Appendix

From: Gunes Acar Gunes.Acar@esat.kuleuven.be

Not all of the mentioned (decentralized personal data architecture) systems designed to achieve widespread adoption, so the criterion for the evaluation (little adoption, thus failure) may miss the point for some of the systems.

Putting such a system to widespread use have something to do with the amount of investment, marketing, public relations, or similar business related factors which are independent of design characteristics / system architecture.

Centralized systems enjoy the network effect as the authors argue, but this may (and does!) lead to monopoly, especially if data portability mechanisms/standards/regulations (if there is any) are extremely weak as they are today.

I doubt if more control "almost inevitably translates to" more decisions. Do decentralized systems need to have thousand of privacy knobs to achieve more control?

Speaking of cost...
users already pay for the access costs in centralized systems.
marginal costs incurred by using a decentralized system may be pretty low, or even may be zero for those with flat rate data plans. Take Torrent users: they happily share their bandwidth with others.

Section 4 :
I think access control is not the ultimate mechanism achievable for "mapping information flow" by decentralized systems. IMO one can devise smarter clients that can offer finer privacy controls.

I think offering nudges, or similar feedback don't require a centralized system.

Speaking of not relying on friends' software implementations, one can never rely on browsers too, but they're the main tool to access centralized SNSs.

One can circumvent the revelation of consumption by private information retrieval or dummy traffic.

Section 4, (including 4.1) lists some difficulties achieving a easy-to-use decentralized system, but all such systems (including the centralized ones) have similar difficulties. Given enough incentive they're (most probably) solvable. Speaking about difficulties of auto-updates or backups does not prove anything about the impossibility of achieving a successful decentralized system. Also some of these drawbacks are equally applicable to centralized systems (e.g. hardware backdoors)

The main problem facing interoperability, IMO, is not the technical/standardization issues, but the lack of incentives. Locking down users to their platforms is clearly more profitable for commercial SNSs.

From: Sonja Buchegger buc@csc.kth.se
Subject: Re: deadline for responses tomorrow
Date: August 8, 2012 1:07:51 PM GMT+02:00
To: Seda Guerses sguerses@esat.kuleuven.be

Hi Seda,

Here are a few quick thoughts. […]

drawbacks according to Arvind's paper:
technical factors:
- analytics hard in DCS (decentralized systems)
at least more costly, yes. This can also be a good thing for privacy preservation.

- higher unreliability (CAP theorem)
CAP is a concern.

- slower, sync need, data duplication minimization challenge
not necessarily, also to be traded off against drawbacks from centralization

- standardization hard (many interoperating protocols etc.)
This concern is not specific to decentralized systems. Centralized services/systems and walled gardens are not good examples of standardization and openness.

economics:
- economies of scale, network effect better in centralized
This depends on the particular application. Several decentralized systems have been developed precisely to get more scalability, e.g. file-sharing.

- path dependence
- unraveling suggests intermediaries might not represent a stable equilibrium (?)
Not sure what is meant by this statement.

cognitive factors:
- require software installation

This concern is not specific to decentralized systems. So do popular applications like Skype or Spotify on computers. Smartphones have found wide-spread adoption despite or because of the apps one can install on them.

- more control means more decisions
This concern is not specific to decentralized systems. Good default values reduce the number of decisions. Besides, not offering features in the assumption that people do not want to make decisions might be a bit overprotective of users and curtailing their control.

- security vulnerabilities when users without expertise configure software
This is of course always a risk. When it happens in a decentralized system, most probably one user's data is endangered. What happens in centralized systems today is well chronicled at datalossdb.org – one security exploit leaking the data of up to millions of users.

- users may be unable to verify privacy guarantees provided through crypto
norms of information flow:
This concern is not specific to decentralized systems. Users are unable to verify privacy guarantees of centralized service providers as well. Open-source communities and researchers can test and verify security and privacy properties. Users are unable to personally verify any highly complex system - medication, technology, transport, etc. This concern is not specific to decentralized systems.

- degrees of publicness (FB stopping search bots even for public data)
This concern is not specific to decentralized systems. With privacy-preserving decentralized systems, the goal is to give users control over their data and to come up with access control mechanisms that allow for both fine and coarse grained control.

- nudges
- reveal access patterns on storage nodes
Yes, and countermeasures are needed (see our Metadata OSN paper at SESOC 2012). Still, the concern remains greater for the centralized provider having theses access patterns plus all user data, including derived behavioral and relational.

