ReDOSN a customizable Decentralised Social Network

Divi De Lacour, Adam Gautier

ReDOSN a customizable Decentralised Social Network

Divi De Lacour
Orange Innovation, IMT Atlantique, Inria
divi1.delacour@orange.com

Adam Gautier
Orange Innovation
adam.gautier@orange.com

Abstract

Social Networks have become a major part of people life, used to communicate and cooperate between humans and even with connected objects. Current social networks are server based, either centralised or federated. This poses security, privacy and performance issues as the server administrators must be trusted and the server is a single point of failures. Decentralised social networks proposed to tackle those issues lack customization. We propose a model to create customizable social network, focusing on the decentralisation and its security. We show how it can be implemented using existing decentralized technologies, arguing that data availability is not a hard constraint.

Keywords: online social networks, decentralization, decentralised online social networks, security

1 Introduction

Online Social Networks (OSN) have been a major innovation of the 21th century. They are used for efficient cooperation and information sharing between humans, for example for sharing alerts [4], or collaborating at work [37].
OSNs are a medium for human-machine interactions and cooperation (symbiotic) [29], they are also useful for cooperation between autonomous agents (social Internet of Things approach[24]) where connected objects imitate human social networks to better cooperate.
Current OSNs are mainly centralised, either with a single server or with a federation of servers. This poses performance and security issues as users are dependent of a server that can see their content and is a single point of failures[6].
Decentralised OSN (DOSN)[27, 31] have been proposed to tackle those issues, however they are designed to reproduce a specific centralised social network and as such lack the ability to be customized. For example Mastodon[33] is designed to imitate twitter and so only allows for small public messages. It cannot scale for video hosting that is handled by PeerTube, the federated alternative to YouTube.
Content hosting and guaranteeing data availability at scale is the main technical challenge for OSN. In centralised OSN, content hosting costs are handled by showing ads to users and using their data for marketing or AI training purposes. However this is hardly feasible in DOSN, culturally and technically, since there is not central authority guaranteeing data availability and DOSN are designed for an improved privacy. Data replication schemes with cryptocurrencies incentives[20, 32] have been proposed for the collaborative hosting of data, but they remain complex to implement and secure.
Data availability is not guaranteed by centralised OSN[8], they may arbitrarily remove any content (e.g. Google has the rights to delete inactive accounts[3]). Since data availability is not guaranteed in centralised OSN, we argue that it is not necessary for decentralised OSN. In our approach, content that is not popular enough to be hosted by anyone can legitimaly be removed from the OSN.
Securing DOSN is also an important challenge as they can be victim of various attacks such as communication privacy breaches [17], attacks on decentralisation [39] or spam. The integrity and privacy of data and communication must be guaranteed without the help of a centralised trusted third-party while protecting from toxic content.
This paper provides the following contributions: We propose a customizable model for social networks and show how it can be decentralised. We identify the security properties of our model and the counter-measures that can be added. We propose an architecture and its technological stack for an implementation of our model.
This paper is structured as follow: In section 2 we review the current OSN approaches. Section 3 presents our proposed model for social networks. Section 4 presents our architecture and the associated technology stack.

2 Related Works

In this section we review the current approaches to decentralising OSN. We identify the core functionalities and security properties an OSN may feature.

2.1 Decentralizing OSN

We distinguish three types of OSN in terms of architecture (Figure 1):

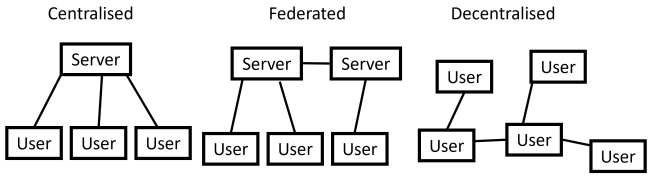

Figure 1. Types of OSN

Centralised – OSN hosted and administrated by a single entity (e.g. Twitter, Facebook). They may be distributed on multiple data-centers for better scalability and resilience as they are limited by the providers infrastructure. They show

privacy and security issues as the administrators have a full control on the OSN and may block or read any content.
Federated – Multiple OSN are hosted and administrated by different entities following the same protocol[35] (e.g. mail protocol, Mastodon). Users of an instance can interact with users of another, for that they are identified with a *user@server* identifier. The W3C [9] has worked on the normalization of such protocols.
Decentralized – The users interact directly with each other without using centralized hosting instances [27]. Each peer is in charge of hosting a part of the social network.
Data hosting is the main challenge of decentralized social network as it must guarantee data availability of the content with a high client churn. Distributing hosting is sensitive to Sybil attacks[14] where a single entity creates multiple identities to get more storage. Decentralized OSN usually limit Sybil attacks on free hosting and spam by using monetary incentives in forms of cryptocurrency[20].
To design DOSN, two technologies are used in existing projects (Figure 2):
Blockchain [21, 40, 41] approaches write all of the content in a distributed ledger copied over all of the OSN users. Users share their content (transaction) by propagating it by gossip in the network. Every x amount of time, the new content is added to the ledger as a block pointing to the previous block of the blockchain. All of the hosted content is copied by each peer, this drastically limits the scalability of the OSN in the number and size of messages.
Distributed Hash Tables (DHT)[28] approaches allow to search for clients sharing a specific content (user, file, service) using its identifier (e.g. file hash in IPFS[38]). The hash table is distributed between the users to allow for resilient and fast look ups. DHT based OSN often feature a protocol to shard the user content into multiple clients for redundancy[16].

