



Stefan Jaindl, B.A. B.Sc.

Social Media Software Integration for the Symfony Web Framework and Android and iOS Versions of the Catrobat Project

MASTER'S THESIS

to achieve the university degree of

Master of Science

Master's degree programme: Software Development and Business
Management

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Ing. Dr. techn. Wolfgang Slany

Institute of Software Technology

Graz, April 2016

This document is set in Palatino, compiled with [pdfL^AT_EX2e](#) and [Biber](#).

The L^AT_EX template from Karl Voit is based on [KOMA script](#) and can be found online: <https://github.com/novoid/LaTeX-KOMA-template>

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis dissertation.

Date

Signature

Abstract

Social Media is a phenomenon that rapidly evolved over the past decade. In 2016 already more than two billion people use social networks worldwide. Therefore, it is important to organizations to be present in social networks. Of course, it is essential to choose the right social networks that fit best to an organization's target group. However, this choice can be difficult for globally active organizations as the social media landscape is very diverse. Therefore, this thesis gives an overview about how social media platforms are split up, especially across different regions and age groups. Based on these insights, the ultimate goal is to give a recommendation for the Catrobat project on how and where to engage in social networks.

Some of the main findings of this thesis are that social media gets increasingly mobile and that there is a significant shift from the big social networks towards messengers and private groups.

Based on these findings, implementations of social media software elements have been done and suggestions for further implementations have been given for the mobile Android and iOS versions of Pocket Code as well as for the Website and Web API. The implementations include social sign-in (Open Authentication) with Facebook and Google+ for Pocket Code and the Website, sharing of Catrobat programs on diverse platforms, liking, automated posts of uploaded Catrobat programs to a Facebook page or program download statistics. In future, it is suggested to add more social elements and to diversify the now existing social elements regionally. Recommendations on which social platforms to engage with from a marketing perspective are given as well.

Kurzfassung

Social Media ist ein Phänomen, das sich über das letzte Jahrzehnt rasant entwickelt hat. Im Jahr 2016 nutzen bereits mehr als zwei Milliarden Menschen weltweit soziale Netzwerke. Daher ist es für Organisationen wichtig in diesen sozialen Netzwerken präsent zu sein. Es ist natürlich essentiell dabei die sozialen Netzwerke zu wählen, die am besten zu der Zielgruppe einer Organisation passen. Diese Wahl kann für globale Organisationen jedoch schwierig sein, da die Landschaft der sozialen Medien sehr vielfältig ist. Daher gibt die vorliegende Masterarbeit einen Überblick wie diese Landschaft, im Speziellen hinsichtlich regionaler Unterschiede und Altersgruppen, zusammengesetzt ist. Aufbauend auf diesen Erkenntnissen, ist das Ziel eine Empfehlung für Catrobat abzugeben in welchen sozialen Netzwerken und auf welche Weise es Sinn macht aktiv zu werden.

Einige der wichtigsten Erkenntnisse dieser Masterarbeit sind, dass Social Media zunehmend mobil wird und dass es eine deutliche Tendenz von den großen sozialen Netzwerken weg hin zu Messengern und privaten Gruppen gibt.

Darauf basierend wurden für die mobilen Android- und iOS-Versionen von Pocket Code, für die Website und die Web API verschiedene Social Media Softwareelemente implementiert und Vorschläge für weitere Implementierungen gemacht. Die Implementierungen umfassen das Anmelden über soziale Netzwerke (Open Authentication) mit Facebook und Google+, das Teilen von Catrobat-Programmen über mehrere Plattformen, Liking, automatisierte Beiträge auf einer Catrobat-Facebook-Seite von auf den Webserver hochgeladenen Programmen und Programm-Downloadstatistiken. Zukünftig wird vorgeschlagen weitere Social Media-Elemente hinzuzufügen und die existierenden Implementierungen regional automatisiert zu differenzieren. Es wird ebenfalls eine Empfehlung aus der Marketingperspektive abgegeben, auf welchen Plattformen Catrobat aktiv werden sollte.

Contents

Abstract	v
Kurzfassung	vii
1 Introduction	1
2 Glossary	3
3 Web 2.0 - Social Web	7
3.1 Web 2.0 Evolution	7
3.2 Web Service Standards	10
3.2.1 XML-RPC	10
3.2.2 SOAP	11
3.2.3 REST	15
3.3 Web 2.0 Technologies	21
3.3.1 AJAX	21
3.3.2 Mashups	23
3.3.3 Content Syndication and Aggregation	23
3.3.4 Folksonomies - Social Tagging	25
3.3.5 Microformats, RDFa and Microdata	27
3.3.6 Cloud Computing	30
3.4 Social Applications	33
3.4.1 Blogs and Micro-Blogs	33
3.4.2 Wikis	35
3.4.3 Podcasts and Videocasts	35
3.4.4 Social Networks	36
3.4.5 Other Social Applications	39

Contents

4	Web 3.0 - Semantic Web	43
4.1	Web 3.0 Evolution	43
4.2	Social Semantic Web	45
5	Social Media Marketing	53
5.1	From Outbound to Inbound Marketing	53
5.2	Traditional vs. Social Media Marketing	59
5.3	Digital Marketing Communication	60
6	Catrobat	65
6.1	Pocket Code Target Group	67
7	Social Media Statistics	77
7.1	Social Networks General	78
7.2	Mobile	84
7.3	Social Media Platforms	88
7.3.1	Social Networks	88
7.3.2	Social Messengers	93
7.4	Countries and Regions	99
7.4.1	Messengers by Region	101
7.4.2	Europe	103
7.4.3	United States	104
7.4.4	Asia	105
7.4.5	Brazil	108
7.5	Recommendation	109
8	Implementation	115
8.1	Open Authentication	115
8.1.1	OpenID	116
8.1.2	OAuth 2.0	118
8.1.3	Facebook	122
8.1.4	Google	132
8.1.5	Design and UX	135
8.2	Facebook Channel Postings	138
8.3	Likes, +1 and Sharing	142
8.4	Program Download Statistics	144
8.5	Further Implementations	147

Contents

Conclusion	151
Appendix	157
Bibliography	163

List of Figures

3.1	UDDI Communication	13
3.2	WSDL Communication	14
3.3	SOAP Message Structure	14
3.4	Classic vs. AJAX Web application model	22
3.5	Climate Justice Peace Geolocation Mashup in Paris	24
3.6	Podcast creation and publication process	25
3.7	Web 2.0 tag cloud	26
3.8	Google Rich Snippet	30
3.9	Basic Technologies of Cloud Computing	31
3.10	Layered Architecture of Cloud Computing	32
3.11	Sociogram: Network of completed passes in the UEFA EURO 2012 final Spain vs. Italy	40
3.12	Physical and conceptual contexts in a mobile environment	41
4.1	Social Semantic Web Dimensions	47
4.2	Socially Enabled Semantic Web: FlickrR	49
4.3	Semantically Enabled Social Software	49
5.1	Customer-Based Brand Equity Pyramid	55
5.2	Inbound Marketing Methodology	58
6.1	Script of a Pocket Code program	66
6.2	Catrobat community website	66
6.3	Catrobat target group	67
6.4	Catrobat persona: Tobias	68
6.5	Catrobat persona: Marie	69
6.6	Community Website Overview	70
6.7	Community Website Country Distribution	71
6.8	Community Website Device Distribution	72

List of Figures

6.9	Pocket Code on Google Play	73
6.10	Number of Active Pocket Code Users	73
6.11	Age and Gender of Pocket Code Users	74
6.12	Country Distribution of Pocket Code Users	75
7.1	Number of Internet and Social Network Users Worldwide . .	78
7.2	Social Network Users Worldwide per Platform	79
7.3	Social Network Growth Worldwide	80
7.4	Social Network Growth Rates Worldwide per Platform	81
7.5	Messengers and Social Apps Growth Rates Worldwide per Platform	82
7.6	Average Number of Social Media Accounts Worldwide	83
7.7	Average Time Spent on Social Networks	84
7.8	Mobile Subscriptions from 2008 to 2015	85
7.9	Mobile Penetration Rates by Age	86
7.10	Mobile Penetration Rates by Gender	86
7.11	Number of Facebook Desktop, Mobile and Mobile-Only Users from 2010 to 2015 Worldwide	87
7.12	Age Distribution of Social Media Users Worldwide by Q3 2014	89
7.13	Gender Distribution of Social Media Users Worldwide by Q1 2013	90
7.14	Usage Reasons for Facebook and Twitter	91
7.15	Global Usage Frequency by Active Users of Selected Social Networks by Q4/2014	92
7.16	Usage Penetration of Selected Social Networks by Region . .	92
7.17	Number of mobile messenger users worldwide.	93
7.18	Most Popular Global Messenger Apps	94
7.19	WhatsApp and Facebook Age and Gender Distribution among German Childrens and Teens	95
7.20	WeChat user distribution by age	96
7.21	LINE user distribution by age	97
7.22	SnapChat user distribution by age	98
7.23	Digital, Social and Mobile Numbers by Region as of January 2015	99
7.24	Digital, Social and Mobile Numbers by Region in Percent of Population as of January 2015	99
7.25	Global Social Network Penetration Rates by Region	100

List of Figures

7.26	Top 3 Messengers by Region	101
7.27	WhatsApp users by Region	102
7.28	Most popular Asian messengers	102
7.29	Share of Internet users accessing Platforms in selected Euro- pean countries in 2013	103
7.30	Leading Social Networks in the USA as of June 2015	104
7.31	Most popular Social Networks in the USA for teens	105
7.32	Most Popular Chinese Social Networks	106
7.33	Most Popular Japanese Social Networks in 2014	106
7.34	Most Popular Indian Social Networks	107
7.35	Most Popular Russian Social Networks	108
7.36	Penetration of leading social networks in Brazil as of 4th quarter 2014	109
8.1	Preferred Platforms for Social Signin Worldwide	116
8.2	Age Distribution for Social Signin Worldwide	117
8.3	OpenID protocol flow	118
8.4	Server-to-server authorization flow	120
8.5	Client-to-server authorization flow	121
8.6	Possible Facebook Login workflows	124
8.7	Catroweb Client Workflow - Secure Server-to-Server Calls. . .	126
8.8	Google+ Sign-In for server-side apps	133
8.9	Old Pocket Code Login Dialog	135
8.10	New Pocket Code Signin Dialog	135
8.11	New Pocket Code Registration Dialog	136
8.12	New Pocket Code Login Dialog	136
8.13	Catroweb Registration Web Page	137
8.14	Catroweb Login Web Page	137
8.15	Facebook Post on Facebook Page.	138
8.16	Facebook Post on the Program Webpage	139
8.17	Like and Share Design for Desktop Version of Website	142
8.18	Like and Share Design for Mobile Version of Website	143
8.19	Program Download Statistics in the Website Admin section .	146

List of Tables

5.1	Pros and Cons of Traditional and Social Media Marketing. Adopted from (SocialTimes, 2015).	60
7.1	Grouping of regions or countries with similar social media platforms.	110
8.1	Server-side vs. Client-side authorization flow. Adopted from (Boyd, 2012)	122
.2	Server-Side API Functions	158
.3	Server-Side API Functions	159
.4	Server-Side API Functions	160
.5	Server-Side API Functions	161

1 Introduction

Nowadays, Social Media has become an important part of everyday life in most regions. Especially for Open Source projects, like Catrobat, social platforms offer a fast and cheap way to reach a large audience and to build communities.

The difficulty thereby is to choose the right social platforms to be present on. Actually, this thesis has two interrelated goals. The first goal is to give a recommendation on which platforms Catrobat should be active from a Marketing perspective. The second goal is to identify which social software elements are meaningful to be added to the mobile Pocket Code apps and to the Website. A part of these elements has been implemented and another part has been suggested as future implementations.

In order to accomplish these goals, this thesis is structured into several theoretical and practical chapters. In the beginning the most important terminology used throughout the thesis is defined in Chapter 2. Chapter 3 describes the theoretical background knowledge of the Social Web, which is also known as Web 2.0. This includes the principles behind the Social Web, explanation of the underlying Web Service standards XML-RPC, SOAP and REST, Web 2.0 technologies as well as some of the most important social applications. This includes technologies such as mashups, content syndication, social tagging or microformats, and social applications such as blogs, microblogs, Wikis, podcasts or whole social networks. Chapter 4 first gives a quick introduction to the Semantic Web, which is also known as Web 3.0, before discussing the connection to the Social Web within the Social Semantic Web. The Social Semantic Web combines structured data of the Semantic Web with user-generated content of the Social Web. Social Semantic Web initiatives such as FOAF, DBPedia, SIOC, as well as the new W3C JSON-LD standard to create an interlinked network of structured data across social Websites, are discussed.

1 Introduction

Chapter 5 gives a brief overview over Social Media Marketing aspects. First, the shift from outbound to inbound marketing as a consequence of Social Media is explained. The main principle behind inbound marketing is that organizations should be present at those places where their customers are present and offer a great user experience there. Inbound marketing techniques such as branding, search engine optimization (SEO) or Social Media Marketing are presented. Finally, a few paragraphs are dedicated to traditional vs. Social Media Marketing and digital marketing communication.

Chapter 6 initiates the practical part of this thesis. In this chapter the Catrobat project is introduced and its target group is defined with the help of Personas, Google Analytics, Andlytics and Facebook Analytics data. Based on the target group, Chapter 7 analyzes the global social media landscape for its potential to Catrobat. This includes general social network statistics (e.g. number of social network to Internet users, leading social networks worldwide or growth rates), mobile usage statistics (mobile subscriptions worldwide, share of Internet users who access social media via mobile by age and gender, and mobile and Desktop user distributions for Facebook), detailed age and gender-related statistics about social networks and social messengers, diverse social networks usage statistics (e.g. usage frequencies) and, finally, statistics related to different countries and regions. The regional statistics are especially tailored at Catrobat's most important target countries and regions, which are Europe, the United States, Brazil and several Asian countries, as outlined in Chapter 6. The final section of this chapter gives a recommendation for Catrobat on how and where to engage in different social media platforms, based on the statistical data retrieved. The recommendations are two-fold, as marketing aspects (e.g. marketing presence on WhatsApp and Snapchat) are recommended on the one hand, and potential implementations (e.g. diversification of social sign-in) on the other hand.

Finally, Chapter 8 illustrates and explains necessary background knowledge of implementations that have been made as part of this thesis. This includes social sign-in with Facebook and Google+, Facebook channel postings, Likes, +1's, sharing of Catrobat programs and program download statistics. At the end, further implementations are suggested. Many of these suggestions are based on the fact, that the social media landscape is relatively diversified across different regions in the world and that mobile usage scenarios get more and more important.

2 Glossary

This chapter gives definitions of important terms used in this thesis.

According to (Kotler et al., 2009) **Marketing** is “the task of creating, promoting, and delivering goods and services to consumers and businesses in order to fulfill wants, needs and demands”

According to (AmericanMarketingAssociation, 2016), a **Brand** is a “name, term, sign, symbol, or design, or a combination of them, intended to identify the goods and services of one seller or group of sellers and to differentiate them from those of competition”.

According to (ManagementStudyGuide, 2016b), **Brand Awareness** is the degree to which consumers precisely associate a brand with a product. Brand recognition and brand recall are part of brand awareness. **Brand recognition** is the ability of consumers to clearly recognize a brand, when they are shown or asked about that specific brand. **Brand recall** is the ability of consumers to recall a brand when they are given a product category.

According to (ManagementStudyGuide, 2016a), the **Conversion Rate** is the percentage of visitors who take a desired action. A desired action can be a membership registration, for example.

A **Landing Page** is a Web page that contains marketing elements and is displayed as first page to Web site visitors. Landing pages are used for email marketing or search engine optimization, for example (OnPageWiki, 2016).

The **Click-Through-Rate** is the ratio of number of clicks (e.g. on a banner or link) to page impressions (OnPageWiki, 2016).

2 Glossary

The **Bounce Rate** is the ratio of number of visitors that visit more than one Web page to the number of visitors that only visit one Web page (OnPageWiki, 2016).

A **Resource** is something that has an identity such as an image, a document or a Web service. A **Uniform Resource Identifier (URI)** is the identifier or name of a resource. The **Uniform Resource Locator (URL)** is a subset of the URI that specifies the access mechanism to the resource (e.g. http://) (Berners-Lee, 2005).

Hyperlinks are texts or media elements (e.g. images) users can click on in order to navigate to different Web pages. **Hypertext** and **Hypermedia** are subsets of hyperlinks. Hypertexts refer to textual and hypermedia to media (e.g. images or videos) hyperlinks (Techopedia, 2016).

According to (W3Schools, 2016), the **Service-Oriented Architecture (SOA)** is a set of components which can be invoked, and whose interface descriptions can be published and discovered.

According to (W3Schools, 2016), **Web Services** are software applications that allow interaction with other systems over the Internet, and are based on open Internet standards, such as XML or JSON, to describe its interfaces and bindings, and are clearly identified by a URI (Techopedia, 2016).

Middleware is a software layer located between operating systems (OS) and applications in distributed systems that simplifies programming by providing a uniform and higher-level interface to applications (Techopedia, 2016).

Universal Description Discovery and Integration (UDDI) is a directory where Web service providers can register their business and Web services. Service consumers can look up these Web services in the directory (Techopedia, 2016).

Extensible Markup Language (XML) is a universal format used to describe and transfer structured data between applications or the Web (Techopedia, 2016).

The **Web Services Description Language (WSDL)** is an XML-based language that describes functionality and details (e.g. parameters) of Web services (Techopedia, 2016).

A **User Agent (UA)** is a program that allows to use network services and is controlled by a user (e.g. Web browsers). The term UA is also associated with the, mainly HTTP, header strings send in requests that help a server identify a user's OS and other request content (Techopedia, 2016).

Linked Data is interconnected and structured data on the Web that aims to facilitate information sharing and retrieval. One of the most important technologies in this context is the Resource Description Framework (RDF) (Techopedia, 2016).

Authentication is a process where a user proves his identity. This is done by passwords, biometrics, digital certificates or other authentication tools such as key cards or USB tokens. **Authorization** is a security mechanism that assigns successfully authenticated users with the respective access rights to resources (Techopedia, 2016).

3 Web 2.0 - Social Web

Traditional marketing includes well-known forms of advertising and can be broadly classified into the categories print (e.g. newspaper advertisements and magazines), broadcast (e.g. radio and television commercials), direct mail (e.g. flyers and catalogs) and telemarketing (direct phone calls). Many companies combine these different marketing channels with an according Cross-Media Marketing Strategy¹.

During the last years many marketers switched their focus increasingly on Social Media Marketing methods and away from traditional Marketing methods. The trigger behind Social Media was the rapid evolution of Internet technologies (Weinberg, 2014) [pp. 5-6]. Before comparing traditional marketing and Social Media Marketing in greater detail in Section 5.2, the technological evolution must be understood.

3.1 Web 2.0 Evolution

Before Web 2.0 evolved, the Internet was basically a collection of static Web pages without any user interaction. This period was called the era of Web 1.0. The burst of the dot-com bubble in 2001 led to the beginning of the Web 2.0 era. There are different definitions of what Web 2.0 is. In this thesis Web 2.0 is referred to as the Social Web and it is about user interaction, and user-generated and dynamic Web content. The following important concepts build the foundation of the Social Web as laid out by Dale Dougherty and Tim O'Reilly at the initial Web 2.0 Conference of this new era²:

¹MarketingSchools, 2015.

²Dougherty and O'Reilly, 2015.

3 Web 2.0 - Social Web

- Web as Platform
- Collective Intelligence
- Importance of Data
- Software Release Cycles
- Lightweight Programming Models
- Cross-Device Software
- Rich User Experience

The term “**Web as Platform**” or, equivalently, “Software as a Service” refers to the requirement that Software should not only be locally installed, but rather be distributed over the Internet. An example is Netscape as Web 1.0 vs. Google as Web 2.0 representative. Netscape’s products were servers and the Netscape Navigator, which was a browser and therefore a Desktop application. The strategy was to control the standards for displaying Web content (e.g. HTML) and the browser itself. However, these Web components are seen as commodities nowadays. In contrast, most Google products are native Web applications which are delivered as Web services. The big advantage of this architecture is the high scalability and the platform-independence. Also, no scheduled software releases are needed. The necessary core competency of Google is the control and management of the huge amount of data. The service Google delivers is to act as middleman between users and desired Web contents with, among others, the Google search engine.

Collective Intelligence means that any user can add new Web content and the more users contribute the more valuable the Web content gets. With the aid of hyperlinking Web content gets connected as users discover and link to new content. Hyperlinking was the key to Google’s success with the Page Rank Algorithm, for example. Other examples are the recommender systems of Amazon or eBay, content creation in Wikipedia or Folksonomies of Flickr³. Folksonomies are described in detail in the Section 3.3.4.

Another important Web 2.0 technology that allows for collective intelligence is blogging. Blogging was made possible by RSS technology that pushes updates to subscribers, which is referred to as “incremental Web”. Hereby strong links between content syndicators and content aggregators are created. RSS technology is also used for other social applications such as podcasts or news feeds that are pushed to a news reader (Hammersley,

³Dougherty and O’Reilly, 2015.

2005) [pp. 1-11]. Blogging, Podcasts and RSS are described in detail in the sections 3.4.1, 3.4.3 and 3.3.3.

Of course, collective intelligence creates huge amounts of data. The **importance of the data** must be understood and managed. Therefore, database management systems are a core competency of Web 2.0 applications. In order to gain a competitive advantage, companies must have a unique and hard-to-recreate source of data as applications are increasingly data-driven. The software should be designed in a way that users are encouraged to add valuable data through an architecture of participation (Anderson, 2012) [pp. 23-33].

Concerning software development methods Web 2.0 caused a fundamental change. **Software release cycles** are dramatically shortened. For example, Google has to crawl the Web, update search indices and respond to user queries continuously. As far as possible, third party data services should be reused and loosely coupled systems should be supported in the network of cooperating data services. Another point is to see users as co-developers or real-time-testers. By doing that, companies can gather valuable user feedback on new features immediately⁴.

Furthermore, companies need to adopt **lightweight programming models** to guarantee for highly reliable and cross-platform distributed applications. This can be achieved with Web services, as defined in the Chapter 2. Communication between Web services and user agents happens through standardized messages (e.g. with XML or JSON serialization) via standardized protocols. In this context, the most important technologies are RSS, Remote Procedure Calls (RPC), Simple Object Access Protocol (SOAP), Representational State Transfer (REST) and Asynchronous JavaScript and XML (AJAX) within a service-oriented architecture (SOA). SOA means that single and easily exchangeable low-level services can be encapsulated into high-level services. Therefore, the complexity of single applications is hidden behind standardized interfaces. Technically, the services are available in the Internet and communication happens with the above mentioned protocols and technologies. The user of a Web service only needs to know what input is needed and which result is delivered. Available Web services may be detected by a directory service. An exact description of Web services

⁴Dougherty and O'Reilly, 2015.

3 Web 2.0 - Social Web

may also be available. For example, in the case of SOAP, such a directory service is the Universal Description, Discovery and Integration (UDDI). Web service descriptions for SOAP can be obtained with aid of the Web Services Description Language (WSDL). The invocation of Web services is done by calling a remote program, which should be possible across companies and the entire Web. For that reason, Web services need to be loosely-coupled. Altogether, Web service architectures have three components - the service requester, the service provider and the service registry (directory service). Furthermore, the architecture is based on a common syntax (e.g. XML), has an interaction convention (e.g. messaging or RPC) and a set of bindings to map messages onto a transport protocol (e.g. TCP/IP, HTTP or SMTP) (Alonso et al., 2004) [pp. 123-155].

Another important Web 2.0 principle is to **support software across devices**, e.g. as Desktop, mobile and Web applications. Finally, **Rich User Experiences** allows to deliver active content to users 3.3.1. A very important technology in this context is AJAX, that will be described in detail in section⁵.

3.2 Web Service Standards

This section gives an overview of the aforementioned Web service standards and where they are located in the Web Service Technology Stack.

3.2.1 XML-RPC

XML-RPC is one of the oldest Web service communication standards and predecessor of SOAP. RPC basically allows remote function calls in a client-server or peer-to-peer architecture. XML is used to describe the syntax of remote calls and HTTP is the medium for transporting messages over the network. Basic functionalities include the representation of data types, generation, analysis and transmission of request and response packages, and the provision of skeletons and stubs. Disadvantages are that XML-RPC

⁵Dougherty and O'Reilly, 2015.

is stateless and tends to be error-prone, inefficient (because it is based on the simple HTTP protocol) and gives no guarantee that Web services are loosely-coupled. Generally, XML-RPC service descriptions are based on WSDL (Hammersley, 2001) [pp. 1-12]. The XML-RPC specification can be found on <http://xmlrpc.scripting.com/spec.html>.

3.2.2 SOAP

In order to understand SOAP, the Web Service Technology Stack must be discussed first. The items are similar to the TCP/IP stack and contain the following layers in a top-down perspective:

1. Discovery
2. Description
3. Packaging
4. Transport
5. Network

The discovery layer provides service requestors with descriptions of available Web services. For example, the UDDI protocol resides on this layer. The description layer provides an exact specification about how Web services can be called and which results are delivered. The WSDL is positioned on this layer. In the packaging layer data is serialized to a standardized message format so that it can be understood by each party. SOAP works exactly on this layer. The transport layer makes the communication between service requestors and service providers possible. Technologies on this layer are for example TCP/IP, HTTP or SMTP. Finally, the network layer is the most basic layer and provides communication, addressing and routing abilities (Snell, Tidwell, and Kulchenko, 2002) [pp. 7-9].

Starting at the top layer, Figure 3.1 shows how Web services can be detected using UDDI. The service provider publishes service descriptions in the UDDI registry. The service requester (client) can then look at development time (static binding) or runtime (dynamic binding) for services in the UDDI registry. Then, the client invokes the service by messaging the service provider directly. Communication is done completely with standardized SOAP messages over SOAP-based middleware. The application objects are

3 Web 2.0 - Social Web

the actual programs involved in calling (service requester) and running the Web service (service provider). Stubs (client-side) and Skeletons (server-side) are responsible for marshalling and unmarshalling the SOAP messages. They act as proxies for the application objects as they are available as local functions but perform remote calls (Alonso et al., 2004) [pp. 123-155].

Figure 3.2 illustrates how the WSDL of a service provider is used to describe services on the description layer. WSDL describes available functions, data, data types and protocols of a Web service. The information is delivered to the client with the help of a WSDL compiler. Communication is also done with standardized SOAP messages over SOAP-based middleware.

As stated previously, SOAP itself resides on the packaging layer. Hence, SOAP is a standardized packaging protocol to allow communication between applications and is based on XML. The specification defines a simple XML-based envelope and certain rules for translating platform-dependent data types into XML-representation. Additionally, SOAP defines conventions about the types of data exchanged, XML-representation of data and how data is to be send. Figure 3.3 shows the SOAP message structure. The SOAP header is an optional element that contains information about routing and delivery settings, authentication and authorization, and transaction contexts. The message itself is packaged into the body. SOAP header and SOAP body are wrapped into a SOAP envelope.

There are two major applications for SOAP:

- Remote Procedure Calls (RPC)
- Electronic Document Interchange (EDI)

SOAP for RPC is a standard application where one program makes a procedure call to a remote procedure passing by some arguments and receiving return values. EDI is designed to exchange documents in business transactions. This technique is often used within Enterprise Resource Planning (ERP) systems (Snell, Tidwell, and Kulchenko, 2002)[pp. 11-13]. The SOAP specification can be found on <http://www.w3.org/TR/soap/>.

3.2 Web Service Standards

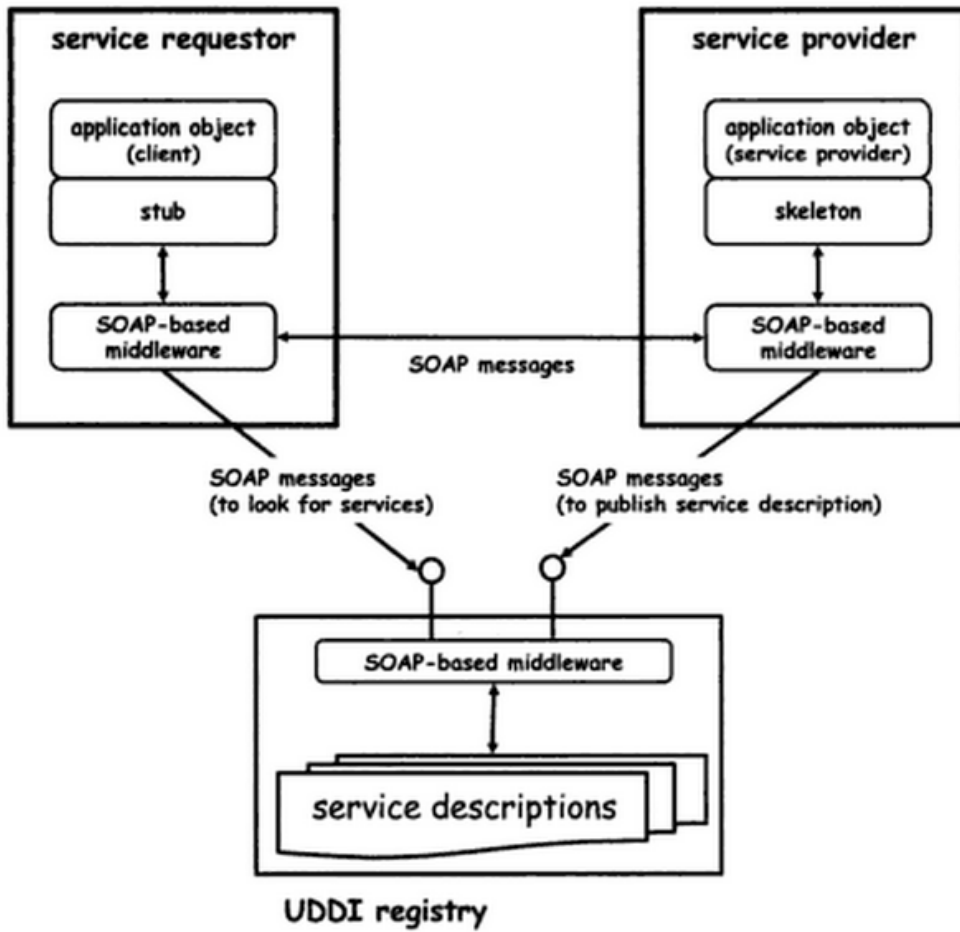


Figure 3.1: UDDI Communication. Adopted from (Alonso et al., 2004)

3 Web 2.0 - Social Web

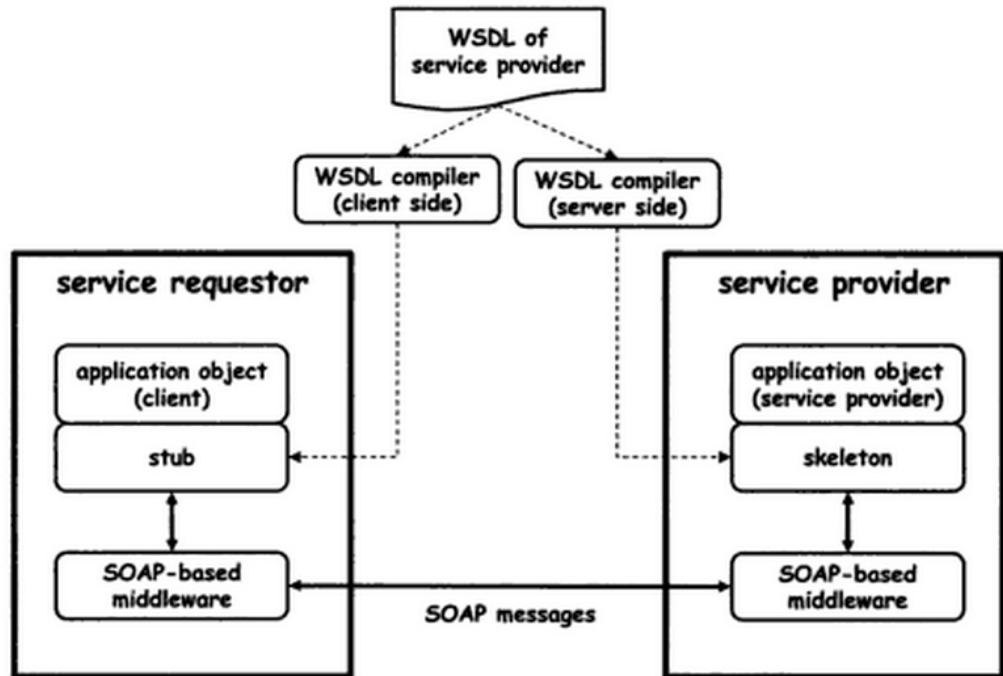


Figure 3.2: WSDL Communication. Adopted from (Alonso et al., 2004)

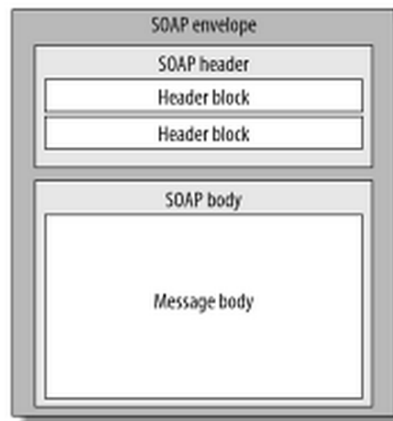


Figure 3.3: SOAP Message Structure. Adopted from (Snell, Tidwell, and Kulchenko, 2002).

3.2.3 REST

REST is currently the state-of-the-art architecture for Web services. It can be grouped into three different Web service architectures that differ in scoping (which Web resource is addressed?) and method information (e.g. GET, POST):

- RESTful resource-oriented
- RPC-style
- REST-RPC hybrid

In the **RESTful resource-oriented architecture** the HTTP method contains the method information (RESTful) and the URI the scoping information (resource-oriented). Consequently, only one URI is necessary for applying different methods on a resource. Some RESTful service providers offer Web Application Description Language (WADL) files. WADL files describe the Web services for RESTful Web services as WSDL files do for SOAP messages. In **RPC-style architectures** method and scoping information are located within the envelope. Therefore, a new vocabulary is probably used for each programme, in contrast to the standardized HTTP methods of RESTful Web services. The only supported method is POST and different methods on the same resource need a different URI, too. XML-RPC is an example of this legacy technology. **REST-RPC hybrid architectures** are a combination of RPC-style and RESTful resource-oriented architectures, mixing the way of how method and scoping information is communicated (L. Richardson and Ruby, 2007) [pp. 13-21].

To put it short, RESTful Web Services have clearly defined standard methods (e.g. GET, PUT) and interactions between stateless resources. Web Services based on REST have four main properties:

- Addressability
- Statelessness
- Connectedness
- Uniform Interface

Addressability means that Web resources are clearly identified by an URI where users can navigate to. This may not be the case within AJAX applications, where Web services are addressable (inside the Javascript client) but

3 Web 2.0 - Social Web

not the Web page itself. Another important point concerning addressability is the representation, which refers to the format of the data. This can be for example XML, JSON, a HTML document or hypermedia. The same resources might have different representations. Resources that do not have an underlying data source must be represented by metadata. The representation should be part of the URI and therefore addressable. As a positive side effect, REST supports clean URLs. Alternatively, the accept header flag in the HTTP request can be set to the requested format. **Statelessness** means that every client request is handled on the server independently from previous requests. This eliminates a lot of possible failures because the server does not need to handle client timeouts. All information is send within a single request. This also allows for easier load balancing, as requests can be distributed to any server without side-effects, and enhances scalability, as the number of servers can easily be increased. Clients can also cache server responses without taking into account previous responses. Cookies may break the statelessness as they represent a certain state on the server. **Links and Connectedness** come into play when users navigate through the Web with links provided by server responses (e.g. from the Google Search Engine). This means that resources are connected by linking to each other within their representations. This makes it possible for users to navigate through the Web without directly entering a Web address into the browser. However, resources in Web services are not always well-connected. Finally, **uniform interface** refers to the ways how resources can be modified. This uniform interface most often includes the following HTTP methods:

- GET
- POST
- PUT
- DELETE

Importantly, RESTful Web services support safety and idempotence. Safety means that requests do not modify resources on the server and is applicable to GET requests. Idempotence means that multiple identical requests lead to the same server answer. This is applicable to the GET, DELETE and PUT methods. Altogether, HTTP methods must be used in the way they are intended to be used and have the same effect on each resource to be RESTful. This requirement is not met when e.g. data is deleted with a GET request. The Uniform Interface property is also violated when POST-requests are

used in RPC-style. This means that method information is somewhere stored within the request message instead of within the HTTP-method header. Notably, the REST specification does not require a specific interface, but it requires that the interface is uniform (L. Richardson and Ruby, 2007) [pp. 79-105].

An example for a RESTful application is the Facebook Open Graph API. The following code is used in Android test code for Open Authentication (Sign-In functionality) with Facebook within the Pocket Code app to simulate different Facebook login behaviours (a detailed explanation is given in the Section 8.1).

