

# Title - to be discussed

Carsten Porth, [carsten.porth@stud.tu-darmstadt.de](mailto:carsten.porth@stud.tu-darmstadt.de), 1804629



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

---

## Meta data

---

**Name:** Carsten Porth  
**Student number:** 1804629  
**Department:** Computer Science  
**Allocation Date:** xx.xx.2018  
**Supervisor:** Jörg Daubert  
**Second Supervisor:** Aidmar Wainakh  
**Category for Digilib:**

---

## 1 Motivation

---

The work will be done as part of the project "Privacy and Trust for Mobile Users". The ongoing penetration of computers in everyday life is used by a large part of the population for their own benefit. Emerging disadvantages, such as the increasing transparency and monitorability of a user with simultaneous lack of transparency of the entire network, are accepted. The aim of this interdisciplinary project is to reverse these trends and adapt privacy to personal interests. Economic interests on the one hand and social interests on the other should be reconciled.

With Web 2.0, the use and perception of the Internet changed. Instead of just consuming content as before, web applications were created that allowed users to place content on the Internet themselves. The big social media networks such as Facebook (2 billion monthly active users<sup>1</sup>) or Twitter (330 million monthly active users<sup>2</sup>) have become particularly popular.

With the evolution of the mobile phone to the smartphone, a range of functions preciously known only by computers has been made accessible mobile. The ability to install any application and to use the device for a variety of purposes opened up new possibilities. The development of the hardware and new sensors make it possible to comprehensively capture the environment. This technology is not only used in smartphones, more and more tracking devices and smart watches are conquering the market. For the user, these changes initially bring advantages. What happens to the collected data and what may be unconsciously shared with servers all over the world and possibly further processed there is in most cases opaque.

Especially in social networks, a lot of data is collected by the respective app. If the user wants to use a certain network, he usually has no choice but to divulge the data. However, recent data scandals have shown that the user should be careful about what data he reveals. Often there are no opportunities to determine this yourself. The aim should therefore be to find a way to be able to use the already used social networks as usual, but to decide for oneself which data are exchanged via the provider servers and which are exchanged via a separate network that is closed to the provider.

---

## 2 Why not Facebook?

---

In March 2018, it was revealed that Facebook had not treated the data of 87 million users confidential. The British company Cambridge Analytica had collected and processed data via Facebook. Both companies suffered immense damage after it became known. Facebook suffered a major damage to its image, Cambridge Analytica had to file for bankruptcy and ceased operations.

Improving privacy and protecting the personal data of Facebook users would be of particular interest against this background. However, the strong limitations of the Facebook API (Graph API) do not allow it to develop your own Facebook client. For example, it is not possible to query a user's news stream or to like a post.

With apps like *Friendly for Facebook* or *Metal* there are alternative Facebook clients in the Google Play Store. Since the official Facebook API may not have been used due to the restrictions, two approaches are conceivable:

1. The website offered for mobile devices (<https://m.facebook.com/>) is displayed in a WebView. Thus, the app only serves as a wrapper around the mobile website. However, the different design indicates that the pages are modified by the Android app. This is technically possible by injecting JavaScript code into the page. Disadvantage is that changes on the mobile website lead to the injected JavaScript code suddenly no longer having the desired effect.
2. The Facebook website is crawled and the content extracted. The display of the data can therefore be freely designed. The disadvantage of this method is its dependence on the HTML structure of the Facebook web

---

<sup>1</sup> <https://www.heise.de/newsticker/meldung/Facebook-meldet-2-Milliarden-aktive-User-3757367.html>

<sup>2</sup> <https://www.heise.de/newsticker/meldung/Twitter-schafft-ersten-Gewinn-Nutzerzahl-stagniert-3963390.html>

---

---

pages. Even small changes to the website could prevent the content from being extracted. Furthermore, requests to the Facebook servers would have to pass through the security precautions on Facebook, for example the presence of any tokens.

The limitations of the Facebook Graph API and the problems with the alternative approaches mentioned were decisive for the decision in favour of Twitter. Twitter releases the full range of functions for developers via its API, allowing them to develop their own clients. Furthermore, a kit is provided specifically for the development of Android apps, which makes the development much easier.

---

### 3 Requirements

---

To develop a Twitter client app for Android based on the objectives, the author should have some knowledge on different topics, which is discussed in more detail below.