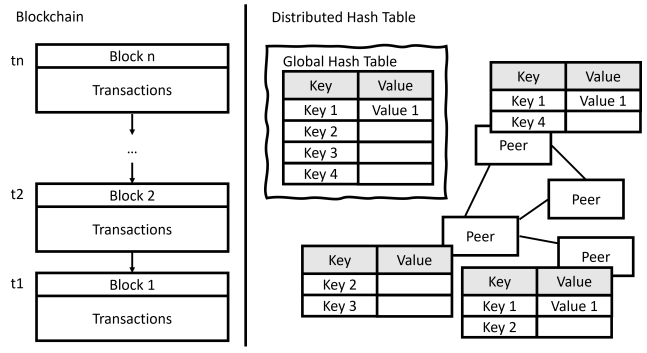

Figure 2. Blockchains and Distributed Hash Tables

2.2 Functionalities of OSN

OSN allow for different types of interactions that may be themselves a composition of other interactions. We identified in OSN the following functionalities:

Communities/recursive: grouping the users in sub-communities with their own set of rules and functionalities. A user may be part of multiple sub-communities as in Discord with its servers.
Profile: a public and unique file is associated to a user, only he can modify. Anyone can find the profile knowing the user (e.g. Facebook wall, profile picture).
Direct messages: sending direct messages be it text or streams between two users (e.g. text, phone/video call).
Groups : direct messages are sent to a predefined group of users (e.g. chat groups). They follow the publish-subscribe (PubSub)[10] pattern where users can subscribe to a subject of interest and receive messages pertaining to this subject.
Inter-OSN interactions : interacting in an OSN with the content of another (e.g. using Facebook authentication to log in another website, sharing a YouTube video on Facebook). The W3C Fedpub standard[1] aims to standardize inter-OSN interactions.
Boards: are spaces where users can write. All of the content written on a board can be found by consulting it (e.g. product comments on e-commerce sites, forum discussions, 4chan boards).
Content Search : searching for a content either by identifier or with semantic search. (e.g. DHT to search content by hash [28, 38], search engines for semantic search[30])
Transactions: establishing contracts with other users, especially to buy or sell content. (e.g. Facebook Marketplace, Ethereum transactions and smart contracts [40])
Other functionalities that improve the user experience include:
Username: provide the possibility to look for user and content through a human readable name (e.g. usernames instead of public key, domain names instead of IP address for websites).
Timestamp: provide a proof of the content existing at a specific time[15].
Other specific functionalities can be added for specific cases by combining existing functionalities (e.g. making votes and polls using boards where each user can only write once).

2.3 Security of OSN

In this subsection, we identify the security properties needed in OSNs.
Integrity – *Content integrity*: the recipient of a content should be able to verify it was produced by the right person and not altered in the transfer.
Functionality isolation: functionalities should not impact the working of other (e.g. the board functionality should not interfere with direct messages). An automatism on a specific functionality should be isolated from the rest of the functions (e.g. a spelling checker not interfering with audio communications).

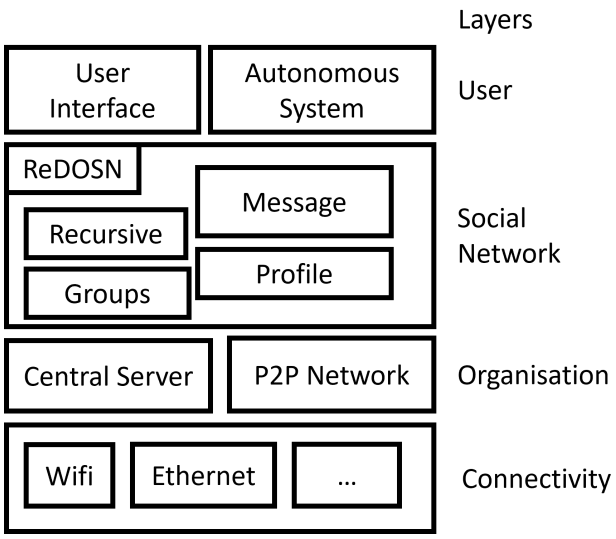

Figure 3. ReDOSN software stack

Availability – *Function availability*: functionalities should keep working in case of system failure and scale in the number of users or exchanged content. *Data availability*: The user content on the OSN should be available at anytime.
Privacy – *Exchanged content*: content should be accessible only to those authorized by the author. *Behaviour*: user behaviour like content consulted and times of connections should not be accessible to others[17].
Malicious users – *Toxic and illegal content* should be hidden from users. The OSN should be able to detect and limit the impact of spam.

