Federated Systems

Jeremy Rubin

jr@mit.edu

December 9, 2015

Abstract

Federation is an amazingly powerful tool in a system designer's tool belt. This work provides a unifying definition for federated. In abbreviation: a system is federated when there are multiple providers of an identical (or nearly identical), interoperable service. The federated paradigm offers much improved fault tolerance, liberty, and privacy over centralized alternatives. The federated paradigm applies to all sorts of systems, from governmental to web services. However, this paradigm seems to be unsustainable, either devolving into centralized systems or being outpaced by them. In this paper, I will delve into multiple examples of federated systems, exploring their successes and failures, as well as delving into what caused the decay of these systems. In reflection, I will also provide several suggestions on how system developers might harden their use of this paradigm.

Contents

1 Introduction:

A natural monopoly is an emergent phenomenon by which a single actor gains complete control over a market; and this monopoly is more efficient than a non monopolistic market. It is easy to imagine scenarios in which natural monopolies exist – the most commonly held example is the telecommunications industry and the rise of AT&T. It is posited that AT&T deserved their natural monopoly with the following logic: building a telephony network is prohibitively expensive, but operating one is cheap, therefore the first to market will form a natural monopoly. However, a closer inspection reveals that these monopolies are far from natural, and are rather a matter (or, failure if you will) of public policy. In the case of AT&T, Thierer drives home that AT&T's monopoly was fragile, propped up by universal access policies, price regulation, and exclusive carrier licensing[1]. The foil of the monopoly is a competitive market, where multiple providers vie for the business of their customers. The case for competition is that, while a monopolized market may be more efficient, a competitive market will innovate. Federated markets seem to be a happy medium between these extremes; participants agree on a common core set of services, and innovate on the edges. Indeed, in the Consent Decrees signed by AT&T through the decades they agreed to federate their service by allowing local providers to tap into their nation wide network, but they still pushed out the competition via other means. Thierer's thesis was that no market should be regulated (or rather, we should not believe that regulation will help prevent monopolies), as they are naturally competitive. Looking at the crucial Internet architecture today, there are many places which are all but competitive. Despite the open nature and inherent

[1] Adam Thierer. *Unnatural Monopoly: Critical Moments in the Development of the Bell System Monopoly.* `http://object.cato.org/sites/cato.org/files/serials/files/cato-journal/1994/11/cj14n2-6.pdf`.

neutrality of the Internet[2], monopolies over certain types of information service seem to form readily. Given that monopolies are indeed bad, it is crucial to understand why federated information services fail, why centralized services succeed in their place, as well as to look at successful federated systems for inspiration.

A little bit about the author: I'm a Senior/Master's student at MIT in Electrical Engineering and Computer Science. My concentration is on Computer Systems engineering and Cryptography, with a passion for cryptocurrency. Cryptocurrency is an exciting (to me at least!) new field which sits at the nexus of systems engineering, cryptography, economics, game theory, policy, and free and open source software. In a nutshell; all of my favorite things. In the realm of cryptocurrency, I was first formally exposed to federated protocols, but quickly realized that this was a common paradigm in many of the critically systems we've built as a society, from our government to the Internet.

However, I also noticed that many of these systems seem to be struggling to stay federated. State's rights seem to have been weakened with recent rulings and legislations, most people I email use Google's gmail as their email provider, and single Internet Service Provider (ISP), Comcast, has 56% market share on broadband[3]. With the federation of these systems weakening, it is critical for us to understand the importance of their federation. What do we stand to lose should they become strictly centralized? What would the consequence be of a singular world government, or singular U.S government? What happens when gmail becomes the only way to send someone text? What happens when a single entity such as Comcast provides the Internet? Is it positive, given the gains in efficiency? Should we encourage it? Or is it a deadly seeming-eventuality, that

[2]Not that this precludes the need for a strong political battle to preserve this openness and neutrality

[3]Jon Brodkin. *Comcast now has more than half of all US broadband customers.* `http://arstechnica.com/business/2015/01/comcast-now-has-more-than-half-of-all-us-broadband-customers/`.

we should fight to avoid?

With my political leanings, and hunches on the importance of federated information systems, I feel that it is our battle. I have a desperate urge to save, preserve, and create new systems which leverage and improve federated paradigms. My hypothesis is threefold: federated systems were important in the past; the principles integral to federation will continue to be critical; the federated systems we have today are dying; therefore, we should try to save them.

In order to contextualize my hypothesis, I'll need to first define what exactly it means to be federated. What is the origin of the term, and how has that definition changed with time? With a definition in hand, I'll then find a few examples of federated systems in the wild which provide critical infrastructure, which will establish that federated systems are indeed important. Having established their importance, I'll then attempt to metric the extent to which they are federated in practice and how that has changed over time, and speculate on why that change has happened. I'll also look at systems that were never important, as they never gained much adoption, such as XMPP. Why did these standard fail in the first place? Are there any vibrantly healthy federated systems in deployment today? With the previous parameters exposed, I will synthesize my thoughts together to answer my hypothesis – and provide suggestions on how to either dismantle or reinvigorate the state of federated systems.

