

Technical Limitations for Designing Applications for Social Media

Christian Reuter, Simon Scholl

Institute for Information Systems, University of Siegen

Summary

Social media content is used in various applications for businesses, organizations and citizens. However, there are technical limitations for analyzing content from social media; these include the way how data can be gained and which safety regulations as well as query limitations have to be considered. They are of specific importance when designing applications for time critical scenarios, such as crisis management. This paper analyzes these limitations (in June 2014) for the most important social media. The selection of social media is based on the Monthly Active Users (MAU), which counts unique users over 30 days. Besides the identification of various limitations, this paper outlines approaches to access the data and summarizes design considerations.

1 Definition and Selection of Social Media

Social media is defined as a “*group of Internet-based applications that build on the ideological and technological foundations of Web 2.0 and that allow the creation and exchange of User Generated Content*” (Kaplan & Haenlein, 2010). Nowadays it is also used in (crisis related) applications to gather, analyze and visualize data, such as Twitcident, SensePlace2, Tweet4act or TwitInfo (Reuter & Ritzkatis, 2014). However, due to the ease of access, most applications as well as most published research papers focus on the use of Twitter. A study on 2012 Hurricane Sandy shows that communications differs across media type (Hughes et al. 2014). Also during the 2013 European Floods Twitter has been used for situational updates and to extend the coverage; Facebook for more complex coordination (Kaufhold & Reuter, 2014). Therefore it may be interesting, to focus on various social media, when designing applications. Caused by huge amount of data that is necessary for real time analyses in time critical contexts, this paper summarizes technical and business oriented limitations that have to be considered when designing applications for various social media. The selection of the most important ones is based on the MAU-Value (Monthly Active Users), which counts unique, active users over 30 days (Ballve, 2013). Users are considered to be active if they are unique and have logged in at least once in the month. We furthermore focus on systems which are well-known in Europe. An additional restriction is

the minimum value of 100 million. This provides the following list: **(1) Facebook (1.15 billion MAUs)**, **(2) YouTube** (1 billion), **(3) WhatsApp** (350 million) **(4) Google+** (327 million), **(5) Tumblr** (300 million), **(6) Twitter** (240 million), **(7) LinkedIn** (184 million) and **(8) Instagram** (150 million).

2 Analysis of Technical and Business Limitations

Based on the aforementioned list, the following section aims at the characterization of their technical and business oriented limitations and potentials. Almost all of the big players offer the possibility to access public data. Access to private user data can also be accomplished in most cases by asking for user authorization.

Currently, the most common architecture to request data is the use of REST-APIs. Representational state transfer is a way to create, read, update or delete information on a server using simple HTTP calls. They provide an easy way to extract heterogeneous datasets. To provide a secure communication between the requester and data source, a large number of requests are based on HTTPS. In addition to the safety of use, the user privacy is one significant aspect for data access. A standard way for APIs is distinguishing between public and private data. A public user profile contains basic information and such information, which is filtered by the privacy settings of the user. An application which wants to use this information does not need any special authorization for making the requests. But if more information is needed the requestor has to ask for the user's authorization. A common way for implementation is the OAuth 2.0 protocol (Internet Engineering Task Force, 2012) that allows applications to access each other's data.

Much of this functionality is wrapped in the provided SDKs and APIs for different programming languages to offer a simple way for integrating social media into new applications. Besides the default functionality for requesting and posting data, the providers offer many standard elements out of their systems for integration. For example, the Facebook "Like" button can be integrated in every website or application. LinkedIn offers the option to invite new members to the network out of any application. An application can integrate these features from different media types and thus shape a cross-system technology. For instance, a Facebook-App could invite people to LinkedIn. The limitations for using APIs and SDKs are nearly equal, because all functionalities which are provided have an underlying layer to query the systems' APIs. In the following we will present the limitations of the various APIs.