3 Proposed model

We propose ReDOSN (Recursive Decentralised Online Social Networks) a model to create customizable decentralised social networks. It is a middleware (figure 3) that allows to reproduce most social networks with an API for either a human user's GUI or for autonomous systems to cooperate. It provides the functions identified in subsection 2.2 based either on a decentralised or centralised organisation of the infrastructure, connecting the members of the organisation using different network protocols.

3.1 Recursive OSN

As illustrated in figure 4, we consider in ReDOSN that a social network is a composition of sub social networks or communities with their own rules. A user participates in multiple OSN as in Helios [23]. Each OSN has its own functionalities (customization), access rights (e.g. a specific house network) and connectivity (e.g. Wifi, 5G). Social Networks can be centralised or decentralised depending of the security and performance needs, for example a road traffic network

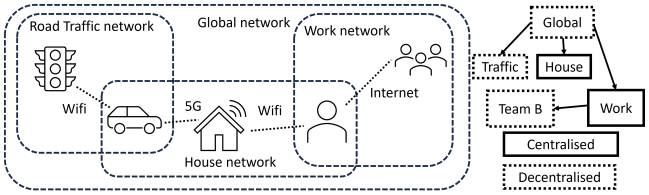

Figure 4. Composition of OSN

may be decentralised for better latency and a house network can be centralised around the WiFi router.
To handle multiple OSN and the possibilities of sub-OSN, we propose a recursive approach where the OSN are composed as a tree (figure 4). This account for the possibility to participate in multiple independent OSN at the same time, or to participate in a sub-OSN inheriting its parent properties (e.g. security policy).
Figure 5 shows how ReDOSN is organised as a tree where functionalities are attached to OSNs, those OSN may handle the functionalities in a centralised or decentralised way. OSNs feature a recursive functionality in charge of handling the sub-OSNs. This defines a path to reach the functionality of a sub-OSN from the root OSN. Calling a remote client follows then the format *functionality:user@OSN1/OSN2*.
Figure 5 illustrates simplified visions of WhatsApp, Microsoft Teams and Reddit with the ReDOSN model. WhatsApp features direct messages and group messages, the profile is used to set a profile picture. Teams features direct messages, users can join teams which are sub-OSNs with their own access rules and discussion channels (groups). Reddit features communities called subreddit which work as sub-OSN with their own rules, they use a main board to list the post made in the community, each post is a board that features the post and its associated comments.

3.2 Decentralising ReDOSN

We observed in the main decentralized content sharing tools like IPFS [12] that data availability is not guaranteed. Hosting depends of the will of the participants to keep content available. So, contrary to other DOSN, we do not aim to guarantee availability of user data. To decentralise ReDOSN, we consider here that content not popular enough to be hosted by anyone can legitimately be lost with limited impact on user experience.
As in Bluesky [22], we consider that each user is identified by a Decentralised identifier (DID) [19]. It can be an ID provided by a third party or a public key working as an ID. This allows to establish encrypted and authenticated communications between users. A sub network can either use its own certificate with a hierarchy of certificates as in https or use the certificate through its parent network.
Providing the functionalities described in sub-section 2.2 for centralised OSNs is a mature field. We describe here how

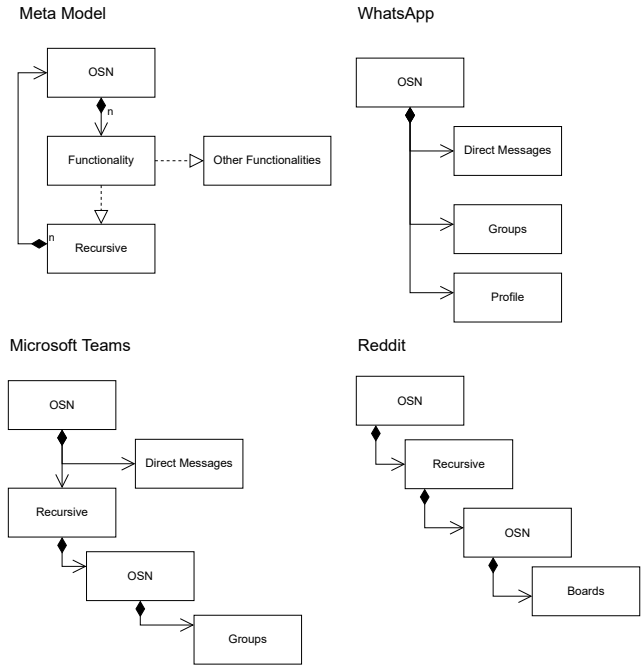

Figure 5. ReDOSN model and illustrations

they can be implemented in a decentralised way using existing P2P solutions. Those technologies are resilient and scalable but at the cost of reduced data availability. Username, Timestamp and Transaction functions are exceptions which use blockchains at their core, but at the cost of limited scalability in the number of calls [41].