2 Defining Federation:

Before it is possible to save federated systems, or even pass judgment if they deserve saving, it must be clear what a federation actually is.

Federation is a term derived from the Latin for covenant. As a contractual agreement, fundamentally federation must have something to do with gover-

nance, or at least adherence to the rules.

Federation has a rich history in society and Computer Science. On the politics side, we have federated governments, on the Computer Science side, federated consensus systems, and in between, identity systems. By exploring these categories we can delineate the principle components of the term federation as it is used, and reconstruct that basis into a sound definition.

2.1 Governmental Federations

Many national governments, such as the United States, Russia, and India, are federations – hence the term "federal" government. A federated government is typically composed of a central federal authority, and several states. The federal authority has certain special privileges and functionality delegated to it by the states, while the rest is left to the state body. For instance, the states may delegate their ability to make foreign and inter state policy to the federal authority, but may retain their rights over domestic policies in the state. Thus, an individual may have a different experience with respect to government functions such as education, healthcare, and law enforcement in an individual state, were they to go to war with another country, it wouldn't matter much (assuming that country does not selectively share a border) which state they lived in were there to be a draft.

In essence, in a governmental federation, a group of states decides upon a body of common problems that they can compromise on, and leaves them to a third party while still retaining control over rights not delegated.

As a note, the opposite of a federation is a monolithic system, where all rights are delgated to a single governing body. The opposite of a federation is not a confederation. One of my favorite jokes of all time is:

> If "con" is the opposite of pro, then isn't Congress the opposite of

progress? Or did we just fucking blow your mind?!?

– Jon Stewart, America (The Book): A Citizen's Guide to Democracy Inaction

Although there is a whole lot of truth imbued in that quip, con does not always imply a purely negating modification to the root of the word. As such, a confederation is not the opposite of a federation, the distinction is merely that a membership in a federation is mandatory whereas membership in a confederation is voluntary. In that respect, they are opposites, however, their similarities are great.

One property that should be clear from the way government is structured is that federations can be recursively nested. For instance, a group of towns may form a federated county, a group of counties may form a federated state, a group of states may form a federated nation, and a group of nations may form a federated planet. It is important to note that the federated qualification is dropped at each level of composition, in other words, the tree of federation is not of uniform depth. Furthermore, membership of a federation is not exclusive, an entity could be a simultaneous member of many federated groups.

It is also important to note that in governance, a federation may select different responsibilities to delegate and this may strongly impact it's viability. A federated group of nations must not delegate all of their responsibilities to the federal authority, lest it become a monolithic group[4], likewise, should it delegate nothing to the federal authority then it would not impact state actions.

[4]Technically, this is not a complete picture as a monolith may be established with a static set of treaties which from then on are used. This would be, in some sense, a federated government

2.2 Computer Science

In Computer Science, federation is used in many contexts, with different connotations. Stijn Peeters, a researcher at the Institute of Network Cultures, argues that because of it's varied use we should redefine federated as an equivalent term to decentralized and distributed[5]. Peeters points out that the term really originates with Paul Baran's seminal work, *On Distributed Communications*[6], and his definition of decentralized. Despite close similarities, there are specific properties of a federated system that differ from pure decentralization, and they are worth mention. A couple examples should help demystify the contexts in which it is used.

2.2.1 Simple Sharded Database

In a simple sharded database, a number of databases are given subsets of the data to process and store. A central authority keeps track of the mappings of which databases should hold which records and can perform critical functions such as load balancing. In other words, each database delegates it's ability to decide what data it holds to the central authority.

2.2.2 Federated Byzantine Agreement

In a Federated Byzantine Agreement, such as the Stellar Consensus Protocol[7], a number of individuals want to agree on a common piece of information (or pieces of information) in the presence of all types of fault (death, malicious lying, random error, etc).

In Stellar, this is accomplished by letting each actor select multiple *quorum*

[5]Stijn Peeters. *Beyond distributed and decentralized: what is a federated network?* `http://networkcultures.org/unlikeus/resources/articles/what-is-a-federated-network/`.

[6]Paul Baran. *On Distributed Communications*. `http://www.rand.org/content/dam/rand/pubs/research_memoranda/2006/RM3764.pdf`. 1964.

[7]David Mazi'eres. *The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus*. `https://www.stellar.org/papers/stellar-consensus-protocol.pdf`.

sets of other actors with whom they would like to agree with should the rest of that set agree[8]. Each actor governs internally by agreeing with the first quorum set that agrees. There is no central authority, unlike in the simple sharded database.

2.2.3 Federated Cryptography

While technically a subset of Federated Byzantine Agreement in some sense, a group of actors can perform a critical responsibility, such as fairly selecting a uniformly random bit, using a cryptographic protocol. In the random bit scenario, each actor can randomly pick a bit, commit to them publicly using a hash, reveal the committed data, and then xor them all together[9]. As long as 1 actor honestly randomly uniformly picked their bit the process was fair[10]. One can imagine such protocols being used as a governing function to fairly distribute the last slice of pizza, or to efficiently combine work[11].

