Decentralized Social Networks: Pros and Cons of the Mastodon Platform

Charlot R. Shaw
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
shawx538@morris.umn.edu

ABSTRACT

Mastodon is a federated social network designed to allow diversity in users, small but interconnected communities, and user-ownership of their data. It builds on standardized protocols, and follows a decentralized model. This architecture allows for more truly public online spaces, and has a number of difference from its closest popular alternative, Twitter. However, the Mastodon network is negatively affected by various forces, and is potentially more fragile than its distributed nature suggests.

Keywords

Mastodon, Social Networks, Decentralized Social Networks

1. INTRODUCTION

Online social networks have become a common platform the world over, a means of communicating and a medium of discovery. Despite their global reach, the most common social networks operate on a centralized model with a single platform provider, operating the entire platform. For example, there is only one Twitter service, and it is the one managed by Twitter. However, the centralized model is not the only option. A decentralized social network known as Mastodon has been operating since 2016, and has accumulated a mesh of independent communities and users (over a million at time of writing). Mastodons decentralized structure changes how its users interact with one another, how the social network grows, and how it behaves under stress. Although far from flawless, Mastodon, and its continuing growth show that the centralized model is not the only option for online social networks.

2. MASTODON

2.1 Background

2.1.1 Federation

Federation is the concept that independent servers can communicate through a common protocol and work together without sacrificing independence. A common example would be email, in which different email providers can send messages between themselves on behalf of their users. A user with a gmail address can message someone with a protonmail address just as easily as they could message a fellow gmail user, and there is no need for a a supervising entity that manages all email.

Email is one exmaple of a federated protocol, but there are others. The World Wide Web Consortium(W3C) published the **ActivityPub** standard [2] in 2018, a federated protocol for social media. This standard is not a social network, it is instead an agreed upon set of signals and actions that allow common interactions in social media, such as following a user, liking some content, or sharing content of one's own. Software that implements the standard can inter-operate with one another, and form a federated network. Data from one program is intelligibly presented on users in other programs, and connections between users on different implementations is as simple as connecting to a user on the same implementation. This network is colloquially termed the Fediverse (Federated universe), and this paper will use the term when discussing the **ActivityPub** network as whole.

2.1.2 Mastodon

Mastodon is a microblogging software built on subset of the **ActivityPub** standard, and published under the Affero General Public License 3.0 in 2016. It will be familiar to users of Twitter, with users posting short messages, known as Toots, and receiving Toots written by other users they are subscribed to. With federation, users sign up for a specific server, also known as an instance. They receive a feed of public Toots written by other users on the server, this is known as the local timeline. Additionally, they see all public Toots from users anyone on the server has subscribed to. This is the federated timeline, and provides the most complete vision of the fediverse as a whole for an individual user. As there is no managing entity, there is no server which sees every message in the fediverse, or even the subset of the fediverse that consists only of Mastodon servers. Because of this, there is no truly global view, each user can only see what their instance can see. Some servers host bots, Mastodon account controlled by software, that systematically follow as many other users as possible. As these bots exist on the server, the content the bot has followed appears in the federated timeline, widening the servers perspective on the fediverse.

2.1.3 Federation Follower Example

Suppose we have a mastodon instance, running at the

UMM CSci Senior Seminar Conference, November 2020 Morris, MN.

URL *xeno.space*, with a user, Alex. Alex finds a blogger they like, Babbar, has posted about his Mastodon account on *yottabyte.rocks*, and wants to follow him. When Alex tells their server this, *xeno.space* opens a connection to *yottabyte.rocks*, and requests *yottabyte.rocks* let *xeno.space* know whenever the account *babbar@yottabyte.rocks* posts a public message, and that Babbar has a new follower named *alex@xeno.space*.

The next time Babbar posts something that is marked as public, *yottabyte.rocks* sends a message to *xeno.space*, which then notifies Alex. *xeno.space* also places Babbars post in the federated timeline, for other *xeno.space* users to see.

