The Good Left Undone: Advances and Challenges in Decentralizing Online Social Networks

David Koll[a,*], Jun Li[b], Xiaoming Fu[a]

[a]*koll@cs.uni-goettingen.de, fu@cs.uni-goettingen.de*
University of Goettingen, Institute of Computer Science
Goldschmidtstr. 7, 37077 Goettingen, Germany
[b]*lijun@cs.uoregon.edu*
University of Oregon, Department of Computer and Information Science
1477 E 13th Ave, Eugene, OR 97403, USA

Abstract

Billions of users are now inter-connected in Online Social Networks (OSNs) and, as they interact with each other, massive amounts of potentially private data are collected at the OSN providers' (e.g., Facebook or Twitter) premises. Unfortunately, provider-initiated privacy violations on this data are frequent and there is little chance that the providers will grant users effective data-protection means. To address these issues and to help users regain the control over their data, decentralized OSNs (DOSNs) have lately been introduced as a competitive paradigm to provider-controlled, centralized OSNs. DOSNs are built to function without the participation of a provider and with the intent to prevent any misuse of private user data. However, all proposed DOSNs still lack widespread adoption. While challenging the market-leading OSNs is difficult for many reasons, in this paper, we set out to understand the *technical* deficiencies behind the absence of a successful DOSN. We focus on the major technical challenge of DOSNs: they need to substitute the datacenter-based infrastructure of centralized OSNs. We first review recent advances in decentralizing OSNs based on how they approach that challenge. In a next step, we analyze the advantages and disadvantages each approach yields, and then derive a series of challenges that a successful DOSN will have to fulfill. Finally, we discuss options of moving forward in designing a new DOSN that could be successful in doing so.

*Corresponding author

Preprint submitted to Computer Communications, vol. 108, August 2017

Keywords: Online social networks, decentralization, survey

1. Introduction

In the past decade online social networks (OSNs) have evolved from small, themed networks into ubiquitous platforms of communication. In July 2014, Facebook, once a small Harvard campus network and now the world's largest OSN, counted one billion interactions related to the FIFA Football Worldcup 2014 [1]. At the same time, Twitter observed 672 million tweets related to the tournament, including over 35 million tweets during a single match [2]. Concurrent to the explosion of content, the number of OSN users has been constantly growing. In 2009, Twitter reported a remarkable 1,400% growth rate [3], and has been continuously growing ever since [4]. By now, 316 million users *tweet*, while Facebook even hosts more than 1.5 billion users [5, 6].

Currently, the main OSN platforms are controlled by single providers in a *centralized* fashion. These providers thus deal with tremendous amounts of user information and can obtain deep insights into the personal interests, social relationships, political opinions, and economical preferences of their users. As an example, Facebook already controls private data of one-fifth of the world population, and is still trying to extend its reach as seen by the multi-billion dollar acquisitions of the Instagram and WhatsApp user bases in 2012 and 2014, respectively [7, 8]. With both deals Facebook obtained photos, phone numbers and messaging data for almost 500 million users, and has since strived to connect this data with its main service to complement the view on the data of already-known users [9]. This data aggregation has raised serious privacy and security concerns, as providers predominantly exploit the data stored at their premises for various purposes that reach far beyond desired improvements of OSN services. Here, potentially private data is, for instance, analyzed for commercial use or sold to commercial partners—with or without notifying users [10–12]. Many providers, including Facebook and Google, have recently granted full access to user data to governmental institutions [13]. Besides, the centralized data repositories can be subject to external misuse as demonstrated by the leakage of millions of passwords from the LinkedIn database in June 2012 [14].

While such practice has led to several class action lawsuits against OSN providers [15], the providers' perspective towards user data privacy has not changed [16, 17]. There is little to no activity by providers to fundamentally

address or even incrementally improve the situation of user privacy. In fact, a fix might not be much of a technical challenge; the providers could encrypt user data and let users decide with whom they want to share what parts of their data. The inertia of providers, however, is more than understandable from an economic perspective, since it is not an option for them to forgo the opportunity of analyzing user data, a main source of their income [18, 19].

To mitigate these issues, researchers and practitioners from academia and industry have introduced the concept of decentralized OSNs (DOSNs). The main idea of a DOSN is to build an OSN without any participation from a central provider, and thus to enable better user data privacy and to reduce the risk of large-scale data leakage [20]. Additionally, due to their distributed architecture, DOSNs can reflect the peer-to-peer nature of online social networking better [11].

A plethora of DOSN solutions that follow many different DOSN construction principles has been proposed recently [21–45]. Yet, all solutions have one common problem: they do not attract a significant user base. This situation can partly be explained by non-technical reasons. That is, OSN providers benefit from the uneven match-up between multi-billion dollar corporations on one side and small research teams on the other. Further, they have already acquired a critical mass of users and offer a wide range of functionality to these users, which is difficult to achieve for DOSN prototypes.

However, technical shortcomings (e.g., slow response time to user requests) could be contributing significantly to the lack of success of DOSNs. Therefore, before producing yet another DOSN that suffers the same fate as the state-of-the-art, we believe a comprehensive study of existing DOSNs is necessary. Specifically, in this paper we survey recent advances in decentralizing OSNs with respect to *the* major challenge of constructing a DOSN: as DOSNs need to function without the support of a central provider, they have to substitute the datacenter-based infrastructure of that provider. With regards to that challenge, we ask the following main questions:

- How do state-of-the-art DOSN solutions build these infrastructure substrates? Can we extract a number of common design choices?
- Are these design choices contributing to their limited success, or why have we not seen a successful DOSN yet?
- Can we learn from the deficiencies of present solutions for designing a future DOSN?

In answering these questions our main contributions are:

- We present a comprehensive taxonomy of over 20 DOSN approaches to substitute datacenter-based infrastructures, with a focus on finding conceptual similarities and common design choices among these approaches (Section 3).
- Based on our taxonomy, we analyze existing DOSN design choices in detail, with a particular attention to the technical reasons why current DOSN approaches have not yet attracted a significant user base (Section 4).
- Our analysis reveals a number of criteria that a full-fledged infrastructure substrate should fulfill. We discuss how state-of-the-art design choices hold up against these criteria and use our findings to sketch possible options to move forward for DOSN research (Section 5).

The remainder of this paper is organized as follows: We first motivate and introduce the concept of DOSNs in Section 2. We then present a taxonomy of existing DOSN solutions in Section 3, assess these solutions in Section 4, and discuss our DOSN design criteria and their implications in Section 5. Finally, we summarize related work on surveying DOSNs in Section 6 before we conclude this paper in Section 7.

2. An Introduction to Decentralized OSNs

Before we dive into the analysis of decentralized OSNs, we introduce the motivation behind DOSNs and briefly describing their characteristics in this section.

2.1. Issues with Centralized OSNs

The remarkable growth of OSNs has inherently led to tremendous amounts of user information being part of these networks. At the same time, this information is collected and maintained by a single instance, which we call the *provider* of the OSNs (e.g., Facebook, Google, or Yahoo). While analyzing user and usage data can improve the OSN service itself, this situation has raised severe privacy and security concerns [10, 11].

First and foremost, the control over huge amounts of user data without restriction of any kind is worrisome itself. Here, providers can obtain a

deep insight into their users' personal interests, opinions, social relationships, and economical or political preferences. For instance, recent lawsuits against Facebook and other OSN providers (e.g., Google and Yahoo) complain about the practice of tapping into the users' private messages for the purpose of content analysis [46, 47]. Moreover, both Facebook and Google have introduced a clear-name policy, which makes the use of real names as user names mandatory; not following the directive will result in an exclusion of the user from the OSN [48].

The providers have good reasons for their actions: Facebook is currently creating 96% of its annual income from personalized advertisements [6]. These advertisements can be customized better—and therefore sold with greater revenue—if precise user profiles are available. By extracting interests or product preferences from user data such as messages, the profile precision can be increased; the clear-name policy further eases the linking of existing OSN profiles with all sorts of information available elsewhere on the internet [49].

The profiling of users does not stop at the boundaries of the OSN providers. Between 2007 and 2009, Facebook and a group of partners (among them, e.g., Amazon, eBay and Sony [50]) used the Beacon application to share sensitive shopping information of users among the partners without the users' consent [12]. Additionally, a large group of major OSN providers—including Facebook, Google and Yahoo—granted full access to user data to government agencies within the PRISM program without any knowledge of their users [13]. Extended cooperation or collaboration of providers with government institutions could thus ultimately result in the *transparent user*, where all available information about each single user is available in a bundle at a single instance, without the user's knowledge and thus also without her consent.

Finally, the providers do not only endanger user privacy. The terms of use are often difficult to process and at the same time invalidate property rights [51]. For instance, for every photo uploaded to Facebook, the user grants a simple usage right for that picture to the company, and the photo will remain on Facebook indefinitely (see Section 2.1 of Facebook's Statement of Rights and Responsibilities [52]). Additionally, the central provider might at some point introduce a usage fee to a previously free-of-charge service. Users would then face the ostensible choice to either lose their social network or to pay the fee to continue using the service. In the worst case, an OSN provider might even shut down its service completely [53, 54]

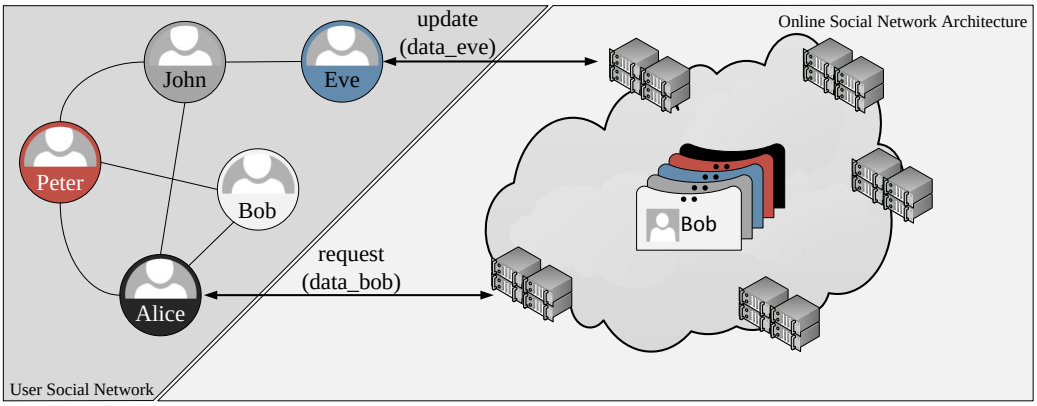

Figure 1: An exemplary centralized OSN. User data is stored across multiple inter-connected datacenters, which are controlled by a central entity. Encryption for user data is non-existent.

Thus, there exists an obvious and urgent need for both increased privacy and long-term guarantees with regards to, e.g., property rights, data availability or terms of use in OSNs. However, OSN providers are unlikely to act in the favor of their users [16]. This would require them to forfeit access to user data, and such a concession would be tantamount to giving up a number of economical advantages, including (i) the opportunity to analyze the data for personalized advertisement; (ii) the possibility to link external publicly available information with the OSN profiles of their users; (iii) the option to exchange data with other providers to complete their own view on the data; and (iv) usage rights on the content. Further, binding themselves to long-term guarantees would reduce the economic flexibility of OSN providers.