2.3 Federated Identity

The realm of identity sits between Computer Science and government. One of the fundamental tasks for many governments is maintaining records on their citizens and providing mechanisms to authenticate operations such as notarization. In computer science, databases keep track of records and cryptography provides authentication mechanisms.

Thus, identity, as a topic separate from government and Computer Science, takes a mixed bag of techniques, methods, and enforcements from both realms in a unique blend.

[8] This is a highly nuanced protocol, please review the Stellar paper for more detail

[9] Manuel Blum. "Coin Flipping by Telephone a Protocol for Solving Impossible Problems". In: *SIGACT News* 15.1 (Jan. 1983), pp. 23–27. ISSN: 0163-5700. DOI: 10.1145/1008908.1008911. URL: http://doi.acm.org/10.1145/1008908.1008911.

[10] for an attentive reader, the must ensure that no two commitments are the same

[11] S. Felter. *An overview of decentralized Kalman filter techniques*. http://dx.doi.org/10.1109/STIER.1990.324634.

Federated Identity is a mechanism by which multiple authorities are able to offer certificates to entities for validation which are recognized globally. For example, passports are a great example of federated identity; they are a somewhat interoperable standard document type, and certain entities do not recognize other entities as issuing valid passports without an additional certification (a Visa). Another example is email, one can register an email address on any domain and be able to email a user on any other domain.

Identity can be required to be multifactor, can be single factor, or even 0 factor (self asserted). Furthermore, not all aspects of Identity are created equal. There is a spectrum of criticality on how much verification is needed. For instance, it may be acceptable to self assert one's favorite pie, but it would not ok to self assert ownership of a bank account. Every type of data can span the full gamut of criticality depending on the context.

Identity makes for a clear distinction of delegation versus federation. As Clay Shirkey emphasizes, a delegated identity is where an entity outsources identity information record keeping to another entity. A federated identity system means an entity can keep their identity information in one of many acceptable providers[12].

2.4 The Definition

Thus a generalized framework for what constitutes a federated system can be constructed. A system is federated when there is a core kernel that is agreed upon for common operations, and a auxiliary layer of behaviors which are not restricted or guaranteed. This kernel could be a set of policies, an algorithm, an API, or some sort of leader. The set of behaviors outside the kernel can be bounded or unbounded (in other words, the kernel could restrict the actions

[12]Clay Shirkey. *Delegated vs. Federated ID.* `https://sites.psu.edu/ntsh/2010/02/15/delegated-vs-federated-id/`.

taken outside of it or not). A federated system is fully recursive, meaning that each actor outside the kernel may be itself a federated system, and the kernel may also contain a federated system. Furthemore, federated systems are not exclusive, an actor may be a part of many simultaneous and heterogeneous federated systems. Lastly, a federated system may be corruptable, meaning it's possible that actors no longer obey the kernel, such as when governments collapse, however it's also possible that it is in a wide range of *fault-tolerant* systems which can not be corrupted with a certain amount of actor corruption (an example of this is cryptographic bit selection seen earlier, from the perspective of any honest actor, the protocol is always fair regardless of others corruption). Wholly incorruptible systems are not included in this definition as it is unclear as to how they might depend on the participants at all.

3 Case Studies

With the space and meaning of federation well defined, we will now examine a number of real world, massively deployed, federated systems and explore their emergent phenomenon. Each of these examples has successes, and each of them failings. This strong set of examples will elucidate actual challenges faced, and the processes by which the responses to these factors caused said success or failure. Critically, success is measured in some respects as a raw quantity of persistence, it can not be left unexamined if the persisted phenomenon is malevolent or benign.

Using case studies is a common method to understand phenomena. Edwin Amenta offers a strong example of how to operate with a case study and provides advice on extracting the most value out of a case study[13]. Critically, Amenta

[13]Edwin Amenta. *Making the Most of an Historical Case Study: Configuration, Sequence, Casing, and the US Old-age Pension Movement*. `http://www.socsci.uci.edu/~ea3/Amenta.2009.byrne.ragin.Ch20.pdf`.

argues that the total correctness of an individual case is not what is important, but rather that the analysis in aggregate leads to a unifying understanding of the joint phenomena. While correctly analyzing a case won't yield incorrect results, nor will incorrectly analyzing a case yield untruths. This is a justification for the style of case study that I will employ here: rather than read too deeply into a single case study, I will take a "shotgun-approach", and briefly sketch a number of case studies to build a theory on top of a broader support basis rather than a deeper one. Not to say that the analysis will not be deep, but rather that emphasis will not be placed on a play by play of each example but rather on elucidating the overarching theme.

3.1 E-Mail

Email is often called the "killer app" of the Internet[14]. It is also a federated service – any domain running a mail server can communicate with any other. It is an amazingly potent tool, and has cemented itself firmly in the work and personal routines of many millions of people. Part of its long term hardiness is in no small part due to its federated nature. As a federated service, the entire network of individual able to email one another did not go down when one or two email providers failed, nor did potential competitors such as MySpace put email out of business because email wasn't a corporate competitor, just a protocol.

