programmierschnittstelle wird aber vor allem in konkreten Anwendungsfällen ersichtlich. Dazu wurde unter http://personalapi.de/beispiel/ eine simple Beispielanwendung eingerichtet, die zeigt, wie sich die API einsetzen lässt. Der Code des Beispiels ist dabei vollkommen unabhängig vom Programmcode der Personal API, es werden nur über GET-Requests Ressourcen aus der API abgefragt. Mit den erhaltenen Daten sollen ‚Stefan's latest activities' dargestellt werden. Neben den letzten drei Statusmitteilungen werden die gelaufenen Schritte der letzten zehn Tage in einem Diagramm dargestellt und der letzte geteilte Aufenthaltsort auf einer Karte angezeigt.[148]

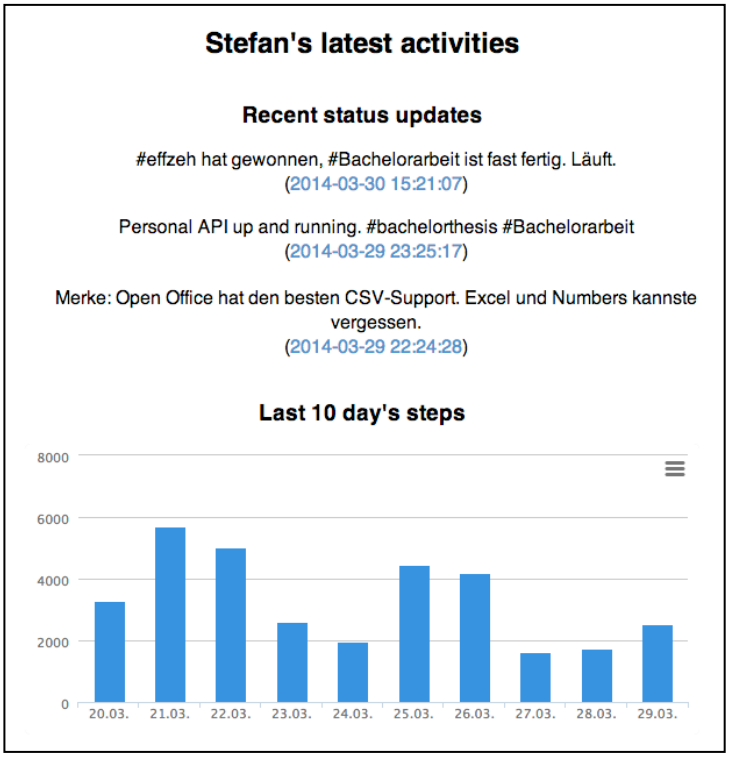

Abb. 12: Darstellung von Statusmitteilungen und Schritten.

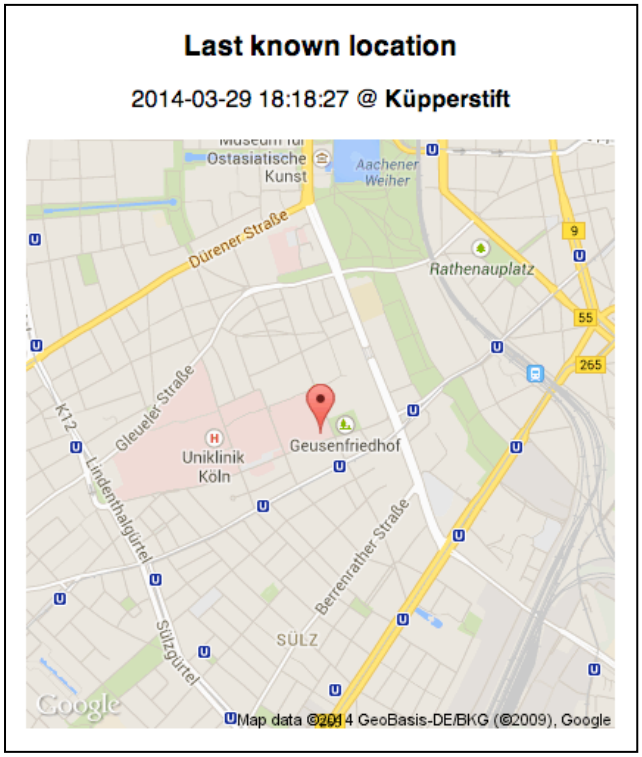

Abb. 13: Anzeige des letzten Aufenthaltsorts.