- reputation systems and privacy through obscurity harder
Probably.

control:
- need storage on personal (as eg. Amazon reveals data on subpoena), but asymmetric bandwidth, NAT, firewalls

Can use encryption for cloud storage and people are working on protection from the cloud storage provider. Other systems (again, Spotify, Skype) deal with the same network-related problems quite well.

- silent updates hard

yes, harder.

- hardware backdoors

This concern is not specific to decentralized systems.

- downstream abuse

This concern is not specific to decentralized systems.

cheers,
/Sonja

From: Sonja Buchegger buc@csc.kth.se
Subject: Re: deadline for responses tomorrow
Date: August 8, 2012 1:11:01 PM GMT+02:00
To: Seda Guerses sguerses@esat.kuleuven.be

forgot to say, I agree with many difficulties mentioned in the paper and focused on the problems I disagree with, most because they are not specific to decentralization. Also, the recommendations do not follow from the rest of the paper but appear rather ad hoc.

From: George Danezis gdane@microsoft.com
Subject: RE: are you there??
Date: August 6, 2012 9:41:02 PM GMT+02:00
To: Seda Guerses sguerses@esat.kuleuven.be

Seda,

Ok, so here it is. I greatly enjoyed the paper, and also composing my reaction to it.
Please do keep making me feel special by asking my opinion on architectures :-)

G

My thoughts on Narayanan et al.'s "A Critical Look at Decentralized Personal Data Architectures":

There is nothing more refreshing than a paper that slaughters a few holy cows. Narayanan's paper picks the idea of decentralized personal data stores and architectures to do exactly this. It brings many interesting points to the table, but in my view, its conclusions offer no more clarity than we had before on the topic of future privacy friendly architectures. Still, I am grateful they started this conversation, and would like to respond to their invitation to discuss by offering my own views on this topic.

First of all, the paper does a very good job to highlight (1) how diverse the space of Decentralized Personal Data Architectures is and (2) how ideas from 10 years ago have been re-hashed and became popular again around 2009. It also does a good job to point out that despite the hype, these architectures have not really taken off, not 10 years ago, and it seems neither this time around. What is the explanation for this? This is where our views somehow diverge. Yet, if we want privacy friendlier services to see deployment one day, answering this question is key.

Doggy Visions of Privacy

First of all some of the decentralized personal (or other) data architectures espouse social visions that are very peculiar, and far away from what a broad number of users would consider acceptable.

Many early censorship resistance systems (tangler, freenet) set as a technical goal for information to be stored on other user's computers, without those users being aware of what they are storing. Some went as far as stating as a security goal hat a user should not be able to discover what they host. This tendency became even more pronounced with the emergence of distributed hash tables as the dominant paradigm for implementing peer-to-peer routing: in DHTs the primary identifier is a random number, that distributes information around peers in as random a manner as possible. This is not an easy architecture to swallow: while, as a user, I am happy to host material for myself, my family, my friends I have no incentives to host random files from random strangers. Furthermore, some types of content would place me in legal trouble (no matter what the legal fictions of the techies building them are). These basic storage architectures assumed there exists a perfect common of peers, without constituting it, or constituting the institutions that will maintain it, regulate it, or police it. This politically and socially naïve vision, in my opinion, was behind the demise of peer-to-peer architectures 10 years ago.

In parallel, a number of services have attempted to place themselves as infomediaries: entities that act as brokers between users, and commercial entities that wish to use personal data from users. First, I share the paper's scepticism as to whether having such entities truly furthers privacy concerns, and the lack of incentives to involve those third parties. If anything the internet tends towards disintermediation, so third parties usually have to provide real value to be included in a transaction. Beside these, I think that what made such services unpopular is the unspoken social assumption they make: personal data is a commodity, that belongs to the individual. As such it can be sold, or maybe just licenced, as part of a contractual agreement for the mutual benefit of all parties. While this sounds win-win it is totally out of line with how people think of their data, and themselves. Privacy, and personal data is about identity; it is about intimacy; it is about the negotiation of social perceptions and of the self; it is about coming out; it is about uncertainty; it is about feeling exposed or vulnerable -- these are not quantities that can easily be packaged, monetised and traded in a market against tangible benefits. Again, the vision underpinning those systems is more akin to a neo-liberal fantasy, and not in line with what I would expect to be doing with my on-line data or identity.