This example request below to the Facebook Open Graph API has a clear scope. The scope is `"/app-id/accounts/test-users"`, which means that there is an operation to be performed on a test user. This fulfills the REST-requirement of addressability with a meaningful URI. The request is also stateless, as each request is handled separately on the Facebook server and no state information is provided within the request. The method information within this request is `HttpMethod.POST`, which means to create the request scope (the test user).

```
private void createFacebookTestUser() {
    AccessToken accessToken = new AccessToken(
        configMap.get(UiTestUtils.CONFIG_FACEBOOK_APP_TOKEN), facebookAppId,
        configMap.get(UiTestUtils.CONFIG_FACEBOOK_ID), null, null,
        null, null, null);
    AccessToken.setCurrentAccessToken(accessToken);
    Bundle params = new Bundle();
    params.putString("installed", "true");
    new GraphRequest(AccessToken.getCurrentAccessToken(),
        "/" + facebookAppId + "/accounts/test-users",
        params,
        HttpMethod.POST,
        new GraphRequest.Callback() {
            public void onCompleted(GraphResponse response) {
                try {
                    String jsonResult = response.getJSONObject();
                }
            }
        })
}
```

3 Web 2.0 - Social Web

```
        facebookTestId =
            jsonResult.getString("id");
        String accessTokenTestUser =
            jsonResult.getString("access_token");

        AccessToken accessToken =
            new AccessToken(accessTokenTestUser,
                facebookAppId, facebookTestId,
                null, null, null, null, null);
        AccessToken.setCurrentAccessToken(accessToken);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    facebookTestUserCreated = true;
}
}
).executeAndWait();
}
```

The representation of Facebook messages is in JSON format. A sample response from the Facebook Graph API server after successfully creating a test user is:

```
{Response: responseCode: 200, graphObject:
  {"id": "125493624464409",
   "access_token": "CAAWxDpi3dEIBADb7ltfr0AZCHXkoZBcpzkFdZC0UompKB5Md
                   2ZAYSm7Vh9Eg0ULUv24ZBNeocL2ndXDnGbu3ntsaGtKfEFRPY
                   ZA5RAK9mg2iitpIvhn1oSyY0C4UvWVHFi7v1zk6cf9fe5d1nD
                   sY6ZAW4b9kW5lPeckQRAs7fNkHhsMegpBb5SFeaHH1P1L1UuU
                   gw4fmjN0wwZDZD",
   "login_url": "https://developers.facebook.com/checkpoint/
                test-user-login/125493624464409/",
   "email": "mewdxtx_rosenthalwitz_1440601876@tfbnw.net",
   "password": "1920546548"},
  error: null}
```

The function `deleteFacebookTestUser()` shows how to delete a previously created test user with a given Facebook ID via the Graph API. The scope is a bit tricky in that case. In order to be RESTful the scope should be the same as above, namely `"/app-id/accounts/test-users"`. However, DELETE-requests are not allowed using this edge of the Graph API as stated in the Facebook documentation⁶. According to Facebook, DELETE-requests for test users must have the scope `"/test-user-id"`. The method information for this request is `HttpMethod.DELETE`.

```
private void deleteFacebookTestUser() {
    new GraphRequest(
        AccessToken.getCurrentAccessToken(),
        "/" + facebookTestId,
        null,
        HttpMethod.DELETE,
        new GraphRequest.Callback() {
            public void onCompleted(GraphResponse response) {
                facebookTestUserDeleted = true;
            }
        }
    ).executeAndWait();
}
```

Concerning the "Links and Connectedness" requirement of RESTful applications, test users are a good example, as users can be seen as the most important edges in social networks. The Facebook user ID, that was retrieved by creating a test user, can be used for example to retrieve this user's friends via the scope `"/user-id/friends"` (despite the newly created test user will probably have no friends) or the user's profile simply via the scope `"/user-id"` ("me" is a shortcut for the Facebook user ID of the currently authenticated user). In the below example, ID, name, locale and email of a Facebook profile are requested:

```
public void getFacebookUserInfo(AccessToken accessToken) {
```

⁶Facebook, 2015c.

3 Web 2.0 - Social Web

```
GraphRequest request = GraphRequest.newMeRequest(
    accessToken,
    new GraphRequest.GraphJSONObjectCallback() {
        @Override
        public void onCompleted(
            JSONObject object, GraphResponse response) {
            if (onGetFacebookUserInfoCompleteListener != null) {
                onGetFacebookUserInfoCompleteListener.
                    onGetFacebookUserInfoComplete(response);
            }
        }
    });
Bundle parameters = new Bundle();
parameters.putString("fields", "id,name,locale,email");
request.setParameters(parameters);
request.executeAsync();
}
```

Finally, there is an issue concerning Statelessness in the Facebook Graph API. There is the possibility to make Batch Requests on Facebook. This allows to request several operations with potentially different scopes and method information. By default, Facebook executes these operations in parallel. However, it is possible to specify dependencies between these operations, so that Facebook executes these specified operations sequentially. For example, such a sequential batch request might be to write a new post to a user's feed and then retrieve the top item of this user's news feed. From a certain point of view, this violates the statelessness requirement of RESTful applications. However, Facebook executes all requests on the server and then returns a single result to the requesting application. This prevents from client timeouts, allows for easier load balancing and enhances scalability⁷. Summarizing, the Facebook Open Graph API is generally a good example of a RESTful API, despite there may be some issues in scoping and addressability worthy of discussion.

⁷Facebook, 2015b.

3.3 Web 2.0 Technologies

This section describes the fundamental technologies that made the success of Web 2.0 applications possible. These technologies are based on the aforementioned Web service standards.

3.3.1 AJAX

An often used technology for dynamic Web pages is AJAX. AJAX is basically a Web service client that runs inside a browser with an AJAX engine. Normally, AJAX is based on Javascript as browser-side language and XML as medium for data interchange and manipulation. However, other languages such as VBScript or representations, such as JSON are possible. The power of AJAX lies in the dynamic display of Web content and interaction between client and server. When a user triggers an action an asynchronous HTTP request to some URI on the server is made by the embedded script with an XMLHttpRequest. Users can navigate without restriction on the Web site in the meantime. When the server response arrives, the user's view is usually updated by modifying the Document Object Model (DOM). The Web site presentation is usually done using XHTML and CSS standards (Hauser, 2007) [pp. 317-370]. Figure 3.4 compares the workflow of the classic Web application model with the AJAX Web application model. The downside of AJAX is that it destroys addressability and statelessness, which are main properties of RESTful applications. This is because every application state has the same URI (L. Richardson and Ruby, 2007) [p. 317].

AJAX is a technical basis for Rich Internet Applications (RIA). Computational power for UI rendering and handling of user input is used client-side, but data resides on the server. In comparison to traditional Web applications, where all calculations are done on the server, RIA are much more powerful - almost as powerful as native desktop applications (Stocker and Tochtermann, 2010) [pp. 6-7].

3 Web 2.0 - Social Web

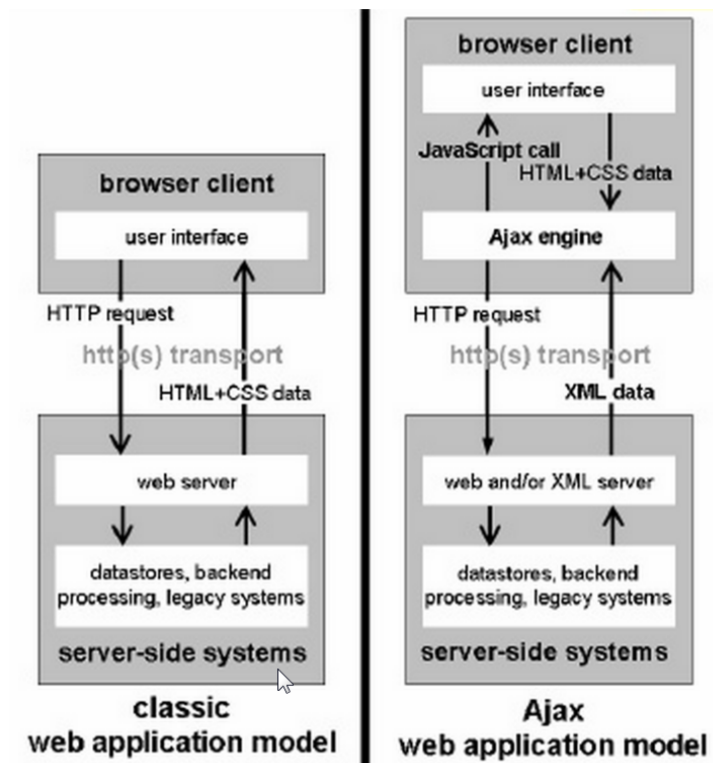


Figure 3.4: Classic vs. AJAX Web application model. Adopted from (Stocker and Tochtermann, 2010).

3.3.2 Mashups

Mashups are important Web 2.0 features that allow to combine multimedia content from diverse sources. This includes remixing of music, images, videos, text or other data. Mashups can be created by using Application Programming Interfaces (APIs) of other Web applications. For example, the Google Maps API allows to embed Google maps⁸ and the Youtube API allows to embed Youtube videos⁹ into Websites. Actually, mashups are extremely popular in combination with geotagging where photos are tagged with their geographical coordinates. An example application is Google Earth with embedded and geotagged photos. Figure 3.5 shows an example where climate activists spelled out 'Climate Justice Peace' during the U.N. Climate Conference 2015 in Paris using geolocation with their mobile phones and Google Maps¹⁰. Mashups may use any of the Web service standards previously described for communicating with the server APIs. Mashups are very powerful on top of huge free data sources such as flickr or other freely available data sources (e.g. with Creative Commons licence). Altogether, mashups are a core feature of Web 2.0 as they strongly promote user-generated content (Carl et al., 2008) [pp. 1-17].

The power of mashups lies in the combination of content from diverse sources to create new Web services with data that has not been available from a single data source. The challenge is to give users the possibility to create mashups (that e.g. combines different RSS feeds) easily without requiring deep technical knowledge (Stocker and Tochtermann, 2010) [pp. 16-17].

3.3.3 Content Syndication and Aggregation

Content syndication refers to the multiple use of content on the Web. A Web site that provides content is referred to as "content syndicator". Websites that take the content from the syndicating Web site are referred to as "content aggregators". The benefit is that users do not need to navigate to

⁸Google, 2015c.

⁹Google, 2015h.

¹⁰Earth Group, 2015.

3 Web 2.0 - Social Web



Figure 3.5: Climate Justice Peace Geolocation Mashup in Paris. Adopted from (Earth Group, 2015).

the Website of the content syndicator, but see the content directly on the Website of the content aggregator. Content is automatically refreshed when the underlying data on the syndicating Web site changes. The most often used data format for content syndication and aggregation is a feed. For example, news feeds deliver current news to users that are subscribed to the syndicating Web site. This content may be read with a Desktop or Web-based feed reader. Other examples are social networks such as Facebook or Twitter that syndicate user activities, blogs or podcasts. Therefore content from diverse sources can be combined and automatically refreshed every time it changes. The most popular standard formats for feeds are Really Simple Syndication (RSS) and the newer ATOM format. The RSS 2.0 specification can be found on <http://cyber.law.harvard.edu/rss/rss.html> and the ATOM specification on <https://www.ietf.org/rfc/rfc4287.txt>. Both are based on XML (Stocker and Tochtermann, 2010) [p. 15].

The aim of RSS and ATOM is to provide a content syndication feed, that is a machine-readable XML-file. The file content may include headlines, links, stories or the entire content of a Website (without layout data). In addition, the newer ATOM format supports diverse MIME types (e.g. plaintext or HTML). This enables content syndication across platforms and devices and pushing of updates on demand. The advantages of content syndication in-



Figure 3.6: Podcast creation and publication process. Adopted from (Farkas, 2006).

clude higher traffic to the syndicating Website, creation of brand awareness, search engine optimization as well as improved relationships between users, the Website and Website communities (Hammersley, 2005) [pp. 1-11].

Figure 3.6 shows the workflow of content syndication of a podcast (podcasts are explained in Section 3.4.3). Prerequisites are that a podcast was created and a Web site that hosts the podcast exists. This Website syndicates the podcast via RSS or Atom feeds to the Web, where the podcast can be aggregated to diverse sources (Farkas, 2006) [pp. 101-117].

3.3.4 Folksonomies - Social Tagging

Folksonomies are used in Web 2.0 applications to categorize and find content. In reality, folksonomies can be located between Web 2.0 and Web 3.0 technologies. In Web 3.0 applications taxonomies and ontologies are most often used to categorize content and give them a meaning. The major difference is that folksonomies are user-generated and ontologies and taxonomies are rather created by professionals. Taxonomies and ontologies will be discussed in Chapter 4. In the following, folksonomies and their applications are explained.

As stated before, folksonomies offer a functionality to structure content of communities and make it searchable and retrievable. Popular communities that use folksonomies are for example flickr (photo sharing), YouTube (video

3 Web 2.0 - Social Web



Figure 3.7: Web 2.0 tag cloud. Adopted from (Stocker and Tochtermann, 2010)

sharing) or del.icio.us (social bookmarking). The term “Folksonomy” is a combination of “folk” and “taxonomy” and means that people create their own taxonomy. Often this is referred to as the “wisdom of the crowds”. Technically, in folksonomies any shared information object can be annotated by tags or keywords. These annotations can be done by any user of the community and are a form of metadata. There are no guidelines on how information objects have to be annotated. There is also no rule on how many annotations an information object receives. This process is called “Social Tagging”. The totality of tags on a platform is the vocabulary that forms the folksonomy and is often visualized by a tag cloud, as depicted for the term “Web 2.0” in Figure 3.7 (Peters and Becker, 2009) [pp.153-156].

Depending on how often tags are used, they are depicted in a larger or smaller font. If a user clicks on a term of the tag cloud, all information objects that were annotated by this tag are displayed. The underlying principle is that the amount of tags forms the semantics of information objects. These tags are also relevant to the results of search engines. Social tagging enables

building of social networks because users that annotate information objects similarly can find each other easily. This is also true for users that annotate the same objects of interest. This allows to build a graph that visualizes relationships between users.

Some of the problems of the free tagging process are that categories tend to be splitted because tags may be used in singular and plural or in different languages by different users. Additionally, the same term, such as "Apache" may have different meanings - in this case the term may refer to the Web server or to the native Americans. To deal with these issues many Social Applications offer an auto-completion with already used tags during the tagging process. Often, there is also an indication on how often tags were already assigned because users tend to rather assign often used tags. This counteracts the splitting of tags (Stocker and Tochtermann, 2010) [pp. 8-11]. Summarizing, folksonomies are collections of user-generated tags in specific communities that allow to search for and retrieve content.

3.3.5 Microformats, RDFa and Microdata

Similarly to folksonomies, microformats incorporate social components of Web 2.0 and semantic components of Web 3.0. The goal of microformats is to add richer semantics to HTML Web sites so that software can better grasp the meaning of Web content. Unlike Semantic Web technologies such as RDF or ontologies, which will be discussed in Chapter 4, microformats do not require developers to learn new technologies. Microformats build directly on established standards. For example, certain HTML class attributes give a meaning to Web content as specified on the community Website <http://microformats.org>. An disadvantage in comparison to Semantic Web technologies is that there is a more limited set of applications. The most important applications are:

- hCard: Marks up contact information of people and organizations.
- hCalendar: Marks up event information (seminars, conferences etc.).
- XFN: Marks up relationships between people (friendships etc.).
- hReview: Marks up published reviews on the Web.
- hResume: Marks up published reviews and CV's on the Web.

3 Web 2.0 - Social Web

- GEO & ADR: Marks up geolocation data.
- hAtom: Marks up syndicated content.

When developing a microformat there are some underlying principles. First, microformats should address specific problems and are thereby designed “for humans first” and “for machines second”. The design should be as simple as possible, established building blocks of widely adopted standards should be reused, microformats should be embeddable and modular, and, finally, decentralized development, content and services should be encouraged. Decentralization in this sense is intended to enable distributed applications such as social bookmarking or Yellow Pages where no centralized service like del.icio.us is necessary anymore. Once again, the flickR photo sharing service is a good example, as it uses hCard in people’s profiles and GEO for describing where a photo was taken with its longitude and latitude (Allsopp, 2007) [pp. 1-20].

Microformats can be transformed into RDF or OWL formats, which are described in Chapter 4, so that Semantic Web agents can automatically process them (Stocker and Tochtermann, 2010) [p. 11]. A simple example of a hCard with the current microformats2-format is as follows (adopted from¹¹):

```
<p class="h-card">
  
  <a class="p-name u-url" href="http://example.org">Joe Bloggs</a>
  <a class="u-email" href="mailto:joebloggs@example.com">
    joebloggs@example.com</a>,
  <span class="p-street-address">17 Austerstræti</span>
  <span class="p-locality">Reykjavík</span>
  <span class="p-country-name">Iceland</span>
</p>
```

Summarizing, microformats are a simple set of specified data formats that build directly on widely adapted Web technologies to give a meaning to Web content.

¹¹Microformats, 2015.

3.3 Web 2.0 Technologies

Alternatives to microformats are RDFa and microdata. Despite the advantage of microformats of its ease-to-use it requires Web developers to rename their Web page elements such as `<div>` and `` as well as CSS classes to fit the specific formats. RDFa offers a bit more flexibility as it is build on XHTML (Extensible HTML). XHTML allows to add namespaces without breaking existing Websites. RDFa added the attributes "property", "role", and "about" for data annotation. Consequently, it is not necessary to change Web elements or CSS in the source code. However, Websites should be well-formed under the more rigid XHTML validation, so that search engines can interpret the content without problems. The newest standard is Microdata. Microdata reuses many already specified class names of microformats (e.g. vCard) and is a new component within the HTML5 standard (C. S. Smith, 2015).

The following simple example of RDFa code shows how the name, telephone number and homepage of a person can be annotated with a given schema (adopted from¹²).

```
<p vocab="http://schema.org/" typeof="Person">
  My name is
  <span property="name">Manu Sporny</span>
  and you can give me a ring via
  <span property="telephone">1-800-555-0199</span>
  or visit
  <a property="url" href="http://manu.sporny.org/">my homepage</a>.
</p>
```

A relatively new addition in the Google search engine are Rich Snippets for products, recipes, reviews, events and software applications. Rich Snippets search for semantic information in crawled Web pages, where RDFa, microdata and JSON-LD are supported. This enables Google to better index content and display detailed information in search results. For example, Figure 3.8 displays a search result for the terms "Catrobat" or "Pocket Code" with a snippet describing the app including a link to Google Play,

¹²W3C, 2015a.

3 Web 2.0 - Social Web

Pocket Code - Google Play

<https://play.google.com/store/apps/details?id=org.catrobat.catroid...> ▼

★★★★★ Rating: 3,5 - 2.598 votes - Free

Pocket Code allows you to create, edit, execute, share, and remix Catrobat programs in a visual, "LEGO-style" programming environment and programming ...

Figure 3.8: Google Rich Snippet. Adopted from (Google, 2015e)

the aggregated rating value of the app at Google Play, the total number of votes and a hint that it is a free app derived from semantic annotations¹³.

3.3.6 Cloud Computing

Another important Web 2.0 technology is cloud computing. Cloud computing combines various existing technologies as illustrated in Figure 3.9 to offer cloud services.

Cloud computing fosters some important Web 2.0 principles as it offers high scalability of software infrastructures and platform independence. Furthermore, it allows to save data on remote servers in order to access it from anywhere and promotes lightweight programming models by offering Web services with standardized protocols hiding the complexity behind them. Cloud computing can also be easily integrated with Extreme Programming (XP) practices and enables shorter time-to-market (Guha and Al-Dabass, 2010) [pp. 1-6].

In a layer-perspective cloud computing is a technique that builds upon existing technologies such as virtualization as shown in Figure 3.10.

Data-Storage-as-a-Service (dSaaS) is at the bottom of the architecture and provides the necessary data storage that is used by customers in the cloud. Next, a **virtualization** layer allows to share the same servers and resources between different operating systems or applications. **Infrastructure-as-a-Service** (IaaS) offers computing resources with guaranteed bandwidth, processing power and data storage to customers. **Platform-as-a-Service** (PaaS)

¹³Google, 2015e.

3.3 Web 2.0 Technologies

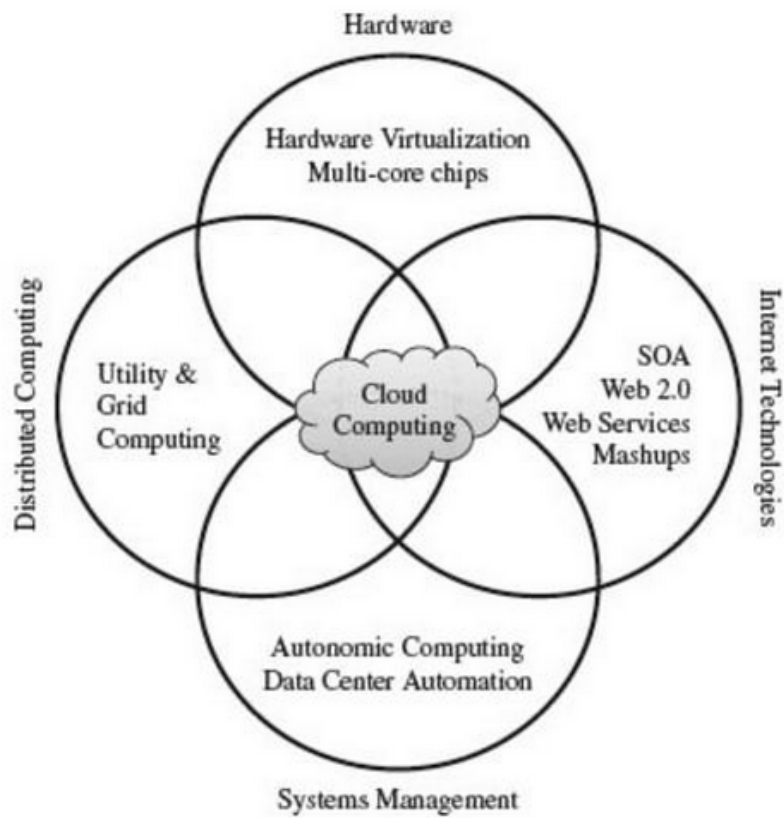


Figure 3.9: Basic Technologies of Cloud Computing. Adopted from (Buyya, Broberg, and Goscinski, 2010).

3 Web 2.0 - Social Web

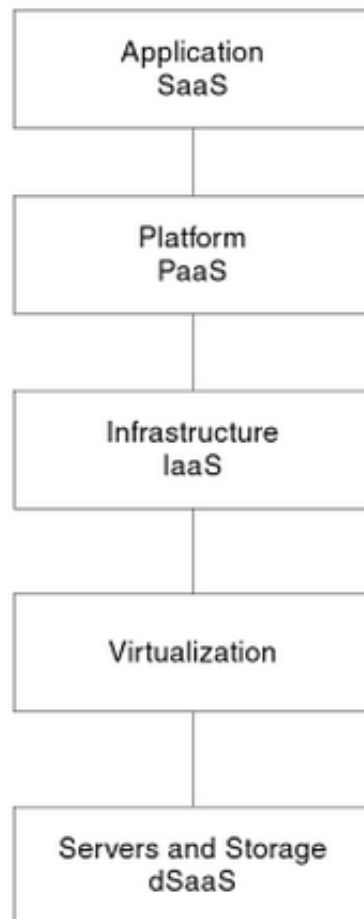


Figure 3.10: Layered Architecture of Cloud Computing. Adopted from (Borko Furht and Escalante, 2010).

3.4 Social Applications

is a service that usually enables access to virtualized operating systems with integrated development environments (IDE) for e.g. Java, PHP or C++. Finally, **Software-as-a-Service** (SaaS) is on top of the stack and provides access to various remote IT services and applications that directly run on the cloud (Borko Furht and Escalante, 2010) [pp. 4-9].

There are three different network scope types of cloud computing:

- Public cloud
- Private cloud
- Hybrid cloud

The public cloud runs on the Internet and offers Web applications and services that are publicly available by third-party providers. Private clouds run in private networks, e.g. business Intranets, and therefore allow for better control over data and higher security. Hybrid clouds are a combination of public and private clouds (Borko Furht and Escalante, 2010) [p. 7].

3.4 Social Applications

This section describes the most important social applications that are based on the technologies discussed in the previous sections. Namely, some of these social applications are blogs, microblogs, wikis, podcasts, videocasts, messengers, social networks, social bookmarks, social news, RSS feeds, media sharing platforms, online communities or forums.

3.4.1 Blogs and Micro-Blogs

Weblogs, simply called Blogs, are Web applications that contain periodic posts about certain topics on a Web page. These posts are generally publicly available and in reverse chronological order with the most recent post on the top. The term blogosphere is commonly referred to as the totality of all blogs and bloggers. Some common features of blogs are that posts have a timestamp (important for the working of content aggregation), posts typically remain unchanged over time, blog content includes links to other sites as

3 Web 2.0 - Social Web

well as multimedia content and commentary on posts can be automatically attached to the original post. Blogging also had a great impact on traditional media as it delivers information already when events have just occurred and because information can be spread by other bloggers.

There are three types of blogs:

- Individual blogs
- Subject blogs
- Organizational blogs

Individual blogs (the original blog style) are written by individuals and are personal by nature. These blogs are like online journals containing stories about a blogger's life or professional experience. Subject blogs are written by one or more authors and focus on particular topics such as traveling, politics or sports. Finally, organizational blogs are written by organizations and represent their views on certain topics (Sauers, 2006) [pp. 1-8].

Technically most blogs build upon Content Management Systems (CMS) and already provide many sophisticated templates to bloggers. Therefore, bloggers do not need to have a technical understanding. Many systems are programmed in simple PHP language with a MySQL database in the back-end. Blogging software can either be installed directly on servers or being accessed by application service providing (ASP). Blog entries also have so-called permalinks (permanent link). Permalinks are URL's that uniquely identify blog entries and enable linking of them on other Web pages. Vice versa, trackback functionality enables to get information about where blog entries have been linked from. This is done by a central and automated notification service of blogging providers and enabled the interlinking of single blogs to the entire blogosphere (Stocker and Tochtermann, 2010) [pp. 3-4].

A newer trend which developed out of blogs are micro-blogs. Micro-blogs are like blogs, but posts are rather short compared to blog entries. Postings are generally public and people can subscribe - or follow, to say it in Twitter terms - to news feeds of persons or organizations of interest to see their postings. These services are used as fast information provider or to communicate with friends and other persons or organizations of interest¹⁴.

¹⁴LMZMediaculture, 2015.

3.4 Social Applications

News feeds and content syndication have already been discussed in Section 3.3.3 as well as the possibility to categorize postings with hashtags (social tagging) in Section 3.3.4. The most popular provider of micro-blogging is Twitter. Twitter will be discussed in Section 7.3.1.

3.4.2 Wikis

Wikis are platforms that allow users to elaborate on topics of interest in a collaborative manner. These topics can be read, written and edited online by all users. There are no sophisticated user permission systems. The documents generally have hyperlinks to each other and are linked in that way together. The name "Wiki" is derived from Hawaiian "wikiwiki" and means "quick" and refers to the fact that people can add content to the Wiki at any time. Wikis generally have an integrated version management system to keep track of modifications and to be able to roll back to previous document versions. Important to be aware of, wiki entries are conceptionally the result of collaborative effort, not individual effort, as content can be freely edited, modified and reused by any user (W. Richardson, 2010) [pp. 55-58].

Popular wikis are MediaWiki (www.mediawiki.org) or, of course, Wikipedia (www.wikipedia.com). MediaWiki is a software that can be used with own server installations or via ASP. Contents created by users are stored in a database on the server (Stocker and Tochtermann, 2010) [pp. 4-5].

3.4.3 Podcasts and Videocasts

Chronologically, podcasts evolved after Blogs and Wikis as social applications and focus not on text but on audio content. The concept is easy - users should be able to listen to radio programs, magazines or other audio content platform-independently when and where they want. This is done by broadcasting audio content via Web to any interested person with a media device. Originally, the term "Podcast" was the combination of the terms "iPod" and "broadcasting". Besides podcasts, videocasts evolved with a focus on video broadcasting. Podcasts as well as videocasts promote user-generated content

3 Web 2.0 - Social Web

in that users are encouraged to broadcast their own multimedia content. Technically, podcast and videocast content is distributed in the Web with RSS or ATOM syndication, which has been described in Section 3.3.3. By using content syndication and aggregation, content gets available instantly to subscribed users. Popular broadcast formats are MP3 for podcasts and MP4 for videocasts (Farkas, 2006) [pp. 1-6].

There are some Web platforms with comprehensible lists of podcast sources by topic. From a social perspective, podcasts can be seen as an evolution of the radio. However, in comparison to blogs, the usage of podcasts is stagnating. Possible causes are that technical understanding is necessary to create podcasts and that podcasts demand more concentration by users, especially for long discussions (Stocker and Tochtermann, 2010) [p. 6].

3.4.4 Social Networks

Social networks are clusters of people connected for specific reasons, such as a group of friends or interest groups for certain topics. Some of the most important characteristics of social networks are, that they allow for instant communication over any distance or sharing of information and multimedia. Actually, many of the previously described technologies and social applications could be part of a social network. Social networks can be browser-based (e.g. Facebook), Desktop/client-based (e.g. Twitter messaging client) or mobile applications (e.g. Facebook app). Technically, the server of the social network provider collects and stores information and multimedia items from users. These information and multimedia items are shared with other users in the social network, based on the permissions granted by the sharing user (Ryan, 2011) [pp. 4-8]. There are three main types of communities in social networks with different focuses. These focuses can be on products, contents or activities (Bell, 2009) [pp. 1-2].

Typical main features of Social networks are:

- Creation and administration of a profile
- Management of contacts
- Instant messaging
- Event notifications (e.g. profile changes)

3.4 Social Applications

- Searching for network data (e.g. information and media of other users)
- Event organization
- Group calendar
- Various integrated tools (Wikis, blogs, newsfeeds, forums etc.)
- Widgets

There are two basic methods to identify and evaluate data in social networks. These methods are Social Media Monitoring and Social Network Analysis. Social Media Monitoring is used to discover (often hidden) data in social networks in order to use it for market research or competitor analysis. For example, the scope of monitoring might be to:

- measure reputation or image of products, services or companies
- discover market trends early
- achieve targeted advertising effects
- conduct Word of Mouth (WOM) studies
- perform issue management

The goal of WOM studies is to automatically assess moods of statements (e.g. posts, integrated blogs or forums) on basis of semantic analysis. The results are measurements, that indicate whether statements are either positive or negative. Of course, there is some error degree, but as measuring techniques get continuously better, results get more and more accurate, too.

Issue management, on the other hand, is used to to discover topics and trends that have recently been discussed in social networks. This information may be used to recognize market trends early (Roebbers and Leisenberg, 2010) [pp. 118-121].

The goal of social network analysis is to understand structures in social networks through the use of network and graph theories. Insights derived from social network analysis can be used for data aggregation, data mining or recommender systems, to name a few important applications. According to these theories social networks are constituted by nodes (e.g. individual actors, people or things) and edges (relationships or interactions) that connect the nodes. Social network analysis is based on relational data (nodes and edges) directly available in social networks, that can be assessed statistically (Scott, 2012) [pp. 1-10].

3 Web 2.0 - Social Web

There are three main types of different networks. These types are ego-centric (networks connected via a single central node), socio-centric (well-connected networks - e.g. specific groups) and open system networks (boundaries not clear). Key concepts used to assess network structures are :

- Connections
- Distributions
- Segmentation

First, connections describe how single nodes are connected to each other. The most important measures are:

- **Propinquity:** Measures the geographical proximity between nodes (nodes are more likely to be connected to each other when they are geographically near to each other).
- **Homophily:** Measures the extent to which similar nodes (e.g. people with same age, gender, race or interests) form ties.
- **Mutuality:** Measures the extent to which two nodes reciprocate each others friendship and interactions (concept of give and take).
- **Multiplexity:** Measures relationship strength in terms of number of different ties (e.g. people who are friends and have the same age have a multiplexity of two).
- **Balance:** Measures the tendency that a third node in a triad (group of three people) complement the other two nodes' characteristics.

Second, distribution measures are based on the whole network structure, as opposed to connection measures. The most important measures are:

- **Number of diads** (groups of two) **and triads** (groups of three) in the network.
- **Density:** Measures whether a network is well-connected by comparing the existing number of ties to the total number of possible ties in a network.
- **Structural hole:** Measures the lack of connections, as opposed to the density measure. A structural hole can be closed by a bridge or weak tie (individual providing the only edge between nodes).
- **Centrality (popularity):** Collection of different measures to assess the influence of particular nodes within a network (e.g. nodes with many connections serve as links between nodes).

3.4 Social Applications

- **Distance:** Measures the minimum number of edges between specific nodes and calculates measures over the whole network.

Finally, segmentation is used to partition networks into groups and specifies connections and boundaries of these groups. Two important types of network segments are cliques and clusters. Cliques are a maximal complete subgraph of at least three nodes. This means that all nodes are linked to each other. Clustering describes grouping of a network into smaller communities (e.g. based on density measures), which may overlap or not. There are also other segment types such as circles, clacks, cabals, coalitions, groups or blocks. Important measures for segmentation are cohesion and structural similarity (Kadushin, 2011) [pp. 13-48].

Popular graphs to visualize the structure of social networks are sociograms, as shown in Figure 3.11.

Another important application of Social Network Analysis is in the mobile environment. There are some approaches to provide useful and relevant contents to users anytime and anywhere. The proposal of contents is based on current user context, which includes physical (e.g. current user location) and conceptual (e.g. preferences or social connections of a user) components. These types of contexts are shown in detail in Figure 3.12 (B. Furht, 2010) [pp. 223-224].

Detailed statistics about social networks and other social applications are given in Chapter 7.

3.4.5 Other Social Applications

There are still many other social applications including messengers, social bookmarks, social news, media sharing platforms, social games, online communities or forums. These social applications are not discussed in detail. However, there are some related statistics in Chapter 7.

3 Web 2.0 - Social Web

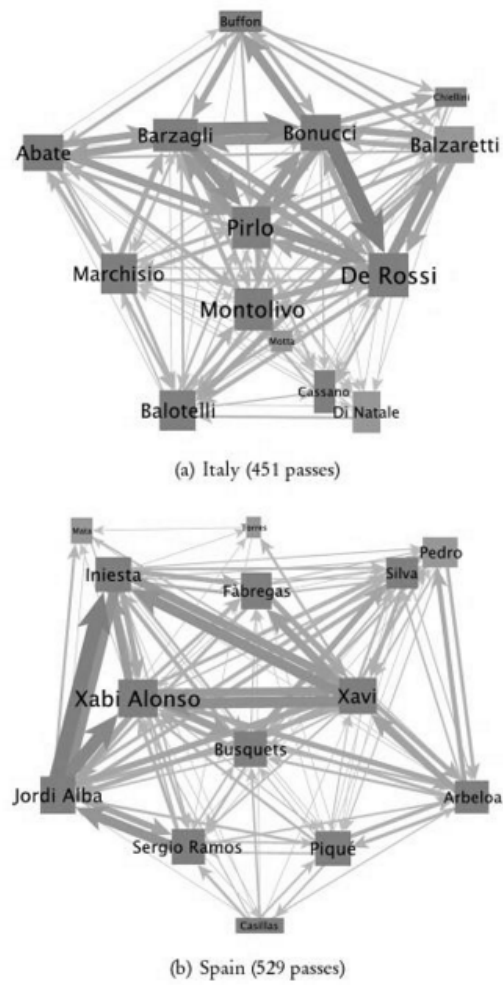


Figure 3.11: Sociogram showing network of completed passes in the UEFA EURO 2012 final Spain vs. Italy. Adopted from (Hennig et al., 2012) [p. 164].

3.4 Social Applications

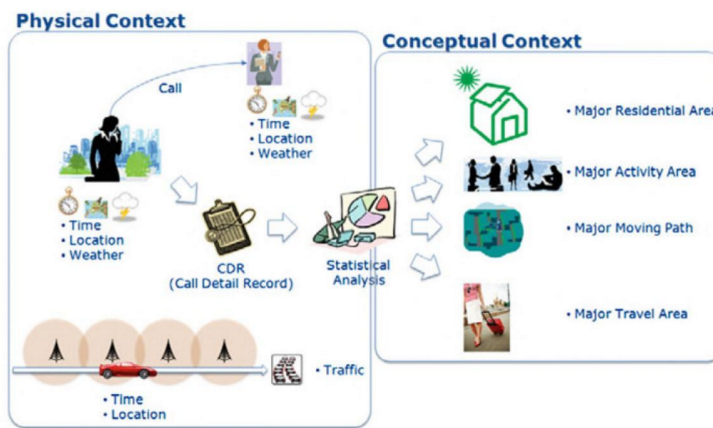


Figure 3.12: Physical and conceptual contexts in a mobile environment. Adopted from (B. Furht, 2010).

4 Web 3.0 - Semantic Web

Web 3.0, also known as Semantic Web or Web of Data, according to the W3C definition¹, is an extension of the Web. First, the Semantic Web in general is explained. Then, the connection to social components of Web 2.0 is discussed.