Unter Verwendung der freien JavaScript-Bibliothek Highcharts.js für das Diagramm und der Google Maps API für die Karte konnte das Beispiel in nur wenigen Zeilen realisiert werden. Wären anstelle der einfach zu handhabenden Schnittstellen der Personal API, die Web APIs von Twitter, Fitbit und Foursquare verwendet worden, wäre die Umsetzung ungleich komplexer gewesen.

---

[147] Hierbei ist vor allem das Web-Tool Hurl.it zu empfehlen, siehe http://www.hurl.it/ (30.03.2014).

[148] Siehe Abb. 12 und 13.

5. Fazit und Ausblick

Durch die hier entwickelte Personal API ist ein Nutzer in der Lage mit geringem technischen Aufwand seine Daten und Inhalte aus den unterschiedlichsten Social-Web-Angeboten zu sichern und über deren Bestehen hinaus weiterhin im Web bereitzustellen. Die Dienste, die für viele Online-Nutzer längst alltäglich geworden sind, lassen sich so weiterhin nutzen, ohne dass der Verlust der eigenen Daten droht. Auch neuartige Produkte können durch den modularen Aufbau des Programms schnell eingebunden und die durch sie erfassten, neuen Datenwelten ebenso komfortabel zur eigenen Datenbank hinzugefügt werden. Nach und nach entsteht auf diese Weise eine umfassende Kopie des digitalen Schaffens des Nutzers, ein Abbild seines digitalen Ichs, das nicht weiter der Kontrolle meist kalifornischer Konzerne unterliegt, sondern vom Nutzer selbst nach Belieben ausgewertet, bearbeitet, verwaltet und veröffentlicht werden kann.

Da die verschiedensten Nutzerdaten über eine zentrale, einfach zu benutzende Schnittstelle abrufbar sind, entstehen völlig neue Einsatzmöglichkeiten, indem sie – im Gegensatz zu vorher – mit Leichtigkeit in andere Systeme eingebunden werden können. So ist denkbar, dass Personal-API-Nutzer ihre Daten miteinander vergleichen oder die Datensätze der verschiedenen Social-Web-Plattformen in Relation zueinander setzen und dadurch neue Erkenntnisse ihres eigenen Nutzungsverhaltens erlangen. Umso mehr Daten zur API hinzugefügt werden, umso interessanter dürfte deren Analyse werden.

Dazu sollen in Zukunft weitere Module entwickelt und bestehende Funktionen verbessert werden, indem z.B. ein Cache für die API eingerichtet wird, damit Antworten auf häufig angefragte Ressourcen nicht immer wieder neu generiert werden müssen. Auch eine automatisierte Authentifizierung, die Nutzern außerhalb des Administrationsbereichs das Erzeugen von Tokens ermöglicht, wäre in zukünftigen Releases der Personal API möglich, um dem Betreiber einen besseren Überblick über die Nutzung seiner persönlichen Programmierschnittstelle zu verschaffen. Schließlich könnte das Programm um eine umfassende Suchfunktion erweitert werden, mit der sich sämtliche Datentypen durchsuchen lassen – etwas, das über die diversen Dienste hinweg nicht möglich ist. Durch die Modularität wären selbst eigene Anwendungen vorstellbar, die unabhängig von externen Diensten operieren. So ließen sich beispielsweise ein Weblog-System oder eben eine eigene Filmdatenbank durch simple Module auf Basis der Personal API realisieren. Und da das Programm bereits zur Kommunikation mit anderen Diensten in der Lage ist, könnten die Daten auch von der Personal API zu den externen Plattformen syndiziert werden, anstatt sie wie bisher nur zu aggregieren.