2.2. Towards Decentralized Online Social Networks

As a consequence of this dilemma, researchers have in the last decade suggested to *decentralize* OSNs. The organization of a centralized OSN usually comprises several inter-connected datacenters to host unencrypted user data, as shown in Figure 1. As one example, Google currently operates thirteen different datacenters across the globe [55]. Although the infrastructure might thus be complex itself, connecting a user to the data is relatively easy in a traditional client-server fashion. In Figure 1, Eve can update her data directly at the corresponding datacenter(s) (e.g., post a status update), and Alice can request Bob's data (e.g., Bob's latest vacation pictures) from there

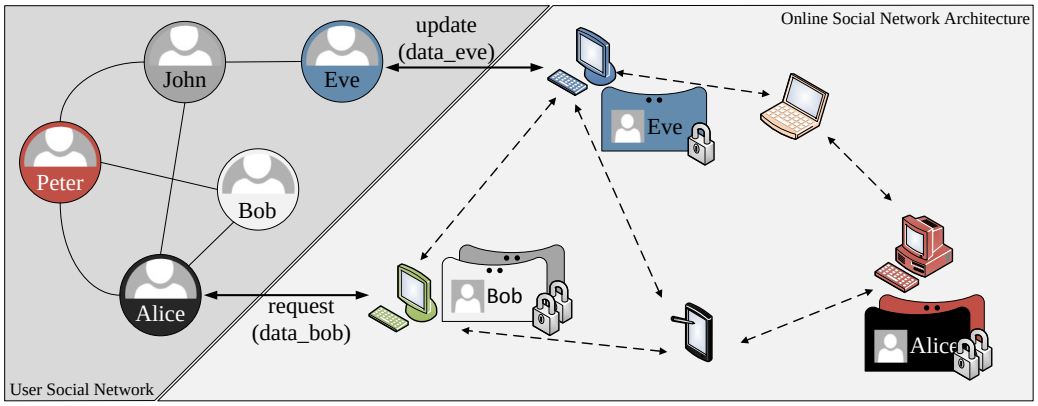

Figure 2: An exemplary *de*centralized OSN. The central provider is removed and encrypted user data is stored among the users themselves.

as well. In fact, in a centralized OSN, the massive datacenter-based backend of the central provider deals with all the technical challenges of social networking, such as storing the massive amounts of user data, efficient content retrieval, averting malicious attacks, and so on.

A decentralized OSN, on the other hand, has to construct a substrate for that infrastructure itself. The way of constructing this substrate is a key characteristic of a DOSN, and may greatly differ from one solution to another.

As one example (we will discuss alternative approaches from Section 3 onwards), Figure 2 shows a decentralized OSN, in which the users themselves absorb the absence of the central provider by storing data across the participants' devices.

However, such decentralization is technically more challenging than following the classical client-server model in a centralized OSN. For instance, connecting a user to data is challenging in DOSNs. As there exists no central user data directory, users first need to locate data of interest in the network before they can request it. The location at which the data is hosted might further be a node that was previously unknown to the requesting user. As an example, in Figure 2, Alice requests Bob's data from a user she does not necessarily know. At the same time, Eve can update her data at her own machine, but has to somehow communicate the information about the update to other users.

Although removing the central provider is thus technically challenging, DOSNs can help users in regaining control over their data. Users are able to encrypt their data such that it is only accessible by eligible users. In Figure 2, Bob's data is encrypted, and Alice can only access it if she owns the proper decryption credentials. In particular, the storage provider itself is not able to analyze any data (unless access is granted by the user).

Further, a decentralized approach is also a natural fit for online social networking for several reasons. First, the communication paths in an OSN do not require the presence of a provider. Although the provider of current centralized OSNs offers the infrastructure to distribute and access the content, it does not significantly contribute to the content itself. In fact, online social networking is rather peer-to-peer in nature, as the users of an OSN generate content for other users (e.g., their friends or subscribers) and not for any third party like a provider. Second, a decentralized system without a single, central data repository also limits the risk of large-scale privacy breaches [11]. Since such a system would rely on a multitude of distributed storage repositories for user data, a breach at a single one of them would not expose all user data to the attacker. Third, DOSNs—if following a P2P paradigm as shown in Figure 2—can inherently provide good scaling characteristics [11]. Fourth and finally, DOSNs can—without decreasing the users' privacy—remove one of the big barriers of today's OSNs, the non-existing interoperability between several OSN applications [20]. Currently, a user needs one account for each OSN she is a member of (e.g., Facebook, Twitter, Flickr, etc.), and additionally inevitably shares her data redundantly with many providers at the same time. Worse, if there were a single-sign-on for all OSNs the user is registered with, the provider of this service could connect all data of users from several OSNs. A generically designed DOSN, however, can allow users to accurately control a single set of encrypted basic user data, and to reduce the hassle of handling multiple accounts, while still granting controlled access to a multitude of OSN applications on a fine-grained basis. Here, the main idea is to maintain reusable data (such as login credentials or basic personal data) within a generic DOSN middleware, and let each social application running on top of that middleware access this data if enabled by the user [41]. By allowing applications to separately store and manage application-specific data depending on the purpose of the respective application, this approach can both realize single-sign-on for users and remain open for extensions for realizing different social networking applications.

3. A Taxonomy of Decentralized OSNs

As DOSNs strive to remove the central provider, they also have to deal with the consequence that they can not rely on any sort of provider infrastructure in their design. All efforts to decentralize DOSNs thus pivot around one fundamental question: *how can that infrastructure be substituted in a decentralized fashion?* In this work, we provide both a taxonomy and an analysis of 22 existing DOSN solutions [21–45] with respect to how each of them builds the infrastructure substrate and what the consequences of the design choices behind each system are.

In general, building the infrastructure substrate can be broken down to one predominant problem: any DOSN needs a way to store the massive amounts of user data in a distributed fashion, and it needs to make sure that this data is available for retrieval. Upon establishing such a storage infrastructure, communication among users can be realized by writing data (e.g., a message) to storage or by direct communication among users.

To structure our analysis, in this section we thus first abstract each solution to one of three more generic decentralization approaches as follows, before we analyze each of these approaches in detail in Section 4.

- The first generic approach to decentralization involves systems that exploit permanently available resources to replace the provider infrastructure. They are built upon the assumption that every user has access to a device that can be online constantly—usually a web-server or a machine in the cloud—to store her data on and to be directly accessed by parties interested in the data. We call these systems *server-based solutions*.

- By relaxing the above assumption, the second category includes approaches that work without such resources; they instead expect a cooperation of users (usually in a P2P fashion). Here, resources for the DOSN infrastructure are solely provided by end-user devices such as laptops or even smartphones. In addition to a more dynamic data storage infrastructure, these approaches also yield additional challenges in retrieving data from changing locations. We call these approaches *cooperation-based solutions*.

- The third class of DOSN solutions consists of systems that contain elements of both categories above. Such systems typically require both

Year	2008	2009	2010	2011	2012	2013	2014	2015	2016
Server-based		Anderson et al. Persona Vis-a-Vis	Diaspora PrPl	Contrail				POSN	
Cooperation	Friendstore	Safebook Peer-SoN	Prometheus	LifeSocial.KOM GEMSTONE	DECENT Cachet	MyZone	ProofBook SOUP	DiDuSoNet	
Hybrid				Confidant	SuperNova				Lilliput

Table 1: A chronologically ordered categorization of DOSN approaches.

permanently available resources *and* the cooperation of users. We call these approaches *hybrid solutions*.

Table 1 shows a chronologically ordered classification of existing DOSN into these categories. It is striking that most (more than two thirds) DOSNs do not assume permanent resources only, but build on some sort of cooperation between users instead (cooperation and hybrid solutions). Below we abstract each DOSN solution into one of these categories and describe their major working principles. Note that to achieve our goal of revealing conceptual deficiencies across all DOSN solutions, we refrain from discussing implementation details on, e.g., data representation, communication protocols or encryption methodologies.

3.1. Server-based Solutions

We begin our discussion with *server-based solutions*. These solutions typically assume that each user has a (web-)server—which can also be hosted in a cloud environment—available to store and retrieve her data [22–24, 26, 27, 29], as shown in Figure 3. The approaches differ in that the servers can either be operated by users themselves, contributed in an altruistic fashion, or supplied by commercial providers.

3.1.1. Server Storage

Anderson et al. [21]: In [21] Anderson et al. propose one of the first approaches towards privacy-enabled social networking. The system strongly focuses on protecting a user's social information from both the provider and other network users via encryption techniques. Here, user data is stored at an unspecified *untrusted server* (e.g., the provider) that has the task to make data available, while the data itself is encrypted and can only be retrieved by users with the appropriate keys. In this context of encrypted data, primitives for message exchanging are proposed as well.

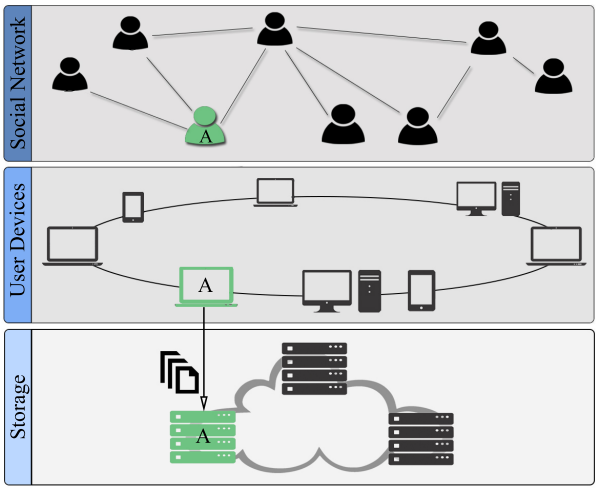

Figure 3: A DOSN can be split into three different layers: the social network between the users (top layer), those users' devices (middle layer), and the infrastructure upon which the DOSN is built (bottom layer). In this example, a server-based solution is shown. Upon creating a new data item, user A instructs her dedicated server to serve that data to requests of others.

Persona [22]: In Persona, the approach is to ask each user to provide a permanently available storage location (e.g., a personal web server) for their own, securely encrypted data. A user can publish references to her data with the help of a storage service running at that location. This service is also responsible for reacting to requests from other users; for that, it offers *get* and *put* operations. Eligibility to modify any data (with *put*) is determined by Access Control Lists (ACL), while the data itself is encrypted by combining Attribute Based Encryption (ABE) [56] with traditional cryptography operations. ABE efficiently allows users to define fine-grained access rules for their data, and subsequently to Persona, several other DOSNs have been employing the technique to improve user data privacy (for details on ABE and its DOSN applications see, e.g., [22, 42, 56]).

Diaspora [23]: Diaspora follows a similar path. It not only allows each user to set up her own server (called *pod* in Diaspora), but also relies on a limited number of altruistically provided servers. Hence, as an alternative to setting up her own server, a user can also select one of the altruistically provided servers to store her data on. Diaspora itself then offers a server-

overarching search function to find other users and to retrieve their data. Currently, Diaspora does not offer data encryption.

3.1.2. Commercial Cloud Storage

With the advent of cloud computing, the focus of DOSN researchers switched from user-provided storage to exploiting existing infrastructures at commercial cloud providers. Here, the general idea is to store and retrieve user data with the help of cloud services rented by the users.

PrPl [24]: PrPl is designed for both user-provided servers and rented cloud machines. On either of these a user can run a *butler* service—a personal service that serves and organizes a user's data based on her preferences on access control. The butler also maintains an index to which eligible users can then issue queries for data. A key concept of PrPl is that the butler also keeps references to possibly several storage locations (e.g., free Dropbox space), at each of which the user can manage access to the local data by the use of OpenID authentication [57]. Here, PrPl has reached query response times from these storage locations of approximately one second for data retrieval.

Vis-à-Vis [25, 26]: Vis-à-Vis gets rid of altruistically- or user-provided servers, and users exclusively store their private data on cloud services like Amazon EC2 or Microsoft Azure. Each user operates an *independent server*—a VM running in the cloud that hosts the user's data—and all independent servers in turn form the network. Content is then shared within groups of user servers, for each of which membership and group communication is administered by the user who created that group. The implication is that, while user data is stored at a provider's premises, communication is implemented on the user side. Due to the use of cloud VMs, searching information is efficient in Vis-à-Vis—typically, search requests are handled within approximately 500ms.

Contrail [27, 28]: Contrail builds on a cloud storage substrate as well, and additionally relies on the cloud to act as a relay for messages between (mobile) users. Here, in contrast to Vis-à-Vis, users do not interact directly between each other, but send and receive messages from so called *cloud relays* in a publish-subscribe fashion. As a consequence, cloud providers are more involved in the DOSN. Contrail also provides measurements on the latency expected from querying mobile devices with a 3G connection. This latency is measured to be approximately 500ms.