4.1 Web 3.0 Evolution

The Semantic Web emerged over the past decade as a further development of Web 2.0. The goal of Semantic Web is to create a "Web of data". This means to link data in the World Wide Web (WWW), so that it can be automatically read and analyzed by computers, to deal with the rapidly emerging amount of WWW data. The goal is to facilitate useful computations on Web data and support trusted network interactions. Diverse Semantic Web technologies allow to create Web data stores, build vocabularies and define rules for processing data. The most important technologies are the Resource Description Framework (RDF), taxonomies, Web Ontology Language (OWL) and SPARQL Protocol And RDF Query Language (SPARQL) for querying linked data. Altogether, linked data is considered to be easy-to-process. However, most of data in the Web is still plain text, which has no automatically extractable meaning by itself. Luckily, there are already approaches to convert plain text to linked data (Euzenat and Shvaiko, 2013) [pp. 1-6].

The **Resource Description Framework** is a simple language that describes structured as well as semi-structured data entries in the Web. RDF uses URIs to describe relationships between nodes. Therefore RDF forms a directed and labeled graph, where edges form links between two resources at a

¹W3C, 2015c.

4 Web 3.0 - Semantic Web

time. Generally, those relationships are characterized by the triple subject, predicate and object. The subject is a resource with an outgoing link to the object. The object might be either another resource or a literal. The edge between subject and object is specified by the predicate².

A valid RDF example that describes for example that the song 'Mountain Man' is from the artist 'Andreas Gabalier' may look as follows:

```
<?xml version="1.0"?>
<RDF>
  <Description about="https://www.youtube.com/watch?v=ACCpAgg1Evo">
    <artist>Andreas Gabalier</artist>
    <song>Mountain Man</song>
  </Description>
</RDF>
```

In this example "https://www.youtube.com/watch?v=ACCpAgg1Evo" is the resource that is being described. Resources always have a URI. "artist" and "song" are properties that define certain aspects of the resource. "Andreas Gabalier" and "Mountain Man" are property values or objects. These property values can be literals, URI's or other resources. From a graph perspective, the node "https://www.youtube.com/watch?v=ACCpAgg1Evo" is connected to the nodes "Andreas Gabalier" and "Mountain Man" with the edges "artist" and "song" (adopted from³).

Taxonomies are used to classify, describe and map specific knowledge domains (people, things, ideas etc.) in a hierarchical arrangement. Taxonomies have three key attributes:

1. Classification scheme
2. Semantic
3. Knowledge map

The classification scheme of a taxonomy is used to group related things together. This makes it possible to find other related things or concepts in that category. The semantic aspect of taxonomies means that a fixed

²W3C, 2015b.

³W3Schools, 2015.

vocabulary to describe knowledge and data exists. The vocabulary should be meaningful and transparent. Therefore ambiguous, alternate and unprecise terms tend to be excluded from taxonomies. As a result, changes of terms are carefully considered and are generally done by domain experts. This is very different from folksonomies which are build by ordinary users. Semantics are expressed as relationships between subjects, which may be 'is-a' or 'has-a'. Finally, a taxonomy should be a comprehensive, predictable and easy-to-navigate knowledge map with a clear structure over the knowledge domain. Taxonomies may have many different representation structures such as lists, trees, hierarchies, polyhierarchies or matrices (Lambe, 2014) [pp. 4-10].

More complex knowledge can be represented by ontologies, which consist of various concepts. The definition of a concept is not very precise - it can be a person, an object, an idea etc. Concepts must be connected to each other in order to be useful for knowledge representation in ontologies. Generally, concepts tend to be connected to the process of human cognition and are abstract (do not differentiate between objects where they apply to) and universal (also valid for subconcepts or subclasses). Based on these concepts, ontologies are able to describe more complex relationships between nodes, as RDF or taxonomies can do. Ontologies organize concepts under a specific application domain (Jakus, Milutinovi, and Omerovi, 2013) [pp. 5-24]. A disadvantage of ontologies is the sophisticated knowledge acquisition process. Knowledge is generally gathered from unstructured or semi-structured data sources. The knowledge acquisition process is to a large extent done by humans and may be aided by machine learning techniques. In a typical process, machine learning techniques initially create the ontology. Then, humans review the ontology and add certain assumptions or axioms to convey the intended meaning of the ontology elements (Maedche, 2012) [pp. 1-12].

4.2 Social Semantic Web

A big challenge in the current Web is to close the knowledge gap between what users want and what computers deliver. For example, search engines

4 Web 3.0 - Semantic Web

are programmed in a way that a diversity of most relevant results is returned in the top ten listings. Then, the search query may be detailed by the user to further limit the results. However, search terms should also not be over-specified because not all searched terms may be present on all relevant pages. Automatic image or sound recognition based on text search is even more difficult (Mika, 2007) [pp. 3-13].

Due to these reasons the Social Semantic Web became more and more popular over time. Figure 4.1 depicts the evolutionary path towards the Social Semantic Web. First, the World Wide Web (Web 1.0) provided the necessary foundation of the Web and allows for hyperlinking between Web pages. Two different paths led to the Social Web and the Semantic Web. The Social Web strongly fosters user-generated content in user-visible front-ends, and has its success in a social perspective. Typical applications are Wikis, blogs or social networks, which have been discussed in Section 3.4. The Semantic Web, on the other side, is positioned in the back-end and gives the Web a meaning and a possibility to analyze data easily with technologies such as RDF or OWL, as discussed in Section 4.1. The Social Semantic Web is an approach to bring the Social Web and the Semantic Web together. The goal is to be able to analyze, customize and reuse social data easily by enriching it with semantics. This means to combine the strict organization of data of the Semantic Web with the scalability and authorship advantages of the Social Web. Prominent examples are Semantically-Interlinked Online Communities (SIOC), DBpedia or Friend of a Friend (FOAF), which will be subsequently described (Breslin, Passant, and Decker, 2009) [pp. 14-15].

FOAF is an information system that originated in 2000 and describes link-ages between people-related terms with aid of structured data (e.g. RDFa, JSON-LD, Linked Data) using a RDF-XML vocabulary. Such people-related terms can be names, occupations, interests, or group memberships, for example. The goal is to enable interlinking of data across different social networks. FOAF data can be automatically generated from profile pages of some social networks. There are already FOAF exporters for the popular social networks Facebook, Twitter and Flickr available⁴. SIOC is an information system that provides an ontology to describe information from online communities, such as blogs or wikis. The goal is to interlink data of these

⁴FOAF-Project, 2015.

4.2 Social Semantic Web

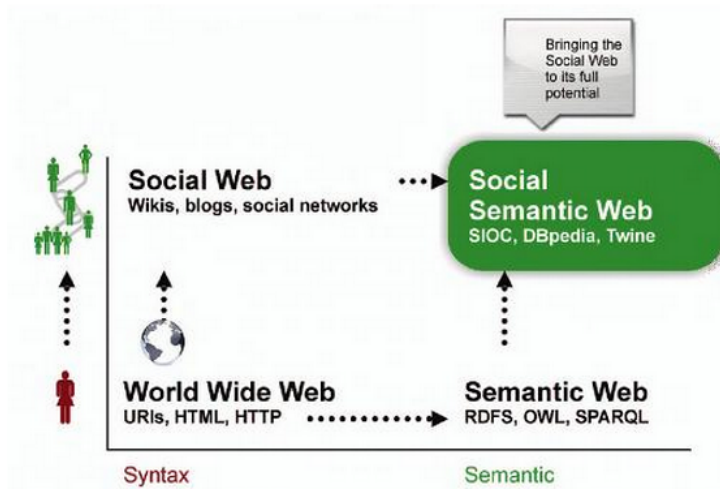


Figure 4.1: Social Semantic Web Dimensions. Adopted from (Breslin, Passant, and Decker, 2009).

online communities semantically. By now, SIOC is used in many free and commercial software applications and often combined with FOAF data to additionally express profile and social network information. There are exporters for communities such as Twitter (microblogs) or Wordpress (blogs) available⁵. DBpedia is an initiative with the goal to extract semantically interlinked data from Wikipedia and make it publicly available⁶.

Social Web and Semantic Web technologies experienced a strong convergence. On the one side, "Semantically Enabled Social Software" focuses on enriching Social Software with semantic functionality in order to generate structured data. On the other side, the "Socially Enabled Semantic Web" focuses on making semantic data available to the public to enable rich content Web applications. The aim is to provide machine-readable data to be able to understand and analyze "Social Data". Prominent examples are FOAF, DBpedia and SIOC. There are different approaches to add semantics to the Social Web. The W₃C developed top-down approaches, whereas critics claimed these approaches are too hierarchical and complicated. In reality, a combination between top-down and bottom-up approaches may

⁵SIOC-Project, 2015.

⁶DBpedia, 2015.

4 Web 3.0 - Semantic Web

be necessary to deal with the large and dynamically changing amount of data.

A big step towards convergence was taken with the "Open Data" initiative. Open Data is a huge distributed and freely available database containing highly-structured data. The "Open Dataset" has sources like Wikipedia or GeoNames database. Companies such as Facebook, Google, Microsoft, LinkedIn, Mozilla, Flickr or Yahoo! joined the Data Portability Group, which has the goal to understand technical, social and legal implications and utilize data transfer between Social Web platforms (Blumauer and Pellegrini, 2008) [pp. 3-22].

Consequently, Semantic Web technologies with agreed upon formats were ideally suited to describe people or objects-of-interest of social networks or other Social Web applications and enable analysis as well as re-usability of data. This in turn enables users to access multiple heterogeneous social Web sites easier and facilitates content-creation. Projects such as FOAF or SIOC, as described earlier, build on that idea. Figure 4.2 shows a prominent example of a Socially Enabled Semantic Web application - flickR. flickR collects person- and object-related data from diverse social Websites and links it all together. This is done by giving the data a meaning with the use of RDF, OWL, SPARQL and metadata technologies.

The other way round, Figure 4.3 shows how semantic applications can be enriched with social components such as Mash-Ups or building of an architecture of participation to foster user-generated content. Ultimately, social data gets interlinked across social platforms in a machine-readable way. This large interlinked graph allows to apply sophisticated queries to gain valuable insights within the social data. Summarizing, technologies such as folksonomies, RSS or ATOM-based feeds, geotagging or microformats and initiatives such as FOAF, SIOC or DBpedia strongly support the Social Semantic Web. Popular areas in which social applications can be enriched with semantics are social networks, online communities, knowledge and information sharing, multimedia sharing or folksonomies (Breslin, Passant, and Decker, 2009) [pp. 1-5].

Finally, a new Social Semantic Web standard for Linked Data, JSON-LD, is discussed. The vision of JSON-LD is to finally enable the shift "from strings to things" in the Web.

4.2 Social Semantic Web

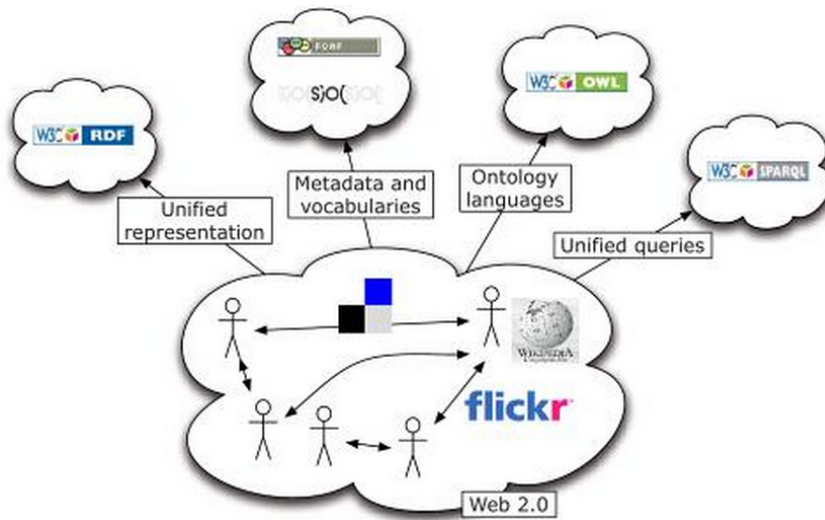


Figure 4.2: Socially Enabled Semantic Web: FlickrR. Adopted from (Breslin, Passant, and Decker, 2009).

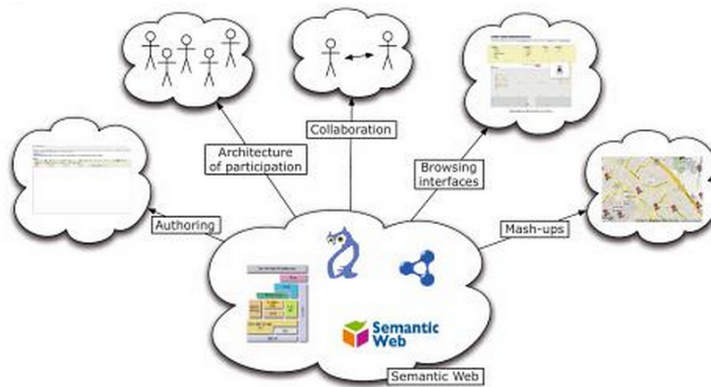


Figure 4.3: Semantically Enabled Social Software. Adopted from (Breslin, Passant, and Decker, 2009).

4 Web 3.0 - Semantic Web

The goal of JSON-LD is to create a network of Linked Data across Web sites that is standardized, machine-readable and human-readable. Linked Data empowers people that either publish or use information on the Web to navigate through the information by following the embedded links across the entire Web. JSON-LD is a lightweight Linked Data format that is based on JSON and enables inter-operating of JSON data at Web-scale. JSON-LD can be used for RESTful Web services, unstructured databases or various programming environments.

Technically, JSON-LD is a standardized format, that can be embedded as markup code into Web pages. The JSON-LD example below shows the markup up code describing a person:

```
{
  "@context": "http://json-ld.org/contexts/person.jsonld",
  "@id": "http://dbpedia.org/resource/John_Lennon",
  "name": "John Lennon",
  "born": "1940-10-09",
  "spouse": "http://dbpedia.org/resource/Cynthia_Lennon"
}
```

Actually, the JSON-LD syntax simply consists of key-value pairs. The first key is '@context', which describes what the document is about. The '@id'-key gives a link to the person by usage of an URI. URIs are good IDs as they can directly link to the entity and simply be pasted into a browser. The other keys are directly related to the person and describe the person's name, birthday and spouse. The link to the person's spouse is another resource represented as URI. This URI again links to another entity.

The '@context' element also specifies which vocabulary to use. A vocabulary commonly used in connection with JSON-LD is schema.org. Schema.org already specifies a large number of different contexts. This can be for example a music event, where the specification can be looked up on <https://schema.org/MusicEvent>. The specification shows the properties (e.g. location), the expected type (e.g. a Place, which is another context) and gives short descriptions for the properties. The specification also shows any inheritances - e.g. that "MusicEvent" is an "Event", which in turn is

a “Thing”. Of course, JSON-LD requires consumers and Web pages to use the same vocabulary. The schema.org vocabulary was initially created by the four big search engine providers, namely Google, Microsoft, Yahoo and Yandex. A main benefit for Web pages to use JSON-LD (or some other Linked Data formats) is that the mentioned search engines crawl Web pages and extract that linked information automatically. For example, Google Rich Snippets use that technology, as described in 3.3.5 by showing elements such as a map with the location or links automatically.

Another use case is JSON-LD markup code in e-mails. For example, many airlines annotate flight confirmation mails with JSON-LD. Gmail is able to extract this data and show detailed flight information of upcoming flights. This information can even be pushed to mobile phones via Google Now, where boarding passes can simply be fetched as QR code.

A main benefit of JSON-LD is that there is a standardized format and vocabulary. Otherwise, integration with all different participants in a semantic network would be necessary⁷. The JSON-LD specification can be found on <http://www.w3.org/TR/json-ld>. JSON-LD is also integrated with the Google Knowledge Graph⁸.

⁷W₃C-JSON-LD-Community-Group, 2015.

⁸Google, 2015g.

5 Social Media Marketing

This chapter gives a basic overview over Social Media Marketing aspects. First, inbound marketing as a superordinate term of Social Media Marketing is discussed. Then, a comparison between traditional and social media marketing is given. Finally, attention is given to digital marketing communication through social media marketing.

5.1 From Outbound to Inbound Marketing

Simultaneously with the evolution of Web 2.0, as discussed in Chapter 3, the marketing world experienced a significant shift from outbound to inbound marketing. **Outbound marketing** is also known as interruption-based marketing. This means that communication is initiated by the company and transmitted to the audience without a possibility for direct feedback (e.g. TV commercials). The underlying logic is that marketing messages are sent to lots of people that roughly fit into the target audience and, as a result, these people should become customers. Contrarily, **inbound marketing** focuses on pulling customers towards a company. The underlying concept is that companies are active where their customers are active and offer a great user experience to them at those places. This, in turn, should ensure regular visits of people and the spread of word-of-mouth to enlarge the reached audience. Inbound marketing can be further categorized into traditional and online inbound marketing. The focus in this section is on online inbound marketing.

Some of the most important techniques of online inbound marketing are:

- Branding
- Content Marketing

5 Social Media Marketing

- Search Engine Optimization (SEO)
- **Social Media Marketing**
- Video Marketing
- E-Mail Marketing

Hence, **Social media marketing** is classified as a form of online inbound marketing. Importantly, these techniques of online inbound marketing overlap each other and can be most effectively used in combination (Pateman and Holt, 2011) [pp. 9-12]. Therefore, these related techniques are subsequently briefly highlighted before social media marketing aspects are directly discussed in the final section of this chapter.

The definition of a brand has already been given in Chapter 2. Besides that definition a brand is associated with more than that - it is something that creates awareness, reputation, prominence and so on. A brand can create differentiation from other products or services that satisfy the same needs and lead to a competitive advantage (Keller, 2007) [p. 5.]. Another important concept built on the top of a brand is **customer-based brand equity**. Customer-based brand equity is defined as “the differential effect that brand knowledge has on consumer response to the marketing of that brand” (Keller, 1993). This differential effect refers to the choice of a brand, recall of copy points from an advertisement and the like by customers. Thereby, the differential effect builds upon brand knowledge, which means that the brand is linked with certain associations in the customers’ minds. The components of brand knowledge are brand awareness and brand image. Usually, goals are creation of high brand awareness by building brand recognition and brand recall as well as creation of a strong, favorable and unique brand image. Based on these points, the “Customer-Based Brand Equity Pyramid” can be built, as shown in Figure 5.1.

Salience is the most basic building block of the pyramid and refers to the brand identity. Brand identity has the goal to create deep and broad brand awareness. Deep brand awareness refers to the ease of recognition and recall, as well as to the strength and clarity of category membership in the customers’ mind. Broad brand awareness refers to the range of usage, consumption and purchase situations in which the brand is considered by customers.

5.1 From Outbound to Inbound Marketing

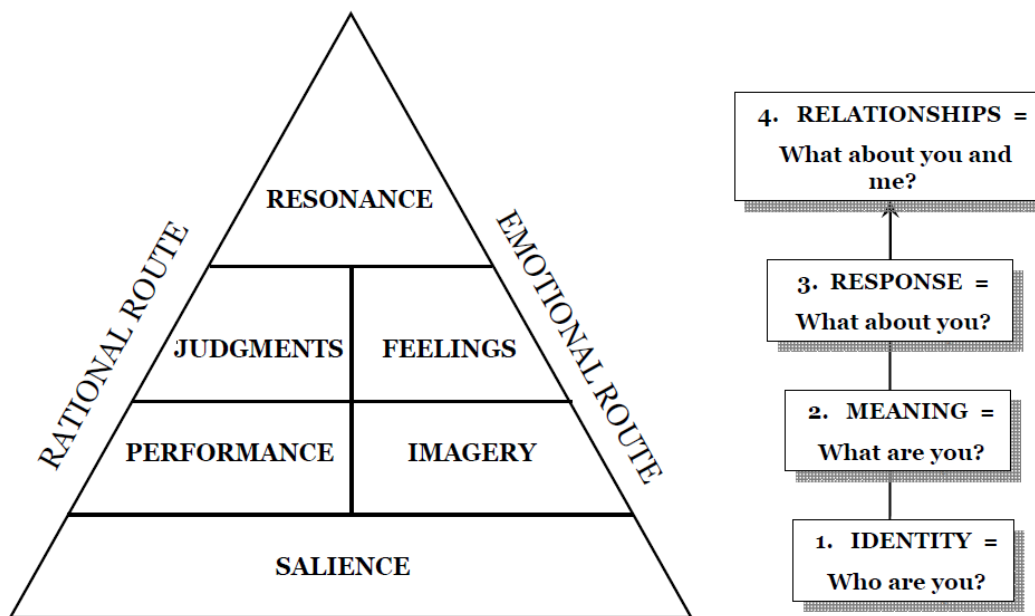


Figure 5.1: Customer-Based Brand Equity Pyramid. Adopted from (Keller, 2007).

5 Social Media Marketing

The next pyramid level describes the brand meaning from a rational and emotional perspective. On the rational side of the pyramid, the **performance dimension** is located. Performance refers to how well a product or service meets customers' needs from a functional perspective. Important features in this dimension are:

- Primary characteristics and supplementary features
- Product reliability, durability, and serviceability
- Service effectiveness, efficiency, and empathy
- Style and design
- Price

The **imagery dimension** describes rather emotional properties of a product or service, including the ways in which a brand tries to meet customers' psychological and social needs. These properties include user profiles (e.g. demographic characteristics, lifestyles or interests), purchase and usage situations (e.g. ease of purchase, types of available channels or time, context and location of usage), personality and values, and aspects related to history, heritage or experiences (e.g. nostalgia).

The third level of the pyramid assesses the brand response from customers, again from a rational and emotional perspective. From a rational perspective, the **judgment dimension** measures customers' opinions about a brand based on its brand identity and meaning. This includes brand quality (e.g. value, satisfaction), brand credibility (e.g. trustworthiness, likeability), brand consideration or relevance, and brand superiority (differentiation). From an emotional perspective, the **feelings dimension** encompasses customers' emotional responses and reactions to a brand (e.g. warmth, fun, excitement or security).

On the top of the pyramid, the **resonance dimension** describes the actual relationship and level of identification customers have with the brand. This includes the following elements:

- Behavioral loyalty (frequency and amount of repeated purchases/usages)
- Attitudinal attachment (extent to which customers love the brand and are proud of it)
- Sense of community

5.1 From Outbound to Inbound Marketing

- Active engagement (e.g. website visits, active seek of information about the brand or membership) (Keller, 2007) [pp. 47-78]

After discussing branding in greater detail, a view words are dedicated to the other techniques of online inbound marketing. First, **content marketing** addresses the need to provide high-quality content on the website such as the availability of useful tools, educational or entertaining content. The goal is to achieve higher conversion rates, return visits, social recommendations, SEO benefits and excited visitors. **SEO** deals with the accomplishment of a high page ranking at search engines. This, in turn, lists the website higher in search rankings for specific keywords and maximizes traffic to the website. Content marketing supports SEO because every new page may act as a landing page in the search results, great content that gets linked by visitors enhances page rankings and Google also takes a great User Experience (UX) into account (e.g. high click-through-rate, low bounce rate or social mentions) (Pateman and Holt, 2011) [p. 18]. **Video marketing** refers to the use of videos to promote brands, products or services for advertising purposes. In the context of online inbound marketing, this means to post videos on social platforms, such as YouTube, with the goal to achieve viral marketing effects¹. Finally, e-mail marketing refers to the promotion of brands, products or services by e-mail. E-mails may be standardized or customized for different target groups².

Completing this section, a model outlining the ideal process of inbound marketing is presented, as depicted in Figure 5.2.

The figure shows how strangers can ultimately be turned to promoters of a business or organization. At the top, necessary actions at the different stages are shown. At the bottom the tools needed to perform these actions are shown. The underlying principle is to publish the right content at the right place at the right time, so that it is relevant and helpful to customers. Major topics in this methodology are:

- Content Creation and Distribution (creation and share of targeted content)
- Lifecycle Marketing (Process to turn strangers ultimately to promoters)

¹TheDigitalMarketingGlossary, 2016.

²TheDigitalMarketingReference, 2016.

5 Social Media Marketing

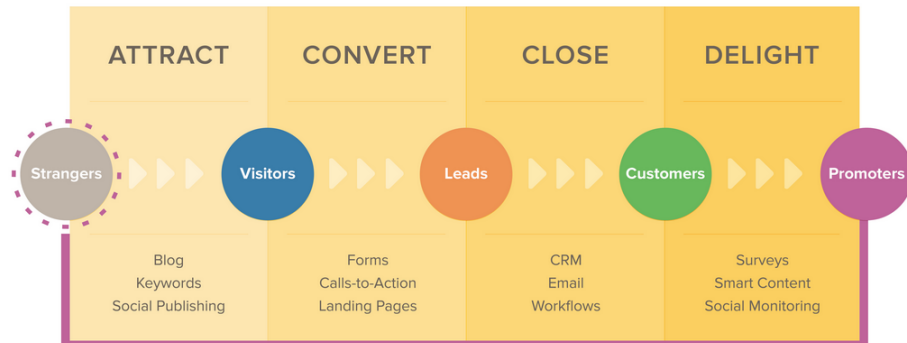


Figure 5.2: Inbound Marketing Methodology. Adopted from (HubSpot, 2015).

- Personalization (Adaption of content towards customers' needs and wants)
- Multi-channel presence (approach of people in channels where they are active)
- Integration (alignment of content creation, publishing and analytics)

Next, the four mentioned marketing actions - attract, convert, close and delight - are discussed. The **attract** action attempts to get the right customers to the website. The right customers are those that are likely to become promoters and are often represented by personas. Important tools, that have already been discussed in detail, to attract the right customers are blogging, SEO, pages (website optimization) and social publishing (e.g. sharing remarkable content in social networks). The **convert** action deals with the conversion of website visitors to leads by gathering their contact information. To enhance the number of lead conversions, counter-trades for gathering contact information, such as provision of special content may be given. Some important tools to facilitate conversion are traditional contact forms, calls-to-action requiring registration (e.g. downloads) on a landing page and contact management in a centralized database. The **close** action is concerned with turning leads into customers. Marketing tools that can be used at this stage are Customer Relationship Management (CRM) systems, closed-loop reporting (analyzing which marketing efforts lead to actual sales - often integrated with CRM systems), e-mails and marketing automation (automatically tailoring marketing content to customer needs). Finally, the

5.2 Traditional vs. Social Media Marketing

delight action deals with the conversion of existing customers into promoters by engaging and delighting them. Tools used at this stage to get insights about customer needs are surveys, smart calls-to-action (based on persona and lifecycle stage of customers), smart texts (provide targeted information about products or service that are relevant to different customers) and social monitoring (keeping track of relevant social conversations indicating customers' questions, likes or comments) (HubSpot, 2015).

5.2 Traditional vs. Social Media Marketing

Before discussing topics directly related to social media marketing, a comparison between traditional and social media marketing is given. Traditional marketing methods have been important elements in the marketing mix since the very beginning of advertising. Therefore, these methods are trusted by customers and are valuable to businesses and organizations. Among others, traditional marketing methods include radio, television, newspaper, direct mail or billboards. However, with new possibilities offered by Web 2.0, customers want to get new information about products, they are interested in, fast and in an interactive way. Furthermore, businesses and organizations want to measure their Return on Investment (ROI) on marketing expenditures precisely.

Traditional marketing does not provide satisfying solutions to both requirements, but social media marketing does. Social networks are a perfect example as methods are available to interactively post, comment or share any information at any time (e.g. on a Facebook business page). Social networks in general also provide detailed statistics to businesses about who is viewing, commenting or sharing their content, and what customer opinions are. As mentioned in Section 5.1, traditional marketing methods are a form of outbound marketing and, conversely, social media marketing is a form of inbound marketing³. Concerning statistics, Social Media Monitoring and Social Network Analysis have been discussed in Section 3.4.4.

³SocialTimes, 2015.

5 Social Media Marketing

Social Media Marketing	Traditional Marketing
Inexpensive	Costly
Interactive/Engaging	One-way/non-reciprocal
Time-consuming	Short-term results
Easily measurable	Hard to measure
Disseminates rapidly	Permanent effect

Table 5.1: Pros and Cons of Traditional and Social Media Marketing. Adopted from (Social-Times, 2015).

Including the points already discussed, Table 5.1 summarizes pros and cons of traditional marketing in comparison to social media marketing.

5.3 Digital Marketing Communication

Concluding this chapter, social media marketing in the context of digital marketing communication is discussed. Digital marketing communication refers to the interaction between companies and their customers via digital channels. An important marketing method in this context is direct marketing. Traditional direct marketing methods, such as telephone calls, often had a rather limited scope and were often regarded as disturbing by customers. In the digital context this has changed completely. Nowadays, people can interact at a global scale instantly, which implies huge possibilities as well as threats for organizations. Furthermore, communication shifts increasingly out of companies' control towards platforms that have been created by others.

An important concept in this context is the **electronic Word of Mouth (eWOM)**, also referred to as **referral marketing**. eWOM refers to the communication and spread of information between people even across cultures and different countries. eWOM naturally has a higher credibility than most other marketing sources as no direct commercial interest is conveyed. This may enable a large viral marketing effect that spreads across linked networks of people when information is shared and shared again. However, for a viral effect to take place, online attention from the target group must

5.3 Digital Marketing Communication

be attracted. This task is often difficult as there is already a huge load of information available on social networks such as Facebook or content sharing sites such as YouTube, as illustrated in Chapter 7. Therefore, companies often promote their products or services first to their customers, which should disseminate it further to prospective customers, and post their marketing material at many different places where the target groups are active. This process is called seeding and may additionally require adaption of content to different cultures and languages. On the one side, **viral marketing** can strongly promote factors such as brand awareness. On the other side, the viral effect can also have a very negative impact when counterarguments are disseminated due to the lack of control for companies.

Other important marketing sources for companies are **brand communities** in social networks. Globally active online brand communities enable promoters to discuss and share information about their favorite brands even across geographical borders. These communities provide a way for companies to reach a large number of consumers, perform market segmentation easily or create consumer trust, among others. There are three types of different brand communities:

1. Consumer-driven brand communities (exclusively managed by consumers)
2. Company-driven brand communities (company manages community, consumers deliver inputs)
3. Hybrid brand communities (managed by companies and consumers coincidentally)

For an online brand community to function, people need to regularly visit the site so that ongoing social interaction can take place and brand loyalty can be enhanced. The social links that are formed by the brand community enable benefits for both, the consumers and the company. There are five major forms how companies can interact with their brand communities:

1. Fight: Community positioning itself as opposition against another group (in the sense: "us vs. them").
2. Role model: Used as something to identify with and create a common sense amongst community members.

5 Social Media Marketing

3. Exchange: Used to share something (e.g. knowledge) or form strategic networks - often constituted as hybrid brand community.
4. Manifestation: Builds upon traditions, rituals, symbols or icons.
5. Progression: Focuses on development and innovation.

Another marketing method that emerged with Web 2.0 is **permission marketing**. Permission marketing requires permissions granted by users before marketing communication is sent, regardless of the communication channel. Typically, consumers provide marketers with information of types of marketing communication they are interested in. This allows marketers to target market communications and consumers to have control over which information they receive. Ultimately, permission marketing should create consumer trust and loyalty towards a company. By creating trust, it is likely that consumers increase the level of permission. There are five levels of permissions:

1. Situation permission: Consumers agree to receive marketing communications for a limited period of time.
2. Brand trust: Consumers have developed confidence towards the brand.
3. Personal relationship: Marketeers can direct customer behavior to sell customized products.
4. Points permission: Consumers allow company to collect personal data and agree to a point-based loyalty scheme.
5. Intravenous: Consumers allow company to make buying decisions on their behalf (Kotler et al., 2009) [pp. 124-131].

Finally, other marketing methods used in a Web 2.0 context should be mentioned. These marketing methods are affiliate marketing, relationship marketing, crowdsourcing and crowdfunding. Affiliate marketing uses networks of individuals or other companies to distribute marketing communications with the goal of a viral marketing effect and is usually compensated with commissions⁴. Relationship marketing is a marketing concept that focuses on attracting customers, position a brand as the top-choice in the customer's mind and build lifelong strong and profitable relationships (M. Smith and Kawasaki, 2011) [pp. 1-2]. Crowdsourcing means for an organization that it asks a community (crowd) to perform a certain task voluntarily that should

⁴EntrepreneurSmallBusinessEncyclopedia, 2016.

5.3 Digital Marketing Communication

lead to mutual benefits for the organization and community. This is often done in various Web 2.0 environments where organizations and the community can interact easily (Brabham, 2013) [pp. 1-3]. Finally, crowdfunding is a way to raise money from a large pool of people to fund specific projects. A well-known crowdfunding platform is Kickstarter (Stegmaier, 2015) [pp. 1-4].

6 Catrobat

After discussing the relevant technical background, the subsequent chapters focus on the practical relevance to the Catrobat project. This chapter gives an introduction to Catrobat.

Catrobat is an open source software project that develops the identically named visual programming language for smartphones, tablets and mobile browsers. The stated goal of Catrobat is to “create free educational apps for children and teenagers”¹. It is based on the programming language Scratch, which is being developed by the MIT Media Lab Lifelong Kindergarten Group. Pocket Code is a mobile application that allows to execute Catrobat programs directly on mobile devices and is available for Android, iOS, Windows Phone and HTML5.

Figure 6.1 shows a script of an example program in Pocket Code. Furthermore, there is the community website <https://pocketcode.org> that allows to directly upload and share programs with other users, as depicted in Figure 6.2. These programs are licensed under the AGPL and are therefore freely editable and reusable for other users. Catrobat also develops a set of creativity tools, such as Pocket Paint and Musicdroid. Pocket Paint is an image editor that is directly integrated with Pocket Code, but can also be used on its own. Musicdroid, which is not released yet, is an Android application to create music. Catrobat won the Austrian National Innovation Award 2013 in the category Multimedia, is featured on Google Play for Education, takes part in the “No One Left Behind” research project funded by the European Union and participated five times in the Google Summer of Code programme. From a social media perspective, Catrobat is currently active on Facebook, Google+, Twitter and YouTube².

¹Catrobat, 2015a.

²Catrobat, 2015b.

6 Catrobat

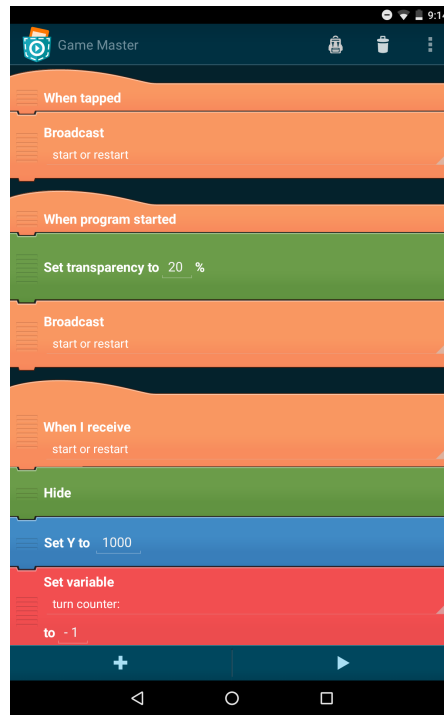


Figure 6.1: Script of a Pocket Code program.



Figure 6.2: Catrobat community website.

6.1 Pocket Code Target Group

This section lies the foundation for the statistical research in Chapter 7. First, Pocket Code’s target group including Personas is defined. Then, a few words are dedicated to marketing strategy and potential alliances. Finally, various current Catrobat statistics based on Google Analytics, Andlytics and Facebook Analytics data are shown.

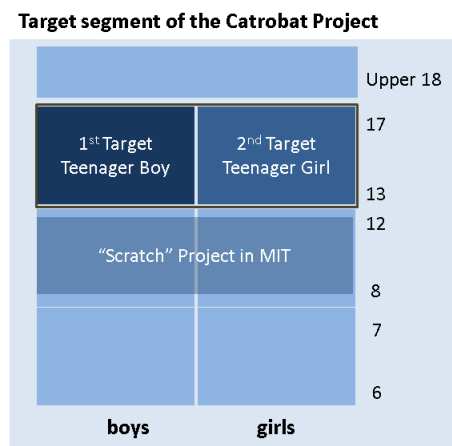


Figure 6.3: Catrobat target group. Adopted from (Tanaka, 2012).

As shown in Figure 6.3, Catrobat’s two target groups are boys and girls aged between 13 and 17. This is located above Scratch’s target group in terms of age. Furthermore, Catrobat focuses on smartphones, not on tablets. Figures 6.4 and 6.5 show Tobias and Marie, two Personas describing typical users in Catrobat’s target group.

From a marketing strategy perspective, Pocket Code competes in the educational and entertainment sector for teenagers. The strength is to train teenagers’ skills through letting them create Catrobat programs. The organization can be supported by financial aid, alliances and partners. Potential alliances include universities, social networks, robot/toys and mobile device companies, among others. The most important triggers for entrance and binders for teenagers to stay, that have been identified, are friends, tools (e.g. robots) and social networks (Tanaka, 2012).

6 Catrobat

lacks patience to read longer tutorials – this does not make fun. Tobias owns a computer with Internet access, too. He uses his PC mainly for playing games and accessing Facebook

Web

Tobias is member on Facebook and MySpace. He is hoping that a band takes notice on his compositions. Furthermore, he reads articles that help him working on his motorbike. During online games with his friends he loves the challenge. It is important for him to measure with and play against others to get better, or the best at the end. Having fun is also important to him. Tobias likes to create animated clips out of photos and videos and add his music. It would be cool, if he can present the clips on a Web platform.