Thus, in my view, the failure of these two families of systems represents a failure of their vision, about what privacy and autonomy is. It does not come as a surprise, and does not need any deeper technical explanation. (What may need some explanation is why they are still so popular amongst some researchers, which is more related to the history and philosophy of science, rather than privacy friendly engineering).

A New Hope?

This brings us to the newest families of systems from Diaspora to status net and FreedomBox. Do these represent a step change in terms of conceptualizing Decentralized Personal Data Architectures?

Since I argue that the failure of other systems is due to poor social vision it is worth noting that Diaspora and Freedom box also were the product of a social vision. This vision originated from a talk given by Eben Moglen on "Freedom in the Cloud" (Feb 5, 2010 -- http://isoc-ny.org/?p=1338), in which the privacy and autonomy problems of cloud services (with a focus on social networks) were described and the FreedomBox architecture was proposed. His criticisms are credible, but of course his architecture does suffer from the fact he is a lawyer, not a distributed systems expert. This is a first problem.

The Resistible Rise of Centralization

Building large distributed systems is very hard. Building such systems on heterogeneous platforms is harder. Building such systems across multiple, mutually distrustful, security domains is harder. Making them work on low-power, sometimes off-line systems is harder. Making them fast, responsive, secure, with slick design is hard. Designing those by committee of a large number of coders that came together to do a community project, instead of the a-team of extremely qualified architects is super-human. Building them on a budget of $200K is laughable. These are some of the obvious problems that Diaspora and freedom box are to face, or are already facing.

The difficulty of building high quality and reliability large systems is also keeping the current paradigms of centralization alive. Most deployed "cloud" services use off the shelf components to achieve their scalability and robustness. They have a distributed data store that scales up, and provides reliability despite failures. They run code on servers that are stateless (in the long run), so that when they fail nothing bad happens. They use simple synchronization primitives (queues and locks) to pipe data from processing to processing to storage in a way that is reliable and slick. Batch processing is facilitated by asynchronous map-reduce engines. That's it really -- these are basically the ingredients that hide the complexity of large data centres behind a few APIs. The existence of high quality components to perform those operations, as well as operators that host them, allows the amazing economies of scale, and in parallel entrenches the paradigm of large centralized data stores. It is not without irony, given Moglen's argument, that some of the best implementations of these mechanisms are now open source.

Besides the availability of these high quality components, that make the job of building a large scale system manageable, this centralized architecture also benefits from an extreme form of the end-to-end principle: all the intelligence is only at one end. This makes service updates trivial to support, and new functionality easy to deploy and test. Javascript is the only credible mobile code out there, and the browser isolation model, for all its flaws is the only credible lightweight isolation model out there. Anyone who has tried to chroot a process, or manage different user account with different privileges on a conventional OS appreciates this. Furthermore, this task is performed automatically, instead of relying on a power-user configuring the system.

As a result of those economies of scale, even systems that have no technical reason to be centralized such as email, personal blogs, personal web-pages, code repositories, have gone centralized. In turn this created network effects for the platforms that drove more usage to them. Meanwhile peer-to-peer solutions were bogged down in weird social visions, and lengthy political battles about copyright, as well as mundane difficulties to even reach peers through NATs.
(Some of these pains are described as early as 2008 by IMC London volunteer Yossarian: http://london.indymedia.org/articles/203)

A lot has been said about the fact that centralization is driven by potential ad revenue, which is the dominant model of monetization today. I am personally sceptical about this. I instead believe that ad revenue is the result of those extremely efficient and scalable architectures, not the opposite. It is clear that in 1999, when Google launched its search engine, they did not have a clue how they will monetize. Ads came years later. Facebook, has been just about in the black, and a significant part of its revenue (about 10%) comes from selling token for addictive games. To me what really drives these platforms is the amazing economies of scale, and ad revenue is just the current, very historical, way of monetizing having user accounts. Tomorrow this may change to be a fee users pay -- yet some of the privacy concerns will remain. (In fact the amount of money some services expect to make out of me is so pathetic, compared with my electricity bill or even ISP, that I have considered sending them a cheque -- see this for more information: http://www.technologyreview.com/graphiti/427964/is-facebook-worth-it/).

