Friend-to-Friend Computing:
Building the Social Web at the Internet Edges

Wojciech Galuba
Ecole Polytechnique Fédérale de Lausanne (EPFL)
wojciech.galuba@epfl.ch

***Abstract*—The current Social Web is centralized. Large information silos store all the users' profiles, their social links and much of the other personal data. In return for the reliable service the users allow their data and activities to be data mined by the service providers, which in this way increase their advertising revenue. As the social applications are storing increasingly more data and attracting more users, many questions about privacy, data ownership and data portability arise.**

In this paper we are going to critically assess the current state of the Social Web, identify several novel research problems and outline the possible solution: friend-to-friend computing (F2F). F2F is a completely decentralized architecture in which two computers can communicate only if their owners know one another. Constraining the connections to friends-only solves many of the security problems of the peer-to-peer architectures. We argue that a reliable social application platform can be built using F2F as the substrate. The platform gives the users much more control over their data than the current Social Web and ensures the level of privacy and security not possible in any centralized architecture. Groups can easily build their own ad-hoc networks and collaborate without the need for any servers or third-party services.

I. Introduction

While it is still an open question whether Web 2.0 is not the next tech bubble and whether profitable business models can be built around it [31], the rise of Web 2.0 has brought with it several important changes. Perhaps the biggest of them is the transformation of the Web from a passive read-only medium to a participatory read-write one. Users are not only consumers but also collaborating producers of content, the prime example of which is Wikipedia. Users also have started bringing their social links to the Web giving rise to a new breed of the *social application*. Thousands of such applications have sprung up enabling, as Clay Shirky [29] puts it: ”ridiculously easy group formation”. However, to be successful each new social application must overcome the chicken-and-egg problem. The value of the social application to its users is the greater the more of them are using it, but applications need the users' social links and the users are required to input it for each new application, which is a slow and repetitive process. This problem has prompted several big on-line social networks to turn themselves into a Social Web platform [5], [9], [8]. Applications have access to millions of users and their social links and can rapidly expand their user bases. However, as the applications are getting tied to the platforms and storing more data in them, many questions about privacy, data ownership and data portability arise. **In this paper we are going to critically assess the current state of the Social Web, identify several novel research problems and outline the possible solutions.**

Interestingly enough, given the decentralized grass-roots social nature of the new Web, from the systems architectural point of view it is still a centralized client-server. Both the social links and the social applications are stored and run in the data centers. But is it the best systems architecture for the Social Web? In the recent years the peer-to-peer (P2P) systems, just like the Web have been rapidly evolving and have become a very efficient and robust communication and content delivery platforms. The popularity and usability of the P2P systems can only increase as more upload bandwidth and other resources are available at the Internet edges [15]. **The central argument of this paper is that many of the current problems of the Social Web can be solved by shifting it from the data centers to the edges of the Internet where the actual users, their data and their social links are.**

II. The motivation & the problems

A. Who owns the data?

As the online social networks are becoming more popular they accumulate more data and it is becoming increasingly more tempting for the third parties to gain access to it. In almost all the systems the Terms of Service prohibit any form of automated data extraction and the users that attempt it are quickly cut off. Several high-profile incidents [10] have sparked the discussion on who exactly owns the social data [13]. One camp claims that users should be allowed to export their social links, while the other argues that a social link is effectively owned by two people and anything that happens to it should be at the consent of both of the owners. Whatever the consensus in this data ownership discussion will be, an immediate research question arises: **What technological means can be used to control access to the social data?**

B. Controlling access to the personal information space.

The Social Web stores not only the data about the social links but also the user's profiles, comments, messages, photos, user action history or in general any social application state [4]. All this, forms the user's personal information space. The user might want to give access to parts of this space to another user or to the general public, while keeping the other parts strictly private. The different parts of the personal information space are modified by the different social applications and some applications might like to share part of the space with other applications [3]. The three actors: users, applications and the online social networks hosting the social data need to coordinate the data sharing process. **How can the personal information space be securely shared between users and between applications? How can we precisely define the access control rules?**