POSN [29]: As a more recent approach, POSN is similar to Vis-à-Vis. User data is stored exclusively on cloud services, while users mainly interact

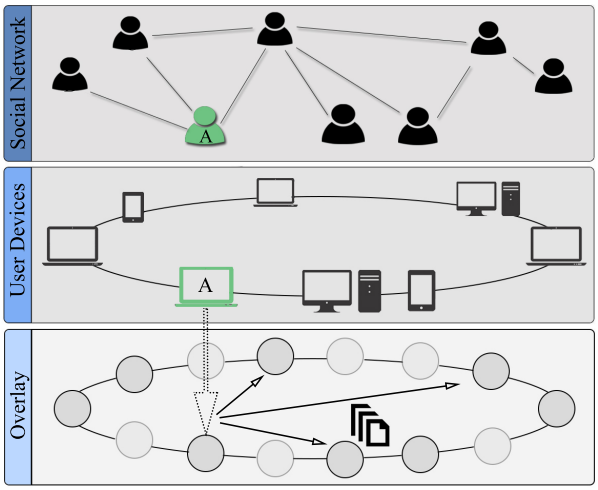

Figure 4: A decentralized OSN that exploits user cooperation replaces the cloud infrastructure with a distributed overlay (bottom layer). In approaches that store user data within that overlay, user A's data *itself* will be stored among overlay nodes according to pre-defined overlay algorithms.

with each other via mobile devices. The key idea is that the cloud services used here are free of charge (e.g., basic Dropbox service), and that data is stored on these services in encrypted form, which makes it difficult for a provider of a cloud service to gain ownership of the data.

3.2. Cooperation-based Solutions

The second category of approaches is the most popular direction among researchers and lets users cooperate and provide temporarily available storage to each other [30–42]. The key difference to server-based solutions is that user-cooperation approaches expect DOSN users to contribute their own devices' resources to build a substrate for the centralized infrastructure and to further reduce the involvement of central elements. A well-designed cooperation-based DOSN can thus enable social networking in which neither a central nor a cloud provider can control a significant amount of data.

However, both storage and retrieval of data are more difficult with this approach, since user devices are usually not as reliable and robust as servers, and can not reliably act as a storage location or host a search index.

In these schemes, the *communication infrastructure* is usually created based on a distributed overlay (e.g., a Distributed Hash Table (DHT)) that

Subclass	Role of Overlay	Storage Procedure
Overlay storage	Searchable data index, direct storage	Handled by overlay internally (e.g., DHT replication)
Friend storage	Searchable data index, pointers to storage locations	Dedicated algorithms for storage at trusted (i.e., friend) nodes
Best-effort storage	Searchable data index, pointers to storage locations	Dedicated algorithms for storage at most capable (possibly untrusted) nodes

Table 2: A subclassification of cooperation-based DOSN approaches. In all subclasses, a distributed overlay acts as a searchable data index.

enables users to search for others (and their data), to initiate communication and to retrieve user data locations. While user communication is thus based on the same idea for many cooperation schemes, they differ greatly in the way they realize the *storage infrastructure*. A consequence of building on unreliable devices is that cooperation approaches usually require replication of user data across multiple locations in the DOSN. In the following, we thus subclassify cooperation-based solutions by the way this replication is approached, as indicated in Table 2.

3.2.1. Data Storage in Distributed Overlay

One solution to create the storage surrogate is to exploit the distributed communication overlay as a data storage repository as shown in Figure 4. The reasoning for doing so is that DHTs often provide integrated replication mechanisms to enable a high availability of data (e.g., PAST [58], in combination with Pastry [59]).

Prometheus [34]: In Prometheus every user operates so-called *social sensors*, applications that collect social data on behalf of the user. Based on the data collected by these sensors, Prometheus then proposes two concepts—(i) to store the collected data in a Pastry DHT (FreePastry [60]), while replicating the data in the DHT with PAST; and (ii) to store a social sub-graph among a user's trusted peers. This sub-graph is a meta-data structure that is generated based on the sensor data and that includes information about, e.g., the intensity of a relation between two users. The goal of Prometheus is to use this data to allow innovation in terms of running social inference queries against this structure. Still, from the system perspective, the OSN data itself is stored and replicated in the distributed overlay (case (i)).

LifeSocial.KOM [35]: LifeSocial.KOM follows the same rationale for data storage and replication (i.e., it uses a combination of Pastry and PAST). To let users communicate with each other, the DHT's routing mechanisms are exploited, and all messages are encrypted by using the principles of public-key cryptography. Social applications (e.g., messaging among users) are implemented as modular plugins on top of this infrastructure.

DECENT [36]: DECENT, similar to LifeSocial.KOM and Prometheus, also stores and replicates data within a DHT and trusts the DHT to handle availability and replication. DECENT realizes that this goes along with storing data on possibly untrusted nodes and thus strongly focuses on security properties. In particular, user data is stored in the form of *objects*, for which DECENT defines protection policies, based on a modified ABE cryptography scheme, that ensure the confidentiality and integrity of the objects.

Cachet [37]: Cachet, as a follow-up work on DECENT, also replicates the data of users as objects within a DHT. Motivated by the high overhead of using a DHT as sole source of information—an application like the Facebook newsfeed requires hundreds of DHT lookups—the key novelty in Cachet is a gossip-based social caching algorithm that pushes updates issued by the data owner to eligible social contacts. These contacts can then hold decrypted versions of the data for other eligible users. As a consequence, Cachet is able to reduce the completion time for both collecting data (by reducing the number of DHT lookups) and cryptographic operations (by reducing the number of operations required in a chain of eligible users) by up to one order of magnitude.

3.2.2. Friend Storage

Another alternative approach in cooperation schemes is to decouple the storage infrastructure from the communication infrastructure. Instead of storing user data *itself* in the overlay (e.g., as seen above in a DHT), many solutions allow their users to store their data at predetermined other OSN nodes. In this case, the communication infrastructure usually only holds *a pointer* to the data location rather than the data itself.

One particularly popular approach is to store data at trusted friend nodes as shown in Figure 5. This ensures that the replication is not carried out by untrusted OSN participants, who might misuse the data or track users accessing it.

Friendstore [30]: Friendstore, an online backup system for user data in which users store data at their friends' machines, was the earliest approach

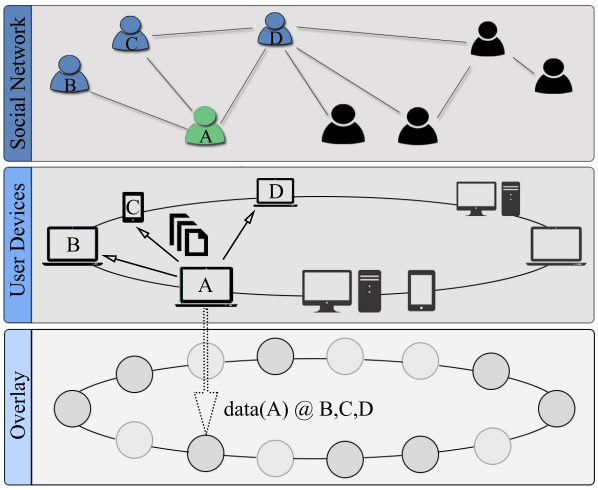

Figure 5: Different from overlay storage, friend storage assigns user data to devices of a user's friends. Here, user A trusts users B, C and D and thus stores her data at those users' premises. In the overlay, only a pointer (A's data can be found at B,C and D) is stored.

towards user-cooperation schemes. More of a backup-system than a DOSN, the idea of replicating data among trusted friend nodes to achieve reliability is the foundation for a variety of DOSNs. Specifically, in Friendstore befriended users enter *offline* contracts to store data for each other before carrying this relationship online. By maintaining a sufficient amount of these contracts, a user can achieve high availability for her data.

PeerSoN [31, 32]: PeerSoN follows a similar, but more tailored-to-DOSN approach. Here, an optimized node selection algorithm is used to form mutual storage agreements. The key difference is that the mutual agreements are formed online and automatically between nodes with similar online times in a tit-for-tat manner. That is, nodes with high online times will match with other high performance nodes, and low performance nodes will team up with other nodes with little online time. To reduce the overhead of maintaining possibly many replicas of user data, the number of agreements a user can enter is limited.

Safebook [33]: Safebook breaks up the concept of storage contracts and more generally tries to mirror a user's data to that user's friends' devices. In contrast to PeerSoN, a user is not restricted in the number of replicas she

distributes to her friends. Safebook also tries to achieve anonymity. That is, a mirror holding some friend's data should not be able to see who is accessing the data in order to prevent tracking access to data. Therefore, Safebook makes user data only accessible through a path of so-called shells. The mirrors themselves form the innermost shell, friends of mirrors form the second shell, and so on. Retrieving a node's data requires traversing the shells along the path of nodes that befriend each other. As a consequence a mirror node will only see requests from friend nodes—it can not know the origin of the request for the data beyond the first shell.

ProofBook [38]: Proofbook follows up on Safebook as it strives for anonymity of users by hiding the source of a data request as well. Here, a user's data is stored in a container structure, which in turn is stored among the user's friend nodes. One key difference to previous works is that Proofbook splits user data into redundant data blocks to ensure that the user data is available even if some parts of it (i.e., some mirrors) are not. Also, ProofBook follows a different path for replication, as it tries to distribute a user's data to *all* (instead of a subset) of her friends, who can then store that data and only need to request updates to it if required. Alongside these features, ProofBook introduces some protections against DoS attacks by introducing a cost to update requests; an incentive mechanism defines if and how users forward these requests to their destination.

MyZone [39]: In MyZone, different from other solutions, a user's data is mirrored (i) on a subset of the user's friends but also (ii) on the (possibly many) devices that the user herself owns. MyZone also tries to combat some malicious behavior in DOSNs. Here, the solution is to essentially limit communication to happen among befriended nodes.

DiDuSoNet [40]: DiDuSoNet is the most recent approach to friend storage and tries—similar to PeerSoN, but orthogonal to Safebook and ProofBook—to minimize the number of replicas each user has to distribute. The main idea is to only have two replicas of each user's data in the network. As soon as one replica becomes unavailable (graceful departure of a node from the overlay or due to node failure), the remaining mirror (called Point-of-Storage, PoS) will elect a new trusted friend node to hold that replica instead. This approach is more dynamic than, e.g., PeerSoN and is capable to adjust to single node failures.

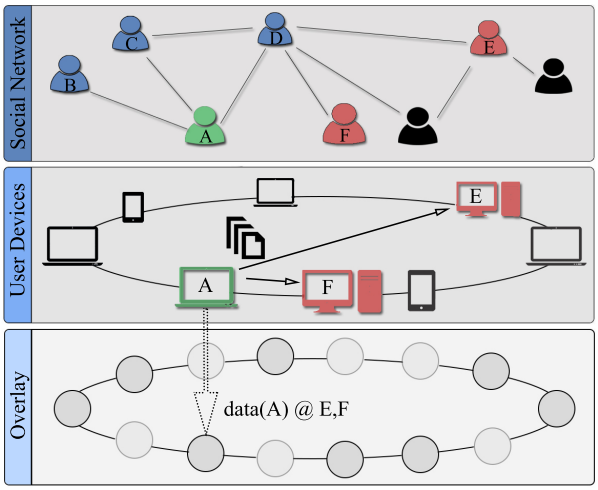

Figure 6: In contrast to friend storage, best effort storage allows a user A to distribute her data to arbitrary nodes in the network. Here, users E and F have more powerful devices than users B, C and D. A thus decides to store her data there. In the overlay, like in friend storage solutions, only a pointer to the data is stored.

3.2.3. Best-Effort Storage

Going one step further, a few approaches try to follow a best-effort strategy for data storage as shown in Figure 6. Here, instead of using trusted nodes as storage locations only, *all* nodes in the DOSN are considered as candidates for replicating user data, with the goal of providing a minimal but sufficient number of data replicas for each user. That is, at least one of the replicas should be available for requests by other users at all times, but the overall number of replicas in the system should be kept as small as possible. The protection of user data is handled by encryption and access control procedures, and the communication infrastructure is also based on a distributed overlay and thus remains similar to solutions exploiting friend- or overlay-storage.