The poverty of privacy engineering

So, the rise and rise of centralized architectures does not come in a vacuum, and neither did the "privacy" solutions fail in a vacuum. There were clear technical advantages to centralization. In my view projects like Diaspora and FreedomBox have a better social vision than other Decentralized Personal Data Architectures, but do not hit at the heart of what makes centralized solutions so tempting. I can illustrate the problems through a $1B service that I want to build:

"I want to make an app, called The Swan (from my local pub) that allows me to advertise where I am to my friends, geared towards serendipitous encounters. The idea is that when I go to some places, like my local pub or a café, the app asks me whether I want to advertise the fact that I am there. If I say yes a red dot appears in my friends maps (they can hover over it to see it's me, Mr Cool having a beer), and a notification appears in their feed, or their phones buzz if I am close by (within 1 mile)."

Building this in a centralized manner is easy. I get an account with amazon EC for compute and an S3 storage account (equivalent with Azure). I store all colocations + friends by user / time in the DB. And make a web page for updating and displaying the info (with a nice REST api). I also make 2 apps, one for iphone and one for android, that buzz and then link back to the mobile browser. I use Facebook connect to seed the user's networks, and allow login. I can do all this in about 2 days.

Now imagine how I could do all this, if I had chosen a de-centralized personal data architecture, ala Freedom Box. Storing data on end-user devices would be a nightmare. I would have to sync between laptop, phone and possibly some encrypted on-line service. Oh, there are no good encryption libraries for python. Making binding for OpenSSL would take 6 hours. Then I would have to find a way for peers to communicate to each other. Well, that could happen through end devices talking to each other (NAT nightmare), or through a user having to run a server part of the service in some public IP address. Ok, so now I do not any more provide a service, but a the software for someone else to run a service. This means installation nightmare = no one will use this. My coding will be all on the client side, which means that I would have to replicate a lot of functionality in Python (for PC app / server component), javascript (for web front end), objective-C and java. And keep those in sync.

To me it is not a surprise that there is not a single service that takes the second option. The existence of status net (to provide status updates) or even Diaspora for some rudimentary social networking does not change anything: the experience of building privacy friendly services, over the equivalent centralized ones, is simply a misery. Even if one is totally clued up on how to build a privacy friendly system (which in itself is hard) building it is the most unpleasant experience.

Thus I find that some of the soul searching in the paper redundant: yes, there are too many standards; yes, adapting / updating systems is important; indeed, context preservation is nice. But I feel the simple explanation that building decentralized services is hard, and made harder by of the lack of high quality components to support the task, is equally adequate.

What future for privacy architectures?

This is my key divergence (although not by much) from Narayanan et al's advice and recommendations: I would advise the architects of Decentralized Personal Data Architectures to focus on software components, infrastructure and services that make the job of building privacy friendly architectures as easy as centralized ones. Killer apps, are of course necessary, but to a first approximation those will be the same as for centralized systems (see the idea above). Users should not automatically be aware from the feature set that the underlying architecture is different!

In particular we need high performance personal decentralized storage with simple APIs. I want to be able to say mystore.write(key, value), and this information to be replicated and synched across my devices and servers (automatically encrypted when it is not in my physical control). Yet, I also want to be able to store information and make it accessible to friends or even public without cumbersome key management. We need to have domain isolated computations on client platforms that can run code that comes from fresh up-to-date sources. We need high performance and reliability event queues that notify clients of interesting events in other devices, belonging to other users. Finally, I want to have social APIs running from and updating my local stores. All of this is high quality software, that is currently missing.

But since the model today is that organizations do not build software, but run services, we need platforms to run the privacy friendly server components. In the same way that I would buy an S3 account and some compute from Amazon, I want to be able to buy the encrypted storage sync component for my users (in case they do not have their own), or users should be able to subscribe to a provider of such storage if they are more sophisticated and *want* their own. Note that I do not want to have to re-write the complex crypto, sync-to-local-storage protocols, or reliable queues, in the same way I do not have to rewrite the replication and reliability protocols for cloud storage.

Finally, some services may be needed to support privacy architectures. One benefit (often overstated) of having a silo of personal data is the ability to extract statistics. Maybe it would be beneficial to collect aggregate statistics in a privacy friendly manner, and third party services could be contracted by the service providers to facilitate this (think: Onion routers!). Identity provision and management could be another one -- although the failures of most identity systems do not speak highly of this possibility.

Note that the strength of mechanism I expect from those technologies are not on-par with previous attempts. For example, while I expect encryption, and not simply access control, to enforce policies on sharing, I would not expect full traffic-analysis resistance for all storage. As we discussed building distributed systems is hard -- building them while requiring perfect security (in terms of leakage not quality of implementation) is impossible currently, even in theory.