Negative Experiences

Tobias does not like to spend much time on fine-tuning, when he wants to do something quickly. Furthermore, once he had the problem that a composition of him was stolen.

Wishlist

Tobias likes to create games with his friends, so that they always have new possibilities for challenges. He does not want to spend days, but have fun quickly. Tobias has no problem that other people play and assess his games, but he wants to have noted his intellectual property. He is also interested into who has seen his games and where this persons come from. Tobias likes to keep an overview over things he did on the computer. He does not like to search for needed program components, but likes to reach his goal without complicated program sequences.

Vision of Future

Tobias wants to prove himself in challenges. He does not know potential jobs, yet. However, the daily challenge is important to him – Tobias does not want to get bored.

Additional Information

Tobias is an outgoing person, that loves to have fun. He is also very ambitious and always likes to be the best.

Tobias Mardorff



15 years, male, Wiener Neustadt (AT)

„I want to do cool stuff and win against others!“

Description

Tobias is 15 years old and studies machine engineering at the polytechnic school. He is very interested in technology and already disassembled toys into its individual parts as child to find out how things work. When he is not at school, he plays online games with his friends, tinkers his motorbike or composes music with his electric guitar.

Social Background

Tobias' parents are divorced. Tobias lives in a flat with his father. The household is fairly assigned between them – Tobias is not really happy with that. He got to know his girlfriend recently.

Technical Knowledge

Tobias easily understands the functionality of things. He got a mobile phone with Internet access on his birthday. He uses it mainly for accessing Facebook and YouTube on it. During idle times he uses his time for playing various games. Tobias would call himself a "computer expert", as he knows the handling and backgrounds. Tobias also has basic programming skills that he wants to improve. However, he

Figure 6.4: Catrobat persona: Tobias. Adapted with kind permission from Cure ©2010

6.1 Pocket Code Target Group

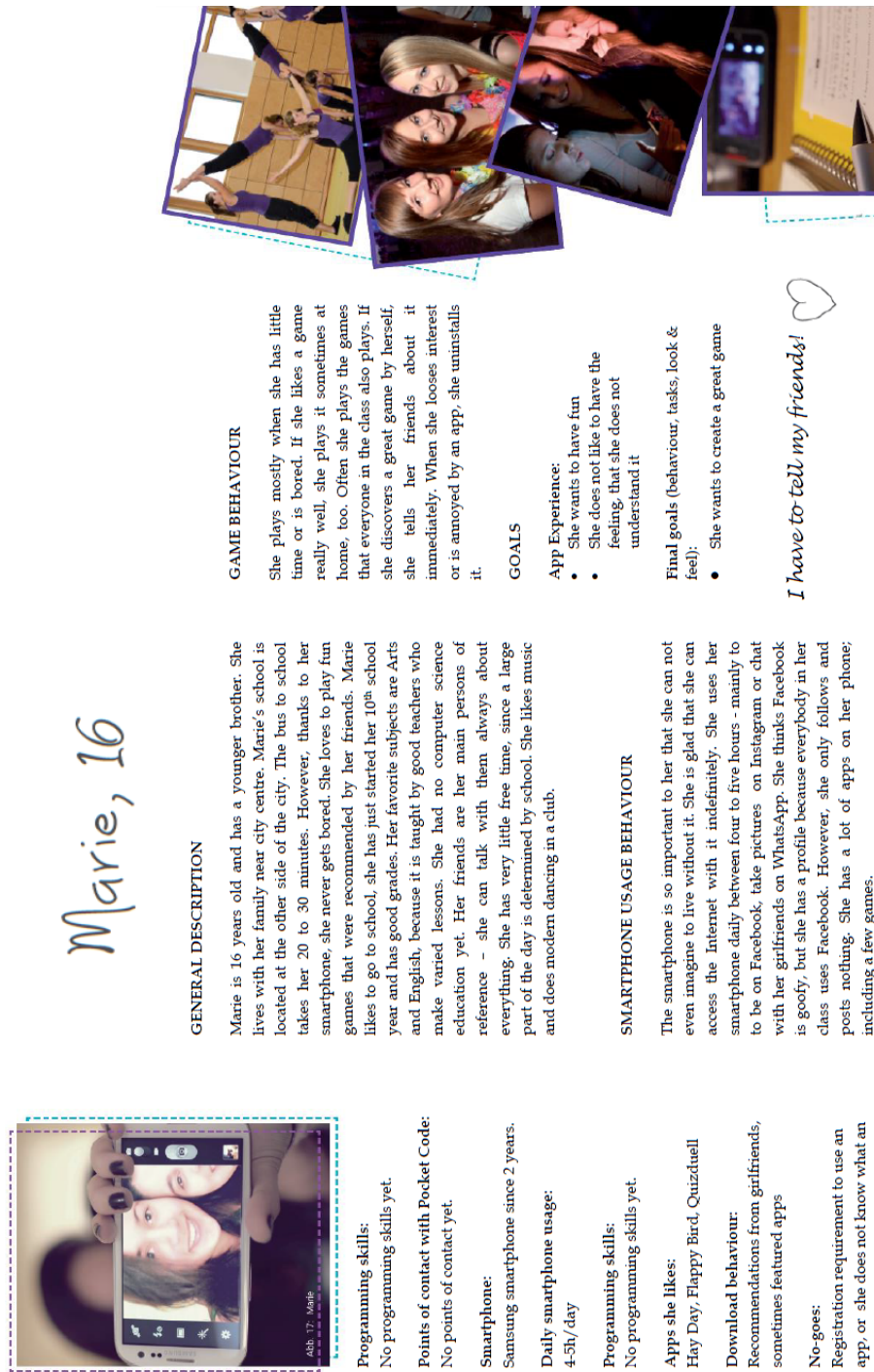


Figure 6.5: Catrobat persona: Marie. Adopted from (Petri, 2014)

6 Catrobat

Subsequently, official Catrobat statistics derived from Google Analytics tracking of the community Website (<https://share.catrob.at>) are shown. The Website is used by Pocket Code users when they download programs, hit the explore button in the Pocket Code app, read tutorials or take actions on their Catrobat profile. Data is taken from the period June 2015 (as Google Analytics has been integrated) to April 2016.

Figure 6.6 shows the total number of users with approximately 56,000 and sessions (including returning visitors) with about 98,000. Subsequently, statistics are based on the number of unique users.

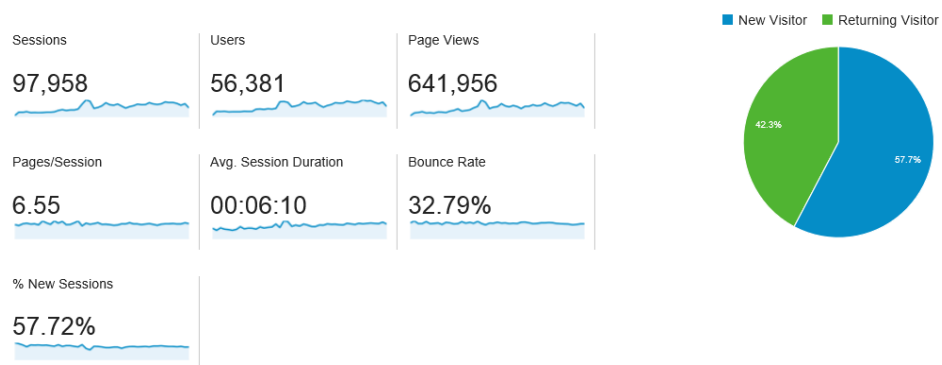


Figure 6.6: Community Website Overview. Adopted from (Catrobat, 2016a)

Figure 6.7 shows a distribution of Website visitors by country and device. Most visitors are from the United States, which is probably due to Google's efforts in supporting Catrobat. Then, Germany and Austria rank on place two and three. Austria has such a high number of visitors as it is Catrobat's home country, is mostly developed there and has a strong Austrian community. Other top European countries are Spain, United Kingdom, Italy, Turkey and France. The distribution shows that a large part of visitors are from Asia too, with the most of them from Russia, Japan and India. China is also included as it may be an interesting market for Catrobat. Brazil ranks in the top ten countries, too. Furthermore, each major country has more mobile than Desktop users. This is expected as the Website is intended to be used from within the Pocket Code app. The regional social media statistics, that will be given in Section 7.4, are based on these findings.

6.1 Pocket Code Target Group

Country	Device Category	Users	Country	Device Category	Users
1. United States	mobile	7,201 (12.70%)	21. Italy	desktop	592 (1.04%)
2. Germany	mobile	4,732 (8.34%)	22. Canada	mobile	553 (0.98%)
3. Austria	mobile	3,734 (6.58%)	23. United Kingdom	desktop	531 (0.94%)
4. Russia	mobile	3,322 (5.86%)	24. United States	tablet	530 (0.93%)
5. Turkey	mobile	2,449 (4.32%)	25. Spain	desktop	503 (0.89%)
6. United States	desktop	2,220 (3.91%)	26. Philippines	mobile	501 (0.88%)
7. Spain	mobile	1,840 (3.24%)	27. Netherlands	mobile	388 (0.68%)
8. United Kingdom	mobile	1,826 (3.22%)	28. Germany	desktop	351 (0.62%)
9. Brazil	mobile	1,657 (2.92%)	29. India	desktop	343 (0.60%)
10. France	mobile	1,276 (2.25%)	30. Sweden	mobile	302 (0.53%)
11. Italy	mobile	1,184 (2.09%)	31. Egypt	mobile	280 (0.49%)
12. Austria	desktop	1,042 (1.84%)	32. Argentina	mobile	278 (0.49%)
13. India	mobile	1,039 (1.83%)	33. Australia	mobile	275 (0.48%)
14. Japan	mobile	964 (1.70%)	34. Malaysia	mobile	273 (0.48%)
15. Mexico	mobile	847 (1.49%)	35. Romania	mobile	256 (0.45%)
16. Poland	mobile	831 (1.47%)	36. Iraq	mobile	237 (0.42%)
17. Indonesia	mobile	733 (1.29%)	37. Thailand	mobile	233 (0.41%)
18. Taiwan	mobile	714 (1.26%)	38. Saudi Arabia	mobile	231 (0.41%)
19. Ukraine	mobile	631 (1.11%)	39. Switzerland	mobile	219 (0.39%)
20. South Korea	mobile	615 (1.08%)	40. Colombia	mobile	213 (0.38%)
			76. China	mobile	92 (0.16%)

Figure 6.7: Community Website Country Distribution. Adopted from (Catrobat, 2016a)

6 Catrobat

Device Category	Acquisition			Behaviour		
	Sessions ↓	% New Sessions	New Users	Bounce Rate	Pages/Session	Avg. Session Duration
share.catrobat.at	97,958 % of Total: 72.57% (134,975)	57.72% Avg for View: 67.97% (-15.09%)	56,541 % of Total: 61.63% (91,748)	32.79% Avg for View: 29.98% (9.39%)	6.55 Avg for View: 5.59 (17.17%)	00:06:10 Avg for View: 00:04:40 (32.02%)
1. mobile	80,319 (81.99%)	56.91%	45,713 (80.85%)	30.26%	6.97	00:06:42
2. desktop	14,654 (14.96%)	61.81%	9,058 (16.02%)	42.25%	4.87	00:03:59
3. tablet	2,985 (3.05%)	59.30%	1,770 (3.13%)	54.51%	3.50	00:02:42

Figure 6.8: Community Website Device Distribution. Adopted from (Catrobat, 2016a)

Figure 6.8 shows that more than 80 percentage of users visit the Website from a mobile device, which is the intended target platform, as said before. About 15 percentage visit the Desktop version of the Website and only a low number of visitors access the Website via tablet. The higher mobile average session duration and lower percentage of new sessions indicate that many users revisit the mobile Website version from within the Pocket Code app and spend a rather long period of time on activities on the community Website³.

Appropriate to the high mobile usage rate, Andlytics and Facebook Analytics data about the mobile Pocket Code Android app is given subsequently. Andlytics data is retrieved from Google Play and Facebook Analytics data from Facebook signin that has been integrated into Pocket code, as described in Section 8.1.3.

Figure 6.9 shows that Pocket Code has more than 160,000 downloads and more than 30,000 installs by April 2016. Subsequently, Facebook Analytics data with a time range from 22th March (as Facebook Login was released + 1 month to get monthly data) to 28th April is given. These statistics include all Pocket Code users, not only those that have signed in with Facebook. The number of active users is relatively stable, as indicated in Figure 6.10. Pocket Code has about 10,000 monthly active, 3,000 weekly active and 500 to 600 daily active users. Altogether, Pocket Code has about 12,800 unique users according to Facebook data.

Figure 6.11 shows the age and gender distribution of Pocket Code users, as far as data is available (54 % of users). Only about 21.5 % of users are

³Catrobat, 2016a.

6.1 Pocket Code Target Group

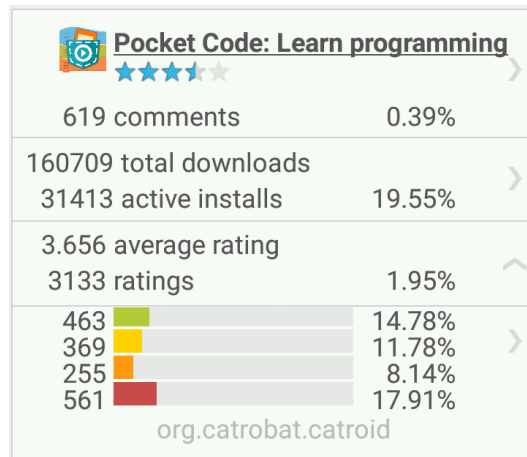


Figure 6.9: Pocket Code on Google Play. Adopted from (Catrobat, 2016b)

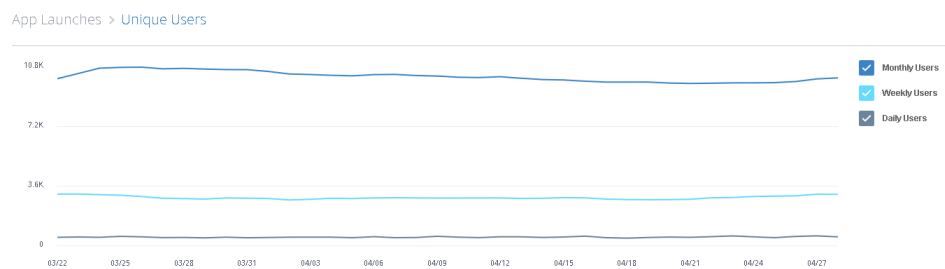


Figure 6.10: Number of Active Pocket Code Users. Adopted from (Facebook, 2016a)

6 Catrobat

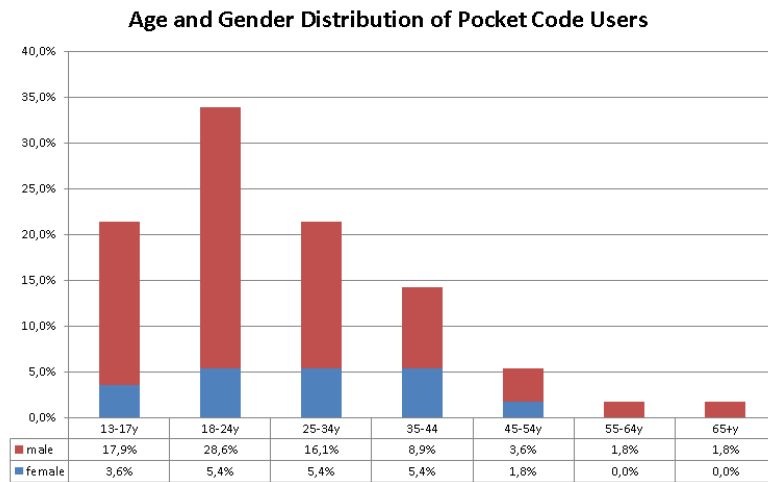


Figure 6.11: Age and Gender of Pocket Code Users. Adopted from (Facebook, 2016a)

from Catrobat's defined target groups of males and females between 13 and 17. Most users are aged between 18 and 24 with about 34 % of total users. Identically to the teen age group, people aged between 25 and 35 represent about 21.5 % of total users. Altogether, there are clearly more male than female users. Male users represent a total of about 78 % and female users a total of about 22 %.

Finally, Figure 6.12 shows a country distribution of Pocket Code users (most important countries are shown). The distribution is very similar to the one retrieved by Google Analytics, that has already been shown in Figure 6.7. According to this statistic, most users are from the United States, followed by European countries such as Germany, Austria, United Kingdom and Spain, Asian countries such as Russia, India and Japan as well as Brazil.

6.1 Pocket Code Target Group

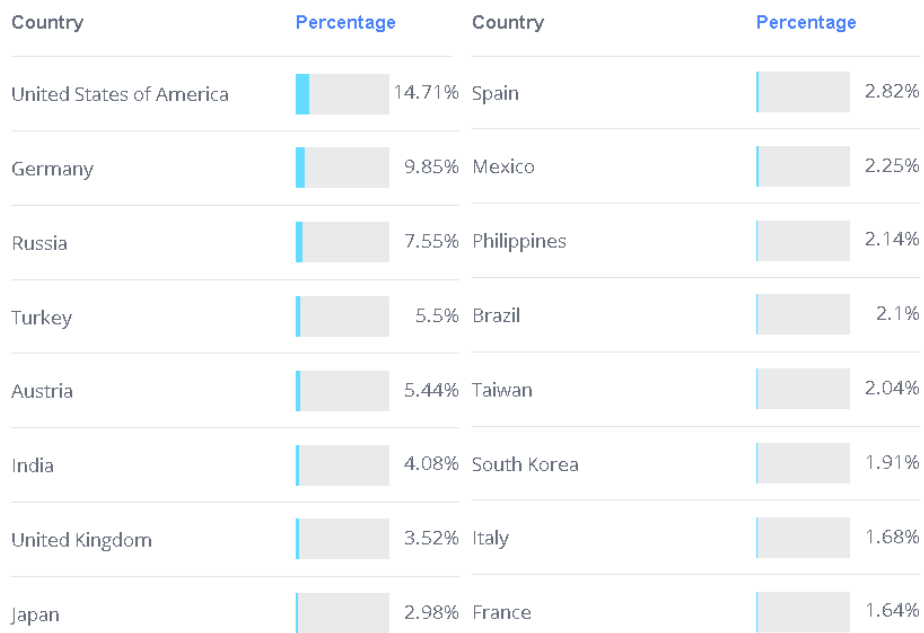


Figure 6.12: Country Distribution of Pocket Code Users. Adopted from (Facebook, 2016a)

7 Social Media Statistics

In this chapter the social media landscape is analyzed for their potential to Catrobat. This includes various social media statistics concerning age, gender, used platforms, usage behavior, mobile access rates and regions. At the end of the chapter a guidance for the Catrobat project on how to deal best with social media platforms in diverse regions of the world is given. The statistical data in this chapter is almost completely retrieved from one of the world's leading statistics portals, namely Statista¹ (otherwise there is an explicit citation).

¹Statista, 2016.

7 Social Media Statistics

7.1 Social Networks General

Before investigating specific platforms, some important general statistics are given in this section. First, Figure 7.1 shows the total number of social network users since 2010, with a projection until 2019, compared with the total number of worldwide Internet users from 2010 to 2015.

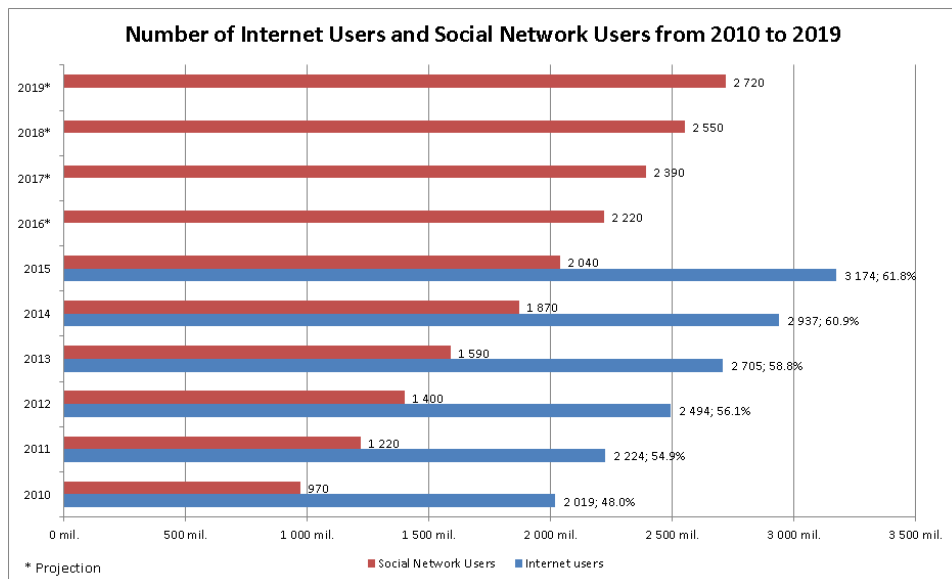


Figure 7.1: Number of Internet and Social Network Users Worldwide. Own illustration, adopted from (Statista, 2016).

Altogether, the percentage of social media users relative to the total number of Internet users has steadily grown from approximately 48 % in 2010 to 61.8 % in 2015. This means that social media got more and more important over the past years. Furthermore, a large number of new social media users to reach a total number of more than 2.7 billion is expected until 2019, as shown in the chart.

Figure 7.2 shows how the total number of active social network users is distributed among social platforms by January 2016.

7.1 Social Networks General

Leading social networks worldwide as of January 2016, ranked by number of active users (in millions)

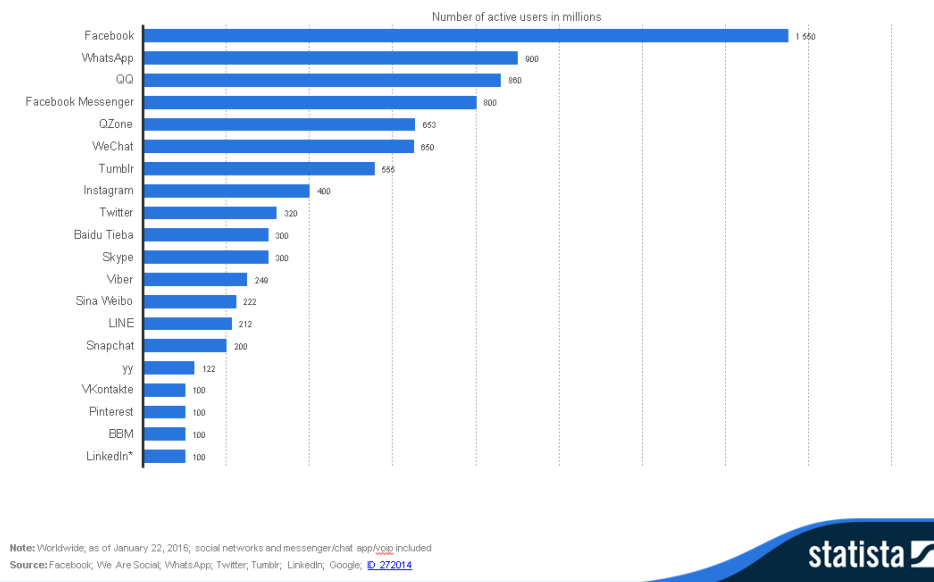


Figure 7.2: Social Network Users Worldwide per Platform. Adopted from (Statista, 2016).

Facebook is by far the platform with the largest number of active users with more than 1.5 billion. Including Facebook Messenger, Facebook even has approximately 2.35 billion (overlapping) users. WhatsApp takes the second place - by February 2016 with already more than 1 billion users (January 2016: 900 millions, as indicated). All in all, the chart indicates that social messengers (WhatsApp, QQ, Facebook Messenger, WeChat, Skype, Viber, LINE, Snapchat etc.) got very popular. The Chinese platforms QQ, QZone, WeChat and Baidu Tieba will be discussed later in Section 7.4. Facebook's mobile-only photo and video sharing platform Instagram takes the eighth place and overtakes Twitter in terms of number of total users. Microblogs, as one of the oldest social media applications, rank at place 7 (Tumblr, 555 millions users) and 9 (Twitter, 320 millions users). Concerning Google+, there are no official numbers on active users. SnapChat and Pinterest will be mentioned later.

Next, Figure 7.3 shows the total growth rates of social networks since 2012,

7 Social Media Statistics

with a projection until 2018.

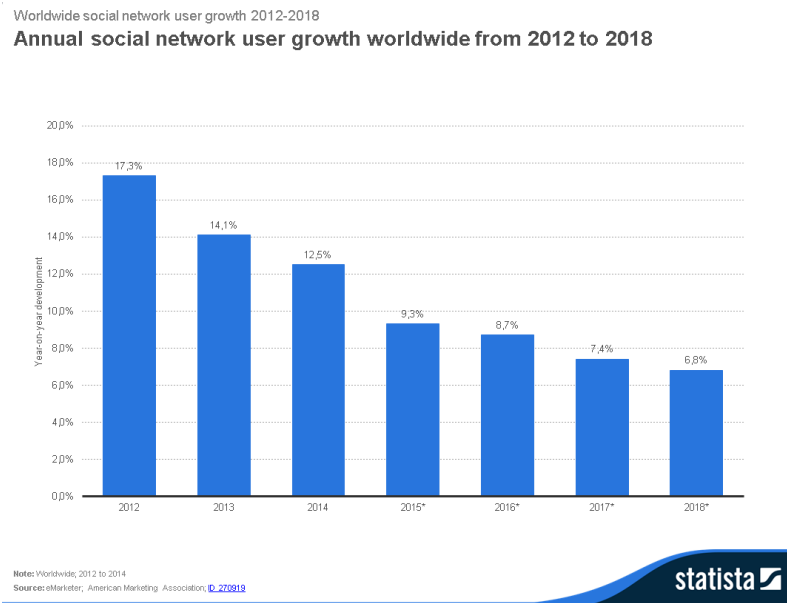


Figure 7.3: Social Network Growth Worldwide. Adopted from (Statista, 2016).

Altogether, overall growth rates of social networks are flattening, as there may be already too many users for such high growth rates to continue.

Figure 7.4 shows how these growth rates are distributed among chosen social networks in 2014.

7.1 Social Networks General

Fastest growing social networks 2014

Fastest growing social networks worldwide in 2014

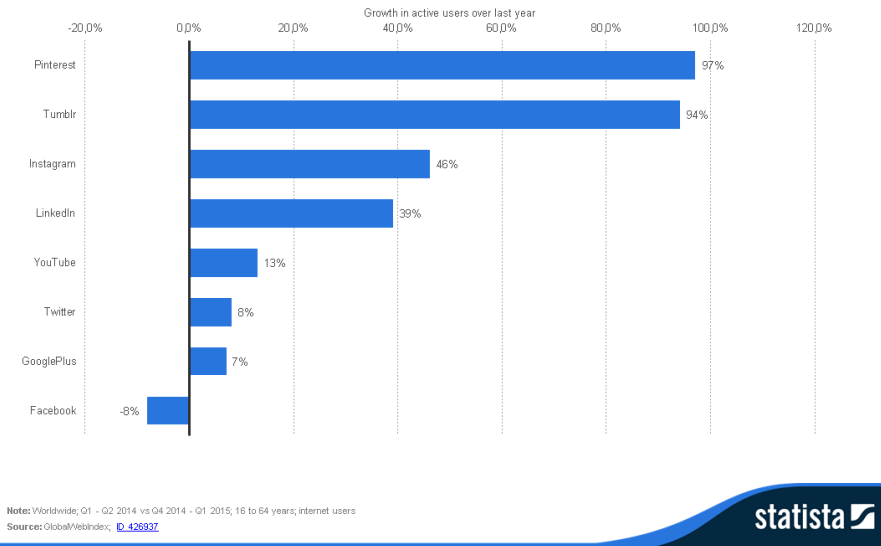


Figure 7.4: Social Network Growth Rates Worldwide per Platform. Adopted from (Statista, 2016).

The social networks with the biggest growth rates are Pinterest and Tumblr, which have almost doubled their number of users. Pinterest is a social network that offers the possibility to share ("pin") photos and videos of interest. The idea is to offer a categorized catalogue of ideas and find others that have the same interests. Pinterest is available as website and as mobile versions for Android and iOS². Tumblr is an U.S.-based microblog that offers a live-feed (dashboard) to users that allows to follow blogs of interest. It is possible to share diverse multimedia content and embed social elements such as hashtags into blog entries. Besides the website, there are mobile versions for Android, iOS, Google Glass and Windows Phone available³. Above-average growth rates are also reached by the multimedia sharing platform Instagram, the business network LinkedIn and the video sharing

²Pinterest, 2016.

³Tumblr, 2016.

7 Social Media Statistics

platform YouTube. Minor growth rates have been reached by Twitter and Google+. Facebook experienced a decline in number of users, but stays the largest platform by far, as stated earlier.

Figure 7.5 shows the growth rates for social apps, including messengers, as of Q1 2015.

Fastest growing social and messaging apps 2015
Fastest growing social and messaging apps worldwide as of 1st quarter 2015

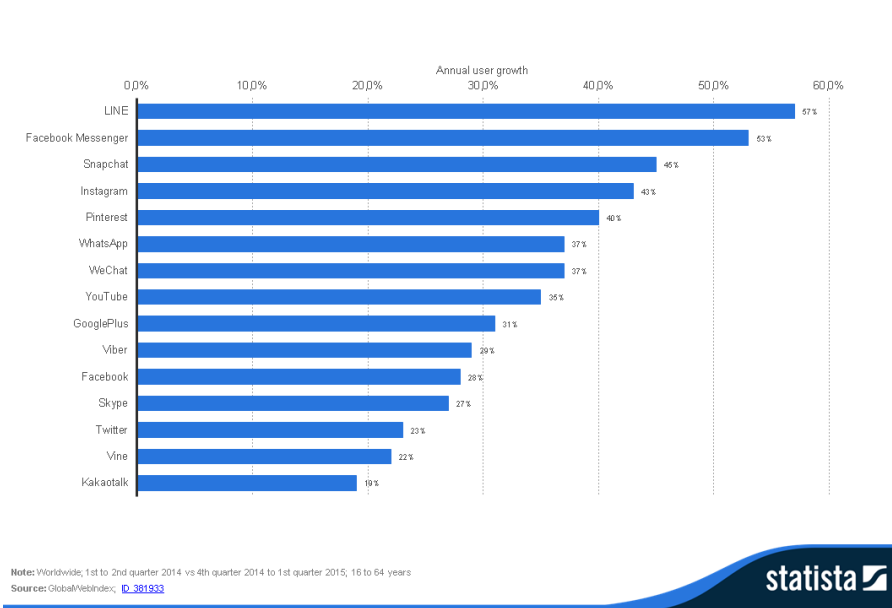


Figure 7.5: Messengers and Social Apps Growth Rates Worldwide per Platform. Adopted from (Statista, 2016).

All in all, the mobile versions offer more stable and higher growth rates than social networks in general. This indicates that the growth continues in the mobile environment. The highest growth rate was reached by LINE (a Japanese messenger), followed by Facebook messenger, Snapchat, Instagram, Pinterest and WhatsApp.

Figure 7.6 shows the global average number of social media accounts per Internet user grouped by age by Q4 2014.

7.1 Social Networks General

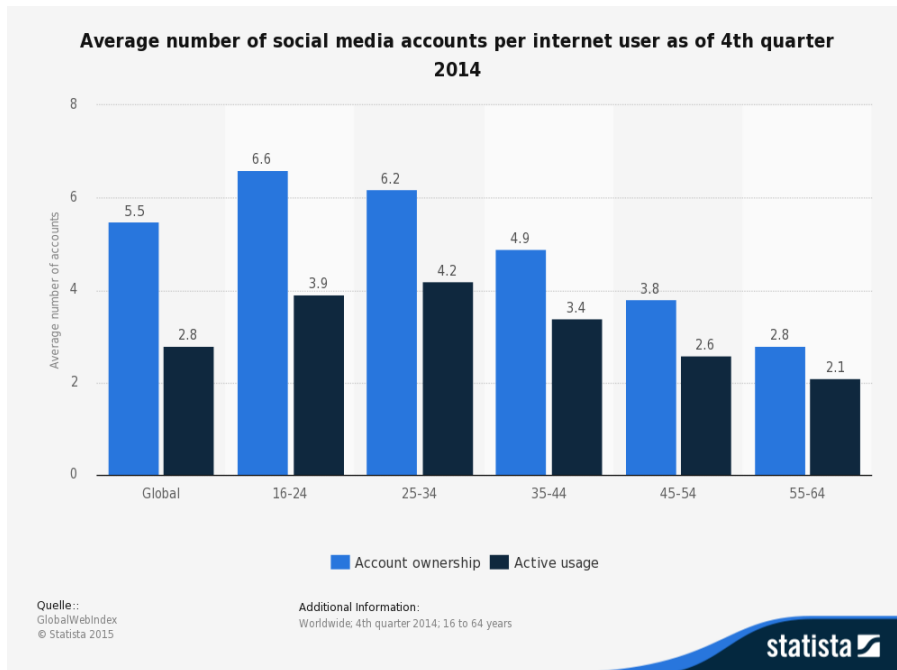


Figure 7.6: Average Number of Social Media Accounts Worldwide. Adopted from (Statista, 2016).

The chart indicates that the younger the users are, the more social media accounts they have. This trend is also true in terms of active usage, except that the age group between 25 and 34 is slightly more active than that between 16 and 24. Altogether, Internet users have a large number of different social media accounts, which is especially true for younger users, that are in Catrobat's target group. However, this age group only uses slightly more than half of the platforms actively.

Figure 7.7 shows the average number of minutes that Internet users spent on social networks worldwide in average from 2012 to 2015.

7 Social Media Statistics

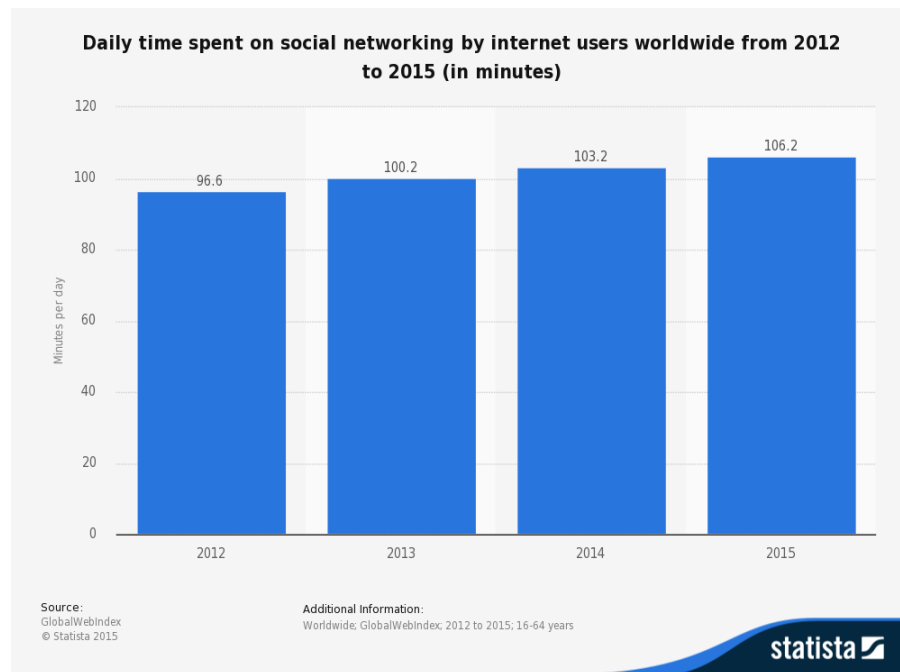


Figure 7.7: Average Time Spent on Social Networks. Adopted from (Statista, 2016).

The chart indicates that the average Internet user spends about 109 minutes per day on social network activities in 2015. It also shows that there is a trend to spend slightly more time on social networks over the past years.

7.2 Mobile

In this section mobile usage statistics, that are especially relevant to Catrobat, are given. First, Figure 7.8 shows the growth rates of mobile subscriptions (including pre-paid cards) from 2008 to 2015.

7.2 Mobile

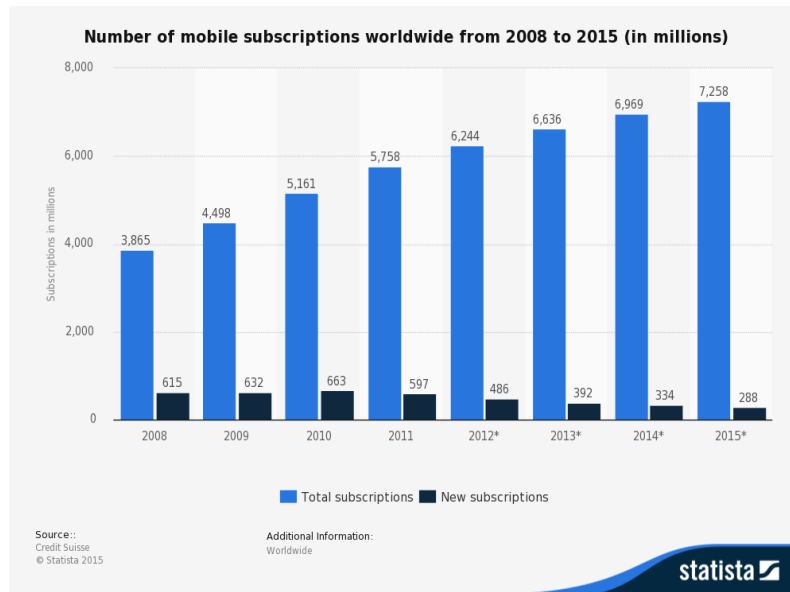


Figure 7.8: Mobile Subscriptions from 2008 to 2015. Adopted from (Statista, 2016).