GEMSTONE [41]: GEMSTONE uses the Pastry DHT as a communication infrastructure. Like friend-storage solutions, the DHT only holds pointers to the data. Unlike friend-storage solutions, however, GEMSTONE tries to find the best nodes to store a user's data out of the set of all nodes known to that user. The key difference is that this set can contain untrusted

nodes as well. In GEMSTONE, the main idea is that a user ranks other nodes based on several criteria (e.g., an observed online time pattern) and subsequently selects the highest-ranked nodes of these to replicate her data until each user's data is available with very high probability. To prevent access to user data by the untrusted nodes ABE is employed.

SOUP [42]: SOUP follows GEMSTONE in the basic rationale but gathers additional information that helps in selecting appropriate replica holders. In essence, while GEMSTONE used the users' own observations to create a ranking of nodes, SOUP instead relied on more accurate experiences of trusted friend nodes. Here, the idea is that a user's data is most commonly accessed by her friends, which should thus be able to measure the quality of the user's mirror nodes. Furthermore, one of the main contributions of SOUP is that the set of mirror nodes can be dynamic, and SOUP is evaluated to be both adaptive to DOSN dynamics (e.g., churn) and resilient against some kinds of malicious attacks.

3.3. Hybrid Solutions

Finally, Some DOSN solutions try to combine elements from both above categories. That is, they distribute some functionality to be handled by user cooperation, while other services are provided by resources like cloud platforms or web servers.

Confidant [43]: Confidant lets users cooperate to provide each other with storage space for their data, while the name resolution service for the data is built by using cloud services. That is, the users take over the task of the cloud in server-based solutions, while a cloud service takes over the task of the DHT in user-cooperation solutions. Confidant requires a lower level of cloud involvement than, for instance, Vis-à-Vis, as the cloud service only runs the name resolution service for data that is otherwise stored on user machines.

SuperNova [44]: SuperNova relaxes the dependency on dedicated servers, but builds on the existence of so called super peers, i.e., nodes with increased resources. These super peers act as a replacement for the dedicated servers, and thus have to implement a variety of functionality to be used by the remaining users, including maintenance of the directory of the network and acting as a storage node for others. At the same time, the regular nodes in the network cooperate by acting as *storekeepers* for each other, i.e., to mirror each other's data.

Lilliput [45]: In Lilliput, as the most recent hybrid approach, different types of data are stored in different ways. While smaller and more recent information is held available by OSN participants in a cooperative P2P fashion, bulk data such as images and videos are stored on servers. For the former, users form small, aggressively self-maintained overlays as a storage substrate. These overlays are dynamic and add additional mirrors for a user's data as required (e.g., on departure of a member of the overlay). For the latter, a user—depending on her own privacy needs—can choose among different options to store her bulk data (e.g., Dropbox or a self-maintained server). The rationale behind this approach is to provide a lightweight, privacy-preserving storage option for more timely and relevant data that can also be provided by less resourceful devices such as smartphones. At the same time, less requested and bulkier data is outsourced to permanently available resources.

Table 3 summarizes the DOSN approaches with regards to our taxonomy. We observe that among the three major classes of DOSNs there exist five ways of realizing the storage substrate. A DOSN can be realized by storing data i) on personal web servers, ii) on cloud services, iii) within a distributed overlay, iv) at friends in the DOSN, or v) on user resources in a best-effort way. Hybrid approaches use a combination of these options. Approaches that follow a common principle usually only differ in details. For instance, solutions that use a distributed overlay for data storage all employ a DHT for this task, while they differ in achieving several side-goals ranging from encryption to performance increases.

With regards to the communication infrastructure, almost all solutions—including several server-based and hybrid solutions—prefer communication between users to be direct instead of using a server-based or cloud proxy. In Section 2 we argued that this is a more natural fit for communication in OSNs, and most approaches naturally follow this philosophy. There are few exceptions (e.g., in Contrail), where cloud services are used to relay messages between users.

Interestingly, hybrid solutions do generally implement a cooperation-based data storage, while servers or cloud services in these solutions take care of, e.g., name resolution. That is, they do not exploit the option of storing user data on permanently available devices. One reason for this trend is that cloud providers are not treated as trusted entities in these solutions, an issue that we will discuss in more detail in the following.

Class	Subclass	DOSN	Key Ideas for Infrastructure Substrate	
Server-based	Personal servers	Anderson et al.	Storing encrypted data on untrusted (provider) server	
		Persona	One personal web server per user used for storage and retrieval	
		Diaspora	Data stored on distributed pods; search function over all pods	
	Cloud services	PrPl	Butler service providing a searchable data index running on server or cloud instance; possibly multiple storage instances per user	
		Vis-a-Vis	Exclusive storage on cloud VMs as independent server; direct communication among users	
		Contrail	Storage on cloud service; cloud also relays user communication	
		POSN	Storage on no-cost services; communication between mobile devices	
Cooperation	Overlay storage	Prometheus	Data collected by social sensors; data replicated in DHT	
		LifeSocial.KOM	Data replicated in DHT; encrypted communication between users	
		DECENT	Data replicated in DHT	
		Cachet	Data replicated in DHT; gossip-based caching to push data to social contacts to reduce overhead	
	Friend storage	Friendstore	Mutual storage contracts among users (agreed upon offline)	
		PeerSoN	Mutual storage contracts among users (agreed upon online)	
		Safebook	Mirror data to all friends; anonymity through need to traverse path of nodes to retrieve data; direct communication among users	
		Proofbook	Data stored at friends in redundant data blocks	
		MyZone	Data stored at friends and replication to multiple user devices	
		DiDuSoNet	Dynamic storage at two possibly changing friend nodes	
	Best-effort storage	Gemstone	Select replication location based on ranking mirror candidates	
		SOUP	Select replication location dynamically based on mirror candidates recommended by friend nodes	
Hybrid			**Server or Cloud Task**	**Cooperation Task**
		Confidant	Name Resolution	Storage of user data
		SuperNova	Super peers with increased; resources handle network operation	Storage of user data
		Lilliput	Bulk data storage	Communication overlay; storage for leightweight and more recent data

Table 3: A summary of DOSN approaches.

4. A Qualitative Analysis of Decentralized OSNs

In the previous section, our taxonomy revealed three major categories to achieve decentralization in OSNs, and that all suggested solutions follow a handful of design principles to replace the infrastructures of centralized OSNs. However, none of these solutions have actually had a perceivable impact on the prevalence of centralized OSNs. While a wide range of reasons contributes to this dilemma (as previously discussed), technical shortcomings could be another major reason for the limited success of DOSNs.

Hence, based on our taxonomy, in this section we will dissect and compare state-of-the-art DOSN solutions to detect the technical drawbacks that hinder users in recognizing DOSNs as a serious alternative to their centralized counterpart. Here, our focus is not to describe and find fault with implementation details, but rather to point out the deficiencies we detect on the conceptual level, i.e., as a direct consequence of the main design choices of each DOSN. For that purpose, we again follow the classification of our taxonomy.

4.1. Server-based Solutions

4.1.1. Advantages

The major advantage of server-based solutions is that they can efficiently replace the centralized storage infrastructure of a single OSN provider with distributed but still reliable storage and communication facilities. The key benefit is that such DOSNs—in contrast to P2P-solutions—are inherently successful in making user data highly available and at the same time do not require any replication of user data. Additionally, the use of servers usually offers a better performance and less synchronization effort than interconnected, more distributed user devices.

4.1.2. Drawbacks

Even though resourceful solutions thus usually do a good job in technically replacing the centralized infrastructure, they suffer from a number of drawbacks that render their widespread acceptance unlikely.

First, in case of exploiting user-provided storage, requiring each user to set up their own server is impractical. For instance, in Diaspora, to run her own server, a user would have to be able to first set up the server physically and then to install Ruby, SQLite3, OpenSSL, and several other libraries required to run a Diaspora server [61]. Similarly, in Persona, each user would have

to set up a webserver. Since this is usually too much effort required from a typical OSN user, most users have to rely on the altruistically provided servers.

However, altruistic provisioning, usually from a limited set of volunteers, is unlikely to meet the demand of a large-scale social network with as many as several hundred million users [42, 44]. This problem can be circumvented by storing user data on a provider's server, but in encrypted form, as proposed by Anderson et al. [21]. Yet, not being able to access user data undermines the providers' business model, which means that they will not provide the required server capacity.

Another option is to motivate server or cloud providers with user payments, as seen in Vis-à-Vis or Contrail [26, 28]. Ultimately, this concept results in imposing monetary costs (e.g., payment for VMs) on users to enable social networking. Because current centralized OSNs not only offer more functionality, but are also generally free of usage fees, public acceptance for such an approach seems unlikely [62, 63].

Second, the dependency on both altruistic and paid servers is also a concern, as data loss can occur when such servers become disengaged abruptly. Such disengagement can happen, for instance, when the subscription for a cloud VM runs out, storage quota are exceeded, if a popular altruistically provided server can no longer be maintained free-of-charge because it creates too much traffic, or simply if the server owner decides to shut down the server.

Third, cloud providers are not necessarily a better alternative than current OSN providers with regards to user privacy. Dropbox, which is for instance suggested as a storage option in POSN, was (alongside with Microsoft and Amazon) also involved in the PRISM program and has been criticized for other privacy breaches [64]. Also, any provider may change the terms of usage for their services at will, or shut down the service completely [54].

Fourth, common among all the approaches are further drawbacks with regards to malicious users. If a user stores her data at a dedicated server, this server might suffer from DDoS attacks at any time or simply get overwhelmed by benign data requests. None of the state-of-the-art systems provide means to react to such scenarios. Exceptions are Vis-à-Vis and PrPI. In the former a user could boot additional VMs to manage the increased load, but the procedure itself is undefined and would also result in higher monetary cost. In the latter a user would theoretically be able to migrate her data to another of her locations, but this requires manual intervention of the data owner and

thus technical understanding of resource overutilization.

4.2. User-cooperation Solutions

4.2.1. Advantages

With the mutual cooperation of nodes and flexible data storage locations, users can be independent of dedicated servers and their drawbacks. Moreover, as the OSN functions with resources that are exclusively contributed by users, it can operate without additional costs. Also, these solutions are more practical in the sense that they do not require the user to set up any server facilities or rent additional machines in the cloud. On the contrary, this approach is more or less plug-and-play, as users can keep using the devices with which they usually explore OSNs.

In addition to the monetary aspects, user-cooperation solutions can be much more flexible, since storage locations can be chosen dynamically: for instance, in case of failure, it is easier for a user to find an additional friend node to host her data than to host a second web-server.

4.2.2. Drawbacks

However, user-cooperation schemes are also much more challenging in terms of achieving the same performance as server-based DOSNs or even centralized OSNs. In general, there exist some trade-offs that user-cooperation solutions have to deal with:

- To maximize data availability in the absence of servers or cloud environments also means to introduce replication of data; increasing the replication factor results in a higher availability, but also in higher replication overhead and synchronization efforts.

- To maximize user data privacy, only trusted and user-chosen nodes should be used for replication of data; however, an increasing trust requirement at the same time decreases the cardinality of the set of possible replication locations. Therefore, cooperation-based solutions that seek to maximize data availability are often less transparent in their storage choices.

- The user-experienced latency of lookups (e.g., in a DHT) can be much higher than in centralized environments [28]; to reduce the latency, distributed caching may help, but requires additional communication overhead.

Finally, all solutions in this category suffer from the trend of OSN users that migrate to mobile devices with increasing frequency [65, 66]. While Facebook reported 21% of their users accessing the service exclusively from mobile devices in 2013 [65], this number has more than doubled to 44% in July 2015 [66].

This trend can raise two issues. First, while mobile devices have recently become computationally more powerful, many of them still offer comparatively low resources. For instance, low-budget smartphones (with low processing and storage capabilities) are widely distributed in developing countries [67], where millions of phones are sold for less than $100 [68]. Approximately one third of the phones in developing countries have CPU speeds less than 500Mhz and less than 10MB of memory [69]. Also, the battery lifetime of mobile devices can be reduced by up to 40% when a device is queried frequently [28], which can make contributing to the DOSN unattractive to users.