2.2 Graphs

The connections between users and those they follow, or between intercommunicating servers form graphs, collections of nodes with connections between them. This data is one of the best ways available to understand the Mastodon network, but we need to make sure we understand what graph we are referring to, as there are two interconnected ones that shape the Mastodon network's activity.

The network graph consists of the connections made by servers when their users interact. These are carried out through HTTP(s) requests, and routed by the DNS servers that underpin the internet. They are utilitarian in nature, created by the software, and as much as possible hidden from the user. The content of these messages has forms specified by the **ActivityPub** standard, thus allowing the different servers to parse the messages automatically.

The social graph is the relations between users: who follows who, who has liked which posts, etc. This graph is made of the connection records in each instance's database. They are created as reactions to signals received through the network graph, and the network graph forms and abandons connections between servers based on them. These records are available on request, to verify that *babbar@yottabyte.rocks* really is being followed by *alex@xeno.space* for example.

The network graph is complex, but not altogether novel. It consists of the connections between servers required to exchange the messages required by the social graph. The social graph is constrained by the structure of the network graph, with users being able to discover new content only by their instance encountering it while serving a fellow users request, no matter how popular that content is elsewhere in the network.

Mastodon has not implemented a recomender algorithm to steer user connections; this community driven discovery process is the only way users encounter each other on-platform. Thus makes Mastodon a valuable data-source for the behavior of communities online, especially decentralized ones. [7]

However, the decentralized nature of Mastodon does make study difficult in some ways. There is no way to access the network wholesale through negotiation with a single entity. Instead, researchers carry out automated explorations of public instance and user data, starting with as broad a list of servers as possible, and following the connections between users to discover new instances and thus map the network. [7]

Some of the crawls considered in this paper were carried out over six months starting in August of 2018, and covered 479,425 users, an estimated 46% of all users in the Mastodon network. [7] The other crawl considered was carried out between April 2017 and July 2018, and found 239 thousand unique users and 62% of all toots in the crawled graph. The remaining could not be accessed due to the instances configuring themselves to block crawling, or the toots being non-public.

These datasets include both snapshots of the network, and long term data on how Mastodon grew and changed. Although not discussed thoroughly here, the overall trend is steady growth, with spikes occurring that likely correspond to popular campaigns on other platforms, including #DeleteFacebook, or with publication of article about Mastodon in mass media, such as Wired Magazine. [3]

3. EFFECTS OF MASTODONS MODEL

With the rough structure of Mastodon understood, and an introduction to the data out of the way, we can now talk about what this means. First, the main issue that Mastodon sets out to address, and how it does so. Second, the differences in its structure compared to Twitter, its closest popular analog. Third, some issues with Mastodons federation model, and some proposed strategies to mitigate them.

3.1 Benefits

To understand what benefits Mastodon provides, and what issues it was designed to avoid, let's compare it with its closest analog, Twitter. Though their internal structures are vastly different, they both are microblogging platforms, focused on short, public messages. Mastodon improves on Twitter in three ways we will enumerate here: public space, privacy and diversity.

3.1.1 Public Space

Twitter is definitely a public space in one sense, with any action occurring before a potentially vast audience. However, as Aral Balkan observes [1], Twitter is closer to a shopping mall. The public is admitted to the property, and quite often people carry out social interactions there. However, the purpose of a mall is not to facilitate a public space, but rather to achieve profits. Socializing at the mall serves this motive, but if the socialization hampers the commerce of the mall, it's in the mall's interests to put a stop to it. Similarly, the mall focuses its amenities on those who contribute to profits. Twitter is in the position of a mall here, maintaining an appearance of being an open and public space, while maintaining the same profit motive.

Mastodon at first glance has merely shuffled this problem around, with each and every instance being a mall-like space. However, the Mastodon network as a whole doesn't belong to any one instance. The public space Mastodon creates is the network between every node, which is a open protocol.