C. Protecting privacy & the attention data.

When users are accessing any of the data in their personal information space, that data is hosted somewhere, either at the servers of the social networking website or at the servers of the social applications. The servers can log the access to any data for any particular user. This data is termed the *attention data* [1] and is typically used to serve more accurate, targeted advertising, which makes the data valuable to the service providers. The attention data is extremely sensitive and any data leaks have serious privacy implications [12], [25], [24]. However, the user might like to have control not only over who can access the user's data but also control who knows about the fact that the data was accessed. **How can we prevent the attention data leaks and privacy violations in the Social Web?**

D. Need for a Social Web platform giving users more control.

Currently the majority of the Social Web is locked in a few information silos, which store the social data on behalf of the users and in return the users provide advertising income. For this reason the social data is valuable to companies and is not easily given away to third parties [11]. To use the social data, the third party applications must use the Social Web platforms and agree to their numerous restrictions on how the data can be used. Although most platforms offer various user-adjustable privacy controls for each application, the ultimate control lies in the hands of the information silo owner. **How can we transfer the full control over the users' data from the information silos hosting that data to the actual users?**

E. Decentralizing the Social Web.

Most of the time in social applications the users either access their own data or the data of their social graph neighbors. The data from the whole social graph does not have to be aggregated and made accessible in a single central location, on the contrary, it is very amenable to distribution due to the localized data access patterns. It might be feasible to completely decentralize the Social Web and move it to the Internet edges, where the amount of available resources has been steadily increasing over the years [15]. **What is the best architecture for the decentralized Social Web? Is the P2P architecture suitable?**

III. The solution: friend-to-friend systems

In this section we are going to suggest a solution to the problem of securing and decentralizing the Social Web. Our solution addresses the issues from the previous section and we inevitably arrive at a new set of problems and open questions, which in our humble opinion constitute an interesting research agenda.

A. The social network becomes the P2P network

Let us assume that there is some social network of people who want to start using a social application. Each of these people has a computer with a browser and an Internet connection. There are no servers. How can we construct the Social Web using only this infrastructure?

The solution that we propose is the *friend-to-friend network (F2F)*. In an F2F network two computers can communicate with one another only if their owners are related in the social network. Since the two owners know one another they can communicate either in person, through an instant messaging service or any other means. This side channel can be used to set up the connections between users' computers, i.e. by exchanging their network addresses and possibly some cryptographic data for securing the link. We claim that the F2F network is a great substrate for building the decentralized Social