Second, mobile devices are limited in their connectivity from two perspectives. On the one hand, they are usually tied to a data plan that limits the amount of data a user can consume in a month when not connected to, e.g., a WiFi access point. By requesting user data from mobile devices that limit could quickly be reached. On the other hand, bandwidth is also an issue. Although high-speed network coverage has improved in the past few years, even 3G connectivity results in request response delays ranging from 500ms to 3000ms on average for a single query [26, 28]. Cooperation-based DOSNs often need to issue multiple queries to different storage locations for a single user request (consider, for instance, a newsfeed application), which can thus ultimately result in much longer delays. Worse, many OSN participants will be located in areas with limited connectivity.[1] For instance, even in the United Kingdom 3G coverage has only reached 90%, and there still exist uncovered areas in major cities [70]. In developing countries 66% of the mobile phones only support GSM and GPRS, limiting their data transmission rate to 40kbps [69]. In these cases, requesting data from mobile devices can result in significant delays or even timeouts.

[1] `https://www.nperf.com/en/map/` gives a graphical overview of worldwide network coverage

4.2.3. Solution-specific Drawbacks

Moreover, each style of storing data in user-cooperation schemes has specific weaknesses.

Replicating data in distributed overlay: Prometheus, LifeSocial.KOM, DECENT and Cachet exploit the data replication features of a DHT to make user data available. While this principle ensures high data availability, it also increases the communication overhead between nodes. As OSNs usually experience high churn rates [71, 72], data often has to be transferred from departing nodes to other DHT members. This is particularly the case for mobile nodes. Also, there are no efforts to minimize the number of replicas. As a result, the overhead to keep all replicas of a user's data up-to-date can be increased.

Friend-storage: In this class of solutions, users can only store their data at friends' devices, which comes with some limitations.

First, a user depends on her social contacts for data storage, as she needs enough suitable friends that qualify as a mirror (in MyZone a mirror is even more trusted than a friend). This is difficult for many users in an OSN who maintain few social relations [73]. As a consequence, such systems typically achieve low data availability rates. For instance, 90% are reported for Safebook and MyZone, which means that every 10th request of a user is unsuccessful.

Second, none of the solutions explicitly considers mobile (i.e., smartphone) devices, which have become one major way of using OSNs. In cooperation schemes some tasks (e.g., maintenance of a DHT, or acting as a Point of Storage in DiDuSoNet) can be difficult for mobile devices with limited connectivity, bandwidth capacity and energy resources.

Third, both defense algorithms against any kind of attack and adaption mechanisms for extended user contributions are missing—with the exception of Proofbook and MyZone, which offer some basic defense algorithms against DoS attacks. However, all friend-based solutions are not protected against compromised friend nodes, which are a global phenomenon in OSNs [74, 75]. These nodes can easily stop serving data of honest users and thereby undermine the system performance. This dilemma could be mitigated by dynamically changing storage locations.

However, and fourth, only DiDuSoNet is able to dynamically adjust the replica assignment on the basis of standard DOSN dynamics. It supports a failover for the case in which one of the two Points of Storage (PoS) fails,

but falls short of a solution for the case in which *both* PoS fail (the current solution is to wait for a returning PoS). In the light of recent research that suggests very short OSN client session times, such a situation may happen frequently [72]. In all other approaches, adaptivity to such scenarios is worse: once a user has made a choice of where to store data, the mirrors remain static. Re-election procedures for the case of persistently failed, malicious or otherwise non-performant mirrors are not proposed yet.

Fifth, most solutions are not making an effort to limit their storage and communication overhead. This usually results in storing many copies of a user's data in the network to achieve a certain degree of data availability (e.g., Safebook requires 13-23 replicas per user). The consequence is increased network overhead and difficult data synchronization efforts. Exceptions are PeerSoN and DiDuSoNet. In PeerSoN nodes enter mutual storage contracts that optimize the number of replicas in the network (PeerSoN achieves approximately 4-6 replicas per user), while DiDuSoNet effectively limits the number of replicas per user to the two Points of Storage (PoS).

Finally, these properties lead to further solution-specific problems:

- The main issue of *PeerSoN* is its inability to construct a robust OSN due to the concept of mutual storage agreement among similarly performing nodes. While the concept of tit-for-tat contracts works well for high-performance nodes (i.e., nodes with high online time), users with an online time of less than eight hours a day achieve less than 90% data availability. Given a power-law distribution of online time in OSNs [71, 76] (or worse [72]), the majority of users will thus be unable to achieve high levels of data availability. The outcome is a frail OSN where even highly contributing users may not be able to find data they want.

- Also due to short user sessions, data has to be transferred often between points of storage (PoS) in *DiDuSoNet*. In particular, every time a PoS leaves the network, the data has to be transferred to the newly elected PoS, which introduces a high communication overhead. Moreover, for the election of a new PoS, the electing nodes need to know a classification of the user's friend relations, which can be a privacy concern.

- *Friendstore* offers a storage substrate, but lacks important OSN functionality (i.e., communication infrastructure).

- In *Safebook* all nodes on the same path towards the innermost shell of nodes need to be online simultaneously in order to provide access to the data, which is unlikely (due to short user session times [72].

Best-effort: Best-effort systems try to fix several of the aforementioned problems. Similar to Safebook and PeerSoN in their architecture, both GEMSTONE and SOUP are more flexible as the storage locations for data can be any device, ranging from those of friends to possibly altruistically provided servers. In SOUP, storage locations are additionally no longer fixed, but rather dynamic and can be changed as, for instance, users are able to detect failing storage locations. These properties relax the dependency on friend nodes and enable the system to react to various DOSN dynamics. Moreover, SOUP provides protection against some kinds of attacks and favors mobile devices with a special treatment: it exempts them from participating in data storage and overlay maintenance and thereby reduces resource consumption on these devices.

However, several problems remain. First, the storage and communication overhead—while reduced in comparison with earlier approaches—is still significant. To *maximize* data availability both GEMSTONE and SOUP follow the rationale that *at least one copy* of each user's data should be available in the network at all times. At the same time, to *minimize* the replica overhead, they follow the rationale that *one online copy is enough*. The result can often be a situation in which exactly one copy of a user is available, while the majority of replica holders each user needs to manage around six replicas on average are offline. Now, if there is an update to that user's data, that update has to be distributed not only to the replica holders, but also to their storage locations recursively. Here, a simple procedure like posting a picture to someone's profile can have significant overhead, especially as the size of multimedia objects is growing.

Second, both systems face deployability issues as they currently do not offer any NAT traversal capabilites, on which they would highly depend in practice. In fact, currently, out of all user-cooperation schemes only MyZone supports NAT traversal.

Third, SOUP needs a significant amount of time (in the scale of days) to converge to a stable solution. This means that new users will usually suffer from limited availability of their data during the time in which users are most active in OSNs [77].

4.3. Hybrid Systems

The third category of infrastructure substrates are hybrid systems, which combine elements from both above categories to get the best of both worlds.

4.3.1. Advantages

By combining server-based solutions with cooperation-based schemes, hybrid solutions can offer the benefits of both classes. In particular, they can offer high user data availability by the use of servers, while keeping communication and data private among users.

4.3.2. Drawbacks

However, currently, hybrid solutions unfortunately also suffer from drawbacks typically experienced by both server-based and cooperation-based approaches. Confidant lets users cooperate to provide each other with storage space for their data, while the name resolution service for the data is built by using cloud services. Thus, Confidant requires a lower level of cloud interaction than Vis-à-Vis. Still, a monetary effort is required by the user. At the same time, data availability from storing the data on user machines tends to be low for weakly connected users, as Confidant also requires these machines to be trusted (i.e., friends), which essentially lets Confidant suffer from many of the drawbacks discussed above (Section 4.2.3).

In SuperNova super peers act as a replacement for the dedicated servers, and thus have to implement a variety of functionality to be used by the remaining users, including maintenance of the directory of the network and acting as a storage node for others. These super peers are supposed to be economically motivated as well, which means that users either have to pay for their services, or that super peers should have access to some parts of the data to make advertisement valuable for them. Both properties are not desirable for DOSN solutions. Further, SuperNova does not provide any information on how to choose other nodes as mirrors (storekeepers) in the first place, and suffers from low data availability similar to Safebook or MyZone. Additionally, it does not provide any encryption means, so that super peers and storekeepers can always inspect the data other users store at their facilities.

Finally, Lilliput splits up user data. A small set of basic information is replicated among user devices in a cooperative fashion, while bulky data is stored on servers to ease the burden on the user devices. Unfortunately, this approach only helps in reducing the communication overhead among user

devices and thus can especially help mobile users, but does not solve the remaining challenges of cooperation approaches. At the same time, users are suggested to store their bulky data (i.e., images and videos) on services like Dropbox, YouTube or their own personal server, which again are controlled by a centralized provider or unfeasible to set up for every OSN user.

5. Towards a Successful DOSN

So far we have summarized the concepts of state-of-the-art DOSN solutions in our taxonomy, and then investigated what has been going wrong with these solutions in our analysis. There, we paid particular attention to the major challenge of decentralizing OSNs, which is to replace the infrastructure currently offered by a central provider. Table 4 summarizes the advantages and disadvantages of each approach. From this analysis, as a next step, we extract ten challenges for building a decentralized infrastructure that we think will enable DOSN success, if fulfilled.

5.1. Challenges for DOSNs

5.1.1. General Architecture

The first question to be answered when designing a DOSN is that of choosing the architecture that will substrate the centralized infrastructure. As previously argued, relying on permanently available, external resources would result in a dependency on the resource provider, which can not only observe communication patterns but also change the terms of usage or shut down their service at will. Furthermore, a large-scale dissemination of any DOSN requires the acquisition of a critical mass of users. As current centralized OSNs are free of fees, getting over this hurdle will be even more difficult in a DOSN that requires user payments, which are expected in cloud-based DOSNs. The same reasoning can be applied to hybrid systems, where both a dependency on super nodes or cloud providers and payment requirements still exist.

Challenge 1 (C1) - Independency: *A DOSN must not depend on any external resource provider, neither commercial nor altruistically motivated.*

C2 - Free-of-Charge: *A DOSN must not incur additional costs on any user.*

Class	DOSN	Approach Advantages	Approach Disadvantages
Server-based	**Common advantages:** high data availability; little to no replication and synchronization overhead; lower latency compared to P2P approaches		
	Common disadvantages: potential loss of data; static storage choices; limited attack prevention; potential privacy issues with cloud provider(s)		
	Anderson et al.	-	Incomplete architecture
	Persona	-	Setup of server or user payment required
	Diaspora	-	Relying on altruistic provisioning
	PrPl	Multiple storage locations possible	User payment required
	Vis-a-Vis	-	User payment required
	Contrail	-	User payment required
	POSN	Free-of-charge storage locations	-
Cooperation	**Common advantages:** No dependency on servers; scalable; free-of-charge; plug-and-play; improved opportunities for privacy		
	Common disadvantages: High latency; lower data availability; replication overhead; synchronization overhead; none to limited mobile support (exception: SOUP); none to limited defense mechanisms (exceptions: MyZone, ProofBook, SOUP); static storage choices (exceptions: SOUP, DiDuSoNet)		
	Friendstore	Offline contracts form trust relation	No DOSN functionality
	Safebook	Preservation of anonymity	Shell structure when faced with churn
	PeerSoN	Relatively low replication overhead	Frail OSN due to tit-for-tat strategy
	Prometheus	Enables inference queries	High communication overhead
	LifeSocial.KOM	-	High communication overhead
	GEMSTONE	Soft transitioning capabilities	Convergence time
	DECENT	-	High communication overhead
	Cachet	Caching improves performance	High communication overhead
	MyZone	Defensive mechanisms	Dependance on friend nodes
	ProofBook	Defensive mechanisms; preservation of anonymity	Dependance on friend nodes
	SOUP	Mobile support; dynamic storage choices; defensive mechanisms	Convergence time; recursive updates
	DiDuSoNet	Dynamic storage choices	PoS concept not feasible in high churn situations
Hybrid	**Common disadvantages:** Inheritance of disadvantages from server-based and cooperation approaches		
	Confidant	Lower cloud involvement compared to server-based solutions	User payment required; dependance on friend nodes
	SuperNova	Lower cloud involvement compared to server-based solutions	SuperPeers economically motivated; lack of actual algorithms
	Lilliput	Dynamic storage choices	-

Table 4: A summary of advantages and disadvantages of DOSN approaches.