3.1.2 Privacy

Twitter holds all its user's data, and tracks them even offsite. [5] Their motivation for this is profit, through selling that data, access to that data, or insights derived from that data. Privacy from this standpoint, is something between users. Although options like blocking unwanted contacts, or protected Twitter accounts offer more privacy, none of them protect the user from Twitter itself. Consider two scenarios: If a hundred random people learned a different fact about you, that would be potentially unpleasant. However, if one person learned one hundred different facts about you, the situation is worse. The odds of something you would

not wish known is a hundred times greater. The amount of information that person can infer, correctly or not, is also increased. In Twitter's case, this same entity is also collecting information on everyone else, not just you. Twitter may claim that no humans read this data, or that it works without exposing user information. The author of this paper does not find comfort in the reassurance that our surveillance is automated and industrialized.

Being decentralized, Mastodon already has a vast advantage in terms of truer privacy. Again, issues arise with the administrators of an instance having access to the data of the instance users, but there is no central entity that sees all information. Furthermore, each instance has less users than Twitter, and so less data to offer if they wished to monetize it. If an instance did monetize its user's data, Mastodon has tools to allow easy switching of instances, so users would have little reason to stay on an instance that did not respect their privacy.

This creates an environment where instance owners have strong incentives to protect users information, and users are not locked into any particular host.

3.1.3 Diversity

Twitter, as singular platform, is placed in the unenviable position of trying to please everyone. With all users existing in more or less the same virtual space, there is bound to be conflict between user's different styles of communication, expectations of behavior, etc. assuming that all parties on the platform are acting in good faith. Twitters terms of service, and various moderation systems are necessarily generic, and whether they are fair guidelines or well applied is beyond the scope of this paper. It is sufficient to note that they exist, and must necessarily take a one-size-fits-all approach.

Mastodon, by its decentralized model, does not have every one on the same platform. Each instance can create and enforce its own code of conduct, with users being able to opt out by switching instances or hosting their own instance if they find themselves in an uncomfortable position. Instance moderators also have effective tools to filter content coming from other servers, so content coming in through the federation can be rejected if it doesn't fit the code of conduct.

This raises an issue of siloization, of users only encountering those who have compatible viewpoints, and creating a fractured perception of reality. This issue is wider than Mastodon, and occurs on most social networks, including centralized ones. The author acknowledges this, but did not focus their research on this issue specifically, and so cannot state whether or not Mastodon handles this issue better or worse than other platforms.

3.2 Differences

Not all of Mastodon's features are clearly beneficial or detrimental. The instance structure provides explicit groups whose interrelations are of interest. Additionally, instances have individual traits, some explicit and others implicit that are visible in the social graph.

3.2.1 Instance Topics

Starting with instances, Zignani et al. collected the topics that instances chose to self-describe with. A very large group of these describe themselves as "General", or do not state any topics at all. Unsurprisingly, for an open source network growing by word of mouth, the specific topics "Technology" and "Programming" were the most common topics, with various forms of science cumulatively taking second place. Instances dedicated to various forms of art, including "writing" and "anime" were also well represented. [7]

Aside from the amount of servers dedicated to a topic, there is also interesting patterns in how users interact with topics. For example, instances dedicated to "Journalism" or "General", are more common than average, but have lower user counts per-instance. [3] From this, one could suppose that servers with less to differentiate themselves from others on a common topic tend not to attract users as readily as a server that is unique in its topic.

The inverse of this patterns exists, with topics served by few instances with a disproportionate number of users. This has it's clearest examples in instances covering erotic or sexual subjects. Though few in number, these instances host a larger number of users than servers not focused on erotic content.

3.2.2 Instance Behavior Patterns

Topics might reveal patterns of behavior across instances, but within an instance, unique patterns also emerge. Looking at the social graph, one finds that users in an instance behave like other users in the same instance, specifically in terms of their interactions and connections.