However, despite email's successes in facing stiff competition, the greatest attacks to it strike by undermining its federated nature. As Benjamin Mako Hill points out, it's impossible to keep his emails off of Google's servers, Google simply too large – almost invariably, his emails will be forwarded, sent directly, or passed through a Google server[15]. This problem goes further than just pri-

[14]Elliotte Rusty Harold. *Email: The Internets First and Last Killer App.* `http://radar.oreilly.com/2013/12/email-the-internets-first-and-last-killer-app.html`.

[15]Benjamin Mako Hill. *Google has Most of My Email Because it has All of Yours.* `https://mako.cc/copyrighteous/google-has-most-of-my-email-because-it-has-all-of-yours`.

vacy, the sheer size of email providers such as Google or Microsoft put email in a precarious position, should they fail then swathes of users would be uprooted from their digital lives. Not only that, but a few large email providers can act as functional gatekeepers to the entire system. Lee Hutchinson, the Senior Technology Editor at Ars Technica, has a series of articles subtitled "Gmail? Apple? The cloud? Forget 'em all-in this series, we take your e-mail back"[16]. In the series, Hutchinson pushes the centralization of email as a failure of it's users, attracted by "gigabytes of space and plenty of cool value-added features". Hutchinson's article concludes its introduction with the following provocation:

> *Why do battle with arcane dragons to roll your own e-mail solution?*
>
> *I'll tell you why: because if it's in the cloud, it's not yours.*
>
> *Because you must rely on others for your security. You have no control over who can read your correspondenceyou must allow your data to be mined and your marketing profile extracted. You won't be told if your metadata is collected or if your inbox is vacuumed up by a secret government request. You consent to be not a customer but a product, and a product has no rights.*
>
> *Well, to hell with that. It's your e-mail. And we're going to take it back.*

The problem is, as noted by Hutchinson, major email providers can act as gatekeepers to the system, blocking smaller or individual email providers. In *The Hostile Email Landscape*, Jody Riboton laments her failure to un-blacklist her email server. Essentially, Riboton had found herself in a catch-22. In order to have her emails delivered to the major provider's users, her IP address had to have built up some reputation with them. In order to build up reputation,

[16]Lee Hutchinson. *How to run your own e-mail server with your own domain, part 1.* `http://arstechnica.com/information-technology/2014/02/how-to-run-your-own-e-mail-server-with-your-own-domain-part-1/`.

she had to have users not mark her mail as spam. But there was no clear way to bootstrap her reputation. Unable to get her mail delivered, Riboton had little choice:

> *In the end, I gave up and switched back to Google Apps. It felt like defeat. This isn't how the internet is supposed to work. As we continue to consolidate on a few big mail services, it's only going to become more difficult to start new servers.*

The problem only compounds itself when regulatory considerations are considered, as they further gatekeep newcomers. For instance, the CAN-SPAM act places harsh penalties on sending spam mail. Under the act, "each separate email in violation of the law is subject to penalties of up to $16,000"[17]. A new email provider could potentially be attacked using this regulation should a malicious party send some spam from their platform.

3.2 Blog

The P2P Foundation considers federated blogging software to be nonexistent, presently nothing more than concept[18]. To be blunt, the P2P Foundation's assessment is largely incorrect. In fact, blogs are already very federated! The basic qualification of a blog is to be a web-based log of events. Misguidedly, the P2P Foundation's definition calls for integration, but blogs are already highly integrated by the power of the hyperlink. A blogger has many options for hosting their content, they could provision a server of their own, or they can use one of a number of services such as Medium, Blogger, or Wordpress. The restrictions on what makes a blog a blog are quite loose, it is less of a machine format and more of a user expectation to see a certain style, typically a sequence

[17] *CAN-SPAM Act: A Compliance Guide for Business.* `https://www.ftc.gov/tips-advice/business-center/guidance/can-spam-act-compliance-guide-business`.

[18] *Federated Blog - P2P Foundation.* `http://p2pfoundation.net/index.php?title=Federated_Blog&oldid=63858`.

of posts containing a title, a date, some text content, and perhaps, a place to leave comments.

One of the core problems that blogging faces is that this "bring-your-own-everything" model of Federation promotes the growth of centralized platforms. In the blogging space, platforms such as Medium draw a large number of users because they simplify a lot of features that any user might want, such as comments, collaborative editing, search engine optimization, etc. Even among highly technical audiences, Medium has a large draw, so it isn't just a matter of ability, it's a matter of friction. Some services, such as Disqus, help independent bloggers deal with the complexities of commenting systems (any time a website changes its content due to user action there are a host of security implications to consider). The content is still hosted and owned by the blogger. This is a nice trade off in a sense, but it is somewhat unsatisfactory in that Disqus is still a centralized service.