So overall the goal should be to allow for innovation in using privacy preserving architectures -- when the simple service I described could be implemented in a couple of days in a privacy friendly, non-centralized manner -- I think we will see a higher rate of adoption of these technologies. Since we start with pretty much zero adoption this is a relatively modest target.

Can this really happen?

My theory is that fashion in centralized and decentralized architectures comes and goes in cycles. This is due to technical disruptions, as well as economic and political considerations -- once a technical paradigm, its software / hardware embodiment, and the largish companies that benefit from it become established they quash innovation to retain their position in the eco-system. At the point the other paradigm re-emerges as a competitor, and new poles of power start constituting themselves.

I still remember the times when "programming" meant building desktop applications. The assumption was that the user data is somewhere around the user's computer, i.e. in some files. Sharing was done by physically moving those files around in floppy disks. Updates had to wait for a new version of the software, that came in disks attached to magazines in the case of freeware. It was a horrible time, and I am glad it's over -- it took a day to build a nice dialog box.

I was glad to be using HTML and CSS for the design of the UI, dynamic languages like ruby and python for the back end, and standard web apis for storage and social stuff. I am so happy that some google / amazon / microsoft engineer has built reliable storage and compute for me, so that I do not have to. I hate javascript, and so do you, but we are stuck with it is seems -- and will have to rely on jQuery and friends to take some of the pain away.

Yet, it also seems that new devices now can also be programmed with those technologies -- yes, native apps, can run in JavaScript against what were thought to be web-apis, and server components being local services. This seems to be the new programming model for Windows 8 that will unify devices: phones, tablets, desktop apps and web-sites will not be programmed at the same high level, and against similar apis. Apps (mobile) are the new thing, and the model of the app store has made it as easy to update client components as downloading web content. Now my phone has about 16GB of memory, and as much compute as my computer 6 years ago. Maybe this is a sign that the times are changing again towards clients? This will only be an opportunity if those supporting privacy friendly architectures, including a mature form of Decentralized Personal Data Architectures, work hard to make it into one.

From: Scott L. David sldavid@uw.edu
Subject: RE: 2012-08-07-idn PDEC FW: request for comments
Date: August 10, 2012 12:10:38 AM GMT+02:00
To: […] Seda Guerses sguerses@esat.kuleuven.be

I think the article is very thoughtful and analytically helpful in calling out some of the challenges.

Since I didn't have a copy where I could interlineate comments, please see general comments below.

These are in no particular order

1. There is mention in the recommendations (number 3) of the need for socio-legal approaches. I think that the article (and future research) would benefit from that emphasis. Identity systems need to be reliable to deliver privacy, security, liability limitation, identity assurance and other desirable qualities consistent with stakeholder expectations and needs. Technology (hardware and software) is made reliable by being built in conformity to specifications. People in the system are made reliable by conformity to rules (policies and laws). The privacy and other desirable qualities will be best achieved not by a more effective "vault" system, but by a "mutual deputiz-ation" that arises when duties are assumed by populations.

2. Some of the issues raised might more easily be addressed if we apply the notion from Shannon's quantitative theory of information that "Data plus meaning equals information." First, Shannon's formula suggests that data and information are not identical. Second, it suggests that data is the "raw material" for information. Meaning in Shannon's formulation is added by the observer/user of the data. For a single piece of data, there may be multiple observers/users. Exposing data to multiple users in a way that is consistent with data subjects interests yields "data leverage." As long as rules for distribution norms and contextual appropriateness (to borrow some of Helen's concepts) are observed, data leverage can be consistent with a data subjects "identity integrity" (aka privacy) expectations. In fact, data leverage yields value that helps to incentivize third party users/observers to abide by the rules set forth (for fear of losing the privilege of access. This "unpacking" of the data and information concepts allows data to flow (and to be managed as a "common resource") consistent with big data needs/expectations, while allowing the protection of the subset of observed data that becomes "personal information" as a result of the observation.

3. The architecture issue is interesting in this context. Getting very theoretical for a moment - Gödel's incompleteness theorem posits that no system can be entirely known entirely from within - The externality is also relevant. With some analytical leaps, that suggests that privacy is not addressed by secrecy, but rather through a mutual desire for privacy held across a population. The externality there is the other folks that might come in contact with data about a particular data subject. Again, it goes back to Helen's concept of distribution norms and contextual

appropriateness. Those concepts do not rely on "secrecy" but rather on proper behavior of others to create a system for data privacy and a system of "pipes" for data flows.