5.1.2. Resource and Usage Diversity

Hence, a DOSN that exploits the cooperation of users appears as the better choice. However, as evaluated earlier, the design of such an approach is more difficult. Here, user communication patterns and resources are diverse and can even be subject to change.

First, the online time patterns of OSN nodes can differ substantially from each other and—equally important—seem to change from time to time. Around five years ago OSNs usually experienced a power-law distribution of online times [71, 76, 78]. Here, the majority of users are seldomly online, but at the same time, a significant fraction of users is available with a certain stability. This phenomenon was exploited by several DOSN approaches when designing algorithms for reliable data storage. Recent research, however, has shown much more short-lived and less reliable session times in OSNs [72]. As a consequence of changing patterns, a DOSN infrastructure that is built based on certain assumptions of user behavior might not be able to function properly once this behavior changes.

C3 - Online Times: *A DOSN must not be dependent on a particular user online time distribution.*

Second, users run diverse hardware configurations. While desktop devices are still one way to use OSNs, social networking on mobile devices has become much more popular in recent years. For instance, Facebook was accessed by almost equal amounts of mobile and desktop users, and a large fraction of users visited Facebook exclusively from their smartphones in the first half of 2016 [6].

C4 - Mobile Support: *A DOSN must acknowledge that a large fraction of users participates from mobile devices. Ideally, it should be able to function even if all users are using mobile devices.*

5.1.3. Efficiency, Scalability, Resiliency

Moreover, our analysis revealed that many DOSNs can, especially in the presence of short-lived sessions, have unregulated overhead—(i) due to the requirement of moving user data between peers often and (ii) due to a large number of user data replicas in the system. Thus, even though storage is a relatively cheap resource, the DOSN must make an effort to reduce this overhead. Note that dedicated solutions for efficient replica management have been proposed separately and could be exploited for this purpose [79–81].

C5 - Efficiency: *A DOSN has to ensure little storage overhead and prevent an excess in communication overhead.*

A new DOSN must further scale to the dimensions of a successful OSN. Here, the goal is to support communication among many millions of users. Unfortunately, full cooperation and resource contribution of all users towards that goal can not be assumed [44]. For instance, when considering replicating user data across the DOSN, many devices (e.g., mobile devices) have to be exempted from that task.

C6 - Scalability: *A DOSN must be scalable to millions of users, even if there are limited resources available.*

At the same time, a successful DOSN would have to be able to adapt to changes, which will be frequent in a multi-million device network. From a system perspective, these changes can be both of positive and negative nature. For instance, the DOSN *should* also be open to the opportunities of altruistically provided resources (e.g., servers similar as in Diaspora) and exploit them if they become available. At the same time, it *must* be resilient when facing more unfavorable situations as well. That is, its performance must not significantly suffer even if resources in the network become—temporarily or permanently—unavailable (e.g., due to node failures, overloading, or attacks by an adversary).

C7 - Resiliency: *A DOSN must be resilient against both DOSN dynamics and attacks towards the network.*

5.1.4. Privacy

Removing the central provider and shifting the data to distributed storage locations across the OSN does not prevent the collection, analysis and misuse of data by a third party if there are no mechanisms to control access to the data.

User content in OSNs is often targeted at a specific audience only. For instance, while a user might decide to share her vacation pictures with her close friends, she does not want her work colleagues to see those pictures as well [22]. Thus, a new DOSN must provide users with means to facilitate access control, to hide the contents of their data from others.

While access control is today partially realized in centralized OSNs as well, in a DOSN it includes controlling access of the storage providers themselves. In this context, one particularly prominent problem is that of private information retrieval (PIR) [82]. Here, any third party hosting user data ideally should not be able to track the patterns of access to the data itself.

C8 - Privacy: *A DOSN must allow each user to control access to every single data item, and to do so on a very fine-grained basis. Access to data should not be tracked.*

5.1.5. Performance

After installing their DOSN client, users often experience limited performance. P2P-based solutions often incur high latency and a user has to wait for up to hundreds of seconds for a query to complete [37]. Further, some systems require a long bootstrapping time to achieve a stable system state. As a consequence, data of a new user might only be highly available after several days. While a distributed OSN can not be expected to perform as good as a centralized version backed by a datacenter infrastructure, it has to offer an environment that is user-friendly. That is, for instance, query delays should be well within 2 seconds to maintain the users' focus [83, 84].

C9 - Performance: *A DOSN has to offer user-friendly performance.*

5.1.6. Usability

Finally, one big limitation of today's DOSNs is that they are often not easy to use:

- Users need to resolve dependencies, install additional software, or are hindered in their participation by the inability of the DOSN client to traverse NAT. Worse, some systems require users to set up web servers or to get familiar with cloud services. An ideal DOSN on the contrary should be easy to set up. That is, they should run in, e.g., a mobile application on mobile devices, or over a web interface without the requirement to install additional software on desktop computers or laptops.

- Further, we believe that one key to DOSN success is the ability of the DOSN to interact with current centralized OSNs. Intuitively, it is hardly possible to create a DOSN that will be able to attract and maintain a significant user base instantly. The reason is simple: the interest in an OSN grows with the amount of users, content and functionality available there. Users will quickly lose interest in a new DOSN with—which is to be expected—low user numbers (in the bootstrapping phase), little content available, and initially less functionality than its centralized counterpart. A soft transition from centralized OSNs towards a DOSN could help by allowing users to concurrently use both

Class	DOSN	C1 - Independence	C2 - Free-of-Charge	C3 - Online Times	C4 - Mobile Support	C5 - Efficiency	C6 - Scalability	C7 - Resiliency	C8 - Privacy	C9 - Performance	C10 - Usability
Server-based	Anderson et al.	X						X			X
	Persona	X	X					X			X
	Diaspora	X					X	X	X	X	X
	PrPl	X						X			X
	Vis-a-Vis	X	X					X			X
	Persona	X	X					X			X
	Persona	X						X			X
Cooperation	Friendstore			X	X	X		X	X		X
	Safebook			X	X	X		X			X
	PeerSoN			X	X	X		X	X		X
	Prometheus			X	X	X		X			X
	LifeSocial.KOM			X	X	X		X			X
	GEMSTONE			X	X	X		X			X
	DECENT			X	X	X		X			X
	Cachet			X	X	X		X			X
	MyZone			X	X	X		X			X
	ProofBook			X	X	X		X			X
	SOUP			X		X					X
	DiDuSoNet			X	X	X		X			X
Hybrid	Confidant	X	X	X				X			X
	SuperNova	X	X	X	X	X		X	X		X
	Lilliput	X		X	X			X			X

Table 5: A summary of the state-of-the-art DOSNs. Cases, in which an approach does *not* meet a certain challenge are marked *X* in the table. For instance, the reading with regards to the challenge *performance* is that no cooperation-based solution tackles this challenge. The only exception is Cachet, in which caching is employed to reduce the user-experienced latency.

networks. Such a transition can be implemented by using provider APIs (e.g., the Facebook Graph API) to mirror user posts on the DOSN to the respective OSN [41, 85].

- At the same time, one of the big opportunities of decentralized online social networking is to remove the need for each user to maintain one set of data for each social networking application [31]. Thus, a new DOSN should also be generic in the sense that it allows a multitude of applications to operate on a single set of shared basic data (e.g., login credentials or general user information). Moreover, these applications should be allowed to be diverse in the sense that they, while operating on a single set of shared data, can implement additional functionality and features, adding application-specific contents to the existing user data. In particular, they should not be limited to current OSN functionality.

C10 - Usability: *A DOSN has to be plug-and-play for the users; it should allow for a soft transition from centralized OSNs, the development of a multitude of applications, and the integration of new features on top of the architecture.*

5.2. Current DOSNs vs Challenges

We now summarize how current DOSN solutions hold up with regards our success criteria. A summary of state-of-the-art DOSNs is given in Table 5. Here, each approach is listed with regards to whether or not it is successful in fulfilling the challenges, with the goal of providing a clear overview of the proliferation of DOSN solutions based on our observations from Section 4.

One finding is that each investigated system has deficiencies in multiple properties, which would be important to fulfill in order to enable the adaptation of the DOSN by a critical mass of users from centralized OSNs. As a consequence, a competitive DOSN is currently lacking.

Our major point here, however, is that we can extract several commonalities among DOSN solutions. On the one hand, server-based systems can efficiently enable high availability for user data with reasonable performance (C3-C6). However, their main unsolved challenge is to provide these features in a both technically and economically feasible fashion, without depending on some sort of (paid-for) centralized entity as, for instance, cloud services or web servers, which might be shut down at their owner's will (C1-C2, C10).

On the other hand, systems which exploit cooperation among users are able to solve these challenges, but have limited success in providing high and robust data availability. These issues are rooted in their dependency on online times of OSN users (C3), where low availability of user devices results in low overall data availability. Additionally, they do not consider mobile devices (C4) in the sense that mobile devices are often expected to contribute their sparse resources to the DOSN.[2] Also, only by assuming certain online time patterns, they are able to provide scalable DOSN solutions without introducing a lot of overhead (C5-C6). Since more recent patterns suggest much shorter online times for OSN participants [72], the overhead of mirroring schemes can increase and may also result in limited scalability. Most cooperation schemes have further problems in achieving *actual* deployment success based on their low performance (C9) or their inability to traverse NAT (C10). Here one major issue can be the delay of requests when, for instance, a query for a certain piece of data needs to traverse several hops on a DHT to only retrieve the location of the data, let alone the data itself.

Moreover, hybrid systems, while trying to extract the best from both worlds, in fact suffer from their combined drawbacks as well. For instance, in Confidant, the usage of cloud services as lookup service still introduces a monetary cost, while the data storage at user nodes is still problematic as it relies on the users' online times.

Finally, arching over all existing solutions resiliency (towards dynamics and/or attacks) is lacking (C7). The reason for shortcomings in these features is mainly that providing a storage substrate for a DOSN is already challenging, and researchers have focused on providing this substrate for operation in benign scenarios until now. Thus, both security (besides the privacy of user data) and adaptivity have—although important—not been considered as issues yet.

There is however an alternative reading of Table 5. If we invert our discussion of weaknesses into one of strengths, the results of several years in DOSN research become apparent. On the bright side, one result of the extensive research in DOSNs is the availability of comprehensive schemes to improve user privacy in these networks (C8). Nearly all DOSNs offer such capabilities or are conceptually extensible (e.g., Diaspora's concept does not

[2]Note that mobile devices can be inherently supported by server-based solutions, in which they do not have to participate in maintaining the DOSN itself.

technically hinder the deployment of encryption mechanisms).

In summary, server-based solutions excel in providing (i) highly available machines, (ii) efficient storage, (iii) mobile device support, (iv) high performance and (v) scalability. Cooperation schemes provide in (vi) independence from (paid) resource providers and (vii) flexibility. One main observation is that these properties are almost mutually exclusive, while an ideal DOSN would offer all of them. Therefore, we next describe the main trade-offs among the challenges and discuss whether these trade-offs inherently prevent the design of an ideal DOSN.

5.3. Challenge Trade-offs

Independence: The first trade-off concerns the level of independence. On the one hand, C1 and C2 rule out any external storage provider and any user cost for the sake of independence and public acceptance. On the other hand, C3 and C4 require a DOSN to work with varying user online times and high participation from mobile devices. That is, the DOSN has to deal with possibly short sessions, high churn-rates and limited storage resources, which makes reliable data storage difficult. Thus, there exists a trade-off between independence and reliable, highly available storage, both of which are required in an ideal DOSN.