We use the Kademlia[28] DHT to find the resources on the network in a $log(n)$ time where n is the number of users. Kademlia has been extensively tested in terms of efficiency and resilience by its implementation in IPFS [38].

Profile: We use an IPNS [5] approach, the hash of the user public key is used to point to the static content. The changes are broadcasted on the network and signed using the user private key to guarantee integrity and prove that the update is the most recent. To provide data availability, policies such as hosting friends profiles when the user is offline can be implemented.

Direct messages: to send messages to other users, the DHT can be used to find the IP address of a user from its public key. Indirect messages can be sent by asking a tier to host and forward a message.

Groups: groups are implemented using IPFS Pub-Subs [10], the group is a topic to which users subscribe. Other subscribers to a specific topic are found through the DHT. The messages are broadcasted through gossiping, this reduces the load on the sender: it will not have to send a message to each member individually. Access rights to groups can be set by sharing a common encryption key between allowed members and setting a list of participants to prevent message gossiping from leaving the group.

Inter-OSN interactions: the user's DID can be used to identify a user on multiple different OSN. Content from other OSN can be shared using their URI. Other OSN may access content through a trusted user acting as a gateway to the p2p network.

Boards: decentralised databases like orbitDB[7, 26]can be used to store messages of boards. Each board (e.g. forum discussion) should be considered as its own database that is collaboratively hosted by those that interacted with it. Meta-boards can be created to list all of the sub-boards.

Content Search: a DHT can be used to search for content by its identifier (e.g. searching with a hash). Decentralised semantic search is an active field of research with tools like YaCy and De-DSI[30].

For functionalities like **usernames** and **timestamping** there is a need for a trusted third party, either centralised or with blockchains approaches (e.g. Ethereum name service[2], smart contracts for time stamps[15]), but at a scalability cost.

3.3 Security countermeasures in a decentralized setting

We described in the previous section how our model can be implemented in a decentralised way using existing technologies. We identify here the security countermeasures applicable to guarantee the security properties identified in sub-section 2.3.

DHT security as been explored separately [39], we consider that they are secure against attacks aiming to prevent honest nodes from discovering and interacting with each other. The use of hashes to identify content and public keys for users guarantees the integrity of content and the user identity.

Integrity – Content integrity: exchanged content is signed by its sender using its private key. Boards and groups can set access rights to control which user can write and what they can write. **Functionality isolation**: the OSN and functionalities are organised as a tree this allows to define access rights between functionalities and isolate them (e.g. a functionality can call sub OSN functionalities but not the opposite). **Consensus**: to prevent Sybil attacks[14] where a single entity creates multiple identity to get more vote rights. The core functionalities are based on a DHT that does not need for consensus. Blockchain based functionalities use Proof of X approaches. A list of identities allowed to vote can also be set in other cases.

Availability – DHT based functionalities remain available by exploiting the Kademlia DHT's resilience and scalability. Data availability is guaranteed by replication [36]. For example using hosting policies where users automatically participate in hosting the content they liked or from their friends. A user can also use paid tiers (e.g. Pinata or Filecoin for IPFS) to host their content. Finding content that is provided by someone in the OSN is guaranteed by the

DHT. Blockchain based functionalities have their availability guaranteed by the blockchain.

Privacy – Exchanged content: message content is encrypted using the recipient public key. Access control policies on groups and boards can be set where updates are only shared between a list of allowed users that also share a common encryption key. As in centralised OSN, were a single member to be corrupted, the whole content of the groups and boards could be leaked. Cryptographic schemes can be set for specific cases (e.g. polls and votes, computing the mean of a private value[25]) to guarantee the privacy.

Behaviour: network anonymization tools like TOR or mixnets [13, 34] and the use of temporary identities can limit network traffic monitoring. Other obfuscation methods like dummy traffic, creating fake content requests and content padding can be added to limit privacy leakages[17].