The chart shows that the number of mobile subscriptions almost doubled from 2008 to 2015 to more than 7.2 billions, approaching the number of total world population. Consequently, many people have more than one mobile subscription.

Figures 7.9 and 7.10 show mobile social media usage distributions by age and gender as of Q4 2014, respectively.

7 Social Media Statistics

Global mobile social network penetration 2014, by age
Share of internet users worldwide who access social media via mobile as of 4th quarter 2014, by age

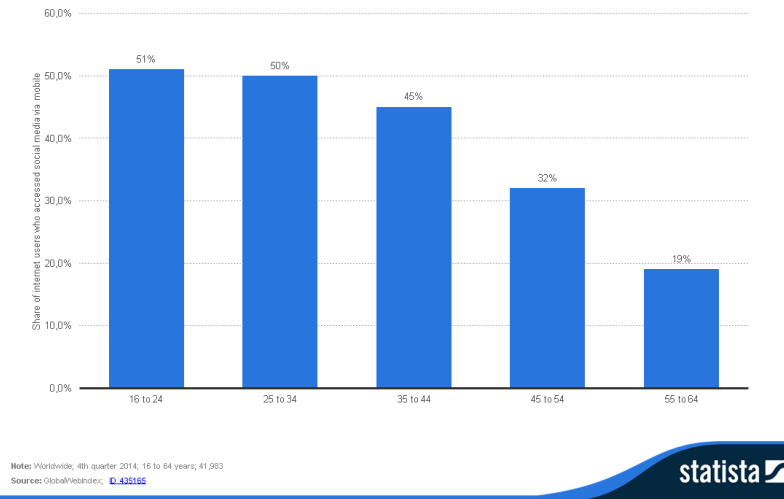


Figure 7.9: Mobile Penetration Rates by Age. Adopted from (Statista, 2016).

Global mobile social network penetration 2014, by gender
Share of internet users worldwide who access social media via mobile as of 4th quarter 2014, by gender

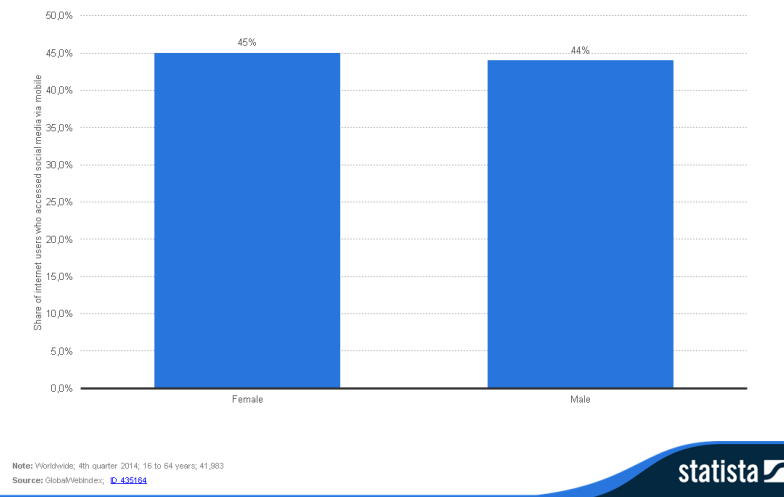


Figure 7.10: Mobile Penetration Rates by Gender. Adopted from (Statista, 2016).

7.2 Mobile

About half of the user groups aged between 16 and 34 access social media via mobile as of Q4 2014. For older age groups, mobile access rates decline. Altogether, this indicates that Catrobat’s target group in terms of age gets increasingly mobile. Concerning gender distribution of mobile social media access, there is no significant difference between males and females.

Figure 7.11 shows the Desktop and mobile usage distribution over active Facebook users from 2010 to 2015 (mobile-only data available from Q4 2011 onwards). Facebook is taken as a representative example as it is the largest social platform and can be used via Desktop and mobile versions.

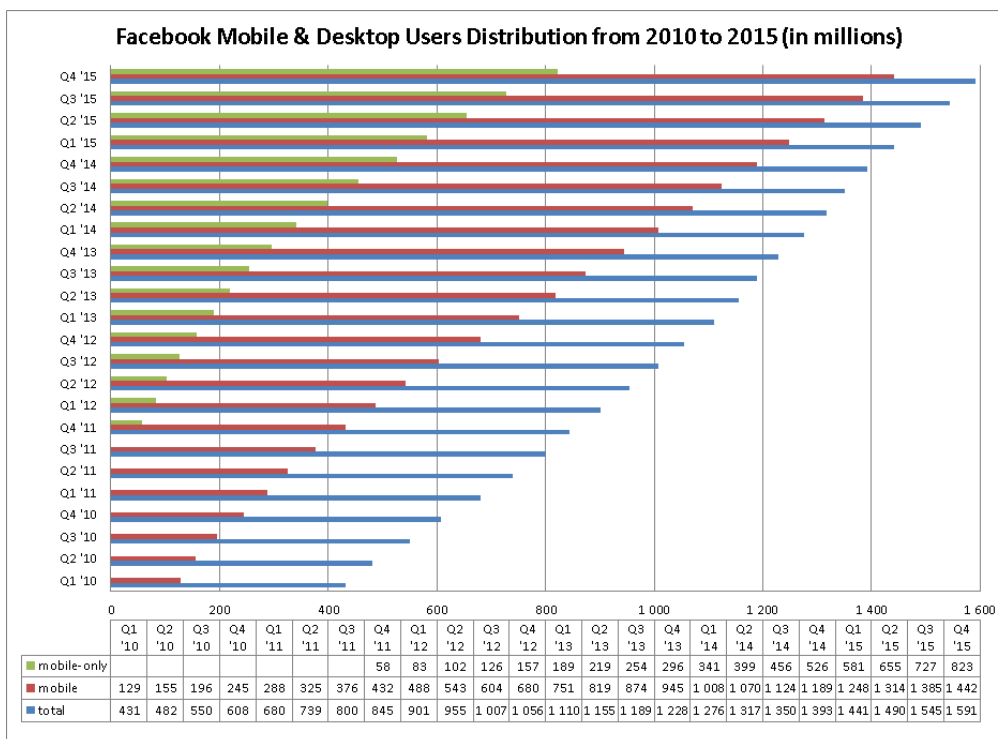


Figure 7.11: Number of Facebook Desktop, Mobile and Mobile-Only Users from 2010 to 2015 Worldwide. Own illustration, adopted from (Statista, 2016).

The chart shows that there has been a large growth on Facebook’s share of mobile users from 2010 to 2015. In Q1 2010 only about 29.9 % (129 mil. of 431 mil. total users) accessed Facebook via mobile. By Q4 2011 already 51.1

7 Social Media Statistics

% (432 mil. of 845 mil. total users) accessed Facebook via mobile and 6.9 % (58 mil. users) mobile-only. By Q4 2015 already 90.6 % (1.442 mil. of 1.591 mil. total users) accessed Facebook via mobile and 51.7 % (823 mil. users) accessed Facebook mobile-only. This approves the finding that mobile social media gets increasingly important.

Mobile statistics by region are given in Section 7.4.

7.3 Social Media Platforms

This section includes statistics about social media platforms that may be relevant to Catrobat. The section is grouped into social networks (including content sharing platforms and microblogs) and messengers.

7.3.1 Social Networks

This section gives statistics about social networks, including content sharing platforms and microblogs. Statistics about the Chinese networks QZone, Tencent Weibo, WeChat and Sina Weibo, the Russian networks VK and Odnoklassniki as well as the Indian network Hike Messenger are given in Section 7.4.

Figure 7.12 shows the age distribution of chosen platforms.

7.3 Social Media Platforms

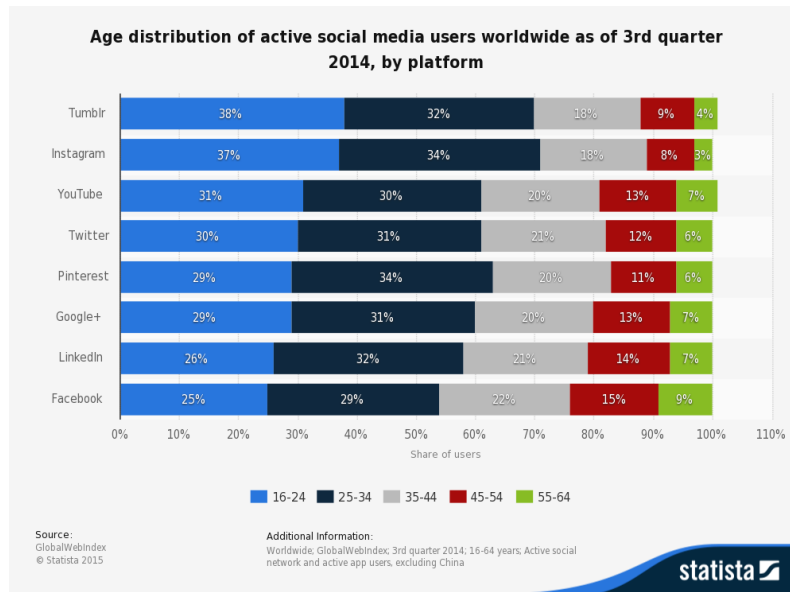


Figure 7.12: Age Distribution of Social Media Users Worldwide by Q3 2014. Adopted from (Statista, 2016).

Three major blocks of age groups can be identified. First, microblog Tumblr and photo sharing platform Instagram have on average the youngest active users with 38 % and 37 %, respectively. Second, YouTube, Twitter, Pinterest and Google+ rank between 29 % and 31 %. Ultimately, Facebook and LinkedIn have a slightly older average audience, with Facebook only having a share of 25 % of the youngest age group. LinkedIn is of no special interest for Catrobat, as it is a business platform.

Figure 7.13 shows the gender distribution of chosen platforms.

7 Social Media Statistics

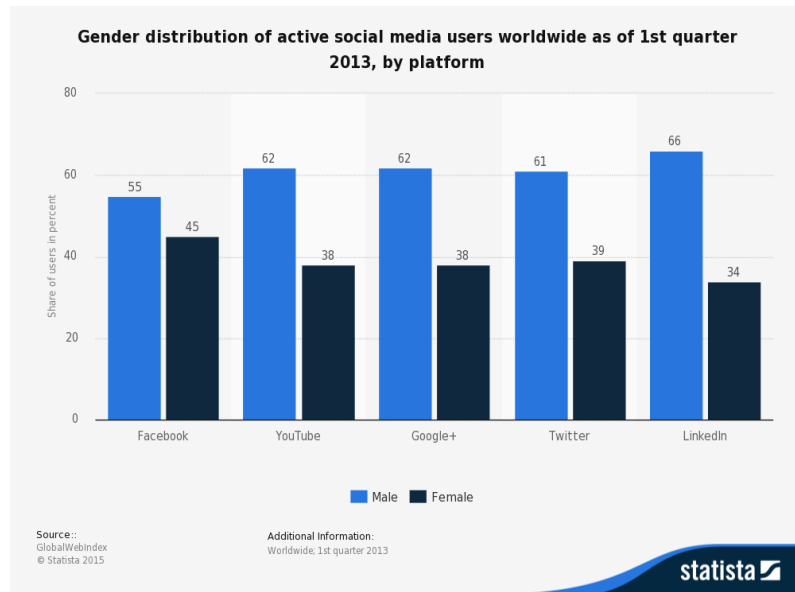


Figure 7.13: Gender Distribution of Social Media Users Worldwide by Q1 2013. Adopted from (Statista, 2016).

The chart indicates that across all listed platforms there are more male than female users. Facebook has the best balance with 55 % male and 45 % female users.

Figure 7.14 highlights main usage reasons of the social network Facebook and the microblog Twitter. The statistic is based on U.S. data, but is probably meaningful to other regions, too.

7.3 Social Media Platforms

U.S. Millennials: Facebook and Twitter usage reasons 2015
Most common reasons for Millennials in the United States to use Facebook and Twitter as of February 2015

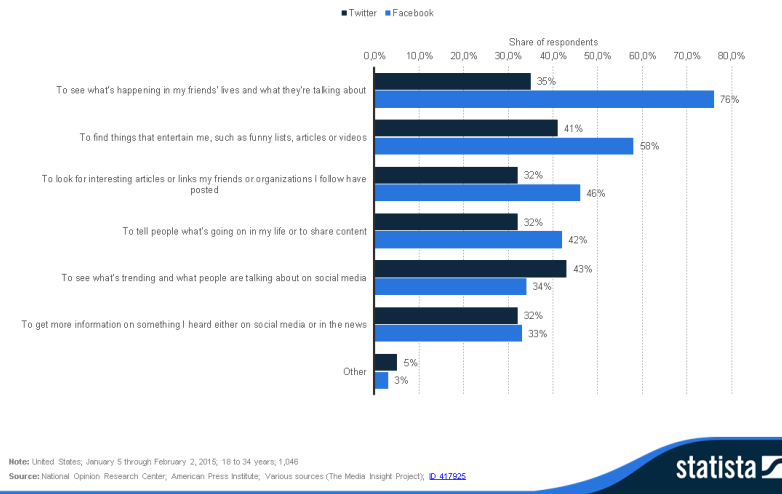


Figure 7.14: Usage Reasons for Facebook and Twitter. Adopted from (Statista, 2016).

The chart indicates that the most common usage reasons are largely the same, but almost always more intensely associated with Facebook than with Twitter. This is true except for trend discovery, which is more associated with Twitter. Conversely, connection with friends, post frequencies and sharing of content is more intense on Facebook. Another common usage scenario of social networks is social login, as discussed in Section 8.1.

Figure 7.15 shows the usage frequency of selected social networks by active users worldwide as of 4th quarter 2014.

7 Social Media Statistics

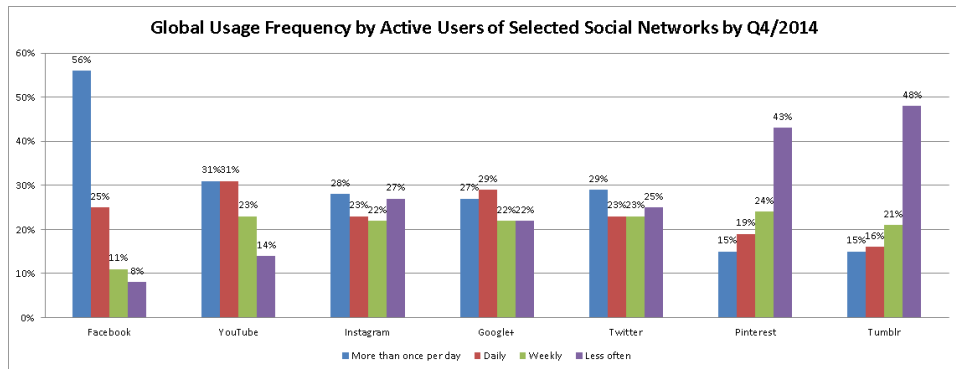


Figure 7.15: Global Usage Frequency by Active Users of Selected Social Networks by Q4/2014. Adopted from (Statista, 2016).

Summarizing, Facebook is clearly used most frequently, followed by YouTube. Then, Instagram, Google+ and Twitter have a relatively similar usage frequency. Pinterest and Tumblr, which also have similar usage patterns, are used least frequently.

Finally, Figure 7.16 shows usage penetrations by all Internet users of the chosen social networks by region (data available for different dates only, as indicated).

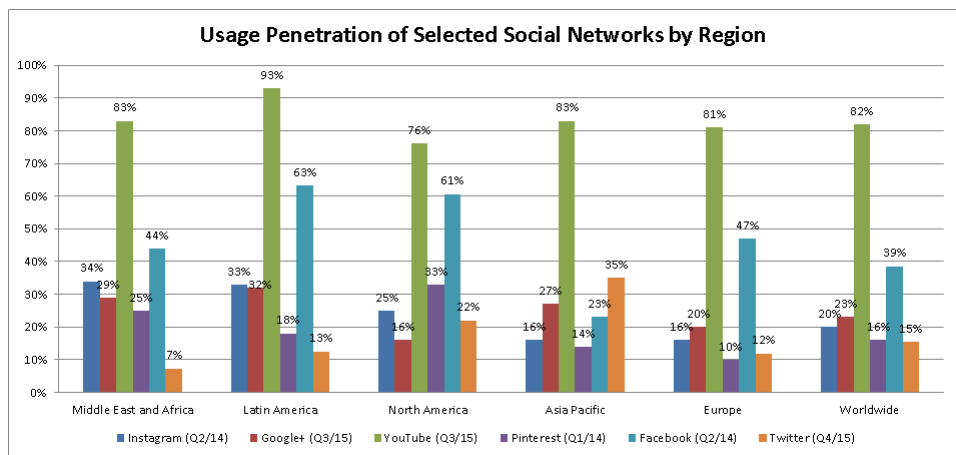


Figure 7.16: Usage Penetration of Selected Social Networks by Region. Adopted from (Statista, 2016).

7.3 Social Media Platforms

The chart shows that YouTube (excluding China) has by far the highest penetration rate with 82 % worldwide. Noteworthy differences are in North America with only 76 % and Latin America with 93 % usage penetration. Facebook, ranking second, is by far most extensively used in Latin and North America, followed by Europe and Middle and East Africa. In the Asia Pacific region only about 23 % use Facebook, which is mainly due to the distinct networks in Russia and China, that will be discussed in Section 7.4. Next, Google+ and Instagram are used most frequently in Latin America and Middle East and Africa. Google+ is less frequently used in North America, and Instagram in Europe and the Asia Pacific region. Pinterest is most popular in North America and least popular in Europe. Finally, Twitter is extensively used in the Asia Pacific region as well as in North America, but relatively little in other regions.

7.3.2 Social Messengers

This section analyzes the usage of social messengers concerning active users and age distributions.

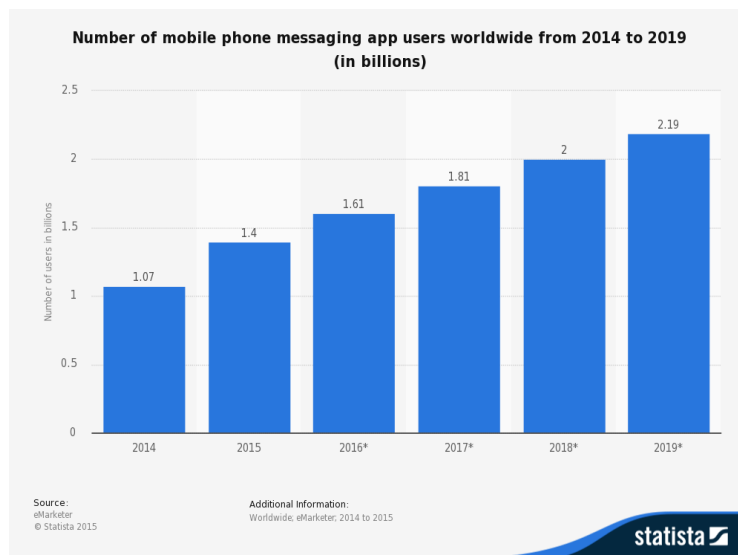


Figure 7.17: Number of mobile messenger users worldwide. Adopted from (Statista, 2016).

7 Social Media Statistics

As shown in Figure 7.17, the number of worldwide mobile phone messaging app users rises sharply. As already noted in previous statistics in this chapter, the growth of messaging apps is bigger than of social networks in general. The number of users is projected to more than double from 2014 to 2019 to approximately 2.2 billion users.

Figure 7.18 shows an overview over most popular messaging apps worldwide in 2016.

Most popular global mobile messenger apps as of January 2016, based on number of monthly active users (in millions)

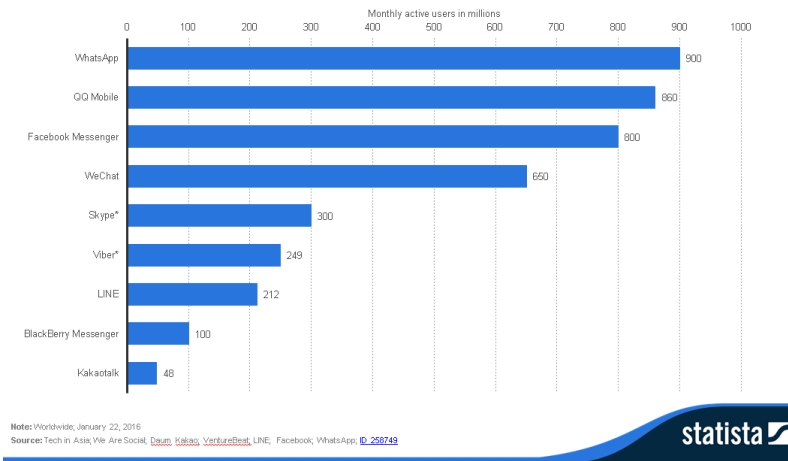


Figure 7.18: Most Popular Global Messenger Apps. Adopted from (Statista, 2016).

As shown in the chart, WhatsApp has currently the most users with 900 millions by January 2016 (and already 1 billion by February 2016, as noted earlier). Second, the Chinese messenger QQ Mobile has overtaken Facebook with 860 million users, pushing Facebook Messenger on third position with 800 million users. WeChat is another Chinese messenger, ranking on fourth position with 650 million users. LINE is another Asian-based (Japanese) messenger that may be of interest for Catrobat, too. It currently has 212 million users with the biggest part coming from Japan.

Subsequently, age distributions of the above mentioned messengers are analyzed. There are no statistics available showing global age distributions

7.3 Social Media Platforms

except for Snapchat. Therefore, data from countries where the respective messengers are popular and frequently used is taken.

First, for Facebook and WhatsApp a study conducted by Bitkom about digital behaviour of German childrens and teens from 10 to 18 is taken. Based on this study, messaging is the most important means of communication with 70 %, even ranking higher than personal communication, that was mentioned by 66 % (top 3 mentions). Social networks have been mentioned only by 32 %. The main reason is that many childrens and teens said that they prefer slim applications (e.g. communication via text, voice and images) over multifunctional social networks. Personal data is also important - 62 % even use nicknames instead of real names. Figure 7.19 compares Facebook and WhatsApp active usages by age and gender.

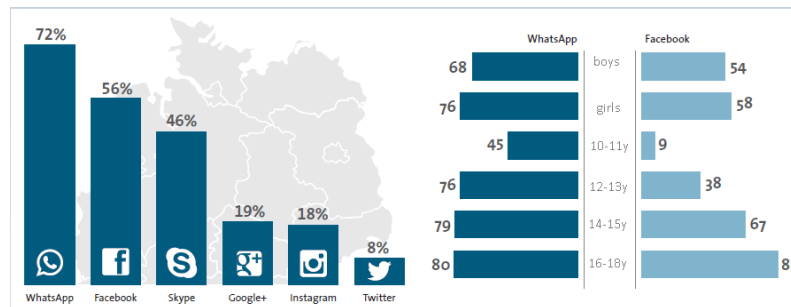


Figure 7.19: WhatsApp and Facebook Age and Gender Distribution among German Childrens and Teens. Adopted from (Bitkom, 2014).

As seen in the chart, WhatsApp has clearly overtaken Facebook as the most popular social network among childrens and teens. Skype, Google+, Instagram and Twitter rank on positions three to six in this study. Concerning gender, WhatsApp is more actively used than Facebook by girls. Boys use WhatsApp also more actively, but the difference is not so big. Altogether, girls use Facebook and WhatsApp more actively than boys. Concerning active usage by age, WhatsApp leads in the sector of 10 to 15 year olds. The older teens get, the less difference in usage is between WhatsApp and Facebook. Finally, the age group between 16 to 18 uses Facebook more actively than WhatsApp. Concluding, WhatsApp is clearly more popular among younger age groups (Bitkom, 2014).

7 Social Media Statistics

Figure 7.20 shows the age distribution of Chinese WeChat users, as the largest user group is from China.

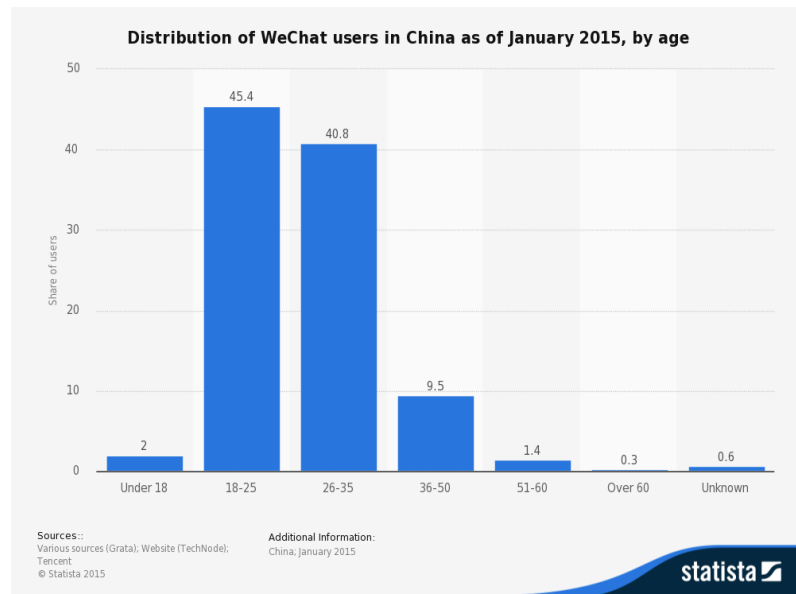


Figure 7.20: WeChat user distribution by age. Adopted from (Statista, 2016).

WeChat is not actively used by teens, as only 2 % of users are under 18. Most users are between 18 to 25 with about 45 %, and 26 to 35 with about 41 % of total users.

Figure 7.21 shows the usage distribution of Japanese LINE users by age, as the largest user group is from Japan.

7.3 Social Media Platforms

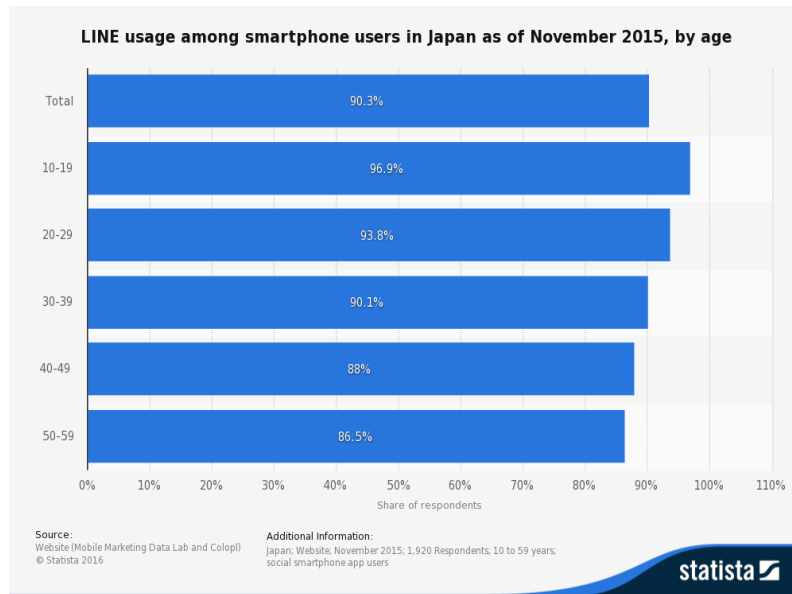


Figure 7.21: LINE user distribution by age. Adopted from (Statista, 2016).

As indicated in the chart, LINE is used on almost any smartphone in Japan. Even 97 % of Japanese teens use LINE. However, the older the age group gets, the less users LINE has.

Figure 7.22 shows the age distribution of Snapchat users worldwide.

7 Social Media Statistics

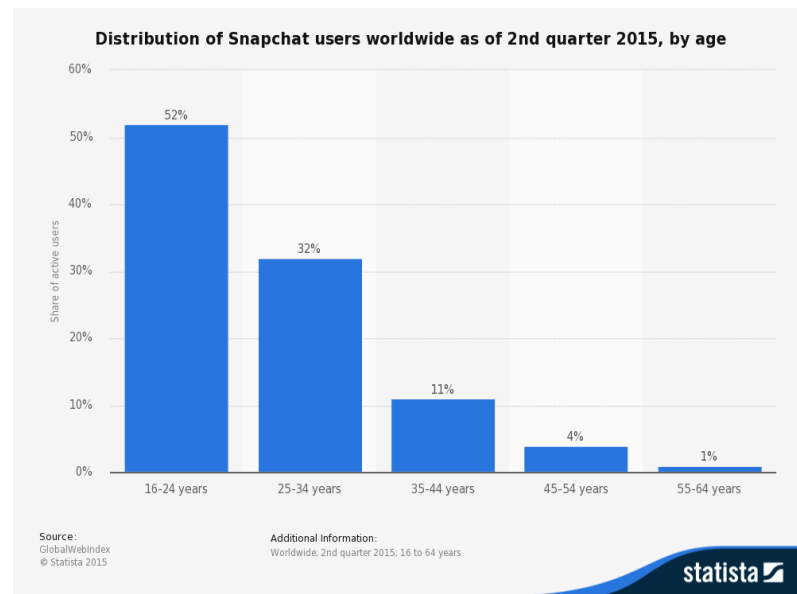


Figure 7.22: SnapChat user distribution by age. Adopted from (Statista, 2016).

Snapchat is clearly most popular for the youngest age group, which is 16 to 24 in this chart. More than half of users are in this age group. For older age groups, the number of users sharply decreases.

Summarizing, it can be said that Snapchat, LINE and WhatsApp are very popular among teens. Facebook Messenger and WeChat are rather popular among older age groups.

There is a trend among teens to use messengers more actively than social networks. Reasons are that teens want to be in a more private context, as many parents joined large social networks such as Facebook where they can see and even comment on their kids photos and other content. Another reason is the popularity of selfies, that are more likely to be shared via messaging services. Especially Snapchat is popular for selfies, as photos are deleted after they have been viewed by friends. Another reason is that the number of messenger users is so big that they already form large social networks (TheGuardian, 2013).

7.4 Countries and Regions

This section shows regional statistics including Catrobat's top countries, as noted in Figure 6.7, and therefore especially focuses on Europe, the United States, China, India, Russia and Brazil.

First, figures 7.23 and 7.24 give an overview over social and mobile usage per region, as of January 2015.

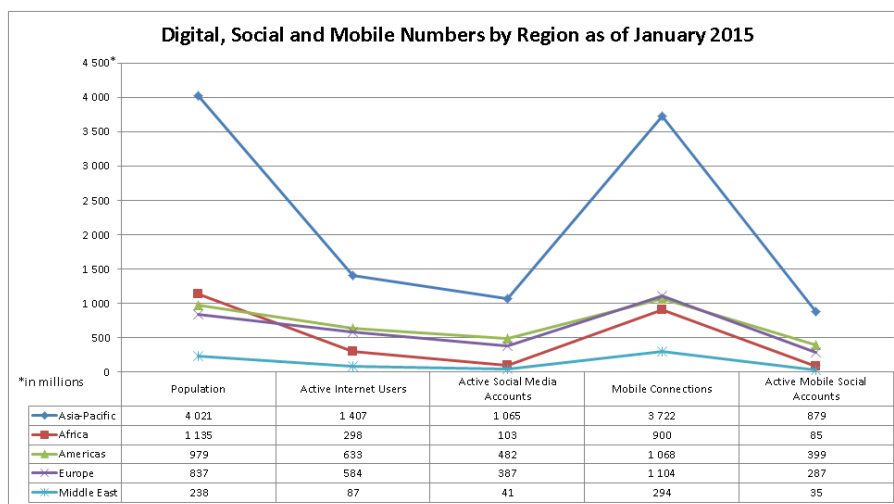


Figure 7.23: Digital, Social and Mobile Numbers by Region as of January 2015. Adopted from (WeAreSocialPteLtd, 2015).

In % of population	Population	Active Internet Users	Active Social Media Accounts	Mobile Connections	Active Mobile Social Accounts
Asia-Pacific	4 021	35,0%	26,5%	92,6%	21,9%
Africa	1 135	26,3%	9,1%	79,3%	7,5%
Americas	979	64,7%	49,2%	109,1%	40,8%
Europe	837	69,8%	46,2%	131,9%	34,3%
Middle East	238	36,6%	17,2%	123,5%	14,7%

Figure 7.24: Digital, Social and Mobile Numbers by Region in Percent of Population as of January 2015. Adopted from (WeAreSocialPteLtd, 2015).

By numbers, the Asian-Pacific region has by far the highest number of Internet users, mobile connections and active social media accounts (including mobile). Concerning penetration measured in % of population, Europe has

7 Social Media Statistics

most active Internet users and mobile connections. In terms of penetration by active social media accounts as well as active mobile social media accounts, the American region is clearly in the lead. In Europe, the Americas and the Middle East the number of mobile connections has already overtaken total population.

Figure 7.25 shows penetration rates of social networks by region in greater detail, including mobile penetration rates.

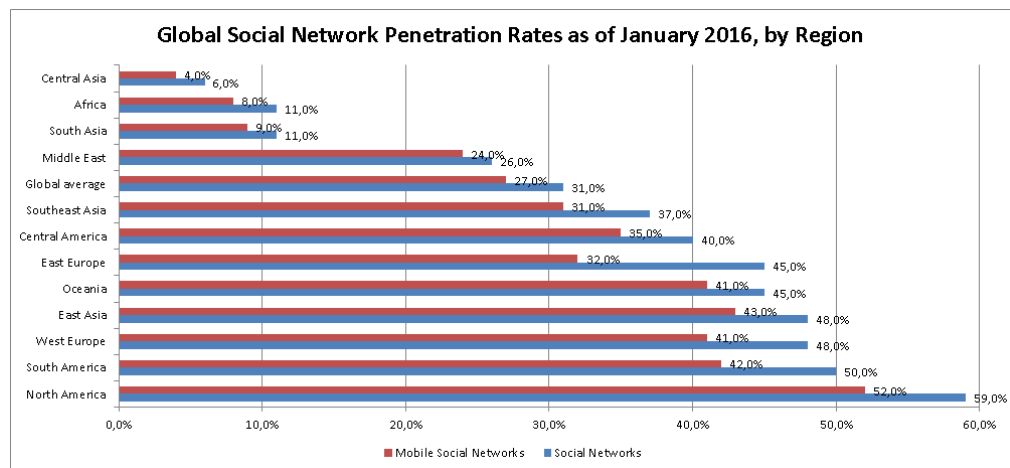


Figure 7.25: Global Social Network Penetration Rates by Region. Adopted from (Statista, 2016).

As already seen before, the Americas have the highest social media penetration. In comparison, North America has a higher social media penetration rate than South America. Then, West Europe, East Asia, Oceania and East Europe have a similar social media penetration rate. Central Asia, Africa and South Asia are relatively little penetrated. In nearly any region mobile usage of social media almost reaches total social media usage. An noticeable exception is East Europe, where only about three fourths access social media via mobile.

7.4.1 Messengers by Region

Subsequently, regional usage statistics of social messengers are given. Figure 7.26 shows the most popular messengers of chosen countries.

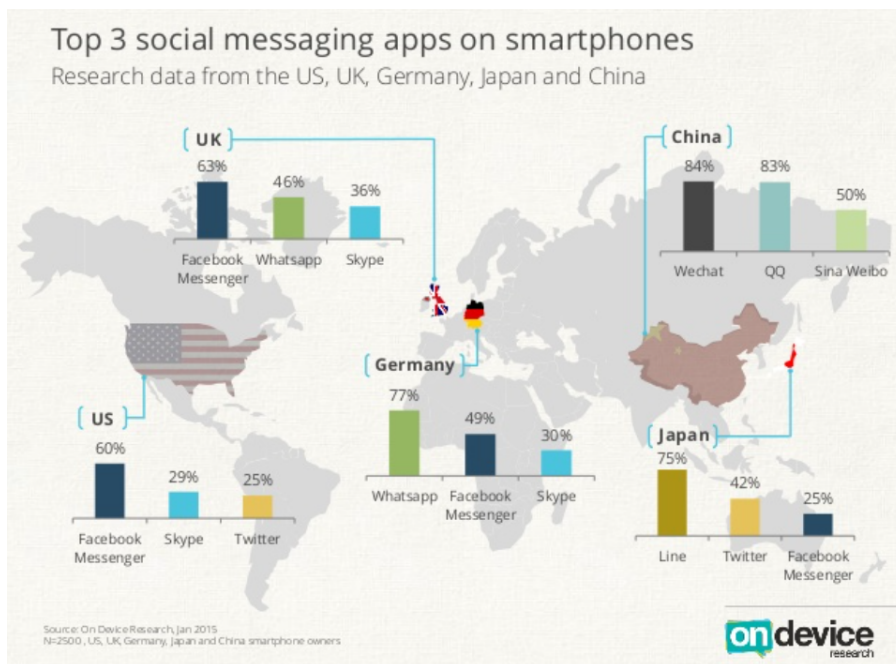


Figure 7.26: Top 3 Messengers by Region. Adopted from (ondeviceResearch, 2015).