In sum, bloggers form a loose federation. It is difficult to point to a specific thing that bloggers all do together, but at the end of the day, the simple and flexible format, along with the hyperlink, allows for many different bloggers to successfully connect and share their world views.

3.3 XMPP

XMPP, the Extensible Messaging and Presence Protocol, is a federated chat protocol first speced out in the late 90's which was designed to be basically the end all be all of instant messaging services. The infrastructure of XMPP was similar to that of email in terms of its federation model, thus much of the analysis is identical, in short, anyone could run their own XMPP server on their domain. XMPP enjoyed great early success, as many chat providers such as Facebook and Google provided support for it, but one by one, its supporters

turned on it, and it is no longer in wide use to customers. However, behind the scenes, XMPP still enjoys use as a core protocol behind many services, albeit, minus the federated inseparability. Let's be clear: XMPP is not dead, far from it. But, in seeking proprietary upgrades, large providers have abandoned it to be more nimble. The problem is that in order to be compatible with many other XMPP nodes, a node cannot implement any radical new features as it will break compatibility.

3.4 DNS

The Internet rests on DNS at its core. DNS provides a system for name resolution, a mapping of human readable names to machine locations. DNS is a simple to understand protocol at a high level, but is much more complicated as one peels back the layers. Essentially, a set of servers keeps track of what IP addresses are associated with which domain names, and provides that information to any client who asks. A more complicated view into this, is that there is a hierarchical tree of servers which are able to direct this at higher bandwidth, provided clients cache higher up portions of the tree (which changes less frequently). This design is implicitly federated because anyone is free to handle their own domain name resolution, but can also outsource it to any number of providers. However, as Pete Keen quickly found out, running a DNS server is not so simple. After running his own server, he quickly found it being utilized by hackers to run attacks[19]. Overall, the federated nature manifests as a strength. In October 2002, a sustained DNS attack took 12/13 DNS Root servers offline. The 13th was able to stay online in part because the operators had set up their infrastructure with over-provisioned resources, capable of mitigating a large attack[20].

[19] Pete Keen. *How and why I'm not running my own DNS.* `https://www.petekeen.net/how-and-why-im-not-running-my-own-dns`.

[20] Paul Vixie. *Events of 21-Oct-2002.* `http://c.root-servers.org/october21.txt`.

One of the major strengths of DNS is that it aims to be a completely composable service. In other words, it is designed to serve as a building block for other services. When a HTTP, SMTP, or FTP request is made, DNS is first consulting. Were DNS not composable, society would need to maintain an entire DNS system for each sub protocol. Instead, DNS centralizes this important name resolution task and makes it easy to build infrastructure on top of it.

3.5 Security Assertion Markup Language

Security Assertion Markup Language (SAML) is industrial framework for identity management. Essentially, SAML provides a standard markup for an identity provider to make assertions about a user. Technically, this is not federated at all, as it is just a language. However, by having a common framework, an individual can select a set of identity providers they recognize and accept SAML statements from that set for authentication or other purposes. What makes this federated is that a user has the ability to be registered with any of a number of different services. While the security policies (such as two factor authentication) of an individual identity provider may differ, they are able to provide a common API to that information. What is neat about this arrangement is that while each identity provider can have a common core set of information about a user, they can easily provide custom assertions as well. This is the big success of SAML in some senses – because the format can be used for many purposes easily, it enjoys use in many cases. This is, to some extent, a part of the Unix philosophy to do one thing, well. SAML doesn't try to solve all the problems, but it does solve one of them, and lends well to composition with other tools.

3.6 McDonald's

I hope in reading this heading, you're thinking, "What the hell does McDonald's have to do with this?". But in fact, McDonald's is a shining example of a massively successful federated service. McDonald's operates as a franchise. A franchise operation is essentially a brand name with federated execution of responsibilities. The brand, in this case, McDonald's, performs or guarantees certain large scale problems, such as supply chain, while the franchisee has domain over small scale hard problems, such as quality of service and hiring. This model has been highly successful for McDonald's, as they are one of the largest companies in the world[21].

I recently spent some time in China, where sometimes it can be difficult to find a "safe" choice of meal. While there, I knew that if I ordered *ji-ro la bu jia* at McDonald's, I would get a spicy chicken sandwich without mayo that was highly unlikely to give me a stomachache, even though my pronunciation of spicy chicken sandwich in Chinese was far from clean. By taking the difficult job of high quality chicken sourcing out of the hands of the individual restaurant, it is possible for the customer to be sure of a minimum quality of service. Consequently, should I have become ill as a result of dining at a single McDonald's, it would not be highly indicative that dining at another location would cause me digestive woes.