Die im Verlauf dieser Arbeit entwickelte Personal API soll in absehbarer Zeit quelloffen unter http://personalapi.org veröffentlicht und zum Down load angeboten werden. Zwar besteht weiterhin das Problem, dass ein Mindestmaß an technischem Verständnis zur Installation und Konfiguration der API auf dem eigenen Webserver vonnöten ist, aber gerade Nutzer, die sich der Tragweite ihres digitalen Kontrollverlusts bewusst sind und daher an Lösungen wie der Personal API interessiert sind, sollte das nicht weiter abschrecken. Technisch wäre es zwar ohne weiteres möglich, anstelle von eigenen Installationen einen zentralen Dienst zum Betreiben zahlreicher Personal APIs einzurichten, die Idee wurde jedoch verworfen, weil sie einerseits von den kommerziellen Plattformen – einen gewissen Zulauf vorausgesetzt – nicht geduldet oder über deren ‚Terms of Service' gleich ausgeschlossen werden würde und andererseits dasselbe Problem aufwirft, das mit der Personal API eigentlich gelöst werden soll: Dass die eigenen Daten nicht im eigenen Geltungsbereich, sondern in dem eines anderen liegen. Dagegen, dass einzelne Nutzer auf *ihrem* Webserver *ihre* Daten abgreifen, können Facebook, Google oder Twitter jedoch nichts ausrichten.

Die Implementierung der hier entworfenen personenbezogenen Programmierschnittstelle erlaubt es dem Nutzer ein Abbild seines digitalen Ichs im Social Web anzulegen, über das nur er selbst Kontrolle ausübt. Der Umstand, dass nicht mehr nur Plattformen und Produkte, sondern nun auch Personen Programmierschnittstellen haben, eröffnet dabei völlig neue Möglichkeiten.

Vielen Dank für's Lesen! Feedback nehme ich gerne per Twitter oder Email entgegen. Wenn Ihnen die Arbeit – vielleicht sogar als eBook – gefallen hat, würde ich mich freuen, wenn Sie sich auf die ein oder andere Weise revanchieren würden, indem Sie über die Webseite sie in den sozialen Netzwerken weiterempfehlen oder per Flattr oder PayPal sogar monetär honorieren.

Mein Dank gilt zudem Peer Bresser, Herbert Töller und Kira Töller, die mir beim Durchdenken und Korrekturlesen der Arbeit tatkräftig zur Seite standen. Weiter möchte ich diese Arbeit – auch wenn es eher unüblich ist – meiner im Verlauf der Anfertigung schwer erkrankten *Oma* Anna Gertrud Linowski widmen, die sich trotz aller Widrigkeiten wie immer täglich über meinen Fortschritt erkundigt hat.

Literaturverzeichnis

- Berners-Lee, Tim: Der Web-Report. Der Schöpfer des World Wide Web über das grenzenlose Potential des Internets. München 1999.
- Berners-Lee, Tim: Long Live the Web. In: Scientific American 165 (2010), H. 12, S. 80-85.
- Ebersbach, Anja; Glaser, Markus; Heigl, Richard: Social Web. Konstanz 2008.
- Fielding, Roy: Architectural Styles and the Design of Network-based Software Architectures. Doktorarbeit, University of California, 2000.
- Grasse, Christian; Greiner, Ariane: Mein digitales Ich. Wie die Vermessung des Selbst unser Leben verändert und was wir darüber wissen müssen. Berlin 2013.
- Hippner, Hajo: Bedeutung, Anwendungen und Einsatzpotenziale von Social Software. In: HMD. Praxis der Wirtschaftsinformatik 43 (2006), H. 252, S. 6-16.
- Kemper, Peter; Mentzer, Alf; Tillmanns, Julika: Wirklichkeit 2.0. Medienkultur im digitalen Zeitalter. Stuttgart 2012.
- Klausnitzer, Rudi: Das Ende des Zufalls. Wie Big Data uns und unser Leben vorhersagbar macht. Salzburg 2013.
- Lobo, Sascha: Die Aufgabe der sozialen Medien. In: Anda, Béla; Endrös, Stefan; Kalka, Jochen; Lobo, Sascha (Hrsg.): SignsBook – Zeichen setzen in der Kommunikation. Wiesbaden 2012.
- Massé, Mark: Rest API Design Rulebook. Sabastopol 2012.
- Maximilien, E. Michael; Ranabahu, Ajith; Gomadam, Karthik: An Online Platform for Web APIs and Service Mashups. In: IEEE Internet Computing 12 (2008), H. 5, S. 32-43.
- Mitchell, Lorna Jane: PHP Web Services. Sebastopol 2013.
- Mulloy, Brian: Web API Design. Crafting Interfaces that Developers Love. San Jose 2012.
- Müller, Thorsten: Habitualisierte Mobilnutzung – Smartphones und Tablets gehören zum Medienalltag. In: Media Perspektiven (17) 2013, H. 9, S. 410-421.
- Richardson, Leonard; Ruby, Sam: RESTful Web Services. Sebastopol 2007.
- Schmidt, Jan-Hinrik: Social Media. Wiesbaden 2013.
- Swan, Melanie: Sensor Mania! The Internet of Things, Wearable Computing, Objective Metrics, and the Quantified Self 2.0. In: Journal of Sensor and Actuator Networks (1) 2012, H. 3, S. 217-253.
- van Eimeren, Birgit: „Always on“ – Smartphone, Tablet & Co. als neue Taktgeber im Netz. In: Media Perspektiven (17) 2013, H. 7-8, S. 386-390.
- van Eimeren, Birgit; Frees, Beate: Ergebnisse der ARD/ZDF-Onlinestudie 2013. Rasanter Anstieg des Internetkonsums – Onliner fast drei Stunden täglich im Netz. In: Media Perspektiven (17) 2013, H. 7-8, S. 358-372.
- von Gehlen, Dirk: Mashup. Lob der Kopie. Berlin 2011.