4. paragraph 3 of section 4 talks about standardization. The market will provide part of the standardization mechanism here, but that will take time.

5. In the sentence following footnote 43 ("However, it is not fully clear why many types. . .") the question is raised about why intermediaries have not arisen here. I think that process is continuing. It depends on a sufficient definition of roles, which in turn depends on clarity of needs. The scope of the potential harms that are captured in the rubric of "privacy" are poorly defined in most jurisdictions, with the result that intermediaries cannot yet step forward to "intermediate." It is like appointing an agent before the principal knows what they want the agent to do. I suggest that there will be a role for intermediaries as the equivalent of "publicity agents," in effect selling agents for personal data about people who are answerable to the data subjects themselves. Some might question the apparent paradox that privacy can be served by making data more accessible. I suggest that it provides greater efficacy for data subjects, who are then in a position to withhold incentives if their requirements are not met.

6. I think the concept of "absolute control" and the emphasis on technical solutions in section 4.1 is distracting. The last sentence of section 4.1 is the most important in my view. Identity is a social construct, data is generated throughout society about its stakeholders, the notion of control is best viewed in this context.

Anyway, those are a subset of my thoughts. I think the paper is great. Very thought provoking and nicely argued and presented.

Please let me know if there is interest in discussing the issues further in this or other contexts.

Thanks
Scott

From: elijah@riseup.net
Subject: Re: request for a response on a paper on distributed/decentralized networks
Date: August 3, 2012 7:53:14 AM GMT+02:00
To: Seda Guerses sguerses@esat.kuleuven.be

just a quick response, more to come later.

in brief: fuck decentralization. i agree with harry: the main problem is people applying their political ideology onto the technology. decentralized tech is a long way off from actually working well, and even longer way off from delivery any anonymity improvement, if ever.

our new project, LEAP, is basically an attack on decentralization and people who fetishize decentralization. how? by showing that a federated system is better. (i am using the term decentralized for p2p and federated for client-to-server-to-server-to-client).

anyway, here is a slightly outdated table (improvements forthcoming) that i have been using to explain our critique of decentralization and our strategy:
https://we.riseup.net/assets/101780/infosec-table.png

our focus is on usability, unmappability, and authenticity--using a federated model (called oddly 'polycentric' in that chart). the logic is that the other security properties are useless without authenticity (everything could be mitm) or usability (people don't use it or use it wrong).

the problem with decentralization is that it makes identity and authentication much much harder. the problem with centralization is that it makes authentication somewhat useless, because the authority that is validating identities can always fake it (a la skype).

so, by process of elimination, we are focused on making federated models as secure as possible.

anyway, i will let you know my response to the paper.

-elijah

From: elijah@riseup.net
Subject: Re: request for a response on a paper on distributed/decentralized networks
Date: August 4, 2012 3:57:10 AM GMT+02:00
To: Seda Guerses sguerses@esat.kuleuven.be

my comments on the paper, use as you will, or not. -elijah

My general response is this: while I agree with many of the critiques of distributed/p2p architectures (and I have a couple dozen more to add) trying to combine p2p and federated in the same category is not useful and leads to very jumbled arguments. The differences far outweigh the similarities.

Inline comments:

> web community

It would be more accurate to say "internet community" instead of "web community". This paper is not about the web, but the internet.

> > Indeed, we suggest that much of the reason for what we see as overenthusiastic claims about decentralized systems is that design characteristics have been confused with values.

True! But mostly just true of p2p.

> > First is functionality: there are several types of computations that are hard or impossible without a unified view of the data. Detection of fraud and spam, search, collaborative filtering, identification of trending topics and other types of analytics are all examples.

These are all really different. Any system, centralized or not, that encrypts user data (or even promises too, a la skype) will have challenges with aggregate functions (search, filtering, analytics). On the other hand, any federated system with cryptographic signatures is highly resistant to spam and fraud.

> > Decentralized systems also suffer from inherently higher network unreliability, resulting in a tradeoff between consistency and availability (formalized as the CAP theorem [57]); they may also be slower from the user's point of view.