One could assume that intraconnections between users on the same server would be more common, as they have immediate access to each other and would share common interests. Some instances, following this pattern, have few connections to other instances. However, this is not the common case, with Zignani et al. finding that 35-40% of all users have significant inter-instance connection. [8] Is the opposite true then? Are users using their home instance simply as a service to access the wider fediverse? Again, Zignani finds this to somewhat untrue, with some instances being very outgoing, and others very insular. (Ignoring the extreme case of the user on a single user instance, who has only outgoing connections.) Mastodon allows a wide variety of instance behavior, and Zignani et al. find that the instances match this diversity.

> In general we confirm that the architecture based on independent instances has a stronger impact on the how users' ego-networks cluster; sometimes instances act as a bound on how the neighborhood of their members clusters, other times instances promote an external clustering.

The culture of an instance has a measurable effect on its users behavior. This also is expressed in how likely users are to form mutual connections, or only one-sided ones (Follower/Followee). This is not correlated with instance sizes; it is not the case that users in smaller instances are seeking connections with larger servers. Nor are large servers seeking small servers in any significantly differing way. Instead, users are still following the unique style of their own server.

A crucial thing to note though, is all the effects of instance style are of lesser import than the effects of instance location. Users have a very strong tendency not to interact with users outside their country; fewer than 10% of users have 50% or more of their connections going to users in instances in other countries. [8]

3.3 Drawbacks

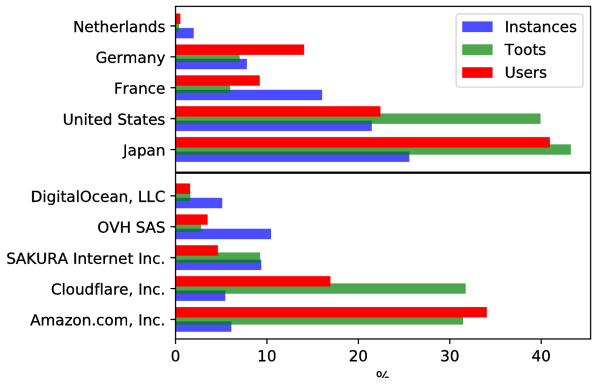

Figure 1: Distribution of instances, users, and toots(messages) across top 5 countries and hosting providers.

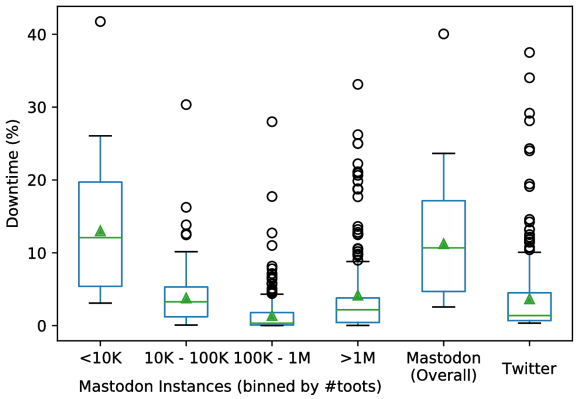

Figure 2: Distribution of downtime over number of toots on an instance, with the average totals and a comparable snapshot of Twitter.

Mastodon is decentralized, a fully functional network of independent servers. However, there are a number of pressures that can cause hidden interdependencies between servers. These drives towards re-centralization are present in the Mastodon network, and we will explore how they can manifest, and the consequences for the network if left unchecked.

3.3.1 *Massive Servers*

Most users coming from a centralized service want to join the biggest server, or don't understand that *mastodon.social* is not the only option. *mastodon.social* itself as the largest server, headed by the primary developer, has tried various tactics to balance between people joining the network at all, and people joining *mastodon.social* specifically. [4]

This is bigger than *mastodon.social*, as there are other servers of enormous size out there. In fact, 52% of all users on Mastodon are hosted in only 10% of all servers. [3] This is a threat to the diversity of the network, as moderators and admins on those 10% of all servers hold far greater sway over the network than a more even distribution would grant them. Though problematic in and of itself, these massive servers will be shown to have other effects later on.