3.7 Social Networks

3.7.1 Failures in Centralized Social Networks

Social networks such as Facebook, Google+, and Twitter pose a major threat to user freedoms. Allowing crucial pieces of social infrastructure to rest in the

[21] *McDonalds (MCD) Stock Price, Financials and News — Fortune 500*. `http://fortune.com/fortune500/mcdonalds-110/`.

hands of the few poses major risks in terms of coercive or otherwise anti-user behavior as these systems are co-opted by various agendas. Two crucial examples of how these centralized systems have been corrupted comes from ZunZuneo and Facebook. While you've hear of Facebook, you likely haven't hear of ZunZuneo. ZunZuneo was a covert US operation to undermine the Cuban government by introducing an initially neutral Twitter like platform to Cuba. Once popular, the plan was to flood the site with propaganda. While ZunZuneo is not a stellar example of large centralized social networks (at most, it had 40k users), what it does underscore is that social networks are seen as an asset to governments, potential tools for implementing political agendas[22]. The Facebook case is in a similar vein, and is relatively well known. Facebook researchers performed an experiment on almost 1 million of their users to track "emotional contagion". In simple terms, they figured out that they can manipulate the emotions of their users to make them more depressed by surfacing depressing content on the user's Facebook feed[23]. This experiment is famously controversial as an example of poor ethics in research as the researchers did not have informed consent for the experimentation from participants[24]. Where ZunZuneo underscores, Facebook makes bold an highlights pink that the platforms we entrust this critical data to are not in user's interests.

[22] Alberto Arce, Desmond Butler, and Jack Gillum. *US secretly created 'Cuban Twitter' to stir unrest*. http://bigstory.ap.org/article/us-secretly-created-cuban-twitter-stir-unrest.

[23] Adam D. I. Kramer, Jamie E. Guillory, and Jeffrey T. Hancock. "Experimental evidence of massive-scale emotional contagion through social networks". In: *Proceedings of the National Academy of Sciences* 111.24 (2014), pp. 8788–8790. DOI: 10.1073/pnas.1320040111. eprint: http://www.pnas.org/content/111/24/8788.full.pdf.

[24] Patrick Coutermarsh. *FACEBOOK: The Psychology Experiment You Consented to in FB's Terms of Service*. http://www.scu.edu/r/ethics-center/ethicsblog/business-ethics-news/20119/FACEBOOK:-The-Psychological-Experiment-You-Consented-to-in-FB's-Terms-of-Service.

3.7.2 Hope in Federation?

Were there to exist a federated version of such sites, crucial components to their operation such as content surfacing algorithms could be federated across many providers, preventing a single provider from controlling the whole pool. Indeed, there are collaborative spam filtering projects that provide a more federated model, such as spamicity[25]. Content surfacing is just a single parameter, there are many crucial axes on which to federate, such as identity and persistence – in the case of ZunZuneo, many users also lost their online identities when the program shuttered.

Richard Esguerra, Development Director at the EFF, recognizes that while the problems caused by centralization may seem mundane, to someone under duress the case would be more clear, "with more user control, diversity, and innovation, individuals speaking out under oppressive governments could conduct activism on social networking sites while also having a choice of services and providers that may be better equipped to protect their security and anonymity."[26]

There are presently a couple major federated social networks, GNU Social[27], diaspora*[28], Friendica[29], pump.io[30], StatusNet, and others. Delineating the history of these networks is a task in it of itself, as they are deeply intertwined. As federated protocols, many of the offer support for one another or have merged their projects together formally (as with GNU Social and StatusNet). The ability to merge is the beauty, in a sense, of federated protocols.

Sean Tilley, community manager at diaspora*, provides a recount of the

[25] *Home - spamicity.info.* `https://spamicity.info/`.

[26] Richard Esguerra. *An Introduction to the Federated Social Network.* `https://www.eff.org/deeplinks/2011/03/introduction-distributed-social-network`.

[27] *GNU social and GNU FM.* `http://webcache.googleusercontent.com/search?q=cache:PilclRI0si4J:https://gnu.io/+&cd=1&hl=en&ct=clnk&gl=us`.

[28] *The diaspora* Project.* `https://diasporafoundation.org/`.

[29] *The internet is our social network — friendica.* `http://friendica.com/`.

[30] *pump.io by e14n.* `http://pump.io/`.

struggles of diaspora* as it sought out relevance[31]. In his writing, what seems to be the largest obstacle is lack of a business model. After struggling to raise money for the project, the team joins Y-Combinator, and "sells out", building a new platform, makr.io. He also documents struggles with Friendica, their chief competitor, which seems to have a more mature and organic development community. For instance, Friendica reverse engineered diaspora*'s protocols, but diaspora* was not able to integrate Friendica support. This behavior manifests the tree-like nature of federated systems, as the set of federated social networks is itself a federated system! However, the nature of competition still applies, projects such as a GNU Social seem functionally dead, and diaspora* on the way out. Thus it seems likely that Friendica, the most mature of these platforms with integration's for most others, will outstrip the competition. This potential for monopolization is not a threat, as the end result would still be federated. However, presently it seems that not much is at stake with low overall adoption – centralized competitors such as Facebook have barely been scratched by these efforts.

4 Conclusion

Federation is a commonly employed mechanism for developing robust platforms which respect their user's autonomy and right of choice, while still providing strong guarantees on the desired properties of such a system. There are many use cases where federation is absolutely the right paradigm to explore. However, federated services are often "ripe for disruption" by a more centralized model, which can reduce cost of operation and provide new features, or lobby for regulation to make it more difficult to run a federated version compliantly.

