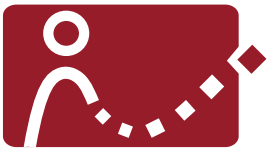
Hybrid Online Social Networks



Masterarbeit von Carsten Porth
Tag der Einreichung: 6. Januar 2019

1. Gutachter: Prof. Dr. Max Mühlhäuser
2. Gutachter: Jörg Daubert, Aidmar Wainakh

Darmstadt, den 6. Januar 2019

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Informatik
Telekooperation
Prof. Dr. Max Mühlhäuser

Abstract

Zusammenfassung

Inhaltsverzeichnis

1 Introduction

1.1 Motivation

1.2 Challenge

1.3 Research Question

1.4 Outline

2 Background

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.1 Network Architecture

Networks are systems that connect several independent computers and enable data exchange between them. The structure of the networks can be represented mathematically as a graph, with the nodes representing computers and the edges representing their connections. Depending on the architecture, two types of networks are distinguished: centralized and decentralized networks (see Figure 2.1). The characteristics of the two different network architectures are described below.

Abbildung 2.1: Schematics of different network architectures

2.1.1 Centralized Networks

Central networks are characterized by a central node, via which all other nodes are connected to each other. Otherwise there is no further connection between the nodes. This is illustrated

in Figure 2.1. For reliability, such a network architecture results in no connection between the other nodes if the central node fails. To join the network, only a connection to the central node is required, which can normally be established via a known, static address.

With such hierarchical structures, the central node is often referred to as a server and the connected nodes as clients. This client-server structure underlies all large social networks, such as Facebook, Twitter and Google+, and is therefore of great importance for the Internet. The users are connected to each other via the server and can thus exchange data. The direct exchange among each other is not possible, so that the data exchange can be controlled by the server, which is under the control of the provider. Each client is therefore dependent on the server. A client node fulfils different requirements than a server node.

2.1.2 Decentralized Networks

Decentralized networks are characterized by the fact that not only one central node has the complete information and control over the system. The nodes are also connected to each other by edges, not all of them with the same node, as shown in Figure 2.1. With such a structure, each node performs the same tasks, which is why we speak of peers as nodes and refer to such networks as peer-to-peer (P2P). In the event of a node failure, the network can continue to function because the information and functionality is distributed. Restrictions and censorship cannot be easily implemented in this type of network.

Popular examples for decentralized networks are file sharing networks like eMule, Limewire, BitTorrent, etc. and blockchains like Bitcoin or Ethereum.

2.2 Peer-to-peer

2.2.1 Structured vs. Unstructured Networks

2.2.2 Decentralized Applications

2.3 Summary

3 Related Work

This chapter gives a comprehensive overview about different projects trying to protect the users' personal data in online social networks. Seven different approaches are presented and their relevance for this work discussed. While some projects are extensions for established online social networks, others are completely new social networks. For each project, the way data is stored and accessed and how communication is realized are investigated. The chapter concludes with a summary of the related work.

3.1 Safebook

3.2 Diaspora

3.3 Akasha

3.4 LifeSocial.KOM

3.5 Twitterize

3.6 FaceCloak

3.7 ActivityPub

3.8 Summary

4 Concept

The past, especially the recent past, has shown that the private data of users in social networks is not secure. On the one hand, social networks are popular targets for hackers and on the other hand, the collected personal data is processed and used, for example, for targeted advertising. For the users of the network it is easy to share personal information, messages and pictures with other users. While the circle of users who can see this data can usually be determined by the author himself, the social network inevitably has access to this data because it is stored on the operator's servers. What happens to the data is not transparent to the user. Although there have been and still are social networks that make it their business to protect users' data in a special way and promise to take care, they regularly fail because the users of the large social networks are trapped on these platforms by a lock-in effect. Since the majority of contacts on the large networks are already linked, it is unattractive to switch to a smaller, new social network where the contacts are missing.

Users are faced with the dilemma that switching to another social network entails the loss of contacts they have collected over years. But staying with the large network entails uncertainty about their own data. So, the question is how to share fewer sensitive data with the provider, despite the usual use of a social network. In the context of this master thesis, this question was dealt with intensively and a solution of a hybrid social network was worked out.