3.3.2 *Social Graph Failure Example*

Supposing we have three users, on three different servers. *alex@xeno.space* who follows *babbar@yottabyte.rocks*, who in turn follows *angela@tea-and-coookies.com*. Angela can see Babbar's posts, and Babbar can see Alex's. If Babbar replies to Alex, Angela would see that reply, and might even reply herself, which would mean anyone who followed her would then see that post. This is how messages can spread beyond the single hop to the followers of the poster and their federated timelines.

Now consider *yottabyte.rocks* fails, perhaps due to a software bug. Alex's posts might now have no way to reach Angela, and Alex can't see Babbar or his posts. Until *yottabyte.rocks* comes back online, Babbar can neither post, nor can anyone see his posts. If *yottabyte.rocks* is unrecoverable, Babbar has ceased to exist.

In a centralized service, the network status tends to be more binary. Either the service is up and running, or it is down. Some grey areas exist in case a regional server goes down, and so traffic must be routed to a distant server at a slower rate.

In Mastodon and other decentralized networks, the network often fails by degrees, which is a significant feature, but also means that the network is practically never going to be in a perfect state.

3.3.3 *Centralization Through Hosting Providers*

As most instance are hosted by volunteers and often funded by themselves or through donations, there is an emphasis on cost-efficiency in hosting providers. This means that only three hosting services support for more than 62% of all users. Amazon alone hosts 6% of all instances, but due to the size of those instances, those account for 30% of all users in the Mastodon network. Cloudflare is the second most common hosting service, and it hosts 5.4% of all servers, and 31.7% of all messages. [3] Though a massive failure in Amazon or Cloudflare would be a nightmare scenario for much of the internet, smaller changes in policy, price or technology could negatively effect large portions of the Mastodon network. Consider Figure 1, which shows how top heavy this distribution is. Something not shown in the graph is the long tail of diverse hosting providers, with the average host holding only 10 instances.

Given these warnings, what is the actual uptime of the Mastodon network? Following the data from Raman et al in Figure 2, we can see that for the average user, Mastodon has an order of magnitude worse availability than Twitter did when it was of a similar size in 2007, 1.25% versus 10.75%. [3] Surprisingly, although less active instances have more downtime, the toot counts of an instance is not a good predictor of it's uptime.[3]

3.3.4 *User Abandonment and the Social Graph*

We can now see the downtime of individual servers, which affect the the users on them, but how is the network affected when connections are severed? This is measured by the Largest Connected Component(LCC), which measure how far a post could spread by sharing alone. Not all users are connected to each other, so there will be separate islands in the social graph, where a post has spread as far as it theoretically could, and there are no more connections to the remaining parts of the graph.

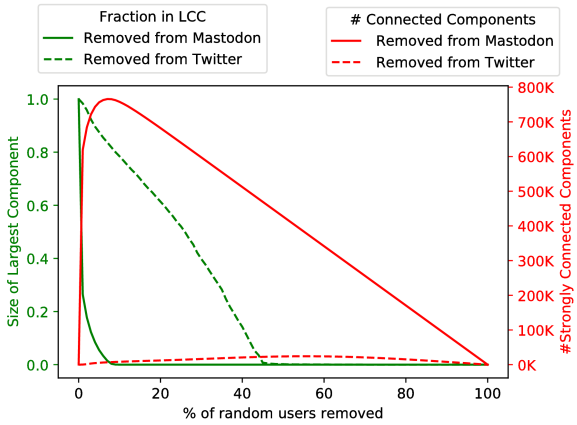

Figure 3: Mastodon's proportion of connected users(solid green), and number of strongly connected components(solid red). A snapshot of Twitter(dashed lines) from 2011 provided for comparison. X axis is percentage of total nodes removed.