Malicious user – To tackle toxic content and spam, reactive approaches can be applied where a trusted authority that publishes lists of whitelisted/blacklisted content/user. This has been implemented in the ublock origin browser extension to block adds through filtering lists, or in the Bad Bits Denylist for IPFS files. This is also the approach adopted in Bluesky with labels emitted by moderation services [1]. Proactive approaches can be applied by a user or a group of users using a reputation system[18], setting real-time detectors using rule-based systems or machine learning to detect and block users having unusual behaviors (e.g. sending thousands of messages, words featured in message content).

Confidential workloads – In cases where a trusted central authority is needed to run confidential workloads, confidential and trusted compute can be delegated to networks of TEE (Trusted Execution Environment) like Ekiden[11]. TEEs are hardware secured execution environments with encrypted memory that guarantees the privacy and integrity of data an execution. They feature remote attestation schemes that allow to remotely verify that a specific code is running in a TEE.

4 Implementation

In section 3 we described the ReDOSN model and identified how the functionalities can be decentralised and secured.
In this section we review how it can be implemented using existing technologies.

4.1 Architecture

Figure 6 shows an UML representation or our proposed architecture. Each functionality has its own methods that are independent of their implementation (e.g. sending a direct message will provide the same interface in a centralised and decentralised implementation). An OSN has a policy defining access rights and the configuration of the functionalities. The private key for communications associated to the DID is held by the OSN object which grants its usage to functionalities and sub-OSNs. The recursive functionality allows to join multiple sub-OSNs that can have different implementations of functionalities (centralised or decentralised).
Receiving messages, either in direct messages or in groups is done by setting a handler function that will be called at each received message. Groups and Boards feature a policy which is used to define specific behaviors like access rights. A user may join a group by subscribing, it can create new boards or read/write on existing ones.
Inter-OSN interactions feature the **GET** and **POST** calls defined in the ActivityPub standard [1], they allow to pull interactions received from other OSN and push content outside to another OSN.
Search can either be done by looking for the content associated to an ID or semantic search, looking for content with a certain meaning. New content can be set to be searchable using the index method, with a policy to set access rights.
A user has a profile with content it may modify, it can also find the content of other users using their identifier.
A user can set a unique human readable name for itself and others ID from their name. A timestamp of a content or a transaction can be generated and existing timestamps/transactions can be verified.

4.2 Technology stack

Figure 6 shows how each functionality can be implemented using existing decentralised technologies and how those technologies linked to each other. We choose technologies that can run directly in the browser, allowing to run on a multitude of environments and specially to run on user device without any software installation. We consider here that users are identified on the OSNs by their public key, this can be extended using veramo for DID.
They are two main families of decentralised technologies proposed:

Blockchain based technologies are built with the Ethereum blockchain at the core. Operations induce a monetary cost to use the smart contracts but guarantee data availability. The library web3.js is used for interaction with the Ethereum blockchain. Etherum name allows to associate a human readable name to a public key, Ethereum attestation service [2] provides time-stamping based on the Ethereum blockchain.

DHT based technologies induce no monetary cost but provide no data availability guarantee. The libp2p library provides a DHT to find other peers and content, it allows authenticated and encrypted direct communication for direct messages. It is fully available to run directly in web browsers using WebRTC and works at the core for IPFS implementations. For groups it provides a pubsub that broadcast messages to the users following a specific subject. IPNS is a sub-system allowing for dynamic pointers in IPFS instead

[1] https://docs.bsky.app/docs/advanced-guides/moderation

[2] https://docs.attest.org/docs/tutorials/timestamping-attestations

of static ones by file hash, this allows to build a profile that can only be modified by the user. OrbitDB is a key-value database built on top of IPFS that provides a global ledger users can read and write on to build boards. Semantic search can be done using solutions such as YaCy and De-DSI.
In both approaches, the DOSN are not directly accessible with an URL, they need a gateway for interactions with other OSN like cloudflare gateway acts for IPFS.

5 Conclusion

We proposed a model to create decentralized social networks that is completely customizable. We identified the key functionalities working as building blocks for any OSN. Arguing that data availability is not a hard requirement, we identified the decentralization techniques that allow for efficient, scalable and secure OSN functionalities. We showed how our model can be implemented, identifying existing technologies for its decentralization and highlighting their ability to be deployed directly in varied environments and especially in web browsers. However, an implementation or ReDOSN and benchmarks of the technologies proposed are necessary to verify that beyond their architecture guarantees, the implementations answer to the performance requirements of the OSN use-case.

References

[1] [n. d.]. ActivityPub. https://www.w3.org/TR/activitypub/.

[2] [n. d.]. ENS. https://ens.domains.

[3] [n. d.]. Inactive Google Account Policy - Google Account Help. https://support.google.com/accounts/answer/12418290?hl=en&sjid=6046569782917084002-EU.

[4] [n. d.]. Introducing Twitter Alerts. https://blog.x.com/en_us/a/2013/introducing-twitter-alerts.