Efficiency: Another issue is that of overhead in the system. Here, C5 mandates low storage and communication overheads. However, this challenge has to be traded-off with C1, as an ideal system without replication and therefore little overhead can only be achieved by exploiting permanently available resources. More generally, reduced overhead results in a higher dependency on specific storage locations.

Resiliency: Further, there is a conflict between C5 and C7, where the latter requires a DOSN to be resilient on several levels. Resiliency (e.g., towards node failures) is usually achieved by increasing the storage overhead (replication), and subsequently also communication overhead in the network (replication synchronization).

Performance: Finally, fulfilling challenges C1 through C4 has an impact on the system performance (C9). As previously discussed, achieving user-friendly performance is difficult in cooperation-based systems, as distributed look-up directories in combination with inadequately connected devices are responsible for high response times.

Due to these trade-offs and conflicts, the design of an ideal DOSN seems unlikely. What we thus need is a good approximation that is able to deal

with most of the challenges, while elegantly handling the inherent trade-offs among them.

5.4. An Outlook to Future DOSNs

Although the concrete design for a novel DOSN is out of scope of this work, we would like to put into discussion two possible ways of moving forward.

5.4.1. Continue the Challenge

The first option is to continue in the design of comprehensive DOSN solutions, and to challenge existing OSNs. Here, instead of opting for server-based *or* cooperation schemes, one possible design could *combine* both styles of decentralized online social networking—however, in a different way than we have seen from hybrid approaches. While these solutions assumed the existence of remote server facilities and the cooperation of user devices, another way forward would be to follow an orthogonal approach. Here, the rationale would be to exploit server-like facilities in the domain of users to relax the dependence on user devices. In particular, the concept of using users' home gateways as storage and communication entities has been proposed repeatedly and represents an option worth investigating [11, 86]. These devices can combine the benefits of both server-based and user cooperation approaches.

A home gateway is a machine with similar availability as a personal remote web-server or cloud VM; such high availability is a key to efficient storage of user data. User data can—when stored on a home gateway—be decoupled from (mobile) user devices and also from their online times. Thus, similar to server-based solutions, diurnal patterns and the prevalence of mobile devices become irrelevant. Furthermore, home gateways are usually not restricted by a NAT that needs to be traversed and their performance (in terms of bandwidth) is generally reasonable.

At the same time, home gateways *remain (even physically) in the domain of the user, and there is no dependence on a resource provider.* This also further limits the problem of access tracking to the data owner herself. Additionally, by carefully designing cooperation schemes among gateways (by learning from existing schemes), flexibility towards DOSN dynamics or attacks can be achieved.

Overall, such a solution has the potential to combine benefits (i)-(vii) as listed above, and in particular to mitigate the trade-offs involving the independence challenge. Still, several hurdles are to take on this path—including ensuring usability and handling users without access to a home

gateway.

5.4.2. Change the Perspective

While in this paper we have mainly discussed the technical issues of DOSNs, one major obstacle towards gaining significant public visibility is the powerful market position of existing, centralized OSNs. When further considering their rich feature lists in addition to this market position, even a DOSN that solves all our challenges is unlikely to supersede Facebook and others.

A second option to obtain market share is therefore to focus on designing new distributed social networking applications that do not have the aspiration to replace market-leading common OSNs, but rather focus on a specific target audience or topic.

6. Related Work

Studies that touch on our understanding of OSNs are omnipresent. However, they usually focus on specific topics, such as evaluating the trust present in OSNs [87] or studying the behavior of OSN users in general [88], while studies that systematically analyze DOSN systems are rare. In a first study on DOSNs, Datta et al. [89] introduced the concept of DOSNs and analyzed early approaches to decentralization. In one of the few recent works in this area Chowdhury et al. [90] focus on a taxonomy of Peer-to-Peer (P2P) DOSNs. In another effort Paul et al. [91] focus on classifying existing DOSNs with regards to security and performance as a follow-up on an earlier study [92].

Our work is substantially different from these studies:

- First, we provide a complete taxonomy of DOSN designs, in which we include the most recent state-of-the-art solutions. All three previous studies listed above focused on a first generation of DOSNs and none of them covers more recent efforts (from 2013 onwards) to decentralize social networks.

- Second, we also look beyond purely P2P-based solutions. In fact, two major classes of approaches to decentralize OSNs—as we have seen above—either incorporate or heavily rely on centralized elements. These classes are not considered in [89, 90].

- Third, we not only list existing solutions, but also abstract DOSNs to design rationales towards one major challenge of DOSNs, i.e., substituting the infrastructure of a central OSN provider. Based on these

rationales, we take another step and provide a thorough in-depth analysis on why current DOSN approaches have not yet succeeded.

- Finally, we present success criteria for a competitive DOSN, and point out a possible way out of the dilemma of the state-of-the-art DOSNs.

7. Conclusion

In this paper we have thoroughly investigated the landscape of comprehensive decentralized online social networks (DOSN). We set out to find those reasons that, from a technical perspective, have hindered DOSNs from being successfully deployed among a significant user base. After motivating the case for DOSN, we presented a taxonomy of over twenty recent advances in decentralizing OSNs. In this taxonomy, we focused on how each DOSN solution tries to solve the major challenge of DOSNs, that is, how to build a substrate for the complex infrastructure of centralized OSNs. We abstracted state-of-the-art solutions to their common design principles, and analyzed the implications of these design principles.

Our main observations are:

- Current DOSNs can be categorized into server-based, cooperation-based, and hybrid solutions. Here, each of these categories follows a different way to realize the substrate for the centralized infrastructure, which mainly differs in the level of involvement of cloud or storage providers.

- Solutions in each category follow similar design choices, each of which comes with a number of advantages and disadvantages. While server-based solutions are typically strong in providing an efficient substrate, cooperation-based approaches are more independent, free-of-charge and scalable.

- While DOSN research has advanced in the past years, a wide range of partially conflicting challenges are yet to be addressed by future DOSNs in order to achieve widespread acceptance. Existing solutions excel in some of these challenges, but no solution has been able to fulfill them sufficiently yet.

Finally, we have pointed out future directions that could be taken to progress towards competitive DOSNs. These directions may work on approximating a solution for all challenges, or change the focus of DOSN research towards more specific applications.

References

References

[1] Facebook, World Cup 2014: Facebook Tops A Billion Interactions, http://newsroom.fb.com/news/2014/06/world-cup-2014-facebook-tops-a-billion-interactions/ (all online sources have been accessed last on March 7, 2017) (June 2014).

[2] Twitter, Insights into the #WorldCup conversation on Twitter, https://blog.twitter.com/2014/insights-into-the-worldcup-conversation-on-twitter (July 2014).

[3] Twitter, Measuring tweets, https://blog.twitter.com/2010/measuring-tweets (February 2010).

[4] GlobalWebIndex.com, Twitter Now The Fastest Growing Social Platform In The World, http://bit.ly/1rJXKgA (January 2013).

[5] Twitter, About Twitter, https://about.twitter.com/company (July 2014).

[6] Facebook, Facebook Statistics 2nd Quarter 2016, https://investor.fb.com/investor-news/press-release-details/2016/Facebook-Reports-Second-Quarter-2016-Results/default.aspx (July 2016).

[7] CNN, Facebook acquires instagram for $1 billion, cnnmon.ie/1mh3vK2 (April 2012).

[8] D. Rushe, WhatsApp: Facebook acquires messaging service in $19bn deal, https://bitly.com/shorten/ (February 2014).

[9] The Guardian: WhatsApp to give users' phone numbers to Facebook for targeted ads, https://www.theguardian.com/technology/2016/aug/25/whatsapp-to-give-users-phone-number-facebook-for-targeted-ads.

[10] B. Krishnamurthy, C. E. Wills, Characterizing Privacy in Online Social Networks, in: Proceedings of the First Workshop on Online Social Networks (WOSN 2008), ACM, New York, NY, USA, 2008, pp. 37–42.

[11] S. Buchegger, A. Datta, A Case for P2P Infrastructure for Social Networks - Opportunities and Challenges, in: Proceedings of the 6th International Conference on Wireless On-Demand Network Systems and Services (WONS 2009), IEEE, 2009, pp. 161–168.

[12] C. Dwyer, Privacy in the Age of Google and Facebook, IEEE Technology and Society Magazine 30 (3) (2011) 58 –63.

[13] G. Greenwald, E. MacAskill, NSA Prism program taps in to user data of Apple, Google and others, bit.ly/193WyKq (June 2013).

[14] LinkedIn, An Update on LinkedIn Member Passwords Compromised, bitly.com/Ni5aTg (June 2012).

[15] E. Barnett, Facebook settles lawsuit with angry users, bit.ly/YYo1eZ (May 2012).

[16] M. Zimmer, Mark Zuckerberg's theory of privacy, http://wapo.st/1gJOqEu (February 2014).

[17] J. Van Dijck, The Culture of Connectivity: A Critical History of Social Media, Oxford University Press, 2013.

[18] L. Andrews, Facebook is using you, nyti.ms/1oVFxtC (February 2012).

[19] Facebook, Facebook Annual Report 2012, bit.ly/1rO7eC0 (January 2013).

[20] C.-M. A. Yeung, I. Liccardi, K. Lu, O. Seneviratne, T. Berners-Lee, Decentralization: The Future of Online Social Networking, in: W3C Workshop on the Future of Social Networking Position Papers, Vol. 2, 2009, pp. 2–7.

[21] J. Anderson, C. Diaz, J. Bonneau, F. Stajano, Privacy-enabling social networking over untrusted networks, in: Proceedings of the 2nd ACM workshop on Online social networks, ACM, 2009, pp. 1–6.

[22] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, D. Starin, Persona: An Online Social Network with User-defined Privacy, ACM SIGCOMM Computer Communication Review 39 (4) (2009) 135–146.

[23] The Diaspora Project, https://joindiaspora.com/ (2010).

[24] S.-W. Seong, J. Seo, M. Nasielski, D. Sengupta, S. Hangal, S. K. Teh, R. Chu, B. Dodson, M. S. Lam, Prpl: A decentralized social networking infrastructure, in: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing Services: Social Networks and Beyond, MCS '10, ACM, New York, NY, USA, 2010, pp. 8:1–8:8.

[25] A. Shakimov, A. Varshavsky, L. P. Cox, R. Cáceres, Privacy, cost, and availability tradeoffs in decentralized osns, in: Proceedings of the 2nd ACM workshop on Online social networks, ACM, 2009, pp. 13–18.

[26] A. Shakimov, H. Lim, R. Cáceres, L. P. Cox, K. Li, D. Liu, A. Varshavsky, Vis-á-Vis: Privacy-preserving Online Social Networking via Virtual Individual Servers, in: Proceedings of the 3rd International Conference on Communication Systems and Networks (COMSNETS 2011), 2011, pp. 1–10.

[27] P. Stuedi, I. Mohomed, M. Balakrishnan, Z. M. Mao, V. Ramasubramanian, D. Terry, T. Wobber, Contrail: Enabling Decentralized Social Networks on Smartphones, in: Proceedings of the 12th ACM/IFIP/USENIX International Middleware Conference (Middleware 2011), Springer, 2011, pp. 41–60.

[28] P. Stuedi, I. Mohomed, M. Balakrishnan, Z. M. Mao, V. Ramasubramanian, D. Terry, T. Wobber, Contrail: Decentralized and privacy-preserving social networks on smartphones, IEEE Internet Computing 18 (5) (2014) 44–51.

[29] E. Erdin, E. Klukovich, G. Gunduz, M. H. Gunes, Posn: A personal online social network, in: IFIP International Information Security Conference, 2015, pp. 1–14.

[30] D. N. Tran, F. Chiang, J. Li, Friendstore: Cooperative Online Backup Using Trusted Nodes, in: Proceedings of the 1st ACM EuroSys Workshop on Social Network Systems (SNS 2008), ACM, 2008, pp. 37–42.