In Europe, WhatsApp, Facebook Messenger and Sykpe are most popular, whereas WhatsApp is clearly most popular in Germany and Facebook Messenger in the UK. In the US, Facebook Messenger is most popular, ranking before Skype and Twitter. WhatsApp is currently not in the top three messengers in the US, but potentially gives a come back with its now free version. China has its own national messengers, which are WeChat, QQ and Sina Weibo. In Japan, LINE is clearly most popular, followed by Twitter and Facebook Messenger.

7 Social Media Statistics

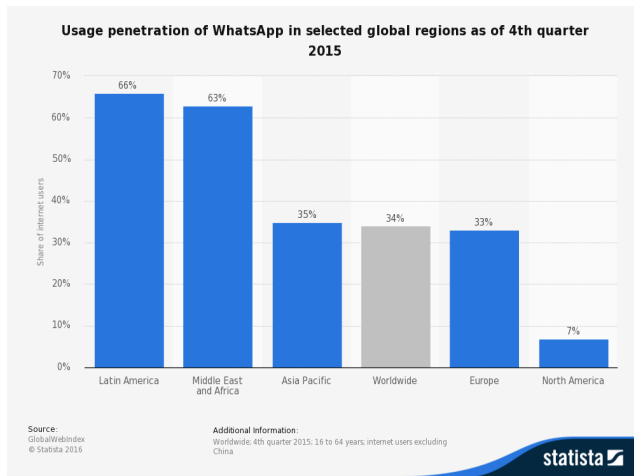


Figure 7.27: WhatsApp users by Region. Adopted from (Statista, 2016).

Figure 7.27 indicates that WhatsApp is extensively used in Latin America as well as in the Middle East and Africa. The Asia-Pacific region (excluding China) and Europe are on the global average of about 34 %. As noted before, North Americans use WhatsApp relatively little with only seven %.

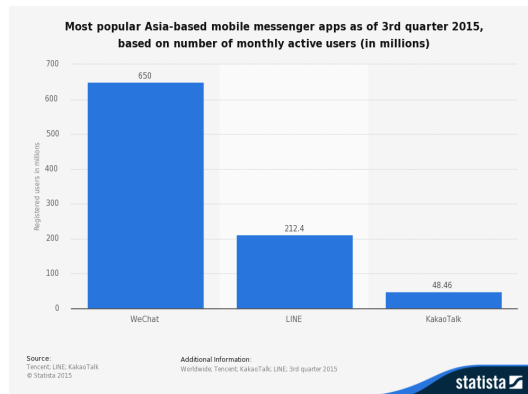


Figure 7.28: Most popular Asian messengers. Adopted from (Statista, 2016).

As shown in Figure 7.28, most used messengers in Asia are WeChat with 650 million, LINE with 212 million and KakaoTalk with 48 million users.

7.4.2 Europe

Figure 7.29 shows the share of Internet users accessing Europe's largest social platforms in selected countries.

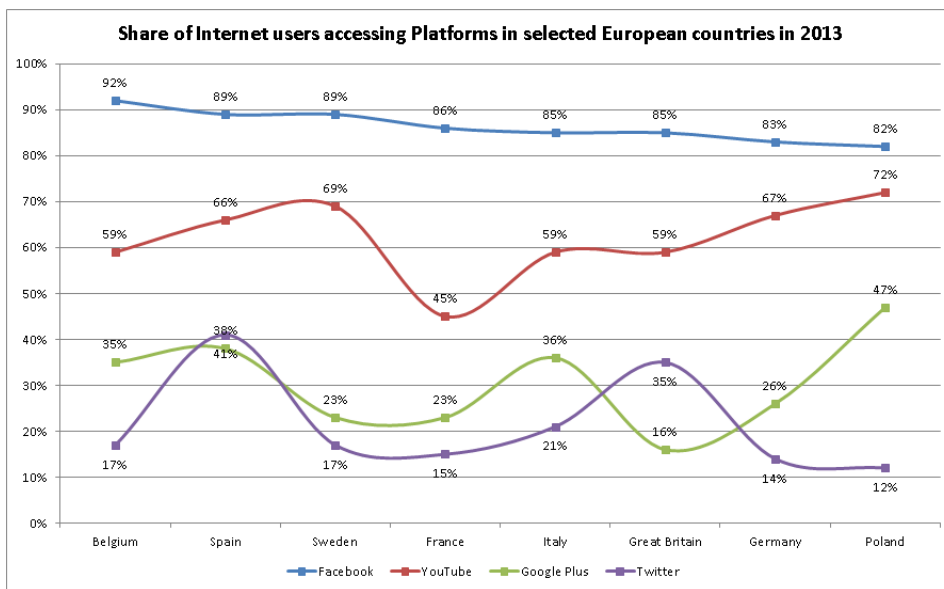


Figure 7.29: Share of Internet users accessing Platforms in selected European countries in 2013. Adopted from (Statista, 2016).

Facebook is clearly the number one social platform across European countries. YouTube is relatively constant the number two platform. However, France is a negative outlier with only 45 % usage. Google+ and Twitter compete each other for place three. However, Twitter is only barely ranking before Google+ in Spain and clearly in the UK. Google+ is most popular in Poland with an impressive usage rate of 47 %, making Facebook serious competition (Facebook has the lowest usage rate in Poland).

7 Social Media Statistics

7.4.3 United States

Figure 7.30 shows the leading social networks in the USA based on visitor numbers.

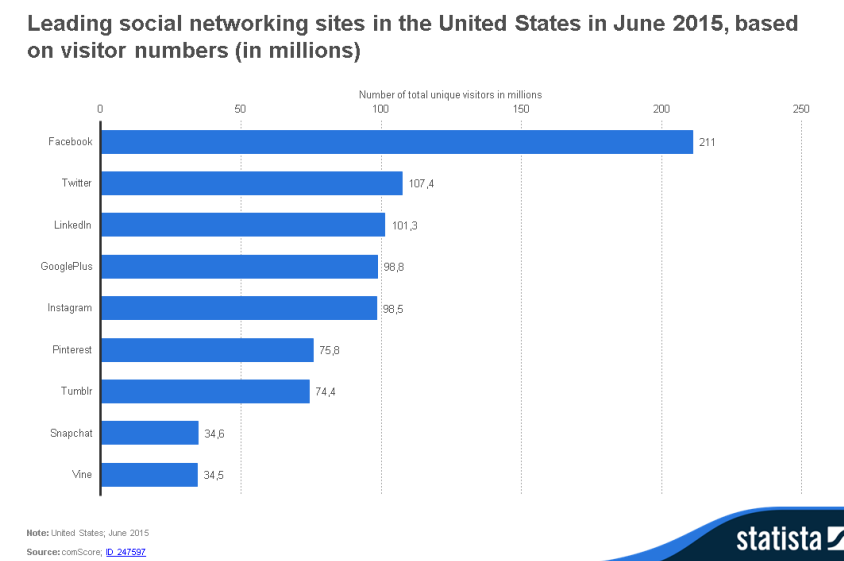


Figure 7.30: Leading Social Networks in the USA as of June 2015. Adopted from (Statista, 2016).

Obviously, Facebook is the number one social network in the USA. Measured on visitor numbers, Twitter is second, LinkedIn third. Then, Google+ and Instagram have almost the same number of visitors. Figure 7.31 shows which social networks have ever been used by young Americans between 12 and 24.

7.4 Countries and Regions

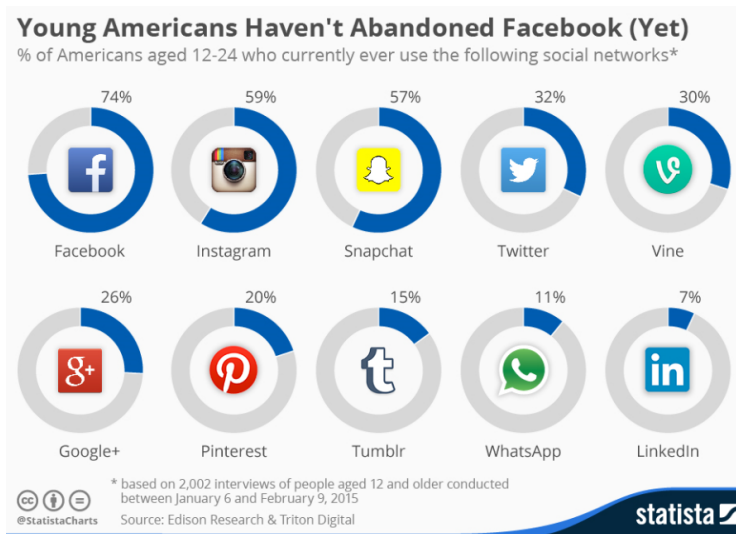


Figure 7.31: Most popular Social Networks in the USA for teens. Adopted from (Statista, 2016).

Accordingly, Facebook is accessed most often by young Americans, followed by Instagram and Snapchat. Twitter, Google+ and Pinterest seem to be less attractive and WhatsApp is even be accessed by a very minor rate of young American. Despite this chart does not say anything about the frequency of use, it shows which networks are considered altogether. As a result, Facebook has not been abandoned by young Americans, as has often been asserted.

7.4.4 Asia

This section addresses social networks in China, Japan, India and Russia.

Figure 7.32 shows most popular Chinese networks based on the number of monthly active users.

7 Social Media Statistics

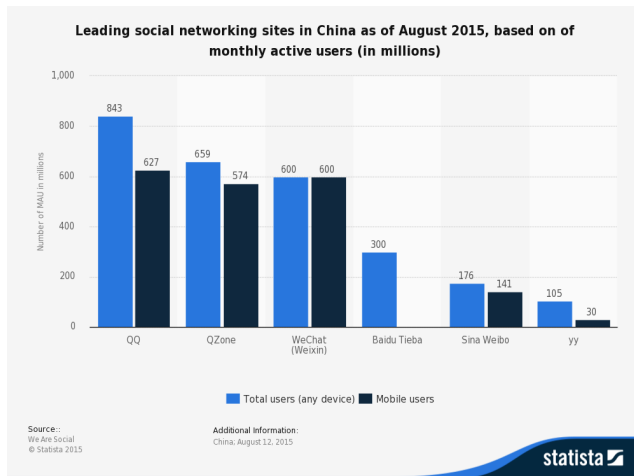


Figure 7.32: Most Popular Chinese Social Networks. Adopted from (Statista, 2016).

QZone is China’s most popular social network with 843 million users, followed by QZone with 659 and WeChat with 600 million users by August 2015. Most users access the respective networks via mobile device. WeChat is even exclusively a mobile messaging app.

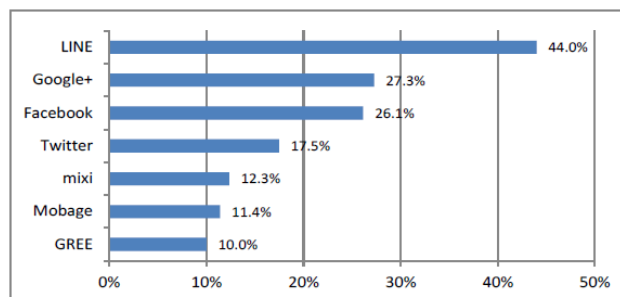


Figure 7.33: Most Popular Japanese Social Networks in 2014. Adopted from (International Affairs and Communications, 2014).

In Japan the most popular social network is the messenger LINE, as already indicated before. Google+ and Facebook follow with a considerable gap.

7.4 Countries and Regions

Penetration of leading social networks in India as of 4th quarter 2015

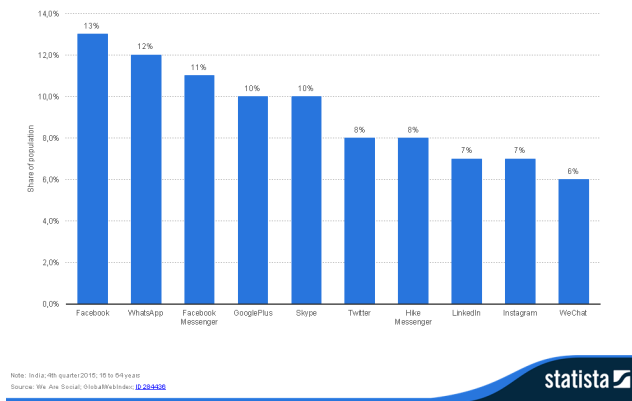


Figure 7.34: Most Popular Indian Social Networks. Adopted from (Statista, 2016).

As shown in Figure 7.34, social networks in India are not yet used by a large percentage of population. Facebook is used most often by about 13 % of Indian population, followed by WhatsApp, Facebook Messenger and Google+. The local Hike messenger is only used by 8 % of population. As a result, India can be identified as an emerging market regarding social media usage, that is, however, already dominated by global leading social networks.

7 Social Media Statistics

Penetration of leading social networks in Russia as of 4th quarter 2014

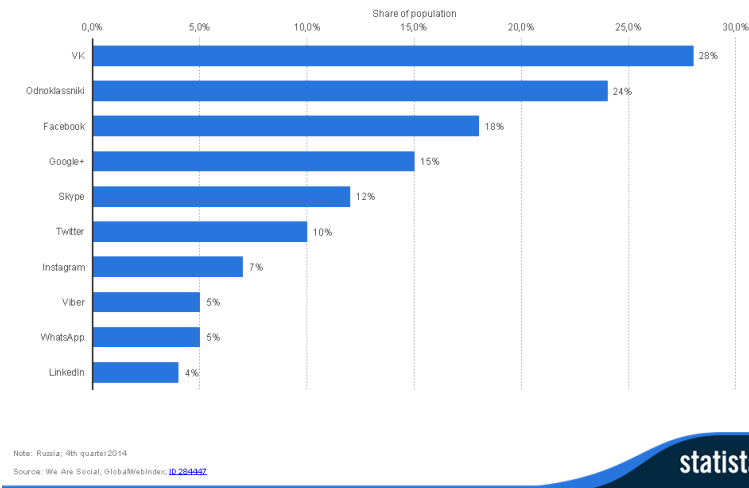


Figure 7.35: Most Popular Russian Social Networks. Adopted from (Statista, 2016).

Russia is, compared to India, obviously more penetrated by social networks than India, as shown in Figure 7.35. However, the most popular social networks are local - VK and Odnoklassniki. Facebook and Google+ rank third and fourth with a clearly lower usage rate, measured in share of population.

7.4.5 Brazil

Finally, the Brazilian social media landscape is shown in Figure 7.36.

7.5 Recommendation

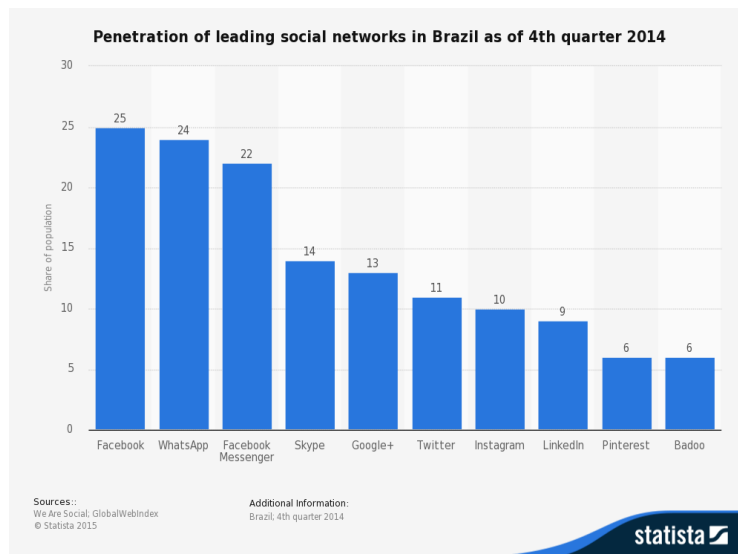


Figure 7.36: Penetration of leading social networks in Brazil as of 4th quarter 2014. Adopted from (Statista, 2016).

Compared to India and Russia, Brazil has the most penetrated social media landscape, measured in share of population. Furthermore, there are no leading locale social networks. Facebook, WhatsApp and Facebook Messenger are the most popular social networks, ranking considerably before Skype, Google+, Twitter and Instagram.

7.5 Recommendation

This final section gives a recommendation for Catrobat on how and where to engage best in different social media platforms, based on the statistical data retrieved in this chapter. From a demographic point of view, age, gender and regions have been investigated. Concerning gender, there were no significant differences found between male and female. Concerning age, platforms with younger participants, better suitable for Catrobat's target group, have been identified. Furthermore, significant differences between used platforms in different regions of the world have been found. Another

7 Social Media Statistics

Table 7.1: Grouping of regions or countries with similar social media platforms.

Region	Platforms	Messengers
Europe, South America, India	Facebook, Google+, Instagram	WhatsApp, Facebook Messenger, Snapchat
USA	Facebook, Twitter, Instagram	Facebook Messenger, Snapchat
China	QQ, Qzone	WeChat, QQ
Russia	Vkontakte, Odnoklassniki	Telegram
Japan	Google+, Facebook	LINE

important criteria for Catrobat is whether or not a platform is actively used via mobile. Actually, there has been no platform identified that is not actively used via mobile. Another differentiation criteria is whether or not a platform is suitable for Marketing activities. Therefore, recommendations per region are given subsequently. The recommendations deal with marketing and technical possibilities.

Before giving recommendations, two predicted social media trends for 2016, that are important to Catrobat, should be taken into account. One trend that continues is that many platforms increasingly monetize, which means organizations have to pay to get more traffic. Social networks such as Facebook (EdgeRank), Google+ (PageRank) and Instagram control which contents are organically displayed to e.g. followers of a page. Only if an organization pays for more visibility, a higher (paid) reach can be accomplished. Twitter is expected to follow this trend in 2016, too. From a marketing perspective, this means that getting visibility on the big social networks will continue to get more difficult and expensive (even to reach followers of pages). The second important trend is that social media gets increasingly private, where communication shifts towards private groups and messaging apps (e.g. WhatsApp, Snapchat, Facebook Messenger, LINE..). As a consequence, social media content should be more private and personal (Jenkins, 2016).

Countries, that have been analyzed and where similar or the same social media platforms are mainly used, are grouped together. Actually, there is one large pool of similar countries and several single Asian countries where the social media landscape is completely different, as shown in Table 7.1.

Considering the above mentioned trends, most important social media platforms and messengers per region have been mentioned separately in Table 7.1. As a result, marketing and technical efforts per region should be diversified. Besides mentioned platforms, **YouTube** is popular in any region, as already seen in Figure 7.16. First, two technical possibilities are highlighted:

- Social Login
- Sharing of Programs

Social login for Facebook and Google+, that has been implemented as part of this thesis, is discussed in Section 8.1. This implementation can be effectively used for Europe, South America, India and Japan. Generally this should be the standard implementation. Different social logins should be made for China (with QQ and QZone), Russia (with VKontakte and Odnoklassniki) and the USA. For the USA, Google+ sign-in should be dismissed and Twitter sign-in added instead (Facebook and Twitter).

Technically, the diversification can be, for example, achieved using gradle product flavors and the respective platform libraries. Therefore, four different product flavors need to be defined in build.gradle and flavor-specific source code has to be added in the flavor-specific src-directories. The resulting .apk-files can be distributed on Google Play using Multiple APK Support. The same package name but a different version code should be assigned to all variants in order to distribute it as one application. To distribute the app versions in the respective countries, country restrictions per app version can be set in the Developer Console⁴. Another possibility would be to check the system default country code of the mobile phone in-app and provide a different UI and source code accordingly. This country code is the preferred locale the user has specified in the Android system settings.

The second recommended technical possibility is to diversify the sharing platforms on Catrobat's community website. Sharing should be possible via the specified platforms and messengers per region, as mentioned in Section 7.1. To check which country the current visitor of the Website is from, the country code in the cookies can simply be checked. Furthermore, it should be distinguished between the Desktop and mobile version of the Website. In

⁴AndroidDevelopers, 2016.

7 Social Media Statistics

the Desktop version social platforms and share via E-Mail are included. In the mobile versions sharing via social platforms and social messengers, as described, should be possible. The standard social sharing implementation, that has been implemented as part of this thesis, is explained in Section 8.3.

Subsequently, certain marketing opportunities are highlighted. As already mentioned in Chapter 6, Catrobat is already active on the big platforms Facebook, Google+, Twitter and YouTube. For the moment, it would rather not be advisable to get active on the unique Chinese and Russian platforms, as Catrobat currently does neither have know-how for these platforms, nor fluent Russian or Chinese speakers in its core team and it is very time-consuming. However, it can be a middle- to long-term goal. Giving attribute to the trend towards messengers as well as their high growth rates and number of users, WhatsApp and Snapchat are recommended platforms to get active from a marketing perspective, too. WhatsApp is already the largest platform in the world and Snapchat has high growth rates and is especially attractive to Catrobat's young target group - teens. Furthermore, WhatsApp is expected to grow in the USA, as it offers a free version now.

WhatsApp can be used as a marketing tool to distribute short messages or breaking news to interested recipients. This can be most effectively done using the WhatsApp broadcast list features where all recipients receive the same message at the same time. The individual recipients do not see other recipients, as each message is retrieved like a regular message. Possible responses are send directly to the broadcaster (in this case to the Catrobat account). Broadcast recipients need to have Catrobat's phone number saved on their phone, so they can easily control whether they want to receive messages or not⁵. Furthermore, messages can more easily be send via browser using the WhatsApp Web feature. The broadcast audience may be build e.g. using platforms where Catrobat is already active (Facebook, Google+, Twitter, hint on YouTube videos) and hints on the community Website. For example, the Austrian newspaper "Kleine Zeitung" already distributes most important daily news via WhatsApp service⁶.

Finally, Snapchat marketing is an interesting alternative to reach teens.

⁵WhatsApp, 2016.

⁶KleineZeitung, 2016.

7.5 Recommendation

Generally, messages, photos etc. are deleted immediately after the recipient has seen it. However, there are also Snapchat stories which are available for 24 hours. Snapchat stories are a series of snaps creating a video narrative that can be used to e.g. tell a story about Pocket Code. Users need to login once within 24 hours to be able to read the story. It is also possible to add music, filters, geo-filters, emojis, text and other multimedia content. Snapchat is especially interesting during events by offering real-time marketing. Another marketing possibility would be, for example, to start a "Snap of the Day" contest with snaps of users while they are using Pocket Code. Snapchat can also be used to "snap" photos behind the scenes (e.g. of people developing with Pocket Code). The biggest benefits for Catrobat using Snapchat is the private nature of marketing and the young target group⁷.

Altogether, Snapchat is the most suitable platform for teens. It is not yet among the biggest platforms but has a higher growth rate than most other leading platforms. Therefore, Catrobat should identify the "Snapchat" trend and offer funny stories on Snapchat from a marketing perspective as well as various implementations within Pocket Code and the Website to e.g. allowing sharing of programs on Snapchat, as discussed in Section 8.3. Furthermore, there are social games such as Habbo⁸ or WeeWorld⁹ that already have millions of teen users. There are no feasible direct marketing possibilities for Catrobat, but these platforms are interesting to monitor or even be active on as a game player to more easily identify trends among teens.

⁷SocialMediaExaminer, 2016.

⁸Habbo, 2016.

⁹WeeWorld, 2016.

8 Implementation

This chapter discusses the implementation of social elements within the practical part of this Master's thesis. Features implemented are authorization with Open Authentication 2.0 (OAuth), automated posts to a Facebook page, Facebook Likes, Google +1's, sharing and program download statistics. The implementation was done for the following Catrobat projects:

- Catroweb - Web server (www.pocketcode.org) with PHP and the Symfony Framework
- Catroid - Android version of Pocket Code

For the iOS version of Pocket Code (Catty), Open Authentication with Facebook and Google+ is still work in progress at the time of submission of this thesis.

Support for the following social platforms was added:

- Facebook
- Google+
- Twitter
- WhatsApp

In the subsequent sections the platform-specific implementations are explained.

8.1 Open Authentication

This chapter discusses the Open Authentication implementation for Catroweb, Catroid and Catty. In the beginning the theoretical background of Open Authentication is discussed.

8 Implementation

Preferred social network ID for social sign-ins as of 1st quarter 2015
Social network ID that social network users worldwide prefer for social sign-ins in 1st quarter 2015

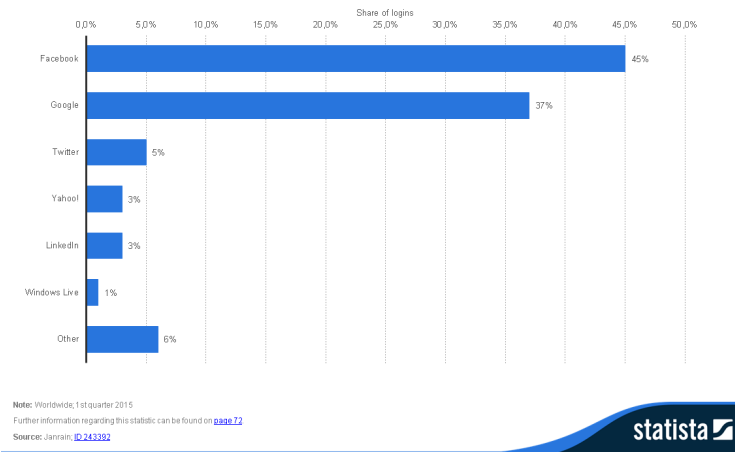


Figure 8.1: Preferred Platforms for Social Signin Worldwide. Adopted from (Statista, 2016).

First, Figure 8.1 shows the most popular platforms that are used for social sign-in. Clearly, Facebook and Google+ are the most popular platforms with 45 % and 37 % share of all sign-ins, respectively. Moreover, Figure 8.2 highlights that the younger users are, the more likely they are to use social sign-in. Based on these facts and findings from Chapter 7, it was decided to add Facebook and Google+ sign-in functionality.

8.1.1 OpenID

Many Internet users have accounts for several different Websites. As a result, these users may have different credentials for their accounts. OpenID was the first attempt to facilitate identity management across Websites and enhance security for users. Technically, OpenID is a protocol which takes an URL as identity. The user has control over the identity URL and over the information (usually OpenID profile information) exposed to various Websites. OpenID does not put any restrictions on the structure of the URL. Among others, OpenID allows a user to login to Websites without entering username and password, to transmit information that Websites may request

8.1 Open Authentication

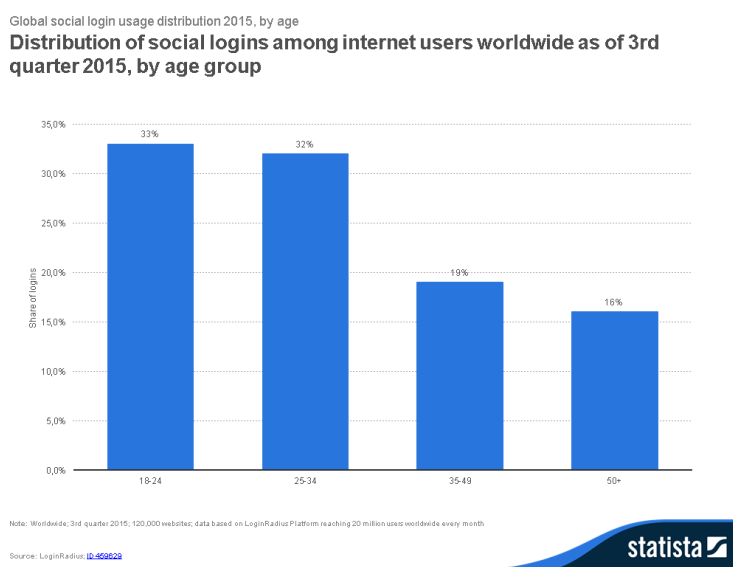


Figure 8.2: Age Distribution for Social Signin Worldwide. Adopted from (Statista, 2016).

or to use single sign on (SSO) mechanism. OpenID-enabled Web sites are referred to as "Relying Party". Registration at the OpenID provider is done with user credentials. Then, the user uses an URL provided by the OpenID provider to login to other Web sites that support OpenID. The user enters the URL into the Web site and is redirected to the OpenID provider Web site to login with his credentials. After successful authentication, the Relying Party Web site usually asks for additional authorization parameters and profile information (Rehman, 2007)[pp. 17-34]. Figure 8.3 shows the OpenID protocol flow in detail.

OpenID was predecessor of the more sophisticated OAuth protocol. Google has built a new system, called OpenID Connect. OpenID Connect is integrated with OAuth 2.0 libraries and can be used to retrieve ID tokens to verify user identity based on the authentication performed by an Authorization Server¹. OpenID Connect will be discussed in more detail at the end of Section 8.1.2. Facebook is no OpenID provider, but has its own build-in security mechanisms that will be described in Section 8.1.3.

¹Google, 2015f.

8 Implementation

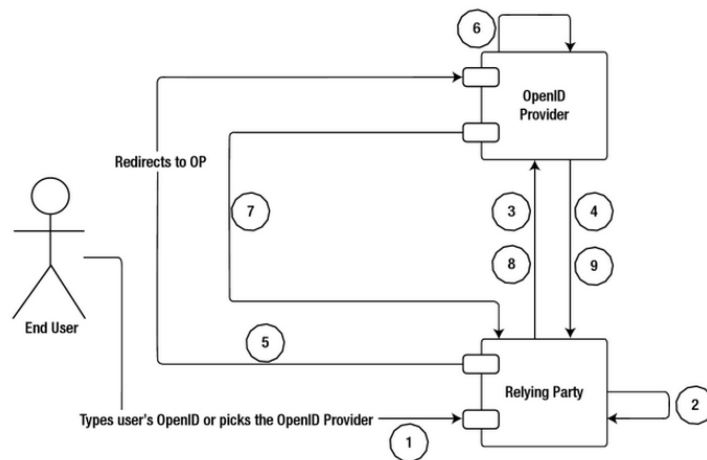


Figure 8.3: OpenID protocol flow. Adopted from (Siriwardena, 2014)

8.1.2 OAuth 2.0

One advantage of OAuth is the standardized mechanism of authentication and authorization for Web-based and native applications with one standard protocol. Also, user credentials are not exposed to requesting applications. Instead, tokens with a limited life-time are issued. This enhances trust, limits risk for phishing and eliminates the need to change the password in order to revoke access from all connected applications. With social networks such as Facebook or Google+, that provide OAuth, also huge centralized data sources are available. These social network data sources give access to the social graph (friends, followers, contacts etc.) and user activities, allow to store data online (e.g. in Google Docs or Dropbox) and create Mashups. Subsequently, the OAuth 2.0 standard will be discussed. OAuth 2.0 is a standardized mechanism that eliminates major deficiencies of OAuth 1.0 by not only defining client-to-server authentication flow, but also authentication flow for mobile apps, desktop and Javascript applications as well as eliminates the need for complex cryptographic signatures by issuing tokens and using SSL/TSL protection (Boyd, 2012) [pp. 1-5].

Technically, the following roles are involved in OAuth protocol flows:

- Resource server

8.1 Open Authentication

- Resource owner
- Client
- Authorization server

The resource server is normally owned by the API provider and stores user data. The authorization server issues tokens to grant access to user data. The resource owner is the user that can grant access to the data. Finally, the client is the application that requests access to user data and needs to register with the OAuth 2.0 API provider. OAuth 2.0 supports server-side Web applications, client-side applications running in a browser and native applications (such as mobile apps) as **client profiles**. Server-side Web applications make API calls with a server-side programming language (e.g. PHP) and are supposed to keep the client secret and access tokens confidential. In contrast, in client-side applications (e.g. JavaScript) the application code is visible to the user. Therefore, client secret and access tokens are not believed to be confidential and certain restrictions from API providers are common. Similarly, native applications are not trusted as they could be decompiled. Independently from the client profile, the ultimate goal of OAuth is to obtain a bearer token (a token without signature) in order to perform API calls on a user's or the application's behalf. Tokens can be send either in or along with requests. The preferred method to send tokens along with requests is within the HTTP authorization header because it is rarely logged or cached by proxy servers or browsers. Alternatively, the token can be send as query parameter or form-encoded body parameter. Concerning the **authorization flow** there are four different grant types within the OAuth 2.0 core specification, depending on the client profile:

- Authorization code
- Implicit grant for browser-based client-side applications
- Resource owner-based grant
- Client credentials

The authorization code is appropriate for secure server-side Web applications. After successful authorization the user is redirected back to the client and an authorization code is send from the authorization server to the client as URL query parameter. This authorization code must then be send to the server and exchanged by server-to-server communication using the client id and client secret. Client secret, client id and token cannot be read by the

8 Implementation

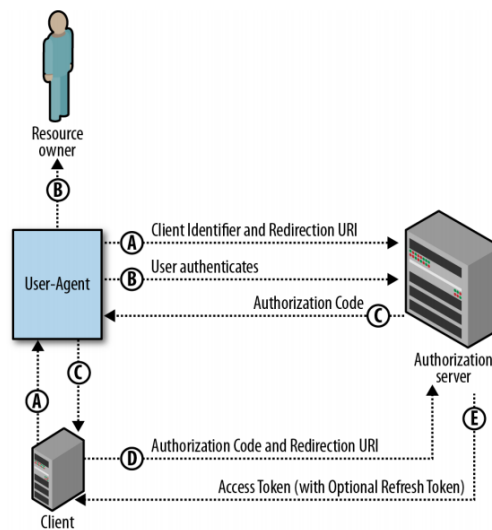


Figure 8.4: Server-to-server authorization flow. Adopted from (Boyd, 2012)

resource owner. This form of authorization allows for both at the same time, long-lived refresh tokens and short-lived access tokens. Figure 8.4 shows the server-to-server flow graphically. The implicit grant for browser-based client-side applications is not secure. The entire code (including client id and client secret) and the access token (eventually leaked by the browser) can be accessed by the user (e.g. within JavaScript). A token is returned immediately after authorization by the user, omitting an authorization code. Only short-lived access tokens are available using this authorization flow. Figure 8.5 shows the client-to-server flow graphically. The resource owner password-based grant still uses user credentials for authentication. However, it is more secure than traditional user name and password authentication because only the token needs to be stored but not the password. Nevertheless, this form of authentication is only used for trusted applications (e.g. Facebook Messenger asking for access to the Facebook server) because the password is exposed to the app. Client credentials are intended for use by clients to receive access tokens for resources owned by themselves (Boyd, 2012) [pp. 6-12].

Server-side and client-side application flows are compared in Table 8.1.

8.1 Open Authentication

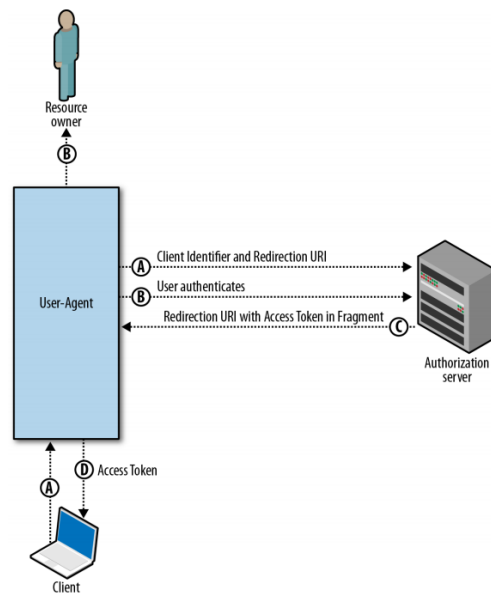


Figure 8.5: Client-to-server authorization flow. Adopted from (Boyd, 2012)

For native mobile apps some special considerations must be taken into account. The choice of authorization flow depends on whether the app has a backend server or not. If there is a Web server backend the client-to-server or server-to-server flow can be used depending on the needs for security and types of tokens for the app, as described in the previous section. Without a Web server backend a native authorization flow must be used. Restrictions are that the `redirect_uri` parameter cannot be used and that client secret values are object to exposure (Boyd, 2012) [pp. 45-50]. The official OAuth 2.0 specification can be found on <http://tools.ietf.org/html/rfc6749>.

An interesting feature is Hybrid Auth, which is the combination of OAuth and OpenID. This combination allows applications to get access to the users' identity and resources in a single step. Using Hybrid Auth a much larger data pool is available to an application (Blanc, 2012) [pp. 501-510].