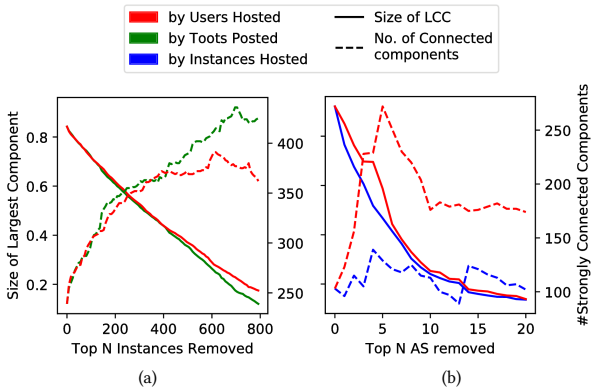

Figure 4: Removals sorted by users hosted (red), toots posted (green), and instances hosted (blue). Raman et al. use the term Autonomous System(AS) to refer to Hosting Providers.

First, Raman et al. considered the effect of users leaving the network, to see how a mass abandonment scenario could spiral, prompting further abandonment. Mastodon has a strong social graph at the outset, with 99.95% of all users in the LCC. With the top 1% of most connected users removed, the LCC only covers 26.38% of all users. This is an incredibly steep drop. For comparison, Twitter in 2011 had an LCC which covered 95% of all users, and with the removal of the top 10%, only dropped to 80%. [3]

It is possible that Mastodons instance system contributes to the creation of these keystone users, in that popular users are followed from many instances, but other connections between those instances are not established. If the popular user vanishes, all those interconnections go with them. Another possibility could be that these users with high amount of connections could be explorative bots. Raman et al. do not state whether they attempted to filter out bot users. If this is true, the social graph could have a different shape for human users.

3.3.5 Infrastructure failure and the Social Graph

Earlier, we noted the unintentional centralization of instances and hosting providers. Raman et al. now examine the issues such centralization can cause.

In the case of instances, the largest instances can be classified by number of messages, or by number of users.

mastodon.social holds the greatest number of users, but *mstdn.jp* has more toots, despite having over 3,000 less users. [3] Similarly, hosting providers can be ranked by number of instances hosted or by number of users those instances support. Both rankings have similar effects on the size of the LCC when used to remove hosting providers, but removal by number of users causes catastrophic failure in the number of components of the LCC. Removal of 5 hosting providers by number of users breaks the graph into 272 components, whereas removal by number of instances only results in 139 components. 3.3.5 This directly shows the effect of massive instances, and how unintentional centralization weakens the fediverse. Losing comparable number of number of random instances would be a linear decay, but with the top 5 hosting providers hosting 20% of instances and 85% of all users [3], failure on the hosting provider level leaves a massive gap in the network.

3.4 Possible Improvements

Though all these issues are problematic, most of them seem rooted in socioeconomic constraints. However, that does not mean that software solutions cannot play a part. Raman et al. studied the effectiveness of replication, of storing toots and other data on servers other than original instance. This adds a significant layer of complexity, as Mastodon is structured around a users' instance being the sole source of truth about them. However, replication strategies hold large potential. All the proposed system relied on a global index of toot replicas being maintained, likely through a *distributed hash table*.

3.4.1 Subscription Replication

One of the simplest methods of replication was having instance store copies of toots they received through subscriptions, with instances looking for a toot on an offline server looking up backup replicas.

This is less effective, as it essentially spreads the centralization out one hop in all directions. According to simulation carried out by Raman et al. under this system a network with the top ten servers by toots removed, the network would lose access to only 2.1% of all toots. Compared to the 62.69% loss without replication, this is already a significant improvement. This improvement largely holds true for the removal of the top ten hosting providers by toots, with 18.66% lost with replication, compared to 90.1% without. [3]

Subscription based replication re-centralizes itself rapidly though, as the majority of replicas would be hosted on servers with a large amount of subscribers. These large servers would already be among the worst ones to lose, even if they were not a key part of a backup mechanism. The same issue is also mirrored in in terms of popular users receiving excessive amount of backup, whereas more out of the way users receive none; 23% of all toots would have more than 10 replicas, while 9.7% would have no replicas at all. [3]

3.4.2 *Randomized Replication*

Avoiding the centralization inherent in following the social graph, the other technique Raman et al. studied was random replication, where every new toot is replicated on one or more servers chosen at random from the whole federation.