[31] Sean Tilley. *Planting a Seed: Diasporas Story (Part 1).* `https://medium.com/anti-fiction/planting-a-seed-what-working-at-diaspora-was-like-cde26fa29364`.

Indeed, many of the federated systems that the Internet holds dear are under threat of centralization. While it is safe to say the federated paradigm isn't dead, the protocols built on it are unhealthy. It's critical to reinvigorate our existing systems, as well as learn from the past in future endeavors.

To establish this, I'd like offer some provocations and food for thought. While I don't know exactly what needs to be done to modernize attempts at federated systems, I can offer some advice synthesized from the close reading into this paradigm on ingredients for success. If it were possible for me to provide an exact prescription, I wouldn't be sitting here writing this paper – I'd be building a system. Each domain will have it's own unique set of constraints and the ideal federated solution will not necessarily closely resemble another, nor should it! As I've come to define federated systems, I know that it is a broad, multifaceted distinction, different facets can be emphasized over others.

4.1 The Federated System Designer's Handbook

1. **Solve the Large Scale problems centrally, and the small scale problem in the federation.**

   This principle comes from the franchise model, where McDonald's does the hard work of guaranteeing the Beef is properly sourced, while the individual restaurant flips the burger. For instance, consider a redesign of a federated e-mail protocol. A chief reason a user such as myself uses gmail is for availability; I believe my email will accessible (as in, Google will still have servers with data) today, and I think will tomorrow as well. If the concern of small email providers losing data could be alleviated, then perhaps the federation can provide other services which let users have more control over how their data is used.

2. **Provide a clean API so that others trying to replace your service**

will make it interoperable.

Friendica is compatible with its competitors. It's competitors, are not. This is an ecosystem win, for both the competitors and the users. Because Friendica could implement diaspora*'s protocol, Friendica did not cause a diaspora of diaspora* users to the new platform. Friendica, on the other hand, was able to bootstrap itself with content, connections, and features from the other platforms.

At the same time:

3. **Don't over-interoperate with other's platforms.** i

   Should a system be overly interoperated, then there will not be large "feature-exclusivity" reason to switch. Many diaspora* users will not switch to Friendica simply because it is overly compatible.

   There two principles require a delicate balance between accessing an existing userbase and asking user's to leave one behind. This is kind of a sad piece of advice – after all, the hope is to develop interoperable platforms for greater user freedom. However, when no platform has high adoption compared to centralized giants like Facebook, it's critical to roll up the sleeves a little bit and displace competitors, after all, centralized corporate attempts will certainly do so.

4. **Have a Business model in mind for providers.**

   Although not quite a failing in some respects, lacking a business model could be a large factor in the failings of a federated system. Money is needed to invest in research and development to improve the protocol, if there is no cash flow then a centralized competitor will be able to more nimbly execute bringing the technology to market. When a federated system's success depends on the generosity of others, it may fail when that

generosity runs short. Importantly, the protocol should live separately from the business model. Should that not be the case, it would be all to easy for a well funded competitor to attempt to displace the federated service.

5. **Beware of Gatekeepers.**

   Gatekeeping behavior is a major threat to otherwise suave federated protocols. Gatekeeping behaviors manifest not only in the design of protocols as seen with the ability for sub networks to censor or blacklist messages, but also in the regulatory requirements lawmakers levy, as seen with seemingly neutral or positive measures which end up serving a protectionist role.

6. **Compose.**

   Federated services are best built as composable blocks. This is because a federated service can be recursively composed of sub-modules of functionality. By striving to solve a single problem, federated systems can enjoy use for multiple purposes. For instance, DNS is used for many name resolution purposes, for HTTP, SMTP, XMPP, etc precisely because it was designed to serve as a composable layer. Having many dependents helps ensure the stability of a platform as more users are invested in that service's operation.

7. **Beware of Customizability.**

   Although customizability is a large selling point of federated systems, it can also be a an adoption barrier. XMPP's large focus on extensibility meant two things: the protocol did not do everything the user's would want it to; and that user's would be encouraged to develop their own extensions, breaking compatibility. Instead, simplicity must be key, and the system should be constrained tightly to prevent strong divergence from

client to client. This seems to be a negative result – if customizability is a selling point, and I'm saying don't use it, then it's not a selling point at all! I assure you, it still is, but it should be taken with the above advice on composability. By focusing on being able to compose a system with another federated system stack, users will be able to customize by swapping between many different networks rather than by breaking protocol compatibility.

8. **Look for customization agnostic opportunities.**

   In contrast to the above advice, there are certain types of benign customization that cannot be used to break compatibility, because they are somewhat format agnostic. Consider the case of blogs – as long as some basic level of format is met, a user will know how to read it. That said, these opportunities are likely rare, but if you come across one, you should definitely try to make use of it.