Another recent technology is OpenID Connect, which was standardized in February 2014. OpenID Connect is successor of OpenID 1.0/1.1/2.0 and OpenID extension for OAuth. It has elements of OpenID as well as of OAuth and is used for authentication and authorization in a RESTful way. Actually,

8 Implementation

	Server-side authorization flow	Client-side authorization flow
Use Cases	<ul style="list-style-type: none">- Long-lived tokens required- OAuth client is a Web server- User should not have access to the token	<ul style="list-style-type: none">- Only temporary data access required- OAuth client running in browser- User regularly logged in at the API provider
Security	<ul style="list-style-type: none">- Client id, client secret and access token only known to server- Long-term tokens stored in database for offline data access may cause risk for compromise	<ul style="list-style-type: none">- Less secure because browser knows tokens- Only short-lived tokens- Reauthentication required to refresh tokens

Table 8.1: Server-side vs. Client-side authorization flow. Adopted from (Boyd, 2012)

OpenID connect builds directly on the OAuth 2.0 protocol and introduces an additional ID token besides the access and refresh tokens. The ID token is a JSON Web token (JWT) that contains user information and is send from the authorization server to the client. Requests for user profile information can either be made within the initial OpenID Connect authentication request (as claim parameters), or later using the authorization server's userinfo endpoint with a valid OAuth 2.0 token (as scope parameters) (Siriwardena, 2014) [p. 181-200].

8.1.3 Facebook

Facebook authentication and authorization with OAuth 2.0 is supported for the mobile Android SDK, iOS SDK and Windows Phone SDK as well as for the Web-based client-side JavaScript SKD and server-side PHP SDK². The following sections explain the most important aspects of the Facebook API integration.

²Facebook, 2015e.

Security Considerations

Security features that are provided by Facebook are the OAuth 2.0 Login Workflow, access tokens and the permission system. The following access tokens are available from Facebook:

- User Access Token
- App Access Token
- Page Access Token
- Client Token

These tokens either identify a user, a page or a client and allow to make Graph API calls on their behalf. The tokens contain information about their expiration date and for which app they were generated. A **User Access Token** identifies a specific user of an app and allows to read, write or modify this user's Facebook data. These tokens are obtained during login with permission of the user. **App Access Tokens** are necessary to read and modify app settings (e.g. create test users) or publish Open Graph actions. This kind of access token can be obtained with a server-to-server call using the pre-agreed secret. **Page Access Tokens** allow to read, write or modify data on a user's Facebook Page. A valid User Access Token is required in order to get a Page Access Token (with `manage_pages` permission) via Graph API call. Finally, a **Client Token** is a less-secure token that identifies an app. It can be used in mobile applications to get access to a rather limited subset of app-level API's.

Generally, short-term (lifetime approximately 1-2 hours) and long-term tokens (lifetime approximately 60 days) are available, depending on the use case and level of security. Tokens are also portable, meaning a token obtained on a client can be passed to the server, and vice versa, in order to make Graph API calls. Pocket Code needs a workflow in which different app types, namely the native mobile clients Catroid and iCatty, the client-based Website of Catroweb, and the Catroweb Web server, are considered. Figure 8.6 shows the possible configurations on Facebook.

In order to meet the requirements of the different app types used for Pocket Code and to provide high security with a comparatively less complex implementation the decision was to take the fourth configuration shown

8 Implementation

Configuration	Advantages	Disadvantages	Security Notes
Login happens in a Web Client. API requests happen in a Web Client (short-term token).	Simple to implement.	No offline posting, no long-term access, must auth often.	
Login happens in a Native Mobile Client or a Web Client with Long Term Token. API requests happen in the Client (with long-term token).	Auth not required often with long-term token.	No offline posting.	
Login happens in a Web Client. API requests happen in the Web Client (long-term token after code exchange).	Extra security in certain situations.	Harder to implement. No offline posting. Shouldn't be used in the usual case.	This is only useful in very specific situations See the docs below for more information.
Login happens in a Native Mobile Client or a Web Client. API requests happen on a Server (with long-term token).	Offline posting. Takes advantage of security features available with server-based calls.	Client must call the server to proxy any calls.	You should use <code>appsecret_proof</code> for any calls.
Login happens in a Native Mobile Client or Web Client. API requests happen on a Server or in the Client.	Offline posting. User-driven posting from the client.	Implementation can be complex.	You should use <code>appsecret_proof</code> for any calls you make from the server.

Figure 8.6: Possible Facebook Login workflows. Adopted from (Facebook, 2015a)

8.1 Open Authentication

in Figure 8.6. This means login happens within the mobile apps of Catroid and iCatty or the Catroweb client Web page. However, tokens are stored in a database on the Catroweb server to mitigate the risk for token hijacking. Moreover, all requests are proxied through the server and signed with the app secret to enhance security. Besides enhanced security this type of configuration allows for offline posting (not used at the moment). The only difference between the clients is, that the Catroweb client receives a short-term token and the mobile clients a long-term token. However, there is the same workflow used for Web and mobile clients.

The Web and mobile clients first receive a user access token when a user logs in and authorizes the app (accepting requested permissions). This user access token could principally be used to make Graph API calls. However, to enhance security Graph API calls are only permitted to be proxied over the Catroweb server. This behaviour is enforced by enabling the `appsecret_proof` option in the Facebook Developer Console. The Web server has to add client secret and client id to any call. Therefore, the client token is send to the server. The server adds the app id and the app secret to the client token and requests a long-term token from the Facebook server. With this long-term token, the Catroweb server can make direct calls to the Facebook Graph API. The Catroweb server acts as proxy for the clients to perform any Graph API calls³. The workflow is graphically shown in Figure 8.7.

Altogether, this workflow ensures that the app secret and the App Access Token is neither included in inspectable JavaScript code of the Catroweb client nor in the decompilable apps Catroid and Catty. It also ensures that client tokens are always regenerated for the Catroweb clients and exchanged again for a long-term server token, when the server token is expired after 60 days⁴. Moreover, Graph API calls that are done on a server-to-server basis have been secured by enabling the `appsecret_proof` parameter to prevent abuse of hijacked tokens. The app secret proof actually is a sha256 hash value of the access token using the app secret as key. This requirement has also been enforced in the Facebook Developer Console⁵. Token hijacking is additionally prevented by checking against a debugging endpoint. The

³Facebook, 2015a.

⁴Facebook, 2015g.

⁵Facebook, 2015i.

8 Implementation

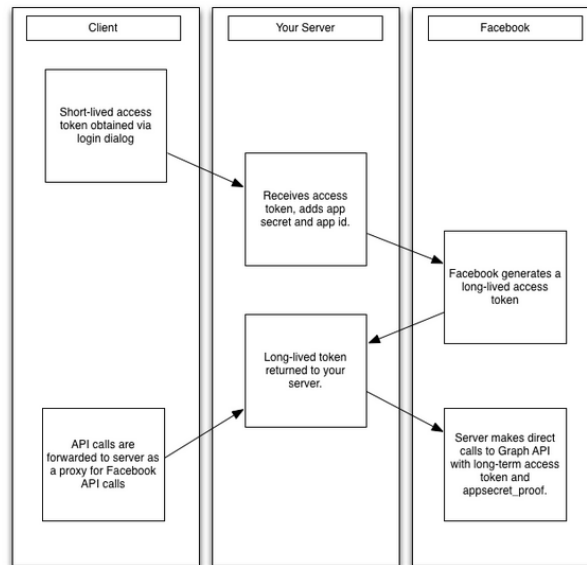


Figure 8.7: Catroweb Client Workflow - Secure Server-to-Server Calls. Adopted from (Facebook, 2015a)

assumption is that a server must never trust that a token received by the client actually comes from the client. The server makes a Graph API request to the debugging endpoint which delivers exact information (app id, application, user id, scope etc.) about the token. Finally, security can be enhanced by locking down specific settings in the Facebook Developer Console. This settings include a whitelist of OAuth redirect URL's to prevent cross-site-scripting (XSS) attacks, the ability to disable certain login workflows, a server IP whitelist from where Graph API calls can be made (does not affect calls with User Tokens), or the app type. The app type does not need to be set to NativeDesktop as communication from the mobile apps to Facebook is delegated through the Catroweb server. This form is more secure and allows to perform more information-critical requests⁶.

⁶Facebook, 2015g.

Permissions

Facebook authorization is handled with permissions for specific resources or user data. Permissions can be asked for during login or API calls. As best practice, permissions should only be asked for when needed. Furthermore, permissions only need to be granted on app-level. This means permissions granted to Catroweb are automatically valid for Catroid and Catty, too⁷. Permission that are required for Pocket Code users are:

- `public_profile` (automatically granted during login)
- `email` (during login - needed for registration and merging with existing accounts)

One permission that will be necessary for the planned next features mentioned in Section 8.5 is:

- `user_friends` (Catroid - asked for when challenging friends)

The `publish_actions` permission for creating posts on Facebook is only necessary in the case an own dialog is used⁸. In the case of Pocket Code and Catroweb, as described in Section 8.3, the standard feed dialog will be used. Therefore the `publish_actions` permission is not necessary.

Best Practices

The following guidelines are recommended by Facebook in order to achieve high conversion rates:

- Ask for permissions in context
- Explain to the user how Facebook data is used by the app
- Use official Login buttons from the Facebook SDK
- Test the Login flow
- Submit the app for review, if applicable

⁷Facebook, 2015h.

⁸Facebook, 2016b.

8 Implementation

As already discussed, permissions should only be asked for when they are needed. This should lead to higher conversions rates and more logins. For example, requesting the `publish_actions` permission during login may negatively impact users. Users understand why certain permissions are needed better in the context they are actually needed. Users should also be informed how their Facebook data is going to be used by the application to give them a better user experience. For example, this can be done in own pre-permission dialogs before permissions are actually requested. Furthermore, the official Login buttons from the Facebook SDK should be used because they are already known and consistent across all platforms. A clearly visible Logout option must also be available. Of course, the login flow should be tested under various conditions. Finally, the application is applicable for review if it requests other permissions than `public_profile`, `user_friends` and `email`. Therefore a review is not necessary for Pocket Code at the moment⁹.

The Login Review is based on the utility and visibility of the extended permissions requested. Utility means that the requested permissions must clearly improve user experience. Visibility refers to the fact that data obtained with the permissions must have a direct and visible use for the user¹⁰.

Login Workflow

The Facebook login workflow has been described in the previous sections and shown in Figure 8.7. To give an impression how this workflow is implemented, the PHP code snippet used for exchanging the Facebook client token for a long-term server token is shown below. This server function is called by the Android, iOS and Javascript clients after a user successfully logged in and received a client token. In-line commentary was added to describe specific code blocks.

```
public function exchangeFacebookTokenAction(Request $request)
{
```

⁹Facebook, 2015d.

¹⁰Facebook, 2015f.

8.1 Open Authentication

```
/**
 * @var $user_manager UserManager
 * @var $facebook_user User
 * @var $user User
 * @var $response FacebookResponse
 */
$ret_array = array();

//Read client parameters send along with request
$session = $request->getSession();
$client_token = $request->request->get('client_token');
$session_state = $session->get('_csrf/authenticate');
$request_state = $request->request->get('state');

$facebook_id = $request->request->get('id');
$facebook_username = $request->request->get('username');
$facebook_mail = $request->request->get('email');
$locale = $request->request->get('locale');

// Ensure that there is no request forgery going on: Check that the user
// sending this request is the user that was supposed to send it
if (!$session_state || !$request_state || $session_state != $request_state) {
    //This might be a Session Hijacking attempt!
    return new Response('Invalid state parameter', 401);
}

//Initialize Facebook SDK with app id and app secret from configuration
$application_name = $this->container->getParameter('application_name');
$app_id = $this->container->getParameter('facebook_app_id');
$client_secret = $this->container->getParameter('facebook_secret');

if (!$client_secret || !$app_id || !$application_name) {
    return new Response('Facebook app authentication data not found!', 401);
}

$this->initializeFacebook();
```

8 Implementation

```
//Set default access tokens for Graph API calls
//Mobile: client token, Javascript: app token
if ($request->request->has('mobile')) {
    $this->setFacebookDefaultAccessToken($client_token);
} else {
    $this->setFacebookDefaultAccessToken();
}

//Perform Graph API call to exchange client token
//for a long-term server token.
//App id, app secret and the token to exchange are necessary arguments.
//Grant for the authorization flow is always 'fb_exchange_token'
try {
    $response = $this->facebook->post('/oauth/access_token',
        array('grant_type' => 'fb_exchange_token',
            'client_id' => $app_id,
            'client_secret' => $client_secret,
            'fb_exchange_token' => $client_token));
    $graph_node = $response->getGraphNode();
    $server_token = $graph_node->getField('access_token');
} catch (FacebookResponseException $exception) {
    return new Response(
        "Graph API returned an error during token exchange for
        'GET', '/oauth/access_token'", 401);
} catch (\Exception $exception) {
    return new Response(
        "Error during token exchange for 'GET', '/oauth/access_token'
        with exception" . $exception, 401);
}

//The validity of the successfully exchanged token is checked:
// - Token was issued for same app (app id, app name)
// - Token was issued for correct user (Facebook user id)
try {
    $token_graph_node =
        $this->checkFacebookServerAccessTokenValidity($server_token);
    $app_id_debug = $token_graph_node->getField('app_id');
```

8.1 Open Authentication

```
$application_name_debug = $token_graph_node->getField('application');
$facebook_id_debug = $token_graph_node->getField('user_id');
} catch (FacebookResponseException $e) {
    return new Response(
        "Graph API returned an error during token exchange for 'GET',
        '/debug_token'", 401);
} catch (FacebookSDKException $e) {
    return new Response(
        "Error during token exchange for 'GET', '/debug_token'
        with exception" . $e, 401);
}

//Make sure the token we got is for the intended user
if ($facebook_id_debug != $facebook_id) {
    return new Response("Token's user ID doesn't match given user ID", 401);
}

//Make sure the token we got is for our app
if ($app_id_debug != $app_id || $application_name_debug != $application_name) {
    return new Response("Token's client ID or app name does not match app's.", 401);
}

//Check whether there is already a user with either the Facebook id
//or email address in the server database
$user_manager = $this->container->get("usermanager");
$user = $user_manager->findUserByEmail($facebook_mail);
$facebook_user = $userManager->findUserBy(array('facebookUid' => $facebookId));

if ($facebook_user) {
    //There is already a user with the unique Facebook id
    //Only the user access token needs to be refreshed
    $facebook_user->setFacebookAccessToken($server_token);
    $user_manager->updateUser($user);
} else if ($user) {
    //There is already a user with a native account using this email address.
    //We need to merge the native and Facebook user and set access tokens.
    $this->connectFacebookUserToExistingUserAccount($user_manager, $request,
```

8 Implementation

```
        $ret_array, $user, $facebook_id, $facebook_username, $locale)
    $user->setFacebookAccessToken($server_token);
    $user_manager->updateUser($user);
} else {
    //Create a new Facebook user account including the token
    $this->registerFacebookUser($request, $user_manager, $ret_array,
        $facebook_id, $facebook_username, $facebook_mail, $locale,
        $server_token);
}

//Check whether there have been any errors and return a json string
//including status code and answer to the requesting client
if (!array_key_exists('statusCode', $ret_array) ||
    !$ret_array['statusCode'] == StatusCode::LOGIN_ERROR) {
    $ret_array['statusCode'] = 201;
    $ret_array['answer'] = $this->trans("success.registration");
}

return JsonResponse::create($ret_array);
}
```

8.1.4 Google

Google authentication and authorization with OAuth 2.0 is supported for the mobile Android SDK and iOS SDK as well as for the Web-based client-side JavaScript SKD and server-side PHP SDK¹¹. The following sections explain the most important aspects for the Google API integration.

Security Considerations

The security considerations of Google authentication with OAuth 2.0 are almost the same as with Facebook. For the implementation the same workflows are used for the Catroweb and mobile clients. Communication is in-

¹¹Google, 2015a.

8.1 Open Authentication

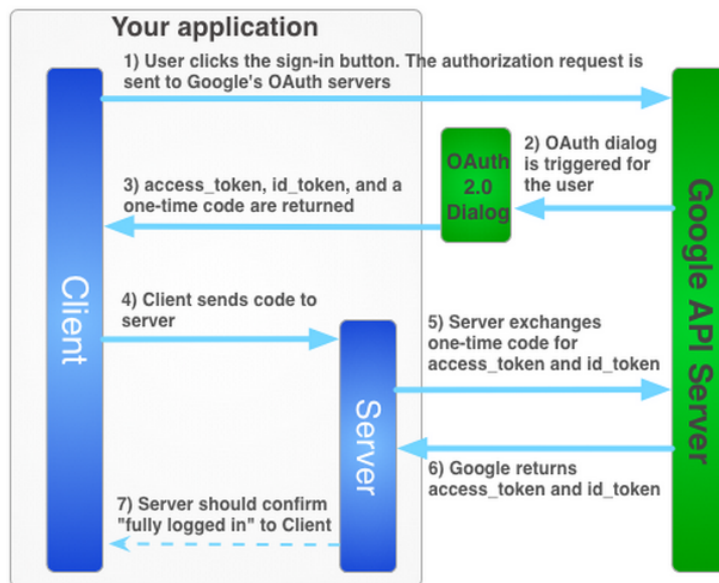


Figure 8.8: Google+ Sign-In for server-side apps. Adopted from¹²

tended to exclusively happen from server to server. This means the Catroweb client first asks the user to login and authorize the Pocket Code application. This request is sent to Google's OAuth authorization server. After successful login and authorization the Google server returns an access token, an id token and a one-time authorization code. The access token is a short-term token, that could be used with the less secure implicit grant flow with client-to-server communication. The id token is part of OpenID Connect that contains additional user information, as discussed in Section 8.1.1. Then, the client sends the one-time code to the Catroweb server. The one-time code must only be used once and has a validity of about ten minutes. If a one-time code is used more than once all tokens of the affected user of the application are invalidated. The Catroweb server, in turn, sends a request including the one-time code and client secret to Google's OAuth server and receives a new server-side access token and id token. These tokens have a longer lifetime as the client-side tokens, are more secure and offer more possibilities to the Catroweb server. For example, the Catroweb server can make Google API requests while the user is offline. At first sign-in of a user, a refresh token is issued, too. This refresh token can be used to renew server

8 Implementation

tokens without asking users to sign in again. Additional security is provided by a anti-request forgery state token that is created by the Catroweb server and is send along with the requests to Google's servers. The state token confirms that the request was done by the user and not by a malicious script to prevent cross-site request forgery (CSRF), for example¹³. The workflow is depicted in Figure 8.8 and is very similar to that of Facebook.

Best Practices

The most important guidelines as stated by Google are:

- Put the user first
- Transparency
- Respect user data

So, the most important guideline for Google is to build an application that users will love and like to share. Transparency means that the application should tell the user for what purposes their data will be used and that the application is explicitly asking for permissions. Finally, Google has a very detailed privacy policy according to user data including rules on using and sharing user data or deletion rules when a user wants to delete personal data or the association between the application and Google. Google has also published a long list of further guidelines. Some of those guidelines, that may be relevant to Pocket Code, are:

- Posts: Users should be informed when a post will be made and given an accurate preview before any posts are made, be able to append their own text, be able to choose with whom to share (e.g. certain circles) and it should be stated that the application is the source of the post.
- Invites: The user should be informed before invites to friends are made.
- Writing app activity to Google: The users should be aware of which activities are written to the Google moments API.
- Logout: A Google Logout button must be provided.

¹³Google, 2015b.

- Google buttons: There are strict Design und Usability rules¹⁴.

8.1.5 Design and UX

This section gives a short overview over design and usability issues of the new sign-in system.

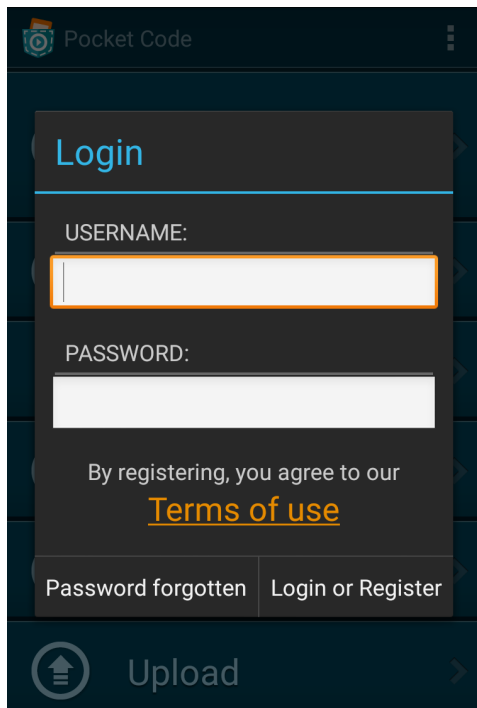


Figure 8.9: Old Pocket Code Login Dialog

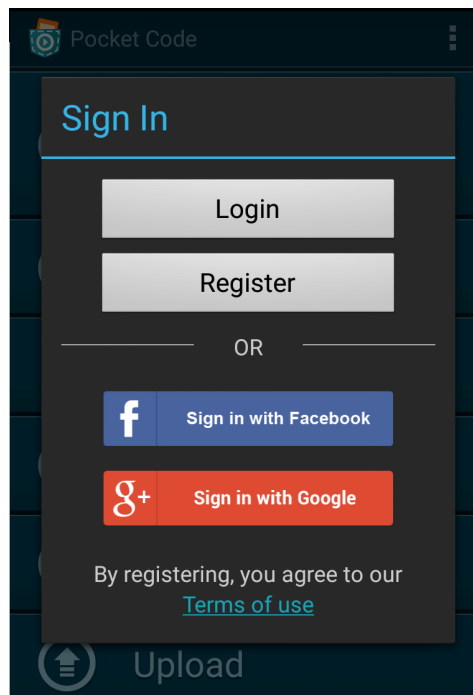


Figure 8.10: New Pocket Code Signin Dialog

Figure 8.9 shows the old Pocket Code login dialog before introducing Open Authentication. On the one hand, there was no Open Authentication and, on the other hand, no distinction between login and registration. When a user clicked "Login or Register" the user was registered when there was no existing account, or logged in otherwise.

¹⁴Google, 2015d.

8 Implementation

The new signin dialog, as seen in Figure 8.10, adds signin functionality for Facebook and Google+, as described in the previous sections. First, a user signs in within a distinct Facebook or Google+ screen. After successful authentication, the Pocket Code app communicates with the Catroweb server to finish the signin process. A complete reference to the implemented server-side API functions is given in the appendix.

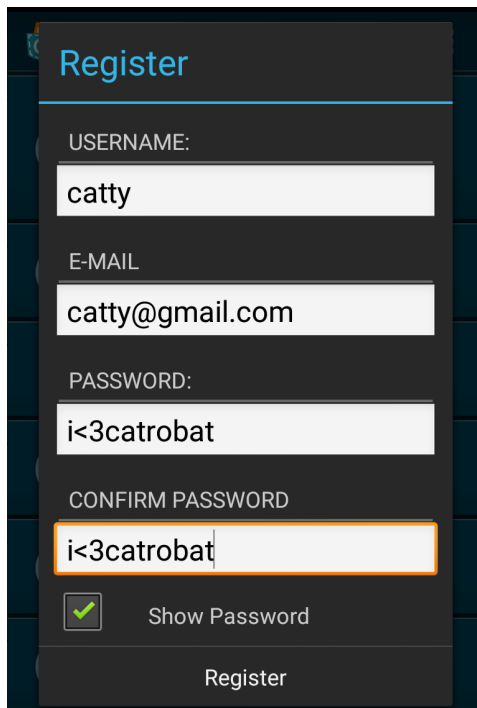


Figure 8.11: New Pocket Code Registration Dialog

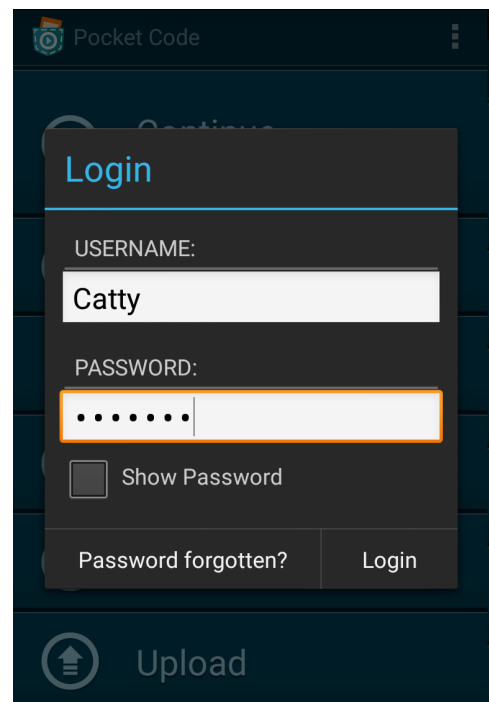


Figure 8.12: New Pocket Code Login Dialog

Furthermore, a clear distinction between the native login and register process has been introduced. Figures 8.11 and 8.12 show the new registration and login dialogs that have been introduced in Pocket Code. A reference to the server-side API functions is given in the appendix, too.

8.1 Open Authentication

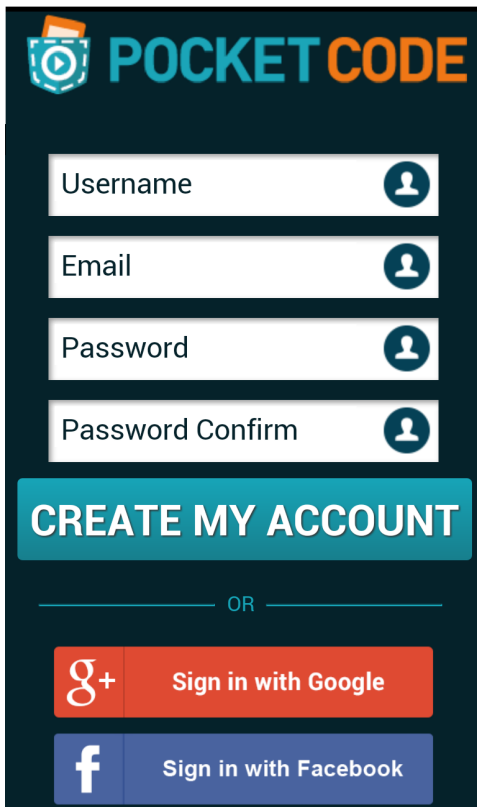


Figure 8.13: Catroweb Registration Web Page

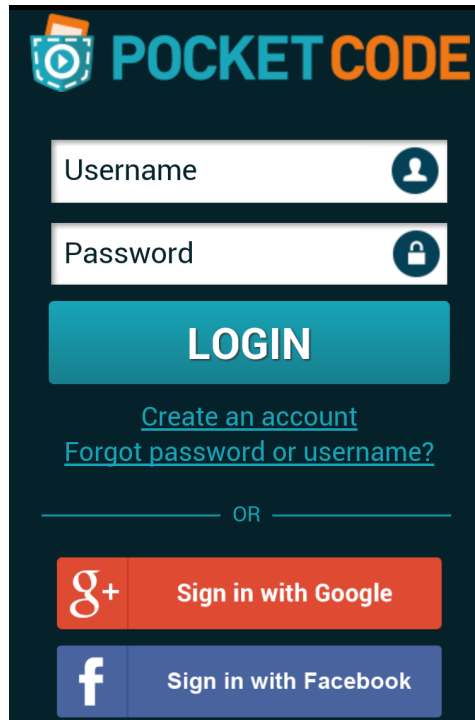


Figure 8.14: Catroweb Login Web Page

Finally, a distinction between login and register as well as signin functionality for Facebook and Google+ has been added on the Website, too. Actually, registration was not possible on the Website - the registration Web page is new to the users. Figure 8.13 and 8.14 show the mobile view of the new registration and login Webpages with the additional possibility to sign in with Facebook or Google+, respectively.

8.2 Facebook Channel Postings

Next, a possibility to directly share, like and comment Catrobat programs was needed. The decision was to create the Facebook page "Catrobat Programs" on which a post for every newly uploaded program is made. The post on Facebook contains the program name and a link to profile of the user, that has uploaded the program, as message, the program thumbnail picture as image, a link to program on the community Website and the program description. A sample post of an uploaded program is shown in Figure 8.15.

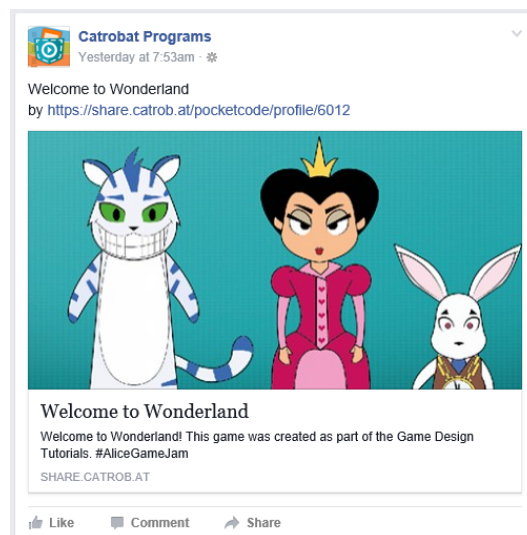


Figure 8.15: Facebook Post on Facebook Page.

Furthermore, an indication on the program page that a Facebook post for that program exists, was added. Therefore, a Facebook symbol with a link to the post on Facebook was introduced, as shown in Figure 8.16.

Besides making posts on Facebook, deletion of Facebook posts has also been implemented. The deletion of a Facebook post is triggered when the program is deleted either by the user or an admin, and when the program is reported as inappropriate. From a security point of view, a valid page

8.2 Facebook Channel Postings

About Pocket Code Tutorials Google Play

POCKET CODE Search... [Profile Icon]

WELCOME TO WONDERLAND rompiesjimmy@gmail .com

DESCRIPTION

Welcome to Wonderland!

This game was created as part of the Game Design Tutorials.

#AliceGameJam

DOWNLOAD **PREPARE APP**

Link

Report as inappropriate
Catrobat Language version: 0.98

2 days ago 3.15 MB 4 downloads 9 views

Tutorials Privacy Policy
About Pocket Code Terms of Use
Google Play Imprint
License to play

English (Australia) Search... © Catrobat

Figure 8.16: Facebook Post on the Program Webpage.

8 Implementation

access token for the Facebook page is necessary. The page access token is based on a valid user access token from an admin of the Facebook page and needs additionally the `manage_pages` permission. This token has an unlimited lifetime and was stored in the secret configuration parameters on the server.

The below code block is the implementation for making a post on Facebook, that is triggered after upload of a program. The link, message and image are directly given as arguments to the post. The description is retrieved by Facebook with the hint of a Open Graph tag in the program HTML page (not shown in the code).

```
public function postOnFacebook($program_id)
{
    $this->initializeFacebook();
    $account_access_token =
        $this->checkFacebookServerAccessTokenValidity();
    $this->setFacebookDefaultAccessToken($this->getAppToken());

    $is_valid = $this->debugToken($account_access_token);

    if ($is_valid) {
        $client = $this->facebook->getOAuth2Client();

        try {
            //refresh access token with long-term admin user-token
            $accessToken =
                $client->getLongLivedAccessToken($account_access_token);
        } catch (FacebookSDKException $e) {
            return $e->getMessage();
        }

        /**
         * @var $program_manager ProgramManager
         * @var $program Program
         * @var $response FacebookResponse
         */
    }
```

8.2 Facebook Channel Postings

```
$program_manager = $this->container->get('programmanager');
$program = $program_manager->find($program_id);

$url = $this->router->generate('program', array('id' => $program_id), true);
$user = $program->getUser();
$profile_url = $this->router->generate('profile',
                                     array('id' => $user->getId()),
                                     true);

$message = $program->getName() . chr(10) . 'by ' . $profile_url;

$program_img = $this->screenshot_repository->getScreenshotWebPath(
    $program->getId());
$program_img_url =
    $this->router->getContext()->getHost() . '/' . $program_img;

$data = [
    'link' => $url,
    'message' => $message,
    'picture' => $program_img_url,
];

$response = $this->facebook->post('/me/feed', $data, (string) $accessToken);
$respBody = json_decode($response->getBody());
$fb_post_id = $respBody->id;
$program->setFbPostId($fb_post_id);

$fb_post_url = $this->getFacebookPostUrl($fb_post_id);
$program->setFbPostUrl($fb_post_url);

$entity_manager = $this->container->get('doctrine.orm.entity_manager');
$entity_manager->persist($program);
$entity_manager->flush();

return $fb_post_id;
} else {
    throw new FacebookSDKException("Invalid facebook user or page token",
```

8 Implementation

```
        StatusCode::FB_POST_ERROR);  
    }  
}
```

However, after some time (about 3 months), we noticed that the Facebook posts are hardly accessed. There have been just a few likes, shares and comments. We think the reason is, that the automated posts are rather unpersonal and the intention of the Facebook symbol is possibly not clear to the users. As a result, we decided to keep Facebook post functionality, but to remove the Facebook symbol from the program page again. A more personal and clearer form to like and share programs should be implemented on the program page. This implementation is discussed in Section 8.3. Nevertheless, special thanks should be given to Patrick Trummer and Claudio Kirchmair for supporting me with the implementation of Facebook post functionality.

8.3 Likes, +1 and Sharing

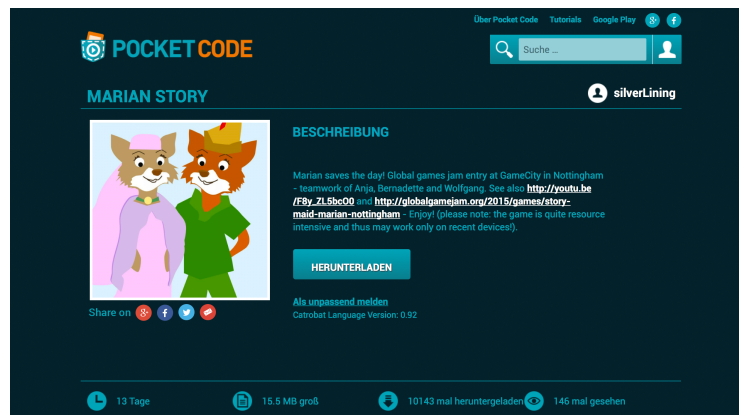


Figure 8.17: Like and Share Design for Desktop Version of Website.

The implementation to share programs and like/+1 the community Website is still work in progress at the end of this thesis. First, it should be possible to like the community Website on Facebook and "+1" it on Google+. These

8.3 Likes, +1 and Sharing

buttons are held in light blue and are located on the top in the Desktop view and at the bottom in the mobile view.

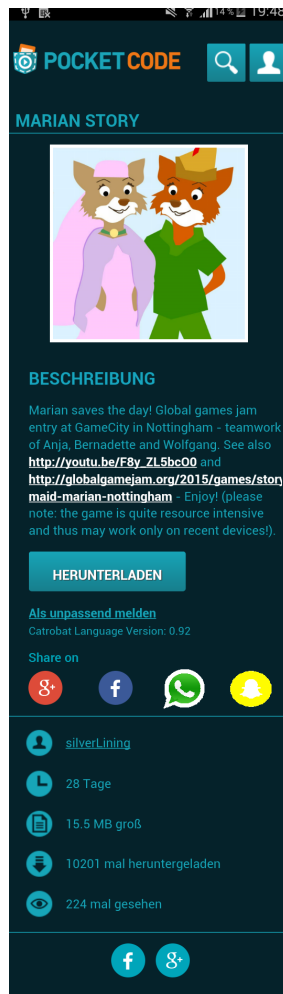


Figure 8.18: Like and Share Design for Mobile Version of Website.

Second, the buttons to share programs should be located below the thumbnail in the Desktop version and just above the footer in the mobile version. In the Desktop version sharing of Catrobat programs should be possible with Google+, Facebook, Twitter and via email. In the mobile version share buttons for WhatsApp and Snapchat should be provided instead of the

8 Implementation

share buttons for email and Twitter. This enables sharing via the popular messengers on mobile devices. Especially Snapchat is the platform that best fits to Catrobat's target group in terms of age. However, an issue that needs to be investigated in this context, is that Snapchat does not provide a public API yet. Figure 8.17 shows the design for the Desktop version and Figure 8.18 for the mobile version. The shares should include a link to the Catrobat program, the program name as title, a screenshot of the program and the program description or a user-defined text. Finally, thanks should be given to Georg Schober for pair-programming with me on this issue.