[5] [n. d.]. IPFS Concepts: IPNS. https://docs.ipfs.tech/concepts/ipns/.

[6] [n. d.]. More details about the October 4 outage. https://engineering.fb.com/2021/10/05/networking-traffic/outage-details/.

[7] [n. d.]. OrbitDB. https://orbitdb.org/.

[8] [n. d.]. Our Digital History Is at Risk | Internet Archive Blogs. https://blog.archive.org/2023/02/07/our-digital-history-is-at-risk/.

[9] [n. d.]. Social Web Protocols. https://www.w3.org/TR/social-web-protocols/.

[10] Pedro Agostinho, David Dias, and Luís Veiga. 2022. SmartPubSub: Content-based Pub-Sub on IPFS. In *2022 IEEE 47th Conference on Local Computer Networks (LCN)*. 327–330. https://doi.org/10.1109/LCN53696.2022.9843795 ISSN: 0742-1303.

[11] Raymond Cheng, Fan Zhang, Jernej Kos, Warren He, Nicholas Hynes, Noah Johnson, Ari Juels, Andrew Miller, and Dawn Song. 2019. Ekiden: A Platform for Confidentiality-Preserving, Trustworthy, and Performant Smart Contracts. In *2019 IEEE European Symposium on Security and Privacy (EuroS P)*. 185–200. https://doi.org/10.1109/EuroSP.2019.00023

[12] Erik Daniel and Florian Tschorsch. 2022. IPFS and Friends: A Qualitative Comparison of Next Generation Peer-to-Peer Data Networks. http://arxiv.org/abs/2102.12737 arXiv:2102.12737 [cs].

[13] Roger Dingledine, Nick Mathewson, and Paul Syverson. [n. d.]. Tor: The Second-Generation Onion Router. ([n. d.]), 18.

[14] John R. Douceur. 2002. The Sybil Attack. In *Peer-to-Peer Systems*, Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Peter Druschel, Frans Kaashoek, and Antony Rowstron (Eds.). Vol. 2429. Springer Berlin Heidelberg, Berlin, Heidelberg, 251–260. https://doi.org/10.1007/3-540-45748-8_24 Series Title: Lecture Notes in Computer Science.

[15] Gabriel Estevam, Lucas M. Palma, Luan R. Silva, Jean E. Martina, and Martín Vigil. 2021. Accurate and decentralized timestamping using smart contracts on the Ethereum blockchain. *Information Processing & Management* 58, 3 (May 2021), 102471. https://doi.org/10.1016/j.ipm.2020.102471

[16] Kalman Graffi and Newton Masinde. 2021. LibreSocial: A peer-to-peer framework for online social networks. *Concurrency and Computation: Practice and Experience* 33, 8 (2021), e6150. https://doi.org/10.1002/cpe.6150 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.6150.

[17] Benjamin Greschbach, Gunnar Kreitz, and Sonja Buchegger. 2012. The devil is in the metadata — New privacy challenges in Decentralised Online Social Networks. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*. 333–339. https://doi.org/10.1109/PerComW.2012.6197506

[18] Ferry Hendrikx, Kris Bubendorfer, and Ryan Chard. 2015. Reputation systems: A survey and taxonomy. *J. Parallel and Distrib. Comput.* 75 (Jan. 2015), 184–197. https://doi.org/10.1016/j.jpdc.2014.08.004

[19] Felix Hoops, Alexander Mühle, Florian Matthes, and Christoph Meinel. 2023. A Taxonomy of Decentralized Identifier Methods for Practitioners. In *2023 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*. 57–65. https://doi.org/10.1109/DAPPS57946.2023.00017 ISSN: 2835-3498.

[20] Cornelius Ihle, Dennis Trautwein, Moritz Schubotz, Norman Meuschke, and Bela Gipp. 2023. Incentive Mechanisms in Peer-to-Peer Networks — A Systematic Literature Review. *Comput. Surveys* (Jan. 2023). https://doi.org/10.1145/3578581 Just Accepted.

[21] Le Jiang and Xinglin Zhang. 2019. BCOSN: A Blockchain-Based Decentralized Online Social Network. *IEEE Transactions on Computational Social Systems* 6, 6 (Dec. 2019), 1454–1466. https://doi.org/10.1109/TCSS.2019.2941650

[22] Martin Kleppmann, Paul Frazee, Jake Gold, Jay Graber, Daniel Holmgren, Devin Ivy, Jeromy Johnson, Bryan Newbold, and Jaz Volpert. 2024. Bluesky and the AT Protocol: Usable Decentralized Social Media. https://doi.org/10.1145/3694809.3700740 arXiv:2402.03239 [cs].