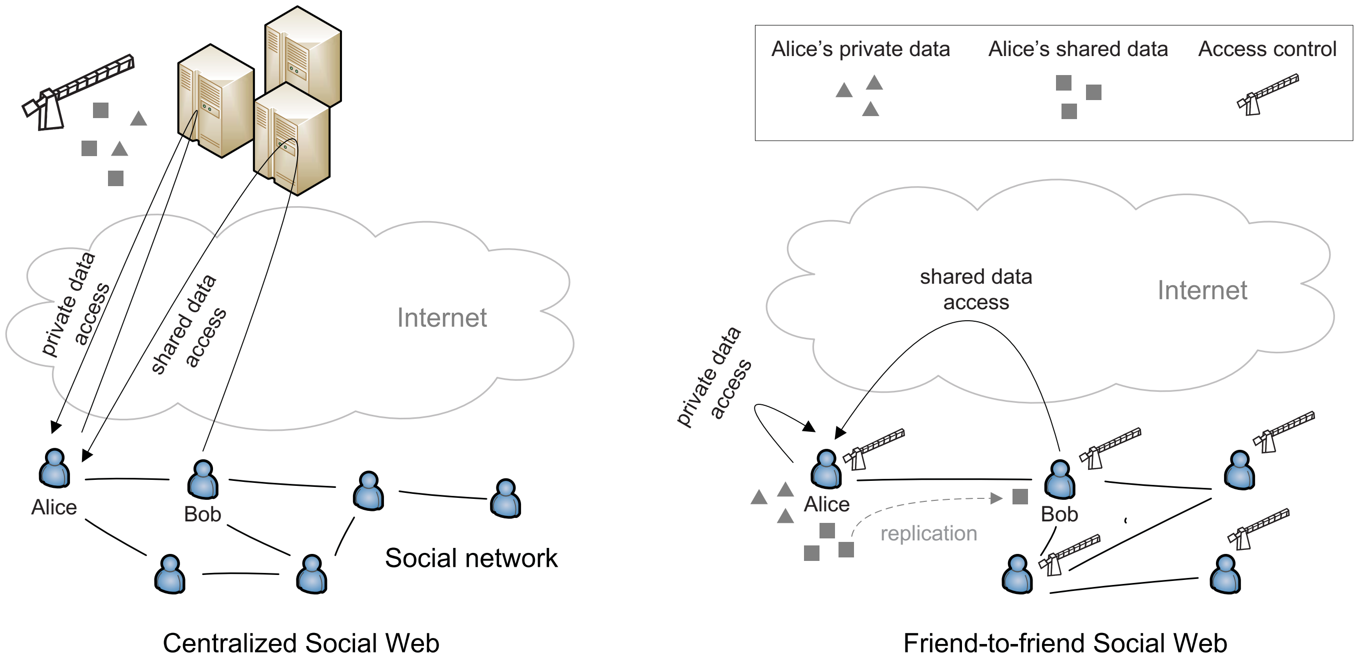

Fig. 1. **The centralized vs. the friend-to-friend Social Web.** Alice wants to access her own private data and share some data with Bob. In the centralized Social Web all the data is stored at the servers. The servers ensure high availability of the data and centrally control the access to it. Both Alice and Bob must make network roundtrips on every data access. In the F2F Social Web the private data stays on the Alice's computer and Alice has full control over the access to it. Alice also controls who can access the shared data and allows only Bob to see it. The shared data is replicated among friends for availability and durability.

Web. **How can we maintain a securely interlinked F2F network under node arrivals and departures and under variable network addresses?**

B. Social trust becomes P2P trust

One of the biggest and largely unsolved problems of P2P systems is security [30]. Because any peer can connect to any other peer this makes the systems extremely vulnerable. Peers can arbitrarily deviate from the P2P protocols, appear in the network under many identities [23] or selfishly use the resources of others while not contributing their own. Even when peer misbehavior is detected the rogue peers need to be isolated from the network, which is extremely hard to do without having a stable peer identity. Open P2P systems are essentially groups of strangers talking with each other, trust is volatile and difficult to build.

In F2F networks the peers have a well defined identity and the users bring the social trust into the system. A user in the system knows that the computers that her computer is communicating with belong to her friends and can be trusted to work as expected. If one of the computers does misbehave and this is detected, the users can be notified about it and undertake the necessary action, e.g. possibly contact the owner of the rogue computer. Strong identity allows for self-policing and user-mediated repair in F2F systems. **How to detect peer misbehavior? How to interact with the users after misbehavior detection?**

C. F2F for data access control

One of the fundamental problems of the Social Web that we have identified in §II is controlling access to the data. Instead of delegating access control to the big centralized information silos as it is done now, the F2F system users can take access control into their own hands (Figure 1). The computer with which the user connects to the F2F network becomes the access controller. Private user data stays private and never leaves the user's computer. The data that the user wants to share with the selected friends can be made available over a direct secure link to the friends' computers. As we mentioned in §II-E the two data access types: private and friend-shared cover the majority of the needs of the social applications.

F2F solves another problem with the centralized Social Web, the attention data leaks. The only parities that are aware of the data access are either the source of the data or the receiver. Since the communication over the F2F links is cryptographically protected, the attention data leaks are greatly limited. **Can we make some formal statements about the potential privacy violations in social applications? What protocols can we use in F2F to enforce data access control and ensure privacy?**

D. F2F for data storage

The main advantages of the existing centralized Social Web are (arguably) high availability and durability of the stored data. The F2F platform should provide a similar or better level of service. Just as it is done in the P2P systems, availability and durability can be achieved by replication [32]. The private user data can be backed up onto her friend's computers. The friend-shared data is by its nature replicated to the friends that are allowed to see it, but can also additionally be replicated to other friends. Replication should of course guarantee confidentiality of the replicated data and use cryptography wherever necessary.