This adds a great deal of complexity, needing a systems to fairly distribute replication duties, and avoid the abuse by over burdening an individual server with replication requests. It also would need a way to index all instances, without that system itself becoming a point of centralization.

If these issues are overcome, then a single randomly placed replica can keep 99.2% of all toots available after removing the 25 instances by number of toots. Subscription based replication would have 95% of toots still available afterwards. More randomized replicas have increasingly better persistence, as demonstrated by Figure 5.

Although complicated, randomized replication could provide a way to avoid the downsides of semi-centralized networks, though it does not actually address the issue of centralization in the network itself.

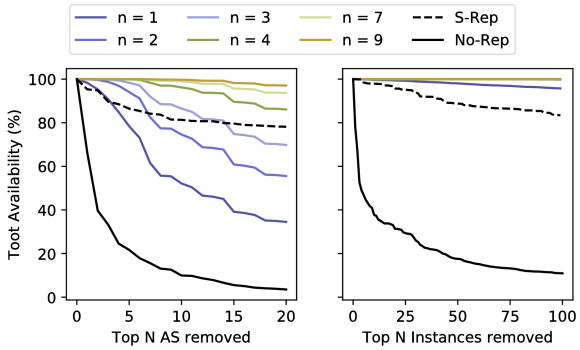

Figure 5: Availability percentage of all toots with the top ten hosting providers and instances removed. Raman et al. use the term Autonomous System(AS) to refer to Hosting Providers.

4. CONCLUSIONS

Mastodon as a platform has its shares of downsides, including the drive towards centralization and the impact of unreliable servers. Some of these problems Mastodon can engineer solutions towards such as replication [3]. Others, such as the growth of megaservers are still a subject of debate, and likely will need to be resolved through community action over software features. [4]

Some of these changes are simply differences to what has come before in the history of social networks. Though not decentralized, one could point towards the micro-communities of reddit as an example of a similar arrangement of users and interests on a significant platform. It is not a stretch to say that the limitations imposed by Mastodon's structure are within the constraints other social networks have imposed, and still been successful.

Mastodon also shows a way out of the non-public gathering places issue that most other social networks represent. This is a significant benefit, besides the emphasis on user control and strong moderation tools that it provides. Access to the source code is also highly beneficial.

The most important detail however, is that Mastodon isn't a theoretical idea. It is a network, growing in the wild, and built to uphold a common standard published by the W3C. [2] Though analysis of how it grows and the issues it faces is important, it also is not waiting for to be theoretically sound. With frequent updates from over eight hundred authors [6], Mastodon is present and active.

Acknowledgments

The author would like to thank their academic advisor, associate professor Elena Machkasova, their seminar advisor, professor Nic McPhee, and seminar teacher, professor KK Lamberty.

5. REFERENCES

[1] A. Balkan. Encouraging individual sovereignty and a healthy commons, 2017.

[2] E. S. A. G. Christopher Lemmer Webber, Jessica Tallon and E. Prodromou. Activitypub. Technical report, World Wide Web Consortium, 2018.

[3] A. Raman, S. Joglekar, E. D. Cristofaro, N. Sastry, and G. Tyson. Challenges in the decentralised web: The mastodon case. In *Proceedings of the Internet Measurement Conference*, IMC '19, page 217–229, New York, NY, USA, 2019. Association for Computing Machinery.

[4] E. Rochko. The role of mastodon.social in the mastodon ecosystem, 2019.

[5] M. Stockley. More relevant ads with tailored audiences, 2013.

[6] TootSuite. Authors.md, 2020.

[7] M. Zignani, S. Gaito, and G. P. Rossi. Follow the "mastodon": Structure and evolution of a decentralized online social network. In *Twelfth International AAAI Conference on Web and Social Media*, 2018.

[8] M. Zignani, C. Quadri, S. Gaito, H. Cherifi, and G. P. Rossi. The footprints of a "mastodon": How a decentralized architecture influences online social relationships. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 472–477. IEEE, 2019.