[31] S. Buchegger, D. Schiöberg, L.-H. Vu, A. Datta, PeerSoN: P2P Social Networking: Early Experiences and Insights, in: Proceedings of the 2nd ACM EuroSys Workshop on Social Network Systems (SNS 2009), ACM, 2009, pp. 46–52.

[32] K. Rzadca, A. Datta, S. Buchegger, Replica Placement in P2P Storage: Complexity and Game Theoretic Analyses, in: Proceedings of the 30th IEEE International Conference on Distributed Computing Systems (ICDCS 2010), IEEE, 2010, pp. 599–609.

[33] L. A. Cutillo, R. Molva, T. Strufe, Safebook: A Privacy-Preserving Online Social Network Leveraging on Real-life Trust, IEEE Communications Magazine 47 (12) (2009) 94–101.

[34] N. Kourtellis, J. Finnis, P. Anderson, J. Blackburn, C. Borcea, A. Iamnitchi, Prometheus: User-controlled p2p social data management for socially-aware applications, in: Proceedings of the ACM/IFIP/USENIX 11th International Conference on Middleware, Springer-Verlag, 2010, pp. 212–231.

[35] K. Graffi, C. Gross, D. Stingl, D. Hartung, A. Kovacevic, R. Steinmetz, Lifesocial. kom: A secure and p2p-based solution for online social networks, in: Consumer Communications and Networking Conference (CCNC), 2011 IEEE, IEEE, 2011, pp. 554–558.

[36] S. Jahid, S. Nilizadeh, P. Mittal, N. Borisov, A. Kapadia, Decent: A decentralized architecture for enforcing privacy in online social networks, in: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on, IEEE, 2012, pp. 326–332.

[37] S. Nilizadeh, S. Jahid, P. Mittal, N. Borisov, A. Kapadia, Cachet: A Decentralized Architecture for Privacy Preserving Social Networking with Caching, in: Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies (CoNEXT 2012), ACM, 2012, pp. 337–348.

[38] S. Biedermann, N. P. Karvelas, S. Katzenbeisser, T. Strufe, A. Peter, ProofBook: An Online Social Network Based on Proof-of-Work

and Friend-Propagation, in: Theory and Practice of Computer Science (SOFSEM 2014), Springer, 2014, pp. 114–125.

[39] A. Mahdian, R. Han, Q. Lv, S. Mishra, Results from a Practical Deployment of the MyZone Decentralized P2P Social Network, CoRR.

[40] B. Guidi, T. Amft, A. De Salve, K. Graffi, L. Ricci, Didusonet: A p2p architecture for distributed dunbar-based social networks, Peer-to-Peer Networking and Applications (2015) 1–18.

[41] F. Tegeler, D. Koll, X. Fu, Gemstone: empowering decentralized social networking with high data availability, in: Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE, IEEE, 2011, pp. 1–6.

[42] D. Koll, J. Li, X. Fu, SOUP: An Online Social Network By The People, For The People, in: Proceedings of the 15th ACM/IFIP/USENIX International Middleware Conference 2014, ACM, 2014, pp. 193–204.

[43] D. Liu, A. Shakimov, R. Caceres, A. Varshavsky, L. P. Cox, Confidant: Protecting OSN Data without Locking it Up, in: Proceedings of the 12th ACM/IFIP/USENIX International Middleware Conference (Middleware 2011), Springer, 2011, pp. 61–80.

[44] R. Sharma, A. Datta, SuperNova: Super-peers Based Architecture for Decentralized Online Social Networks, in: Proceedings of the 4th IEEE International Conference on Communication Systems and Networks (COMSNETS 2012), IEEE, 2012, pp. 1–10.

[45] T. Paul, Mitigating adverse effects of using online social networks, Ph.D. thesis, PhD Thesis, Technische Universität Darmstadt, Darmstadt (2016).

[46] Wall Street Journal, How Private Are Your Private Facebook Messages?, http://on.wsj.com/ZASqje (October 2012).

[47] C. Riley, Facebook faces suit over private messages, http://money.cnn.com/2014/01/03/technology/facebook-privacy-lawsuit/ (January 2014).

[48] A. C. Madrigal, Why Facebook and Google's Concept of 'Real Names' Is Revolutionary, theatln.tc/1uEzeNo (May 2011).

[49] S. T. Peddinti, K. W. Ross, J. Cappos, "On the Internet, Nobody Knows You'Re a Dog": A Twitter Case Study of Anonymity in Social Networks, in: Proceedings of the Second Edition of the ACM Conference on Online Social Networks, COSN '14, ACM, 2014, pp. 83–94.

[50] M. Dickmann, Inside//Out: Facebook Beacon, http://technomarketer.typepad.com/technomarketer/2007/11/insideout-faceb.html (November 2007).

[51] O. Smith, Facebook terms and conditions: why you don't own your online life, http://www.telegraph.co.uk/technology/social-media/9780565/Facebook-terms-and-conditions-why-you-dont-own-your-online-life.html (January 2013).

[52] Facebook, Statement of Rights and Responsibilities, https://www.facebook.com/legal/terms (November 2013).

[53] TechCrunch: Yang decides to shut down Yahoo 360 - nobody notices, https://techcrunch.com/2007/10/23/yang-decides-to-shut-down-yahoo-360

[54] The BBC: Ask.fm owners considered shutting down social network, http://www.bbc.co.uk/newsbeat/article/31249209/askfm-owners-considered-shutting-down-social-network.

[55] Google, Locations of Datacenters, http://bit.ly/YhZqAF (September 2014).

[56] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-Policy Attribute-Based Encryption, in: Proceedings of the 28th IEEE Symposium on Security and Privacy (S&P 2007), 2007, pp. 1–14.

[57] D. Recordon, D. Reed, OpenID 2.0: a platform for user-centric identity management, in: Proceedings of the second ACM workshop on Digital identity management, ACM, 2006, pp. 11–16.

[58] P. Druschel, A. Rowstron, Past: A large-scale, persistent peer-to-peer storage utility, in: Hot Topics in Operating Systems, 2001. Proceedings of the Eighth Workshop on, IEEE, 2001, pp. 75–80.

[59] A. Rowstron, P. Druschel, Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems, Lecture Notes in Computer Science 2218 (2001) 329–350.

[60] FreePastry 2.1, http://www.freepastry.org (2009).

[61] Diaspora, Installation Guide, http://bit.ly/1LTlKFa (January 2014).

[62] Digital Trends: We asked, you answered: Would you actually pay for Facebook?, http://www.digitaltrends.com/social-media/we-asked-you-answered-would-you-actually-pay-for-facebook/ (July 2013).

[63] CNBC: Would you pay to use Twitter?, http://www.cnbc.com/id/100918000 (August 2013).

[64] Z. Whittaker, Snowden: 'Wannabe PRISM partner' Dropbox is 'hostile to privacy', http://zd.net/1r84eDz (July 2014).

[65] Facebook: Annual Report 2013, http://investor.fb.com/annuals.cfm (2013).

[66] Facebook: Quarterly Results Q2 2015, http://investor.fb.com/results.cfm (2015).

[67] M. T. Review, Android Marches on East Africa, https://www.technologyreview.com/s/424454/android-marches-on-east-africa/ (2011).

[68] ZDNet, Google strikes out with Android One in India, http://www.zdnet.com/article/google-strikes-out-with-android-one-in-india/ (June "=!

[69] S. Ahmad, A. L. Haamid, Z. A. Qazi, Z. Zhou, T. Benson, I. A. Qazi, A view from the other side: Understanding mobile phone characteristics in the developing world, in: Proceedings of the 2016 Internet Measurement Conference, IMC '16, ACM, 2016, pp. 319–325.

[70] A. Sathiaseelan, J. Crowcroft, Internet on the move: challenges and solutions, ACM SIGCOMM Computer Communication Review 43 (1) (2013) 51–55.

[71] F. Benevenuto, T. Rodrigues, M. Cha, V. Almeida, Characterizing User Behavior in Online Social Networks, in: Proceedings of the 9th ACM SIGCOMM Internet Measurement Conference (IMC 2009), ACM, 2009, pp. 49–62.

[72] T. Paul, D. Puscher, T. Strufe, The user behavior in facebook and its development from 2009 until 2014, arXiv preprint arXiv:1505.04943.

[73] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, B. Bhattacharjee, Measurement and Analysis of Online Social Networks, in: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC 2007), ACM, 2007, pp. 29–42.

[74] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, Y. Dai, Uncovering Social Network Sybils in the Wild, in: Proceedings of the 11th ACM SIGCOMM Conference on Internet Measurement (IMC 2011), ACM, 2011, pp. 259–268.

[75] D. Koll, J. Li, J. Stein, X. Fu, On the State of OSN-based Sybil Defenses, in: Proceedings of the 13th IFIP International Conference on Networking (NETWORKING 2014), IFIP, 2014, pp. 1–10.

[76] L. Gyarmati, T. A. Trinh, Measuring User Behavior in Online Social Networks, IEEE Network 24 (5) (2010) 26–31.

[77] C. Wilson, A. Sala, K. P. N. Puttaswamy, B. Y. Zhao, Beyond social graphs: User interactions in online social networks and their implications, ACM Trans. Web 6 (4) (2012) 17:1–17:31.

[78] C. Wilson, B. Boe, A. Sala, K. P. N. Puttaswamy, B. Y. Zhao, User Interactions in Social Networks and their Implications, in: Proceedings of the 4th ACM European Conference on Computer Systems (EuroSys 2009), ACM, 2009, pp. 205–218.

[79] K. Rzadca, A. Datta, G. Kreitz, S. Buchegger, Game-theoretic mechanisms to increase data availability in decentralized storage systems, ACM Transactions on Autonomous and Adaptive Systems 10 (3) (2015) 14:1–14:32.

[80] P. Skowron, K. Rzadca, Flexible replica placement for optimized p2p backup on heterogeneous, unreliable machines, Concurrency and Computation: Practice and Experience 28 (7) (2016) 2166–2186.

[81] D. Koll, J. Li, X. Fu, With a little help from my friends: replica placement in decentralized online social networks, Technical Report: TR-IFI-TB-2013-01, University of Goettingen, Germany.

[82] B. Chor, E. Kushilevitz, O. Goldreich, M. Sudan, Private Information Retrieval, Journal of the ACM 45 (6) (1998) 965–981.

[83] F. F.-H. Nah, A study on tolerable waiting time: how long are web users willing to wait?, Behaviour & Information Technology 23 (3) (2004) 153–163.

[84] I. Arapakis, X. Bai, B. B. Cambazoglu, Impact of response latency on user behavior in web search, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '14, ACM, New York, NY, USA, 2014, pp. 103–112.

[85] P. Busby, Three different ways to import json from the facebook graph api, SAS Global Forum (2014).

[86] M. Marcon, B. Viswanath, M. Cha, K. P. Gummadi, Sharing Social Content from Home: A Measurement-driven Feasibility Study, in: Proceedings of the 21st International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2011), ACM, 2011, pp. 1–6.

[87] W. Sherchan, S. Nepal, C. Paris, A survey of trust in social networks, ACM Computing Surveys (CSUR) 45 (4) (2013) 47.

[88] L. Jin, Y. Chen, T. Wang, P. Hui, A. V. Vasilakos, Understanding user behavior in online social networks: A survey, Communications Magazine, IEEE 51 (9) (2013) 144–150.

[89] A. Datta, S. Buchegger, L.-H. Vu, T. Strufe, K. Rzadca, Decentralized online social networks, in: Handbook of Social Network Technologies and Applications, Springer, 2010, pp. 349–378.

[90] S. R. Chowdhury, A. R. Roy, M. Shaikh, K. Daudjee, A taxonomy of decentralized online social networks, Peer-to-Peer Networking and Applications 8 (3) (2015) 367–383.

[91] T. Paul, A. Famulari, T. Strufe, A survey on decentralized online social networks, Computer Networks 75 (2014) 437–452.

[92] T. Paul, B. Greschbach, S. Buchegger, T. Strufe, Exploring decentralization dimensions of social networking services: adversaries and availability, in: Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research, ACM, 2012, pp. 49–56.