"Hybrid" in this case means that an existing (social) network is extended by another network for data exchange. This is a P2P network among the users of the social network, which allows data to be exchanged directly so that the operator of the social network is not aware of it. When using the social network, the P2P network is to be superimposed like a privacy layer. During the actions carried out, the user can then decide for himself via which network the data should be exchanged. In order to exchange data via a P2P network, the usual client must be extended or manipulated for the respective social network or a separate client must be developed for use. Using a P2P network would also have the advantage that the data could not be censored by the operator of the OSN.

The following sections describe the requirements for the respective components OSN, P2P network and client.

4.1 Requirements to the OSN

Social networks are usually used via a web frontend or associated apps. While there is no way to influence the content and extend the functionality of apps, web pages can be manipulated by browser add-ons to add the desired functionality. Another possibility is to display the contents of the social network in a separate, self-developed application. In this case one would have complete control and could adapt the functionalities accordingly. There are two possible ways to access the original content of the social network. First, the websites could be crawled and the content extracted. Second, some social networks provide interfaces that allow developers to

communicate directly with the servers and thus exchange data easily. These interfaces are called API.

Since not all social networks offer an API, and APIs provided do not usually provide the full functionality and are limited by query limits, crawling offers a way that does not rely on the operator. A clear advantage is that this way always works. However, much more effort is needed to extract the desired data from the website. The data is expected in a certain structure in order to be extracted. If this structure is changed by the operator of the social network, the crawling fails and the data can no longer be extracted. In addition, social networks today are complicated web applications. The content is dynamically reloaded and the page changes dynamically depending on the action. This is difficult to capture with crawlers. Furthermore, networks protect themselves from crawlers and try to prevent them from crawling content. In the best case, the network operator does not notice that his page is being crawled because ideally the crawler behaves like a normal user and can therefore not be distinguished. To publish content in the social network, the requests sent to the network would have to look as if they had been created by the forms in the web frontend. Here, too, various security mechanisms attempt to identify and block such ”faked” requests, as this could also be used to load malicious code onto the servers. All in all, it is very difficult nowadays to communicate with the servers of a social network via crawling and fake requests.

It is much easier if an interface is provided and clearly documented how data can be retrieved or added. It is important that the essential functions are provided via the API, so that a client application can be developed that is not inferior to the original website in terms of functionality. Otherwise the users won't switch to the alternative client. By using an API, one enters into a certain dependency on the network. There is no guarantee that the provider will keep up the API and offer all the functionality. Since they are the provider, they decide what happens to the API and the developers using the API and build their applications are completely dependent on them.

To use an API, usually a previous registration is necessary. After registration, the developer receives an API key, which is sent with every request to the API and checked for validity. The number of negative incidents with different social networks has led to providers gradually restricting their APIs more and more. One is the removal of endpoints for certain actions, the other is the introduction of limits. By sending the API key, it is possible to see which developer or which application communicates with the API. Limits can then be used to determine how many requests are processed in a certain time interval or how many users can use the network via this application. It is also possible that the developer has to pay for the API key.

Regardless of the method chosen to exchange data with the social network, a unique content identifier is required. This is necessary so that interactions with it can be related. By knowing the ID of a post, associated comments can be loaded - both from the provider's server and from the private P2P network.

4.2 Requirements to the Hybrid Client

In order for users to be ready to use an alternative hybrid client for social networking, this client must meet a number of requirements. First, all the functionality that the user knows from the

official app/website must be available. If this is not the case, they will always be forced to go back to the original application. Using more than one app for a network is not user friendly and should be avoided. Basically, great importance should be given to user-friendliness. Complicated procedures should be avoided as best as possible or simplified in such a way that really every user is able to understand and use them correctly. This also means that great importance must be attached to a good user experience, so that content is presented in an appealing way and operation is intuitive.

The hybrid client must also meet a number of technical requirements in order to be accepted by the user and contribute to a good user experience. This includes, among other things, the availability for common operating systems. It is necessary that the hybrid client can be used wherever the social network can otherwise be used. If messages are exchanged via the P2P network, but there is no client for Apple's iOs, for example, users with an iPhone cannot retrieve this message on the move. This would result in the message being exchanged via the OSN's server despite the sensitive content that is to be protected from the OSN's operator, since only in this case can the recipient receive the message. It is also important that the hybrid client uses resources such as memory, battery and data volume sparingly. Since client devices (peers) play a special role in a P2P network, the careful use of resources is particularly challenging. This is because the data must be stored on the peers and ideally made available to other peers at any time. It should also be possible for a user to log on to several devices with his account and still benefit from all the advantages. This parallel use and the synchronicity of the data must be guaranteed.