[23] Kevin Koidl, Ville Ollikainen, and Jarkko Kuusijärvi. 2024. HELIOS a Decentralized Online Social Network Framework. In *2024 IEEE International Symposium on Technology and Society (ISTAS)*. 1–8. https://doi.org/10.1109/ISTAS61960.2024.10732337 ISSN: 2158-3412.

[24] Mozhgan Malekshahi Rad, Amir Masoud Rahmani, Amir Sahafi, and Nooruldeen Nasih Qader. 2020. Social Internet of Things: vision, challenges, and trends. *Human-centric Computing and Information Sciences* 10, 1 (Dec. 2020), 52. https://doi.org/10.1186/s13673-020-00254-6

[25] Mohamad Mansouri, Melek Önen, Wafa Ben Jaballah, and Mauro Conti. 2023. SoK: Secure Aggregation Based on Cryptographic Schemes for Federated Learning. *Proceedings on Privacy Enhancing Technologies* 2023, 1 (Jan. 2023), 140–157. https://doi.org/10.56553/popets-2023-0009

[26] Alessandro Margara, Gianpaolo Cugola, Nicolò Felicioni, and Stefano Cilloni. 2023. A Model and Survey of Distributed Data-Intensive Systems. *Comput. Surveys* 56, 1 (Aug. 2023), 16:1–16:69. https://doi.org/10.1145/3604801

[27] Newton Masinde and Kalman Graffi. 2020. Peer-to-Peer-Based Social Networks: A Comprehensive Survey. *SN Computer Science* 1, 5 (Sept. 2020), 299. https://doi.org/10.1007/s42979-020-00315-8

[28] Petar Maymounkov and David Mazières. 2002. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In *Peer-to-Peer Systems*, Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Peter Druschel, Frans Kaashoek, and Antony Rowstron (Eds.). Vol. 2429. Springer Berlin Heidelberg, Berlin, Heidelberg, 53–65. https://doi.org/10.1007/3-540-45748-8_5 Series Title: Lecture Notes in Computer