2.1 Facebook (Social Networking Service)

The *Graph API* allows accessing all data via several root nodes in Facebook. The permissions are regulated by tokens. A lot of user content, such as messages, is shared privately and requires a specific user token to be accessed. If you request a node, not all available fields will be returned automatically. It is necessary to define the parameters to filter the result set. The API can be called by HTTP requests or by using the PHP, JavaScript, iOS or Android SDK which wraps the HTTP calls. In Addition to that you have to generate an access token

which will temporarily provide a secure access. A token is able to identify a user, app or page. There are several use cases with different token types: The (1) User Access Token is used to call the API to read or modify personal data. An (2) App Access Token is used to request the Graph API for an application, not for a user. It is possible to change settings and parameters of an app, additionally the users can be managed, too. The (3) Page Access Token, based on the user access token, is used for API calls to read and modify a page. To generate such a token an admin has to grant extended permission to the page. A (4) Client Access Token is an identifier for mobile or desktop apps and gives access to a subset of API calls of the app access token. The token lifetime is managed by the Facebook SDK. One limitation for using the Graph API is the rate of 600 queries in 600 seconds (Stackoverflow, 2012). Furthermore a request has to pass several limitation layers: (1) User Level Rate Limiting: Each request gets assigned a score; once the entire score has reached the maximum limitation the call is blocked. (2) Specific API Rate Limiting: An allowed limit for API calls per second exists. If a user has reached the request limit for an app, the call fails.

In the *Keyword Insights API*, data include aggregated, anonymous insights about people mentioning a certain term. The *Public Feed API* provides a stream of posted user status updates and page status updates. It has no HTTP API endpoint; updates are sent to a server through a dedicated HTTPS connection. Access for both APIs “is restricted to a limited set of media publishers and usage requires prior approval by Facebook. You cannot apply to use the API at this time” (Facebook, 2014a, 2014b). *Facebook Query Language (FQL)* and *Chat API* are not supported anymore.

2.2 YouTube (Video-Sharing Website)

YouTube Data API (v3) allows accessing several resource types, such as channels, videos or playlists. In connection with these, the API offers standard operations on the resources like list, insert, update and delete. It is possible to request the API via HTTPS. It offers the possibility to integrate YouTube functionality into applications: Fetch search results or insert, update and delete videos or playlists. To perform these actions a Google account is required. Besides that, it is necessary to register the application with an API-Key at Google. After that the API must be activated as a service for the project. Requests to the API are limited by a quota. Every request gets a quota assigned. The value depends on the requested resource type and the selected operation. For example, a simple read operation has a cost of 1 unit, a write operation has a cost of 50 units and a video upload costs 1600 units. The API sets a limitation of 30,000 units per second per user. The entire quota per day is at 50,000,000 units (Google, 2013a).

YouTube Analytics API offers the possibility to retrieve viewing statistics, popularity metrics and demographic information for YouTube videos and channels. It is included in the Google APIs Client Library for .NET, Java, JavaScript, C, PHP, Python, Go, Google Web Toolkit, Node.js and Ruby. Each API request retrieves data for a particular YouTube channel or content owner. The quota is set to 50,000 requests per day (Google, 2014a).

YouTube Live Streaming API aims at creating, updating and managing live events on YouTube. The API connects events (broadcasts) and video streams, which represent the

actual broadcast content. Some API requests need user authorization via OAuth 2.0. Since the API is available in the Google-Developer area, the 1,000 subscriber limitation for channels to use the API seems to have been cancelled (Google, 2013b).

2.3 WhatsApp (Mobile Instant Messenger)

It is not allowed to access data from WhatsApp. There have been some Open-Source-Projects like DISA or WhatsAPI, but they were removed from GitHub because they have –according to WhatsApp - violated the copyrights of WhatsApp (Heise, 2014).

2.4 Google+ (Social Networking Service)

Google+ API provides access to public data and supports Android, iOS, C#/.Net, Go, Java, JavaScript, PHP, Python and Ruby. The core is formed by a REST-API, which can be queried over HTTPS. Therefore all calls require a token based on the OAuth 2.0 protocol or an API key. Based on a REST-Architecture the API offers retrieving and manipulating of resources via HTTP-Requests. Resources which can be accessed are data about people, activities, comments and moments. If developers are using the API, they can define app activities as ‘moment resources’ to reflect user actions. Users of an application have to allow the access to their user data via OAuth 2.0. The request rate is limited to 10,000 calls per day and 5 calls per second. Calls which belong to Google+ API (Sign-in) have a limit of 20,000,000 requests per day. Google+ API (Sign-in) covers calls out of the scope. Via the API-console developers can request a higher limit of API calls. Additionally developers can define a limit for user requests per second for their own application (Google, 2013c).