An interesting related problem is how to handle the *mutable* replicated data. Although there are many existing solutions to this problem, most of them assume the full n-to-n connectivity between the replicas [19] while in F2F the replication is happening on a subgraph of a social graph with much less than n-to-n connectivity, which might pose serious challenges to maintaining data consistency. In addition, in F2F there usually is a clearly defined owner of the data, an authoritative source, which can be leveraged when designing the replication schemes. **How can we replicate mutable data in the social graphs? What guarantees on availability, durability, consistency and confidentiality can we make?**

E. F2F as a social application platform

Having a securely interconnected F2F network and a data storage layer built on the top is just a first step towards the complete Social Web platform. As any other platform the Social Web needs an application runtime environment. This is probably one of the biggest open problems. Shifting the Social Web from the servers to the edges means that the application is distributed as well. F2F gives the users more control over their data, the data is less mobile. The data cannot always get to the application code, the application code must get to the data and be executed where the data is.

At first this might seem to be too constraining for the applications. There are two main problems: 1) the application cannot conceal the code execution from the user and 2) there is no data that is completely private to the application. These two points might be false in the case of client-server applications, but are true for the classical desktop applications. In that sense the F2F runtime model would be the same as the desktop model: applications execute outside the control of the application creator giving the user full control over what data is fed into the application. **What is the best application runtime model suitable for the Social Web that does not force the users to give up control over their data?**

IV. Related work

A. Friend-to-friend systems

The concept of friend-to-friend systems is not new, the term has been coined by Dan Bricklin [7] already back in 2000. Since then several efforts have been made to build F2F systems. We unfortunately do not have space to mention all of them here but excellent introductions [6] and surveys [28] are available. Perhaps, the most well known system is Freenet [20] which is a distributed key-based data storage network. It's main design goals are censorship resistance and anonymity. Freenet can operate in the classical F2F (darknet) mode where the users can only join the network when they know someone who can let them in, but also in the opennet mode in which anyone can join. Another, more recent effort, Friendstore [22] looks at how to ensure availability and durability in a social backup system.

In general, F2F systems have received relatively little attention in the research community. Several systems have been built by enthusiasts and private companies, but they focus on specific applications such as file-sharing or messaging. In contrast, in this paper we are calling for research into F2F as a general social application platform which is an entirely novel problem.

B. Peer-to-peer systems

When building the F2F systems we can to a large extent rely on the results from the P2P community. The problems of overlay routing and distributed storage have been well understood and a multitude of systems have been proposed [21], [26], [27], [16], some of them are widely deployed at the edges [26]. However despite many proposed solutions [30], [18], [17] P2P security remains an important open problem. F2F greatly reduces most of the P2P systems vulnerabilities by bringing the social trust into them. This happens at a cost though. The classical P2P systems rely on being able to connect to any peer in the network while in F2F connections are only possible to the friends. It is an interesting open research question whether F2F can be turned into a reliable routing and data storage substrate despite these constraints.

C. Cloud computing

The dominance of the client-server architecture on the Web has led to significant advances in virtualization and consolidation technologies. Some companies have accumulated large amounts of hardware in their data centers and developed software to manage it. Amazon started selling raw computation and storage services thus creating a new Hardware as Service (HaaS) market [14]. The advances in the browser technology and Web 2.0 have brought applications with similar look-and-feel to desktop applications. A trend termed Software as Service (SaaS). These and other concepts are being thrown under one umbrella term of cloud computing [2].

F2F is the antithesis to cloud computing, it decentralizes data access control and places them in the hands of users by which it significantly improves privacy and security. It remains to be seen, however, whether F2F proves to be a feasible alternative to the centralized Social Web. Ultimately, we envision cloud computing and friend-to-friend computing working together to produce the right balance between privacy, user empowerment and the quality of service.

V. Conclusions

In the seventies the dominant computing platform were the mainframes processing batch jobs dispatched from dumb terminals. Ten years after that the personal computer arrived, placing computing power at the fingertips of the individuals, who no longer had to rely on the mainframes. With the growth of the Web and its reliance on the client-server architecture we are repeating the mainframe history again. We have become too dependent on the servers. We need to decentralize and the best place to start this process is the Social Web. The new Social Web operating at the edges of the Internet can potentially be as empowering to the collaborating groups of people as the personal computer has been to the individuals.

References

[1] "Attention economy," http://en.wikipedia.org/wiki/Attention_economy.