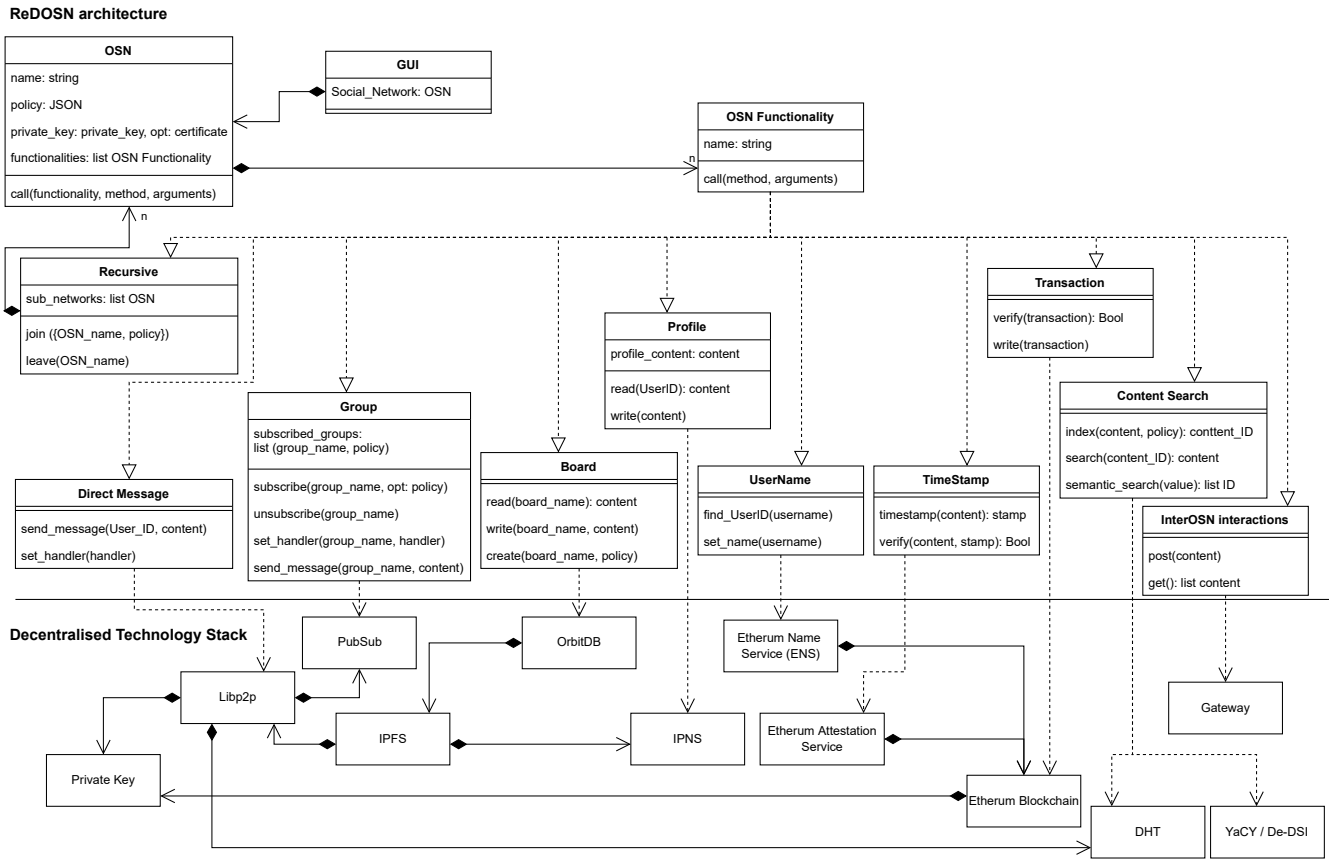

Figure 6. ReDOSN architecture and technology stack

Science.

[29] Chiara Valentina Misischia, Flora Poecze, and Christine Strauss. 2022. Chatbots in customer service: Their relevance and impact on service quality. *Procedia Computer Science* 201 (Jan. 2022), 421–428. https://doi.org/10.1016/j.procs.2022.03.055

[30] Petru Neague, Marcel Gregoriadis, and Johan Pouwelse. 2024. De-DSI: Decentralised Differentiable Search Index. In *Proceedings of the 4th Workshop on Machine Learning and Systems*. 134–143. https://doi.org/10.1145/3642970.3655837 arXiv:2404.12237 [cs].

[31] Thomas Paul, Antonino Famulari, and Thorsten Strufe. 2014. A survey on decentralized Online Social Networks. *Computer Networks* 75 (Dec. 2014), 437–452. https://doi.org/10.1016/j.comnet.2014.10.005

[32] Yiannis Psaras and David Dias. 2020. The InterPlanetary File System and the Filecoin Network. In *2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*. 80–80. https://doi.org/10.1109/DSN-S50200.2020.00043

[33] Aravindh Raman, Sagar Joglekar, Emiliano De Cristofaro, Nishanth Sastry, and Gareth Tyson. 2019. Challenges in the Decentralised Web: The Mastodon Case. In *Proceedings of the Internet Measurement Conference*. ACM, Amsterdam Netherlands, 217–229. https://doi.org/10.1145/3355369.3355572

[34] K. Sampigethaya and R. Poovendran. 2006. A Survey on Mix Networks and Their Secure Applications. *Proc. IEEE* 94, 12 (Dec. 2006), 2142–2181. https://doi.org/10.1109/JPROC.2006.889687

[35] Gabriel Silva, Larissa Reis, Antonio Terceiro, Paulo Meirelles, and Fabio Kon. 2017. Implementing Federated Social Networking: Report from the Trenches. In *Proceedings of the 13th International Symposium on Open Collaboration*. ACM, Galway Ireland, 1–10. https://doi.org/10.1145/3125433.3125455

[36] Evjola Spaho, Leonard Barolli, and Fatos Xhafa. 2014. Data Replication Strategies in P2P Systems: A Survey. In *2014 17th International Conference on Network-Based Information Systems*. 302–309. https://doi.org/10.1109/NBiS.2014.74 ISSN: 2157-0426.

[37] Inside Track staff. 2024. Microsoft Teams increases collaboration in the modern workplace at Microsoft. https://www.microsoft.com/insidetrack/blog/microsoft-teams-increases-collaboration-in-the-modern-workplace-at-microsoft/

[38] Dennis Trautwein, Aravindh Raman, Gareth Tyson, Ignacio Castro, Will Scott, Moritz Schubotz, Bela Gipp, and Yiannis Psaras. 2022. Design and evaluation of IPFS: a storage layer for the decentralized web. In *Proceedings of the ACM SIGCOMM 2022 Conference (SIGCOMM '22)*. Association for Computing Machinery, New York, NY, USA, 739–752. https://doi.org/10.1145/3544216.3544232

[39] Guido Urdaneta, Guillaume Pierre, and Maarten Van Steen. 2011. A survey of DHT security techniques. *Comput. Surveys* 43, 2 (Feb. 2011), 8:1–8:49. https://doi.org/10.1145/1883612.1883615

[40] Jie Xu, Cong Wang, and Xiaohua Jia. 2023. A Survey of Blockchain Consensus Protocols. *Comput. Surveys* (Jan. 2023). https://doi.org/10.1145/3579845 Just Accepted.

[41] Qiheng Zhou, Huawei Huang, Zibin Zheng, and Jing Bian. 2020. Solutions to Scalability of Blockchain: A Survey. *IEEE Access* 8 (2020), 16440–16455. https://doi.org/10.1109/ACCESS.2020.2967218