Hangouts-API – a special API for Google+ Hangouts Video calls – can be requested with a JavaScript interface and accesses data about hangout participants, shares data between app instances during a hangout session and controls the required hardware components. Because this API offers functionality around the Hangout-Services, there are no public limitations for using it in contrast to the Google+ API (Google, 2014b).

Google+ JavaScript API is easy to integrate into webpages via script loading. The API also supports multilingualism and asynchronous API loading. It is possible to integrate the +1-Button similar to Facebook’s “Like”-Button. To use such plugins it is necessary to comply with the Google guidelines, which regulate privacy and design aspects of the elements. The box is a further plugin resource, which provides the displaying of Google+ profiles. Additionally, Google+ - login can be integrated for the authentication and authorization of users, Google+ - sharing provides user interaction with Google+ on custom websites and snippets can display a rendered link preview of an URL. There are no direct limitations; the requested data come from the Google+ API. So the default rate limits also apply for this API (Google, 2013d).

Google+ Play SDK/ Google+ iOS SDK supports iOS and Android application development. The SDK provides the integration of UI-elements, the API calls for the JavaScript API. Additionally, it is possible to request other Google services such as location and activity

services. Here again, the limitations are determined by the default rate limits of the Google+ API (Google, 2013e, 2013f).

2.5 Tumblr (Microblogging and Social Networking Service)

Tumblr API is JSON-based and provides reading and writing permissions. It supports JavaScript, Ruby, PHP, Java, Python and C. The API distinguishes between blog-data and user-data. The access is regulated based on the authentication-level with three different levels of authentication. The first level is the public access which doesn't need any authentication, the second level makes it necessary to generate an API key and the third level needs a signed request based on the OAuth 1.0a protocol. Additionally, the number of requests to 250 posts per day, across the entire owned blogs for each account, are limited. This includes the main blog which is created at registration and the secondary account blogs. Within these 250 posts and re-blogs, it is possible to create 75,150 original photo posts per day. Tumblr has a publishing option for staggered posts with a queue limit of up to 50 posts per day (Tumblr, 2014). It allows to like up to 1,000 posts per day from the own account (Novak, 2014).

2.6 Twitter (Microblogging Service)

REST-API is based on the REST-Design, which allows requests based on HTTP/HTTPS. It offers the possibility to query and modify a user account. Modifying an account requires user permission via OAuth 2.0. The rate limit of 15 minutes' duration defines how many requests for an API-Call are allowed per user and per application. For instance, requesting the friends list of a given user can be called 15 times by a single user and 30 times by an application within 15 minutes (Twitter, 2014a). An overview on all the different calls can be found at the Twitter-developer page (Twitter, 2014b).

Streaming-API delivers live tweets based on a search term or specific users which get requested. Similar to a push service, the streaming-API requires a persistent HTTP connection. The streamed tweets are limited by a streaming cap. The calculation of this cap is based on the total amount of tweets at a certain time. If there are more tweets that match your filter criteria, then you will receive the number of the tweets which are not streamed. For example, you are allowed to get 1% from 30,000 tweets for a given set of hashtags (Twitter, 2014c).

2.7 LinkedIn (Business Oriented Social Networking Service)

LinkedIn-Rest-API supports HTTP requests over REST; for development purposes, LinkedIn offers a JavaScript wrapper via the default REST-API. The API supports XML and JSON response format. The available information, which is provided by the LinkedIn API, is divided into several resources: people, companies, groups, connections, jobs and separate resources for searching people, companies and jobs. This results in Sub-APIs which describe the access to the resources individually. Based on a large set of different parameters, it is possible to request explicit data and metadata about information resources. This allows making complex requests to the API in an easy way. Besides the read access to LinkedIn

data, some Sub-APIs offer special functionalities. The connections-API gives the possibility to exchange messages between several members, the invitation-API allow users to invite other people to LinkedIn out of custom applications and the share-API let users distribute shares on LinkedIn while the custom app defines the appearance of the content. LinkedIn divides data into public and private. For example, a requested profile can be called as standard or public. The public request returns all public user data, based on the privacy settings of the profile owner. The standard requests filter the response even on the privacy settings of the owner, but also on the privacy setting of the requestor and the relationship between requestor and owner. In addition to using OAuth for generating access tokens for API calls, LinkedIn has a defined throttle for applications, users and developers (LinkedIn, 2014a). The usage of the capacities is cleared every day. For instance, posting shares has a daily limit per application of 125,000 calls, per user 25 and per developer 100 calls (LinkedIn, 2014b).