[2] "Cloud computing," http://en.wikipedia.org/wiki/Cloud_computing.

[3] "Dataportability.org - share and remix data using open standards," http://www.dataportability.org/.

[4] "Dear web applications: Where are my files?" http://publishing2.com/2008/05/17/dear-web-applications-where-are-my-files/.

[5] "Facebook platform," http://developers.facebook.com/.

[6] "Friend-to-friend," http://en.wikipedia.org/wiki/Friend-to-friend.

[7] "Friend-to-friend networks," http://www.bricklin.com/f2f.htm.

[8] "Hi5 developer center," http://www.hi5networks.com/developer/.

[9] "Myspace developer platform," http://developer.myspace.com.

[10] "The scoble scuffle: Facebook, plaxo at odds over data portability," http://news.cnet.com/8301-13577_3-9839474-36.html.

[11] "The social network wars begin in earnest: Facebook bans google friend connect," http://www.techcrunch.com/2008/05/15/the-social-network-wars-begin-in-earnest-facebook-bans-google-friend-connect/.

[12] "Update: Facebook caves in to beacon criticism," http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9051119.

[13] "Who owns your social data? you do, sort of," http://www.eweek.com/c/a/Enterprise-Applications/Who-Owns-Your-Social-Data-You-Do-Sort-of/.

[14] "Why amazon's haas (hardware as a service) strategy is a winner," http://www.readwriteweb.com/archives/ amazon_haas_hardware_as_a_service.php.

[15] *Broadband Growth and Policies in OECD Countries*. OECD Publishing, 2008.

[16] K. Aberer, P. Cudré-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, and R. Schmidt, "P-grid: a self-organizing structured P2P system," *SIGMOD Record*, vol. 32, no. 3, pp. 29–33, 2003.

[17] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," in *CIKM*. ACM, 2001, pp. 310–317.

[18] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach, "Secure routing for structured peer-to-peer overlay networks," *ACM Operating Systems Review*, vol. 36, no. si, p. 299, 2002.

[19] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *OSDI*, 1999, pp. 173–186.

[20] I. Clarke, S. G. Miller, T. W. Hong, O. Sandberg, and B. Wiley, "Protecting free expression online with freenet," *IEEE Internet Computing*, vol. 6, no. 1, pp. 40–49, 2002.

[21] F. Dabek, J. Li, E. Sit, J. Robertson, M. F. Kaashoek, and R. Morris, "Designing a DHT for low latency and high throughput," in *NSDI'04*. USENIX, 2004, pp. 85–98.

[22] J. L. Dinh Nguyen Tran, Frank Chiang, "Friendstore: cooperative online backup using trusted nodes," in *First International Workshop on Social network systems*, 2008.

[23] Douceur, "The sybil attack," in *International Workshop on Peer-to-Peer Systems (IPTPS), LNCS*, vol. 1, 2002.

[24] R. Gross, A. Acquisti, and I. H. John Heinz, "Information revelation and privacy in online social networks," in *WPES '05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society*. New York, NY, USA: ACM, 2005, pp. 71–80.

[25] H. Jones and J. Soltren, "Facebook: Threats to Privacy," *Project MAC: MIT Project on Mathematics and Computing*, 2005.

[26] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the XOR metric," in *IPTPS*, ser. LNCS, vol. 2429. Springer, 2002, pp. 53–65.

[27] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Computer Com-*

munication Review, vol. 31, Berkeley, CA, USA, 2001, pp. 161–172.

[28] M. Rogers and S. Bhatti, "How to disappear completely: A survey of private peer-to-peer networks," in *SPACE 2007*, 2007.

[29] C. Shirky, *Here Comes Everybody: The Power of Organizing Without Organizations*. Penguin Press, 2008.

[30] Sit and Morris, "Security considerations for peer-to-peer distributed hash tables," in *IPTPS, LNCS*, vol. 1, 2002.

[31] B. Urstadt, "The business of social networks," *MIT Technology Review*, vol. Jul/Aug, 2008.

[32] Weatherspoon and Kubiatowicz, "Erasure coding vs. replication: A quantitative comparison," in *International Workshop on Peer-to-Peer Systems (IPTPS), LNCS*, vol. 1, 2002.