Internetquellen

- Anderson, Chris; Wolff, Michael: The Web Is Dead. Long Live the Internet. In: Wired, 17.08.2010, URL: http://www.wired.com/magazine/2010/08/ff_webrip/
- Carmichael, Alexandra: Kevin Kelly on The History and Future of QS. In: Quantified Self.com, 14.10.2012, URL: http://quantifiedself.com/2012/10/kevin-kelly-on-the-history-and-future-of-qs/ (30.03.2014).
- Cross, Andrew: The Personal API. In: Medium, 13.06.2013, URL: https://medium.com/better-humans/f9fba15cdcb2
- Dash, Anil: The Web We Lost. In: Anil Dash. A Blog About Making Culture, 13.12.2012, URL http://dashes.com/anil/2012/12/the-web-we-lost.html
- Diverse: A Focus on Efficieny. A whitepaper from Facebook, Ericsson and Qualcomm. In: Internet.org, 16.09.2013, URL: http://internet.org/efficiencypaper
- F., Elbert: How to store passwords safely with PHP and MySQL. In: Alias.io, 25.04.2013, URL: http://alias.io/2010/01/store-passwords-safely-with-php-and-mysql/
- Facebook: Quaterly Earnings Slides Q4 2013, 29.01.2014. Abrufbar unter: http://investor.fb.com/results.cfm
- Fielding, Roy et al.: Hypertext Transfer Protocol – HTTP/1.1. In: W3C Architecture Domani, Juni 1999, URL: http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html
- Haeusler, Johnny: 2013: Das Web zurückerobern. In: Spreeblick, 28.12.2012, URL: http://www.spreeblick.com/2012/12/28/2013-das-web-zuruck-erobern/
- Hamblen, Matt: Google CEO preaches 'mobile first'. In: Computerworld, 17.02.2010, URL: http://www.computerworld.com/s/article/9157778/Google_CEO_preaches_mobile_first_
- Lobo, Sascha: Abschied von der Utopie. Die digitale Kränkung des Menschen. In: FAZ.NET, 11.01.2014, URL: http://www.faz.net/aktuell/feuilleton/debatten/abschied-von-der-utopie-die-digitale-kraenkung-des-menschen-12747258.html
- Lobo, Sascha: Die Mensch-Maschine: Euer Internet ist nur geborgt. In: Spiegel Online, 17.04.2012, URL: http://www.spiegel.de/netzwelt/web/sascha-lobos-kolumne-zum-niedergang-der-blogs-in-deutschland-a-827995.html
- Mikkonen, Tommi; Taivalsaari, Antero: Apps vs. Open Web: The Battle of the Decade. In: Proceedings of the 2nd Workshop on Software Engineering for Mobile Application Development, 27.10.2011, URL: http://livelykernel.cs.tut.fi/publications/BattleOfTheDecade-Mikkonen-Taivalsaari.pdf
- Quast, Christina: Im Blickpunkt: Das digitale Ich (Broschüre des Grimme Instituts). Oktober 2012, URL: http://www.grimme-institut.de/imblickpunkt/pdf/IB-Das-digitale-Ich.pdf
- Red Dot: Design Team of the Year 2003: Nokia Design Team. In: Red Dot Design Award, 2003, URL: http://red-dot.de/pd/celebration/design-team-of-the-year-2003-nokia-design-team/