2.8 Instagram (Mobile Photo and Video Sharing Service)

Instagram-API is based on a REST-architecture and the API-endpoints can be requested via HTTPS. For special support they offer a python and a ruby library. The API supports access to data about users, relationships, media, comments, likes and locations. Especially Instagram refers to core terms for the use of the API. Among others, defined API-terms are (Instagram, 2014a) that users own their media and that it is not allowed to use the Instagram API to crawl or store media without the express consent of the owner. Too many requests in a short time will get your access turned off. 5,000 calls per authenticated user (access token or client) per hour is another limitation. API actions such as commenting, following/unfollowing and liking have its own limit of 350 calls per hour (Instagram, 2014b). The standard calls are signed with a default scope. Scopes provide permissions to advanced API calls. It is possible to extend permission with scopes for creating and deleting comments, scopes for following and unfollowing a user as well as to like and unlike items (Instagram, 2014c).

Instagram-RealTime-API supports the subscription via server. It works with an integrated push-service to notify applications about new content. For that, developers can subscribe Instagram-objects, like users, tags and locations. This is limited to 30 real-time-subscriptions (Instagram, 2014d).

3 Summary of Design Considerations

Access to information from social media is possible. Besides simple data, such as profile information, it is possible to request additional information about the connection of users and groups. The technical possibilities for the integration of other applications are widely diversified by the provider; especially in the mobile sector the trend goes towards mobile SDKs, which make standard functionalities accessible to each type of program similar to Facebook's "Like" button. The access to internal data is subjected to security measures such as the OAuth protocol. The protection of personal data is in the foreground, so that access to

it must be approved by the users themselves. In addition to safety aspects, the use of these services is only possible in the context of the granted query rate. If, for example, a search query have met the capacity in a specific time interval, then its use is only be possible after the end of this interval. These rate limits are dynamic for many requests and systems; if the given amount of possible requests is not sufficient, it will be necessary to make a separate request to the system owner. In sum the most important limitations are as follows:

1. *Facebook* allows 5M MAUs and 100M API calls or 50M impressions per day.
2. *YouTube* sets limits to 30,000 units per second per user and 50,000,000 units per day.
3. *WhatsApp* **does not allow accessing data – however it is technically possible.**
4. *Google+* API sets a limitation of 10,000 requests per day and 5 requests per second.
5. *Tumblr* limits the amount of posts per day for all owned blogs for each account to 250.
6. *Twitter* distinguishes between several requests in a 15-minute interval whilst it is possible to query 180 tweet-searches per user and 450 tweet-searches per app.
7. *LinkedIn* sets daily limits with the same distinction as Twitter. So it is possible to request in the role as application, developer or user: An application can make 100,000 status updates per day, a developer and a user only can make 250 updates per day.
8. *Instagram* sets a defined limit of 5,000 requests per hour per access-token.

An application which reaches one of these general limits may negotiate with the providers – however small and not business related tools for crisis management without income might have difficulties to pay for it. It is also possible to request higher rate limits on a non-monetary base, for that the providers, like Google, demand access to the corresponding application to validate the need of an increase of the limitation. A defined pricing table is not available, even Google just offers pricing tables for services like “App Engine” (Google, 2014c). Using other elements like login buttons has to deal with the specific guidelines for integration, which are dictated from social media owners. These guidelines define the look and functionality of given resources.

A new trend seems to be streaming- and live-APIs that allow developers to integrate live information out of social media. Requestors subscribe specific content like YouTube channels or tweets and get notified about new content. This is “the next step” to build rich social media apps, which not only are able to request and interact with other social media services, but also may observe content and react on specific events. Especially for crisis management, where specific events have to be observed e.g. to create community maps (Goolsby, 2010), this might be of value. Besides this, another trend is based on the increasing usage of mobile devices. The amount of users which access the internet via mobile devices will further raise in 2015 (eMarketer, 2014). Social media are aware of the mobile context and offer functionality supporting the integration of their platform into third-party mobile apps. Finally it must be clear that the exchange of data with social media alone cannot improve crisis management. However, data exchange considering the technical limitations is a necessary pre-condition for every social media related application.

Funded from the EU FP7 project “EmerGent” (grant agreement 608352).