8.4 Program Download Statistics

Finally, program download statistics have been implemented. The goal was to create a system that specifically tracks certain data about downloads from the community Website. This data includes geolocation data (country, locality etc.), ip address, referrer, a download timestamp and links to the associated program and user of the download. A functionality was added, that triggers the creation of program download statistics each time a program download is requested. The ip address is used to retrieve geocode data. Latitude and longitude of the geocode data are used again for reverse geocoding via Google maps request to retrieve rough address data (of course this might be inaccurate or the server, from which an ip address is taken, can be located elsewhere). Importantly, user agent and referrer are added from browser cookies. The user agent gives information about the used browser of the user and whether the download was started from a mobile device (including device name and version). The referrer indicates from where the user navigated to the download page. This is especially interesting for the Facebook posts. If a user was referred from a Facebook post to the download page, the referrer contains a Facebook URL. The main functions to create the program download statistics is shown below.

```
public function createProgramDownloadStatistics($program_id, $ip,
                                             $user_agent, $user, $referrer)
{
    $results = $this->geocoder
```

8.4 Program Download Statistics

```
->using('geo_plugin')
->geocode($ip);

$result = $results->first();

$latitude = $result->getLatitude();
$longitude = $result->getLongitude();
$country_code = $result->getCountry()->getCode();
$country_name = $result->getCountry()->getName();

$program = $this->programmanager->find($program_id);

$program_download_statistic = new ProgramDownloads();
$program_download_statistic->setProgram($program);
$program_download_statistic->setUserAgent($user_agent);
$program_download_statistic->setUser($user);
$program_download_statistic->setReferrer($referrer);
$program_download_statistic->setDownloadedAt(new \DateTime());
$program_download_statistic->setIp($ip);
$program_download_statistic->setLatitude($latitude);
$program_download_statistic->setLongitude($longitude);
$program_download_statistic->setCountryCode($country_code);
$program_download_statistic->setCountryName($country_name);

$this->entity_manager->persist($program_download_statistic);
$program->addProgramDownloads($program_download_statistic);
$this->entity_manager->persist($program);
$this->entity_manager->flush();

$this->addGoogleMapsGeocodeData($latitude, $longitude,
    $program_download_statistic);
return true;
}

private function addGoogleMapsGeocodeData($latitude, $longitude,
    $program_download_statistic) {
    $results_google = $this->geocoder
```

8 Implementation

```
->using('google_maps')
->reverse($latitude, $longitude);

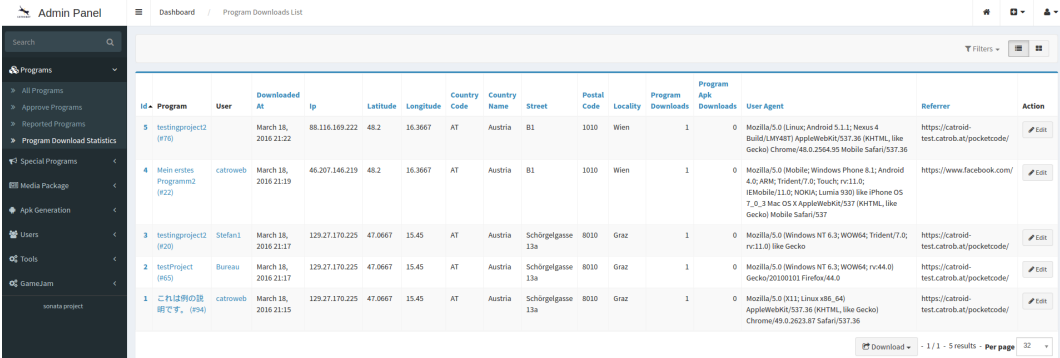
$result = $results_google->first();

$street = $result->getStreetName() . ' ' . $result->getStreetNumber();
$postal_code = $result->getPostalCode();
$locality = $result->getLocality();

$program_download_statistic->setStreet($street);
$program_download_statistic->setPostalCode($postal_code);
$program_download_statistic->setLocality($locality);

$this->entity_manager->persist($program_download_statistic);
$this->entity_manager->flush();
}
```

Finally, Figure 8.19 shows how the retrieved download statistics are displayed in the admin section of the Website.



ID	Program	User	Downloaded At	Ip	Latitude	Longitude	Country Code	Country Name	Street	Postal Code	Locality	Program Downloads	Apk Downloads	User Agent	Referrer	Action
5	testingproject2 (F76)	catroweb	March 18, 2016 21:22	88.116.169.222	48.2	16.3667	AT	Austria	B1 1010 Wien			1	0	Mozilla/5.0 (Linux; Android 5.1.1; Nexus 4 Build/LMY48T) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/46.0.2584.55 Mobile Safari/537.36	https://catroid-test.catrob.at/pocketcode/	Edit
4	Main erstes Programm3 (F22)	catroweb	March 18, 2016 21:19	46.207.146.219	48.2	16.3667	AT	Austria	B1 1010 Wien			1	0	Mozilla/5.0 (Mobile; Windows Phone 8.1; Android 4.0.4; ARM; Trident/7.0; Touch; rv:11.0; [EMobile/11.0; NOKIA; Lumia 930] like iPhone OS 7_0_3 Mac OS X AppleWebKit/537 (KHTML, like Gecko) Mobile Safari/537	https://www.facebook.com/	Edit
3	testingproject2 (F20)	Stefani	March 18, 2016 21:17	129.27.170.225	47.0667	15.45	AT	Austria	Schörgelgasse 13a	8010 Graz		1	0	Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko	https://catroid-test.catrob.at/pocketcode/	Edit
2	testProject (F63)	Bureau	March 18, 2016 21:17	129.27.170.225	47.0667	15.45	AT	Austria	Schörgelgasse 13a	8010 Graz		1	0	Mozilla/5.0 (Windows NT 6.3; WOW64; rv:44.0) Gecko/20100101 Firefox/44.0	https://catroid-test.catrob.at/pocketcode/	Edit
1	これは別の投稿です。 (F94)	catroweb	March 18, 2016 21:15	129.27.170.225	47.0667	15.45	AT	Austria	Schörgelgasse 13a	8010 Graz		1	0	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.87 Safari/537.36	https://catroid-test.catrob.at/pocketcode/	Edit

Figure 8.19: Program Download Statistics in the Website Admin section.

8.5 Further Implementations

There are still lots of other social media features that can be added. Besides Open Authentication with Facebook and Google+ for iOS and sharing/liking on the program Website, that are yet work in progress, I propose to add the following features:

- Geo-customized Open Authentication
- Geo-customized Sharing of Programs
- Posts: Upload and Share of Programs
- Facebook Notifications
- Facebook Invitations
- Social Challenging System
- Adding of Looks and Sounds from Instagram, flickr, Facebook etc.
- Youtube Recording and Uploading

Geo-customized Open Authentication means to add different social sign-in platforms for different countries within Pocket Code and on the community Website. For example, sign-in with Vkontakte and Odnoklassniki in Russia, QQ in China or Twitter in the USA. The same is valid for providing sharing buttons for different platforms for different countries on the Website. This can be done using gradle product flavors in Android and cookies on the Website. It has been described in Section 7.5 which platforms make sense for which regions.

Upload and share of programs means to add a possibility within Pocket Code for users to upload and share a program. For example, a dialog that asks a user whether he or she wants to share the program on Facebook or Google+ after successful upload can be added. A general option that determines whether a user generally wants to share programs can additionally be added in the settings. With Facebook it is possible to share content on the Facebook Timeline or via Facebook messenger, and for Google+ on a user's Google+ page. The post should contain the URL to the Catrobat programme as link, the program name as title, the program screenshot and a text. The text can be the program description by default, but should be editable by users. Facebook App Links and Google+ Deep Linking are also of special interest in this context. If they are embedded into posts, a connection to

8 Implementation

the Pocket Code app is added. If a Facebook or Google+ user clicks on the post, the user is then redirected to the Pocket Code app. If the app is not installed, the user is redirected to the app download page of Pocket Code on Google Play.

Similarly to the above suggestion, a Facebook notification system can be implemented within Pocket Code. The idea is to connect users with their Facebook friends that also use Pocket Code - requiring the Facebook friends permission - and to automatically broadcast notifications to a user's friends when e.g. a user uploaded a new program. Therefore users stay up to date with their friends' activities within Pocket Code. Of course it should be possible to deactivate the sending and receiving of broadcasts in the settings. Another suggestion in this context is to offer a possibility to invite Facebook friends, that are not yet Pocket Code users, to install and try out Pocket Code. This option can be made available in the ActionBar settings for authenticated Facebook users, for example.

The idea of the social challenging system is to offer a possibility to challenge friends within programs directly created in Pocket Code. A challenging option could be integrated directly in the stage, for example after a game has ended in the stage menu. Alternatively, this could also be done by creating a "Share Brick". This brick could store a simple text, which can be set to custom text or the value of a uservariable (highscore). This brick can, e.g., be added when a program ends and asks the user whether he wants to share the program or highscore with a friend. The Facebook SDK already provides great possibilities for this purpose. First, it would be necessary to extend the Facebook implementation to retrieve a user's friends via Open Graph requests. Next, Facebook offers options to invite other users via private notifications to e.g. play a game (Game Requests) and has a Scores & Achievements API. This API can e.g. be used for highscore-based challenging, when highscores are stored in user variables. However, the Pocket Code app needs to be officially listed as game in order to use these APIs. This category can be set in the Facebook Developer Console. The current category is "Education". Another issue to be thought of is, that the program must be uploaded on the server. A further extension would be to introduce highscore counting on the server to add a basic form of gamification.

8.5 Further Implementations

Next, adding of looks and sounds can be built into the add dialogs of the LookFragment and SoundFragment within Pocket Code, respectively. A chooser dialog could offer appropriate platforms, such as flickr, Facebook or Instagram. If a user selects a platform, the appropriate API functions of the selected platform should be called to search for and retrieve multimedia content. For example, the flickr API can be called via REST service or via the flickr Java library wrapper.

Finally, recording of programs on the stage and uploading of these videos on YouTube can be added. The YouTube APIs for Android and iOS already offer simple functionality to record, upload and share YouTube videos. Furthermore, the YouTube API supports Open Authentication and playback functionality (GoogleDevelopers, 2016).

Conclusion

Finally, the conclusion recaps the most important findings and underlines suggestions that have been given to Catrobat in the practical part of this thesis. The practical part directly built upon the theoretical part, in which knowledge about the technological background and applications of the Social Web and Semantic Web as well as important Social Media Marketing knowledge has been conveyed.

In Chapter 6, the Catrobat project has been presented and its target group has been identified. The intended main target group of Catrobat are boys and girls aged between 13 and 17. Furthermore, Google Analytics data of the community Website as well as Andlytics data and Facebook Analytics data of the Pocket Code Android app have been analyzed to get an impression of current user demographics. Concerning age, it could be confirmed that the Website has rather younger visitors. However, the average visitor age group is older than those of the intended target group. Besides the age group, there are more male than female users and most users are visiting the Website from a mobile device. Regarding geographical data, most visitors are from diverse European countries, the USA, Russia, India, Japan and Brazil.

Based on this target profile, general social network and messenger data, mobile usage data, age- and gender-related data, and geographical data have been collected and evaluated. On the basis of these evaluations, recommendations have been given in Section 7.5 on "how and where to engage best in different social media platforms" from a marketing and technical perspective. The main findings have been that there are significant differences concerning popular social platforms in different regions of the world, that mobile usage scenarios get increasingly important and that there is a strong shift towards more private platforms such as messengers.

8 Implementation

Due to this projected more private nature of the social media landscape and that getting visibility on the big social platforms, such as Facebook or Twitter, gets more difficult (mainly due to control of platforms which content is displayed to users and monetarization), it has been recommended to get active on social messengers, too. From a technical perspective, the most important recommendation was to make one standard implementation for social elements with the globally most popular platforms, and to diversify the used platforms in the implementations for some selected regions. The standard implementations that have been made, such as social sign-in with Facebook and Google+, liking, sharing or automated posts to a Facebook page, have been discussed in Chapter 8. Suggestions and instructions for further implementations, including geo-customized social sign-in and sharing of programs, have been given in Section 8.5.

Table 7.1 has shown how social networks and messengers have been grouped into five different regional blocks. The standard group includes the platforms Facebook, Google+, Instagram, WhatsApp and Snapchat and is especially targeted at Europe, South America and India. Facebook is the largest social network and WhatsApp the largest messenger throughout these regions. Google+ ranks just behind Facebook and WhatsApp in most of these countries concerning user penetration. Due to this fact and the largest number of young users, these networks have been selected. Instagram is yet not so popular in Europe and India, but very popular in South America. Due to the already existing popularity in South America, high growth rates and good marketing possibilities it has been decided to add Instagram to the standard group. Finally, Snapchat was added to this group because it is among the platforms with the highest growth rates in these regions and matches Catrobat's target group best with the youngest users. Actually, WhatsApp and Snapchat have been recommended as platforms to get active from a marketing perspective because of the popularity, high growth rates, young users and the observed shift towards messengers.

For the USA, China, Russia and Japan other social platforms have been recommended. In the USA, Google+ and WhatsApp have relatively few users. Therefore, it is more suggestive to focus on Twitter, which is more popular in the USA, instead. The social media landscape in China and Russia is completely different as the most popular social platforms are native to these countries. In Russia, VKontakte, Odnoklassniki and Telegram, and

8.5 Further Implementations

in China, QQ, QZone and WeChat are the most popular platforms. In Japan, in turn, Google+ and Facebook are again the most popular social networks. Therefore, some standard implementations can be used in Japan, too. However, the most popular messenger in Japan is LINE by far.

Finally, it needs to be said that the social media landscape continuously changes. Therefore, it is important to stay up-to-date to be able to react to these changes from a marketing as well as technical perspective.

Appendix

Web Signin API Reference

Subsequently, an overview over the added server-side API functions for login, registration and Open Authentication with Facebook and Google+ is given. The overview includes function names, function parameters, the HTTP method and address URL as well as a short function description.

Table .2: Server-Side API Functions

Method and Parameters	HTTP Request	Description
registerNativeUser() registrationUsername registrationPassword registrationCountry registrationEmail	POST /api/register/ Register.json	Registers a new native user.
loginNativeUser() registrationUserName registrationPassword	POST /api/register/ Login.json	Logs in an existing native user.
isOAuthUser() username_email	POST /api/ IsOAuthUser/ IsOAuth User.json	Checks whether the specified user is a Facebook or Google+ user.
checkEMailAvailable() email	POST /api/ EMailAvailable/ EMail Available.json	Checks whether a user with the specified email address is available.
checkUserNameAvailable() username	POST /api/Username Available/ Username Available.json	Checks whether a user with the specified username is available.

Table .3: Server-Side API Functions

Method and Parameters	HTTP Request	Description
checkFacebookServerTokenAvailable() id	POST /api/ FacebookServer TokenAvailable/ Facebook ServerToken Available.json	Checks whether a user with the specified Facebook id with an access token is available.
exchangeFacebookTokenAction() client_token id username email locale state mobile	POST /api/exchange FacebookToken/ exchangeFacebook Token.json	Exchanges a given client token, retrieved by a user from the signin dialog, for a long-term server token.
loginWithFacebookAction() username id email locale	POST /api/ loginWith Facebook/ loginWith Facebook.json	Logs in a registered Facebook user and renews upload token.
getFacebookUserProfileInfo() id token	POST /api/ getFacebook UserInfo/ getFacebook UserInfo.json	Requests and returns information such as name, profile link, email or locale about a specified Facebook user.

Table .4: Server-Side API Functions

Method and Parameters	HTTP Request	Description
isFacebookServer AccessTokenValid() id	POST /api/ checkFacebook ServerToken Validity/ checkFacebook ServerToken Validity.json	Checks whether the server access token of a specified Facebook user is valid.
checkGoogleServer TokenAvailable() id	POST /api/ GoogleServer TokenAvailable/ Google ServerToken Available.json	Checks whether a user with the specified Google+ id with an access token is available.
exchangeGoogle CodeAction() code id username email locale state mobile	POST /api/exchange GoogleCode/ exchange GoogleCode.json	Exchanges a given authorization code, retrieved by a user from the signin dialog, for a long-term server token.
loginWith GoogleAction() username id email locale	POST /api/ loginWith Google/ loginWith Google.json	Logs in a registered Google user and renews upload token.

Table .5: Server-Side API Functions

Method and Parameters	HTTP Request	Description
getGoogleUserProfileInfo() id	POST /api/ getGoogle UserInfo/ getGoogle UserInfo.json	Requests and returns information such as Display name, Image URL or profile URL about a specified Google+ user.
loginWithTokenAndRedirectAction() fb_id gplus_id	POST /api/ loginWithToken AndRedirect/ loginWithToken AndRedirect	Logs in a registered Facebook or Google+ user on the login and registration Web pages, authenticates the user in the session and makes a redirect.
getFacebookAppId()	GET /api/getFacebook AppId/ getFacebook AppId.json	Returns the Facebook Web application id.
getGoogleAppId()	GET /api/getGoogle AppId/ getGoogle AppId.json	Returns the Google+ Web application id.
generateCsrfToken()	GET /api/ generate CsrfToken/ generate CsrfToken.json	Returns a CSRF token, that is used as session state on the Web site.
deleteOAuthTestUserAccounts()	GET /api/ deleteOAuth UserAccounts/ deleteOAuth UserAccounts.json	Deletes all test user accounts on the Web server.

Bibliography

- Allsopp, John (2007). *Microformats: Empowering Your Markup for Web 2.0*. Springer-Verlag New York Inc. (cit. on p. 28).
- Alonso, Gustavo et al. (2004). *Web Services: Concepts, Architectures and Applications*. Springer Verlag GmbH (cit. on pp. 10, 12–14).
- AmericanMarketingAssociation (2016). *Dictionary*. URL: <https://www.ama.org/resources/Pages/Dictionary.aspx> (visited on 03/21/2016) (cit. on p. 3).
- Anderson, P. (2012). *Web 2.0 and Beyond: Principles and Technologies*. Chapman & Hall/CRC Textbooks in Computing. CRC Press. ISBN: 9781439828687. URL: <https://books.google.at/books?id=rRrOBQAAQBAJ> (cit. on p. 9).
- AndroidDevelopers (2016). *Configuring Gradle Builds*. URL: <http://developer.android.com/tools/building/configuring-gradle.html> (visited on 03/04/2016) (cit. on p. 111).
- Bell, G. (2009). *Building Social Web Applications*. O'Reilly Media. ISBN: 9781449379414. URL: <https://books.google.at/books?id=YUk7swNT3rQC> (cit. on p. 36).
- Berners-Lee, Tim (2005). *Uniform Resource Identifier (URI): Generic Syntax*. URL: <http://tools.ietf.org/html/rfc3986> (visited on 03/24/2016) (cit. on p. 4).
- Bitkom (2014). "Jung und vernetzt: Kinder und Jugendliche in der digitalen Gesellschaft." In: URL: <https://www.bitkom.org/Publikationen/2014/Studien/Jung-und-vernetzt-Kinder-und-Jugendliche-in-der-digitalen-Gesellschaft/BITKOM-Studie-Jung-und-vernetzt-2014.pdf> (visited on 02/10/2016) (cit. on p. 95).
- Blanc, Jonathan Le (2012). *Programming Social Applications: Building Viral Experiences with OpenSocial, OAuth, OpenID, and Distributed Web Frameworks*. O'Reilly & Associates Inc. (cit. on p. 121).
- Blumauer, Andreas and Tassilo Pellegrini (2008). *Social Semantic Web: Die Konvergenz von Social Software, Web 2.0 und Semantic Web*. Springer Science & Business Media (cit. on p. 48).

Bibliography

- Boyd, Ryan (2012). *Getting Started with OAuth 2.0: Programming Clients for Secure Web API Authorization and Authentication*. O'Reilly & Associates Inc. (cit. on pp. 118, 120–122).
- Brabham, D.C. (2013). *Crowdsourcing*. Crowdsourcing. MIT Press. ISBN: 9780262518475. URL: <https://books.google.at/books?id=JgJWk6xY9RcC> (cit. on p. 63).
- Breslin, John, Alexandre Passant, and Stefan Decker (2009). *The Social Semantic Web*. Springer Science & Business Media (cit. on pp. 46–49).
- Buyya, Rajkumar, James Broberg, and Andrzej Goscinski (2010). *Cloud Computing: Principles and Paradigms*. John Wiley & Sons Inc. (cit. on p. 31).
- Carl, Denny et al. (2008). *Mashups programmieren*. O'Reilly & Associates Inc. (cit. on p. 23).
- Catrobat (2015a). *Catrobat Marketing website*. URL: <http://www.catrobat.org/> (visited on 12/21/2015) (cit. on p. 65).
- Catrobat (2015b). *Developers' website of the Catrobat project*. URL: <http://developer.catrobat.org/> (visited on 12/21/2015) (cit. on p. 65).
- Catrobat (2016a). *Google Analytics for Catrobat community Website*. Google Analytics (cit. on pp. 70–72).
- Catrobat (2016b). *Google Play Statistics for Pocket Code*. Online (cit. on p. 73).
- DBpedia (2015). *DBpedia*. URL: <http://wiki.dbpedia.org> (visited on 12/19/2015) (cit. on p. 47).
- Dougherty, Dale and Tim O'Reilly (2015). *Traditional Marketing*. URL: <http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html> (visited on 01/23/2015) (cit. on pp. 7–10).
- Earth Group, Friends of the (2015). *Climate Justice Peace*. URL: <http://climatejusticepeace.org/> (visited on 12/12/2015) (cit. on pp. 23, 24).
- EntrepreneurSmallBusinessEncyclopedia (2016). *Affiliate Marketing*. URL: <http://www.entrepreneur.com/encyclopedia/affiliate-marketing> (visited on 01/03/2016) (cit. on p. 62).
- Euzenat, Jerome and Pavel Shvaiko (2013). *Ontology Matching*. Springer (cit. on p. 43).
- Facebook (2015a). *Facebook Developers: Access Tokens*. URL: developers.facebook.com/docs/facebook-login/access-tokens (visited on 02/10/2015) (cit. on pp. 124–126).

Bibliography

- Facebook (2015b). *Facebook Developers: Batch Requests*. URL: <https://developers.facebook.com/docs/graph-api/making-multiple-requests> (visited on 08/26/2015) (cit. on p. 20).
- Facebook (2015c). *Facebook Developers: Graph API for Test Users*. URL: <https://developers.facebook.com/docs/graph-api/reference/v2.4/app/accounts/test-users> (visited on 08/26/2015) (cit. on p. 19).
- Facebook (2015d). *Facebook Developers: Login Checklist*. URL: <http://www.developers.facebook.com/docs/facebook-login/checklist> (visited on 02/10/2015) (cit. on p. 128).
- Facebook (2015e). *Facebook Developers: Login for Websites*. URL: developers.facebook.com/docs/facebook-login (visited on 02/10/2015) (cit. on p. 122).
- Facebook (2015f). *Facebook Developers: Login Review*. URL: <http://www.developers.facebook.com/docs/apps/review/login> (visited on 02/10/2015) (cit. on p. 128).
- Facebook (2015g). *Facebook Developers: Login Security*. URL: developers.facebook.com/docs/facebook-login/security (visited on 02/10/2015) (cit. on pp. 125, 126).
- Facebook (2015h). *Facebook Developers: Permissions Reference - Facebook Login*. URL: developers.facebook.com/docs/facebook-login/permissions (visited on 02/10/2015) (cit. on p. 127).
- Facebook (2015i). *Facebook Developers: Securing Graph API Requests*. URL: developers.facebook.com/docs/graph-api/securing-requests (visited on 02/10/2015) (cit. on p. 125).
- Facebook (2016a). *Facebook Analytics*. URL: <https://www.facebook.com/analytics/> (visited on 04/29/2016) (cit. on pp. 73–75).
- Facebook (2016b). *Feed Dialog*. URL: <https://developers.facebook.com/docs/sharing/reference/feed-dialog/v2.5> (visited on 03/25/2016) (cit. on p. 127).
- Farkas, Bart G. (2006). *Secrets of Podcasting, Second Edition: Audio Blogging for the Masses*. Peachpit Press. ISBN: 0132712075, 9780132712071 (cit. on pp. 25, 36).
- FOAF-Project (2015). *FOAF*. URL: <http://www.foaf-project.org> (visited on 12/19/2015) (cit. on p. 46).
- Furht, B. (2010). *Handbook of Social Network Technologies and Applications*. Springer US. ISBN: 9781441971425. URL: <https://books.google.at/books?id=7kC0wGMC46cC> (cit. on pp. 39, 41).

Bibliography

- Furht, Borko and Armando Escalante (2010). *Handbook of Cloud Computing*. Springer Science & Business Media (cit. on pp. 32, 33).
- Google (2015a). *Google+ API*. URL: <https://developers.google.com/%5C+//> (visited on 02/11/2015) (cit. on p. 132).
- Google (2015b). *Google Developers: Google Sign-In for server-side apps*. URL: <https://developers.google.com/%5C+//web//signin//server-side-flow> (visited on 02/15/2015) (cit. on p. 134).
- Google (2015c). *Google Maps API*. URL: <https://developers.google.com/maps/?hl=de> (visited on 01/31/2015) (cit. on p. 23).
- Google (2015d). *Google+ Platform Developer Policies*. URL: <https://developers.google.com/%5C+/policies> (visited on 02/15/2015) (cit. on p. 135).
- Google (2015e). *Google: Structured Data - Rich Snippets*. URL: <https://developers.google.com/structured-data/rich-snippets/> (visited on 02/15/2015) (cit. on p. 30).
- Google (2015f). *OpenIDConnect*. URL: <https://developers.google.com/accounts/docs/OpenIDConnect> (visited on 02/10/2015) (cit. on p. 117).
- Google (2015g). *Structured Data*. URL: <https://developers.google.com/structured-data/> (visited on 12/19/2015) (cit. on p. 51).
- Google (2015h). *YouTube API*. URL: <https://developers.google.com/youtube/?hl=de> (visited on 01/31/2015) (cit. on p. 23).
- GoogleDevelopers (2016). *Resources for Mobile Development with YouTube*. URL: https://developers.google.com/youtube/articles/youtube_mobileresources (visited on 03/19/2016) (cit. on p. 149).
- Guha, R. and D. Al-Dabass (2010). "Impact of Web 2.0 and Cloud Computing Platform on Software Engineering." In: *Electronic System Design (ISED), 2010 International Symposium on*, pp. 213–218. DOI: 10.1109/ISED.2010.48 (cit. on p. 30).
- Habbo (2016). *Habbo Homepage*. URL: <https://www.habbo.com/> (visited on 04/16/2016) (cit. on p. 113).
- Hammersley, Ben (2001). *Creating Web Application Gateways: Programming Web Services with XML-RPC*. O'Reilly & Associates Inc. (cit. on p. 11).
- Hammersley, Ben (2005). *Developing Feeds with RSS and Atom*. O'Reilly Media Inc. (cit. on pp. 8, 25).
- Hauser, Tobias (2007). *AJAX: Web 2.0-Anwendungen mit Javascript und XML*. Pearson Education Deutschland GmbH (cit. on p. 21).

- Hennig, M. et al. (2012). *Studying Social Networks: A Guide to Empirical Research*. Campus Verlag. ISBN: 9783593397634. URL: <https://books.google.at/books?id=Yp128W2LstQC> (cit. on p. 40).
- HubSpot (2015). *The Inbound Methodology*. URL: <http://www.hubspot.com/inbound-marketing> (visited on 12/27/2015) (cit. on pp. 58, 59).
- International Affairs, Ministry of and Communications (2014). *Research on information and communication media usage*. URL: <http://www.socialmediaexaminer.com/2016-social-media-marketing-predictions-from-the-experts/> (visited on 03/01/2016) (cit. on p. 106).
- Jakus, Grega, Veljko Milutinovi, and Sanida Omerovi (2013). *Link Concepts, Ontologies, and Knowledge Representation*. Springer (cit. on p. 45).
- Jenkins, Lisa D. (2016). *2016 Social Media Marketing Predictions From the Experts*. URL: <http://www.socialmediaexaminer.com/2016-social-media-marketing-predictions-from-the-experts/> (visited on 02/29/2016) (cit. on p. 110).
- Kadushin, C. (2011). *Understanding Social Networks: Theories, Concepts, and Findings*. Oxford University Press. ISBN: 9780199920815. URL: <https://books.google.at/books?id=yVcTmVeE3j0C> (cit. on p. 39).
- Keller, Kevin Lane (1993). "Conceptualizing, Measuring, and Managing Customer-based Brand Equity." In: *Journal of Marketing* 57, pp. 1–22 (cit. on p. 54).
- Keller, Kevin Lane (2007). *Strategic Brand Management: Building, Measuring and Managing Brand Equity*. Third. Pearson. ISBN: 9780132336222 (cit. on pp. 54, 55, 57).
- KleineZeitung (2016). *Die Kleine Zeitung auch auf WhatsApp lesen*. URL: <http://www.kleinezeitung.at/s/lebensart/multimedia/4750497> (visited on 03/04/2016) (cit. on p. 112).
- Kotler, Philip et al. (2009). *Marketing Management*. 13th ed. Pearson Prentice Hall (cit. on pp. 3, 62).
- Lambe, P. (2014). *Organising Knowledge: Taxonomies, Knowledge and Organisational Effectiveness*. Chandos Knowledge Management. Elsevier Science. ISBN: 9781780632001. URL: <https://books.google.at/books?id=z1mpAgAAQBAJ> (cit. on p. 45).
- LMZMediaculture (2015). *Mikroblogging: Twitter & Co*. URL: <https://www.lmz-bw.de/mikroblogging.html> (visited on 07/19/2015) (cit. on p. 34).
- Maedche, A. (2012). *Ontology Learning for the Semantic Web*. The Springer International Series in Engineering and Computer Science. Springer

Bibliography

- US. ISBN: 9781461509257. URL: <https://books.google.at/books?id=Vz7VBwAAQBAJ> (cit. on p. 45).
- ManagementStudyGuide (2016a). *Dictionary*. URL: http://www.marketingterms.com/dictionary/conversion_rate (visited on 03/21/2016) (cit. on p. 3).
- ManagementStudyGuide (2016b). *What is Brand Awareness?* URL: <http://www.managementstudyguide.com/brand-awareness.htm> (visited on 03/21/2016) (cit. on p. 3).
- MarketingSchools (2015). *Traditional Marketing*. URL: <http://www.marketing-schools.org/types-of-marketing/traditional-marketing.html> (visited on 01/22/2015) (cit. on p. 7).
- Microformats (2015). *Microformats2 h-card example*. URL: <http://microformats.org/wiki/microformats2> (visited on 07/26/2015) (cit. on p. 28).
- Mika, Peter (2007). *Social Networks and the Semantic Web*. Springer Science & Business Media (cit. on p. 46).
- ondeviceResearch (2015). *Messenger Wars 2: How Facebook climbed back to number 1*. URL: <http://de.slideshare.net/OnDevice/messenger-wars-2> (visited on 02/10/2016) (cit. on p. 101).
- OnPageWiki (2016). *Dictionary*. URL: <https://de.onpage.org/wiki> (visited on 03/21/2016) (cit. on pp. 3, 4).
- Pateman, Nick and Dan Holt (2011). *Inbound Marketing*. Bookboon. ISBN: 9788776818791. URL: https://books.google.at/books?id=z0%5C_mPOYQKmAC (cit. on pp. 54, 57).
- Peters, I. and P. Becker (2009). *Folksonomies: Indexing and Retrieval in Web 2.0*. Knowledge & information : studies in information science. De Gruyter/Saur. ISBN: 9783598251795. URL: https://books.google.at/books?id=Aeib%5C_wy18gkC (cit. on p. 26).
- Petri, Anja (2014). *Mitmach-Design - Neugestaltung der Lernapplikation Pocket Code unter Anwendung partizipativer Designmethoden im nutzerzentrierten Gestaltungsprozess*. Master Thesis (cit. on p. 69).
- Pinterest (2016). *About Pinterest*. URL: <https://about.pinterest.com/en> (visited on 01/31/2016) (cit. on p. 81).
- Rehman, Rafeeq Ur (2007). *The OpenID Book: A Comprehensive Guide to OpenID Protocol and Running OpenID Enabled Web Sites*. Conformix Books (cit. on p. 117).
- Richardson, Leonard and Sam Ruby (2007). *Web Services for the Real World: RESTful Web Services*. O'Reilly & Associates Inc. (cit. on pp. 15, 17, 21).

- Richardson, Will (2010). *Blogs, Wikis, Podcasts, and Other Powerful Web Tools for Classrooms*. SAGE Publications. ISBN: 1452271380, 9781452271385 (cit. on p. 35).
- Roebers, F. and M. Leisenberg (2010). *WEB 2.0 im Unternehmen: Theorie & Praxis - Ein Kursbuch für Führungskräfte*. tredition. ISBN: 9783868506341. URL: <https://books.google.at/books?id=Fnu49-d7dFYC> (cit. on p. 37).
- Ryan, P.K. (2011). *Social Networking*. Digital and Information Literacy Series. Rosen Publishing Group, Incorporated. ISBN: 9781448823468. URL: <https://books.google.at/books?id=rshCY6J7E0QC> (cit. on p. 36).
- Sauers, Michael P. (2006). *Bloggng and RSS: A Librarian's Guide*. Information Today, Inc. ISBN: 1573872687, 9781573872683 (cit. on p. 34).
- Scott, J. (2012). *Social Network Analysis*. SAGE Publications. ISBN: 9781446259450. URL: <https://books.google.at/books?id=MJoIGBfYDGEC> (cit. on p. 37).
- SIOC-Project (2015). *SIOC*. URL: <http://sioc-project.org> (visited on 12/19/2015) (cit. on p. 47).
- Siriwardena, Prabath (2014). *Advanced API Security: Securing APIs with OAuth 2.0, OpenID Connect, JWS, and JWE*. Apress (cit. on pp. 118, 122).
- Smith, Chris Silver (2015). *What's Best: Microformats, RDFa, or Micro Data?* URL: <http://www.semclubhouse.com/microformats-rdfa-or-micro-data/> (visited on 07/26/2015) (cit. on p. 29).
- Smith, M. and G. Kawasaki (2011). *The New Relationship Marketing: How to Build a Large, Loyal, Profitable Network Using the Social Web*. Wiley. ISBN: 9781118135006. URL: <https://books.google.at/books?id=4iDulhKgZsAC> (cit. on p. 62).
- Snell, James, Doug Tidwell, and Pavel Kulchenko (2002). *Building Distributed Applications: Programming Web Services with SOAP*. O'Reilly & Associates Inc. (cit. on pp. 11, 12, 14).
- SocialMediaExaminer (2016). *5 Ways to Use Snapchat for Business*. URL: <http://www.socialmediaexaminer.com/5-ways-to-use-snapchat-for-business> (visited on 03/04/2016) (cit. on p. 113).
- SocialTimes (2015). *Marketing 101 – Social Media vs Traditional Media*. URL: <http://www.adweek.com/socialtimes/social-vs-traditional-media-marketing/466873> (visited on 12/27/2015) (cit. on pp. 59, 60).
- Statista (2016). *Statistical Data about Social Media Platforms*. URL: <http://www.statista.com/> (visited on 01/06/2016) (cit. on pp. 77–87, 89–94, 96–98, 100, 102–109, 116, 117).

Bibliography

- Stegmaier, J. (2015). *A Crowdfunder's Strategy Guide: Build a Better Business by Building Community*. Berrett-Koehler Publishers. ISBN: 9781626564107. URL: <https://books.google.at/books?id=TI1zCQAAQBAJ> (cit. on p. 63).
- Stocker, Alexander and Klaus Tochtermann (2010). "Anwendungen und Technologien des Web 2.0: Ein Überblick." In: (cit. on pp. 21–24, 26–28, 34–36).
- Tanaka, Kenji (2012). *NPO Marketing for Project "Catrobat"*. Presentation (cit. on p. 67).
- Techopedia (2016). *Technology Dictionary*. URL: <https://www.techopedia.com/dictionary> (visited on 03/24/2016) (cit. on pp. 4, 5).
- TheDigitalMarketingGlossary (2016). *What is Video marketing?* URL: <http://digitalmarketing-glossary.com/What-is-Video-marketing-definition> (visited on 01/02/2016) (cit. on p. 57).
- TheDigitalMarketingReference (2016). *Email Marketing*. URL: http://www.marketingterms.com/dictionary/email_marketing (visited on 01/02/2016) (cit. on p. 57).
- TheGuardian (2013). *Teenagers say goodbye to Facebook and hello to messenger app*. URL: <http://www.theguardian.com/technology/2013/nov/10/teenagers-messenger-apps-facebook-exodus> (visited on 02/10/2016) (cit. on p. 98).
- Tumblr (2016). *About Tumblr*. URL: <https://www.tumblr.com/> (visited on 01/31/2016) (cit. on p. 81).
- W3C (2015a). *RDFa Lite 1.1 - Second Edition*. URL: <http://www.w3.org/TR/rdfa-lite/> (visited on 07/26/2015) (cit. on p. 29).
- W3C (2015b). *Resource Description Framework (RDF)*. URL: <http://www.w3.org/RDF/> (visited on 06/10/2015) (cit. on p. 44).
- W3C (2015c). *Semantic Web*. URL: <http://www.w3.org/standards/semanticweb> (visited on 12/18/2015) (cit. on p. 43).
- W3C-JSON-LD-Community-Group (2015). *JSON for Linking Data*. URL: <http://json-ld.org/> (visited on 12/19/2015) (cit. on p. 51).
- W3Schools (2015). *XML RDF*. URL: http://www.w3schools.com/webservices/ws_rdf_example.asp (visited on 06/10/2015) (cit. on p. 44).
- W3Schools (2016). *Web Services Glossary*. URL: <https://www.w3.org/TR/ws-gloss/> (visited on 03/21/2016) (cit. on p. 4).
- WeAreSocialPteLtd (2015). *Digital, Social & Mobile in 2015*. URL: <http://de.slideshare.net/wearesocialsg/digital-social-mobile-in-2015> (visited on 02/10/2016) (cit. on p. 99).

Bibliography

- WeeWorld (2016). *WeeWorld Homepage*. URL: <http://www.weeworld.com/> (visited on 04/16/2016) (cit. on p. 113).
- Weinberg, Tamar (2014). *Social Media Marketing: Strategien für Twitter, Facebook & Co*. O'Reilly Verlag GmbH & Co KG (cit. on p. 7).
- WhatsApp (2016). *FAQ - Using the Broadcast List feature*. URL: <https://www.whatsapp.com/faq/en/general/23741782> (visited on 03/04/2016) (cit. on p. 112).