Basic knowledge of the used programming language Java should be present, which is necessary for the development of the Android application. Experiences with the development of Android apps are beneficial.

To exchange data between the app and the Twitter servers, knowledge of REST interfaces and the Twitter API is required, which must be acquired at the beginning of the work, if they are not available. Furthermore, knowledge of peer-2-peer networks is required to implement the protected data exchange between the clients.

Since the actual work with LaTeX is written, previous knowledge should also be available there. Basically, the author must be willing to acquire and expand missing knowledge independently.

---

### 4 Approach/Goal

---

The goal of the thesis is the development of a Twitter client app for Android. In this app it should be possible to use the traditional functions of Twitter as usual, so that communication with any other Twitter user is possible without problems.

In addition, it should also be possible to decide for yourself whether a tweet is shared with the followers via a peer-2-peer network and is therefore not stored on a Twitter server.

The Twitter API is mainly used via the Twitter Android Kit<sup>3</sup>, which already includes UI components to display the essential elements. However, it will be necessary to modify the kit to include tweets from the P2P source. This probably cannot be achieved only by extending the contained classes, so that the code provided on Github has to be adapted. Since the Twitter Android Kit is licensed with the Apache License 2.0, such modifications are possible.

Due to the limited time the app should have the following functionalities:

- Writing tweets in text form
- Retweet Tweets
- Like Tweets
- Show Timeline (Home)
- Show user profiles
- Search for Tweets and users

As a result, the following functionalities are not implemented in the context of this thesis:

- The notification system
- The direct message system
- Changing settings
- Update the user's profile
- Writing tweets containing any multimedia attachment like videos and pictures

---

<sup>3</sup> <https://github.com/twitter/twitter-kit-android>

The peer-2-peer network should not only extend to a local network, but to a global network over the Internet. In doing so, the problem with the network address translation (NAT) has to be conquered. The tweets exchanged via the peer 2 peer network are serialized into a JSON or XML format and then exchanged among the clients via the network.

Two approaches are conceivable for the implementation of the P2P network:

1. The Java Library Hive2Hive<sup>4</sup> (based on TomP2P<sup>5</sup>) is suitable for setting up a P2P network for secure file synchronization.
2. WebRTC<sup>6</sup> is a approach for real-time communication that is supported by Google, Mozilla and Opera, among others. In addition to modern web browsers, Android and iOS are also supported as platforms.

The first step is to evaluate the strengths and weaknesses of the two approaches and the extent to which they are better suited for their intended use.

Peer discovery is the most difficult task in both approaches. Since synchronization should work over the Internet and in most cases the clients should not have a static address, a solution must be found for clients to discover other clients. For Hive2Hive, this problem can be solved by using a bootstrap node that has a fixed address and is first contacted by the clients. The software for this node would also have to be written and deployed on a server with appropriate properties. WebRTC also requires a central contact point. This could also be implemented by own software, but for this purpose there are also third-party solutions that can be used free of charge.

The thesis is written in English.

## 5 Development Environment

The Android app is to be developed in Android Studio. The minimum SDK version results from the minimum requirements of the used libraries.

*Comment on Git repo, license, etc.*

## 6 Schedule

The following schedule should only give a rough outline. Where appropriate, individual phases may also overlap and change slightly over time. In addition, regular meetings are scheduled with the supervisors to discuss progress and possible problems.

Month	Task	Description
	Induction	Analysis of the provided APIs of different social networks. Familiarization with the problems of a P2P network beyond the local network.
Start	Discussion of the task and goal	
1 - Beginning		Masterand begins his literary work. Development of a concept for the implementation of P2P synchronization.
1 - Middle	Hand in exposé	Acceptance of the exposé by the supervisor
2 - Beginning	Start of implementation	Results of the literature search, beginning of the development of concepts, architecture...
3 - End	Prototype	
4 - Middle	Trial period	Two-week test phase with about 10 users
5 - Middle	Implementation completed	
6 - Middle	2nd presentation	
6 - End	Submission of written thesis	

<sup>4</sup> <https://github.com/Hive2Hive/Hive2Hive>

<sup>5</sup> <https://github.com/tomp2p/Tomp2P>

<sup>6</sup> <https://webrtc.org/>