Contact: christian.reuter@uni-siegen.de; simon1.scholl@student.uni-siegen.de

Literature

- Ballve, M. (2013). The World's Largest Social Networks. <http://www.businessinsider.com/the-worlds-largest-social-networks-2013-12#ixzz30MjZz5TK>
- eMarketer. (2014). Smartphone Users Worldwide Will Total 1.75 Billion in 2014. <http://www.emarketer.com/Article/Smartphone-Users-Worldwide-Will-Total-175-Billion-2014/1010536>
- Facebook. (2014a). Keyword Insights API. https://developers.facebook.com/docs/keyword_insights
- Facebook. (2014b). Public Feed API. https://developers.facebook.com/docs/public_feed/
- Google. (2013a). YouTube Data API (v3). <https://developers.google.com/youtube/v3/?hl=de>
- Google. (2013b). YouTube Live Streaming API. <https://developers.google.com/youtube/v3/live/?hl=de>
- Google. (2013c). Google+ API. <https://developers.google.com/+api/?hl=de>
- Google. (2013d). Google JavaScript-API. <https://developers.google.com/+web/api/javascript?hl=de>
- Google. (2013e). Google+ Platform for iOS. <https://developers.google.com/+mobile/ios/getting-started>
- Google. (2013f). Google+ Platform for Android. <https://developers.google.com/+mobile/android/getting-started>
- Google. (2014a). YouTube Analytics API. <https://developers.google.com/youtube/analytics/?hl=de>
- Google. (2014b). Hangouts-API-Referenz. <https://developers.google.com/+hangouts/api/>
- Google. (2014c). App Engine Pricing. <https://developers.google.com/appengine/pricing?hl=de>
- Goolsby, R. (2010). Social media as crisis platform: The future of community maps/crisis maps. *ACM Transactions on Intelligent Systems and Technology*, 1(1), 1–11. doi:10.1145/1858948.1858955
- Heise. (2014). WhatsApp kickt Open-Source-Projekte von Github. <http://www.heise.de/open/meldung/WhatsApp-kickt-Open-Source-Projekte-von-Github-2112486.html>
- Hughes, A. L., Denis, L. A. S., Palen, L., & Anderson, K. M. (2014). Online Public Communications by Police & Fire Services during the 2012 Hurricane Sandy. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. Toronto, Canada: ACM.
- Instagram. (2014a). API Terms of Use. <http://instagram.com/developer/#>
- Instagram. (2014b). Instagram API Endpoints. <http://instagram.com/developer/endpoints/#>
- Instagram. (2014c). Hello Developers. <http://instagram.com/developer/#>
- Instagram. (2014d). Real-time Photo Updates. <http://instagram.com/developer/realtime/#>
- Internet Engineering Task Force. (2012). The OAuth 2.0 Authorization Framework. <http://tools.ietf.org/html/rfc6749>
- Kaplan, A. M., & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of Social Media. *Business Horizons*, 53(1), 59–68. doi:10.1016/j.bushor.2009.09.003
- Kaufhold, M.-A., & Reuter, C. (2014). Vernetzte Selbsthilfe in Sozialen Medien beim Hochwasser 2013 in Deutschland. *i-com - Zeitschrift für interaktive und kooperative Medien*, 13(1), 20–28.
- LinkedIn. (2014a). Throttle Limits. <https://developer.linkedin.com/documents/throttle-limits>

- LinkedIn. (2014b). API Overview. <http://developer.linkedin.com/apis>
- Novak, M. (2014). Unwrapping Tumblr. <http://unwrapping.tumblr.com/tagged/tumblr-limits>
- Reuter, C., & Ritzkatis, M. (2014). Adaptierbare Bewertung bürgergenerierter Inhalte aus sozialen Medien. In *Mensch & Computer 2014*. München, Germany: Oldenbourg-Verlag.
- Stackoverflow. (2012). Facebook graph API rate limit and batch requests. <http://stackoverflow.com/questions/8805316/facebook-graph-api-rate-limit-and-batch-requests>
- Tumblr. (2014). Tumblr API. <http://www.tumblr.com/docs/en/api/v2>
- Twitter. (2014a). The Twitter REST API. <https://dev.twitter.com/docs/api>
- Twitter. (2014b). REST API v1.1 Limits per window by resource. <https://dev.twitter.com/docs/rate-limiting/1.1/limits>
- Twitter. (2014c). The Streaming APIs. <https://dev.twitter.com/docs/api/streaming>