Acknowledgements

Thanks to Neha Narula for reviewing a draft of this paper. Thanks to Ethan Zuckerman and Chris Peterson for their excellent instruction in the Internet as a Social Artifact course at MIT, and to the rest of the IASA class – it has been a pleasure getting to know the group!

References

[1] Edwin Amenta. *Making the Most of an Historical Case Study: Configuration, Sequence, Casing, and the US Old-age Pension Movement*. `http://www.socsci.uci.edu/~ea3/Amenta.2009.byrne.ragin.Ch20.pdf`.

[2] Alberto Arce, Desmond Butler, and Jack Gillum. *US secretly created 'Cuban Twitter' to stir unrest*. http://bigstory.ap.org/article/us-secretly-created-cuban-twitter-stir-unrest.

[3] Paul Baran. *On Distributed Communications*. http://www.rand.org/content/dam/rand/pubs/research_memoranda/2006/RM3764.pdf. 1964.

[4] Manuel Blum. "Coin Flipping by Telephone a Protocol for Solving Impossible Problems". In: *SIGACT News* 15.1 (Jan. 1983), pp. 23–27. ISSN: 0163-5700. DOI: 10.1145/1008908.1008911. URL: http://doi.acm.org/10.1145/1008908.1008911.

[5] Jon Brodkin. *Comcast now has more than half of all US broadband customers*. http://arstechnica.com/business/2015/01/comcast-now-has-more-than-half-of-all-us-broadband-customers/.

[6] *CAN-SPAM Act: A Compliance Guide for Business*. https://www.ftc.gov/tips-advice/business-center/guidance/can-spam-act-compliance-guide-business.

[7] Patrick Coutermarsh. *FACEBOOK: The Psychology Experiment You Consented to in FB's Terms of Service*. http://www.scu.edu/r/ethics-center/ethicsblog/business-ethics-news/20119/FACEBOOK:-The-Psychological-Experiment-You-Consented-to-in-FB's-Terms-of-Service.

[8] Richard Esguerra. *An Introduction to the Federated Social Network*. https://www.eff.org/deeplinks/2011/03/introduction-distributed-social-network.

[9] *Federated Blog - P2P Foundation*. http://p2pfoundation.net/index.php?title=Federated_Blog&oldid=63858.

[10] S. Felter. *An overview of decentralized Kalman filter techniques*. http://dx.doi.org/10.1109/STIER.1990.324634.

[11] *GNU social and GNU FM*. http://webcache.googleusercontent.com/search?q=cache:PilclRI0si4J:https://gnu.io/+&cd=1&hl=en&ct=clnk&gl=us.

[12] Elliotte Rusty Harold. *Email: The Internets First and Last Killer App*. http://radar.oreilly.com/2013/12/email-the-internets-first-and-last-killer-app.html.

[13] Benjamin Mako Hill. *Google has Most of My Email Because it has All of Yours*. https://mako.cc/copyrighteous/google-has-most-of-my-email-because-it-has-all-of-yours.

[14] *Home - spamicity.info*. https://spamicity.info/.

[15] Lee Hutchinson. *How to run your own e-mail server with your own domain, part 1*. http://arstechnica.com/information-technology/2014/02/how-to-run-your-own-e-mail-server-with-your-own-domain-part-1/.

[16] Pete Keen. *How and why I'm not running my own DNS*. https://www.petekeen.net/how-and-why-im-not-running-my-own-dns.

[17] Adam D. I. Kramer, Jamie E. Guillory, and Jeffrey T. Hancock. "Experimental evidence of massive-scale emotional contagion through social networks". In: *Proceedings of the National Academy of Sciences* 111.24 (2014), pp. 8788–8790. DOI: 10.1073/pnas.1320040111. eprint: http://www.pnas.org/content/111/24/8788.full.pdf.

[18] David Mazi‘eres. *The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus*. https://www.stellar.org/papers/stellar-consensus-protocol.pdf.

[19] *McDonalds (MCD) Stock Price, Financials and News — Fortune 500*. http://fortune.com/fortune500/mcdonalds-110/.

[20] Stijn Peeters. *Beyond distributed and decentralized: what is a federated network?* http://networkcultures.org/unlikeus/resources/articles/what-is-a-federated-network/.

[21] *pump.io by e14n*. http://pump.io/.

[22] Clay Shirkey. *Delegated vs. Federated ID*. https://sites.psu.edu/ntsh/2010/02/15/delegated-vs-federated-id/.

[23] *The diaspora* Project*. https://diasporafoundation.org/.

[24] *The internet is our social network — friendica*. http://friendica.com/.

[25] Adam Thierer. *Unnatural Monopoly: Critical Moments in the Development of the Bell System Monopoly*. http://object.cato.org/sites/cato.org/files/serials/files/cato-journal/1994/11/cj14n2-6.pdf.

[26] Sean Tilley. *Planting a Seed: Diasporas Story (Part 1)*. https://medium.com/anti-fiction/planting-a-seed-what-working-at-diaspora-was-like-cde26fa29364.

[27] Paul Vixie. *Events of 21-Oct-2002*. http://c.root-servers.org/october21.txt.