Finally, the security of the data exchanged between the hybrid clients must be guaranteed. The OSN must not be able to use these data and even less not be able to assign these data to a certain user. It must also be ensured that the data can only be read by those users who are authorized to do so. These security aspects must have the highest priority when designing the system.

Most social networks can be used free of charge. Ideally, the use of the hybrid client is also free. Otherwise it could be the first argument against using this client.

4.3 Requirements from the OSN

Basically it can be assumed that the operator of an OSN does not like the idea of a P2P extension of his network. Therefore, information from the P2P network should be offered to the operator without establishing a reference to a user. With this anonymous data, the operator still has important information at his disposal, for example for the recognition of trends.

4.4 Summary

todo...

5 Proof of Concept

The ideas for a hybrid OSN presented in Chapter 4 were implemented with the development of a prototype. The resulting hybrid client for Twitter is presented in this chapter and the important design decisions are presented in a comprehensible way. Essential aspects such as the choice of OSN, the choice of technologies used to develop the app and important features of the client are explained.

5.1 Choosing an OSN

When selecting a suitable OSN for the development of a hybrid client, Facebook was the obvious choice due to the numerous negative headlines about data protection. With over 2 billion users per month, it is currently the most widely used social network in the world. In the recent past, it has often been criticized for its handling of its users' data. In particular, the scandal surrounding the data analysis company Cambridge Analytica, which had access to the data of up to 87 million Facebook users, hit Facebook hard. As a result, CEO Mark Zuckerberg had to face the US Congress and the EU Parliament in question rounds and did not leave a good impression by avoiding many questions. As a result of this scandal, the Facebook API was further restricted.

However, the Facebook API is not suitable for developing your own client anyway. The functionalities offered by the API offer the possibility to develop an app that can be used within Facebook. For example, it's not possible to make a like for a post using this API. As discussed in chapter 4, it is possible to access the data through crawling. However, the constant and rapid development would make this an arduous undertaking. In a blog entry (https://code.fb.com/web/rapid-release-at-massive-scale/) Facebook writes that the code changes every few hours. So it is almost impossible to adjust the crawler fast enough and roll out the adjusted code.

If you take a look at the app stores, you will see alternative Facebook clients. Because Facebook doesn't provide an API and crawling isn't an alternative, these clients have to use a different way. This other way is to display the mobile Facebook page (m.facebook.com) and manipulate it with injected CSS and JavaScript files. The app itself therefore only offers a container around the mobile Facebook website and modifies it. The changes then often refer only to color changes and links to frequently used pages. In order to adapt the design, knowledge of the page structure and the structure of individual components is necessary. The fast and frequent code adjustments of Facebook lead to the fact that the design changes of the app do not work in some places and have to struggle with the same problems as with crawling.

For this number of reasons, Facebook was dropped as OSN for the prototype, despite special interest. Another candidate was the OSN Google Plus, which was dropped from Google because Google announced it would discontinue its OSN in October 2018 (https://www.blog.google/technology/safety-security/project-strobe/).

Ultimately, the OSN's choice for the prototype fell on Twitter. With 336 million active users per month, it is one of the largest social networks. It is particularly well suited for the development of a hybrid client for two reasons: on the one hand, it has a detailed API that provides almost the full functionality free of charge, and on the other hand, compared to Facebook, it offers only a few simple functions. These are the ideal prerequisites for a first proof of concept. In order to use the API, an application must be registered for which an API key is then assigned. This API key entitles the user to use the API and must be sent with every request to the API in the request header. Use of the API is limited. The limitation depends on the app and the user.

5.2 Summary

6 Evaluation

7 Conclusions

7.1 Summary

7.2 Contributions

7.3 Future Work

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

Erklärung zur Abschlussarbeit gemäß § 23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Carsten Porth, die vorliegende Masterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs.2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Masterarbeit stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung überein.

Bei einer Masterarbeit des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, den 6. Januar 2019

Unterschrift