- Rosenbach, Marcel: Tim Berners-Lee über das Web: "Nichts ist perfekt". In: Spiegel Online, 12.03.2014, URL: http://www.spiegel.de/netzwelt/web/tim-berners-lee-ueber-das-internet-und-25-jahre-www-a-957978.html
- Schwenzel, Felix: Reclaim Social Media – FAQ. In: Wirres.net, 25.02.2014, URL: http://wirres.net/article/articleview/7082/1/6/
- Selvadurai, Naveen: A Personal API. In: Naveen Selvadurais Weblog, 28.05.2013, URL: http://x.naveen.com/post/51808692792/a-personal-api
- Social Media (und Unterseiten). In: Webseite der ARD/ZDF-Onlinestudie, 2013, URL: http://www.ard-zdf-onlinestudie.de/index.php?id=397, http://www.ard-zdf-onlinestudie.de/index.php?id=434
- Stöcker, Christian: Identität im Netz: Das digitale Ich liegt in Scherben. In: Spiegel Online, 29.07.2008, URL: http://www.spiegel.de/netzwelt/web/identitaet-im-netz-das-digitale-ich-liegt-in-scherben-a-567899.html
- Tencent: Q3 2013 Results, 13.11.2013, URL: http://www.tencent.com/en-us/content/ir/news/2013/attachments/20131113.pdf
- The Nielsen Company: The Mobile Consumer. A Global Snapshot. In: Nielsen.com, Februar 2013, URL: http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2013%20Reports/Mobile-Consumer-Report-2013.pdf
- Twitter: Twitter Reports Fourth Quarter and Fiscal Year 2013 Results. In: Twitter Investor Relations, 05.02.2014, URL: https://investor.twitterinc.com/releasedetail.cfm?releaseid=823321
- Werdmuller, Ben: The IndieWeb as a minimally viable social web ecosystem. In: W3C Open Social, 08.08.2013, URL: http://www.w3.org/2013/socialweb/papers/MVP.html

Alle Webseiten wurden zuletzt am 30.03.2014 auf Erreichbarkeit hin überprüft. Zudem wurden sie in Kopie bei Archive.org (https://archive.org/web/) gespeichert.

Besprochene Anwendungen, Produkte und Webseiten

- Amazon: http://www.amazon.com/
- Facebook: http://www.facebook.com/
- Fitbit: http://www.fitbit.com/
- Flickr: http://flickr.com/
- Foursquare: http://foursquare.com/
- Frostbox: http://www.frostbox.com/
- Google: http://google.com/
- Google Glass: http://www.google.com/glass/start/
- Google Plus: http://plus.google.com/
- highlight.js: http://highlightjs.org/
- Hurl.it: http://www.hurl.it/

- ICQ: http://www.icq.com/
- IndieWebCamp: https://indiewebcamp.com/
- Instagram: http://instagram.com/
- Instaport: http://instaport.me/
- jQuery: http://jquery.com/
- Line: http://line.me/
- LinkedIn: http://linkedin.com/
- MediaWiki: http://www.mediawiki.org/
- Microformats: http://microformats.org/
- Netflix: http://www.netflix.com/
- Nike Fuelband: http://www.nike.com/de/de_de/c/nikeplus-fuelband
- Pebble: http://getpebble.com/
- Personal API von Naveen Selvadurai: http://api.naveen.com/
- Personal API von Stefan Grund: http://api.stefangrund.de/
- ProgrammableWeb: http://www.programmableweb.com/
- Reclaim Social Media: http://reclaim.fm/
- Re:publica: http://re-publica.de/
- Scribd: http://www.scribd.com/
- Social Safe: http://www.socialsafe.net/
- Soundcloud: http://soundcloud.com/
- Tumblr: http://www.tumblr.com/
- TweetNest: http://pongsocket.com/tweetnest/
- Twitter: http://twitter.com/
- Vine: http://vine.co/
- WeChat: http://www.wechat.com/
- WhatsApp: http://www.whatsapp.com/
- Wikipedia: http://www.wikipedia.org/
- Withings: http://www.withings.com/
- WordPress: http://wordpress.org/
- Xing: http://www.xing.com/
- YouTube: http://www.youtube.com/