Centralized systems face this as well, obviously (since all the big ones are effectively a private federation among far flung datacenters). It is only a matter of degree. In the case of federated approaches, the network is not that much slower, the availability can be made very high, and the consistency measured in a matter of tens of seconds rather than seconds. From the perspective of the user, there is not a substantive distinction. The user experience with federated jabber is the same as the user experience with facebook chat (at least in terms of consistency and availability). Email is a good example of a federated protocol that has consistency and availability problems. These problems have mostly been solved, from the user's perspective, despite how decrepit and ancient the protocol is. I think it would be more accurate to say that p2p systems generally have much lower availability, which is certainly true. This is good example where the distance between centralized and federated is less than the distance between federated and p2p.

> > The need for synchronized clocks and minimizing data duplication are other challenges.

I am not sure what this is getting at. It is not possible to run even a centralized service with incorrect clocks.

> Shapiro notes two benefits of standardization: greater realization of network effects and protection of buyers from stranding, and one cost: constraints on variety and innovation, and argues that the impact on competition can be either a benefit or a cost [50].

Yep, standards take time, sometimes a very long time. And thus far, they usually win in the long run.

> Centralized systems have significant economies of scale which encompasses hosting costs, development costs and maintenance costs (e.g., combating malware and spam), branding and advertising.

Up to a certain point. As hosting becomes commodified, this is less and less of an issue. Development scale is a real problem, but not as much as people imagine. I think most senior technologies would tell you they could reproduce the core functions of any site or service given a talented team of ten programmers. Centralized systems face extreme difficulties when dealing with systems that must support tens or hundreds of millions of users. Other systems face scaling issues, but they are a magnitude more manageable.

> A variety of cognitive factors hinder adoption of decentralized systems as well. First, the fact that decentralized systems typically require software installation is a significant barrier. Second, more control over personal data almost inevitably translates to more decisions, which leads to cognitive overload. Third, since users lack expertise in software configuration, security vulnerabilities may result. A related point is that users may be unable to meaningfully verify privacy guarantees provided through cryptography.

These are all true now, but these properties are NOT inherent to a federated approach and some of them are not true of a distributed/p2p approach (p2p does suffer from inherent usability problems with existing technology when it comes to cryptographic identity verification, e.g. Zooko's Triangle). Our project, LEAP, is designed specifically to address these current limitations by reducing user decisions, removing configuration errors, and establishing strong cryptographic validation without user intervention. These are not hard problems, because they are not technical properties of federated systems but rather problems with existing implementations.

> 4.1. On Control over Personal Data

This thought experiment only applies to distributed p2p systems, not federated ones, although no distinction is made.

> Finally almost all decentralized architectures face the problem of “downstream abuse” which is that the user has no technical means to exercise control over use and retransmission of data once it has been shared.

This is true of all digital data, regardless of distribution architecture.

> > One major impediment is that there are too many standards to choose from. For the most basic, foundational component—identity—there are many choices: OpenID, WebID and others.

OpenID and WebID (and BrowerID) are all trying to solve different problems than the problem of federated identity for social networks or messaging. These protocols are trying to solve the one problem of the user authenticating with the provider, they do not attempt to solve the problem of users authenticating one another. Also openid/webid/browserid are mostly intended for systems that would be considered centralized in this context. Same with OAuth, although it too is addressing a very different problem.

If you want to talk about a foundational 'identity' component of federated or p2p networking, I would say there is nothing serious that anyone has even put on the table.

> > We conclude that while federated social networks have the potential to converge on a reasonably interoperable collection of software—subject to the caveats of differing feature sets and parameters—it is not simply a matter of making some technical decisions, but instead needs serious developer commitment as well as the involvement of standards bodies with significant authority.

I think it is unfair to measure the process of standardization on the same timescale as innovation in the silo. Of course it will take a long time, and of course the problems are more social than technical.

Unlike most people, I feel that it is a good thing that there are no dominant proposals for social networking. If one cares about privacy, I don't think any of the current proposals offer much solace. If one cares about confidentially or social graph mapping, none of them offer any solace whatsoever (except for a few theoretical proposals that I also think won't work). The current crop of social networking protocols are all designed to maximize the properties of control and interoperability, at the expense of other properties. This is a bad trade-off from many perspectives. Fortunately, in the long run, I don't think this is a necessary trade off.

So, I would say it is great that none of them have taken off. Death to them all.

Also, I think it is a misreading of history to suppose that what is needed is a standard body with authority to sort out the mess.

The more I think about it, despite your impressive efforts to tease out a meaningful dialog, I feel like the paper is hopelessly muddled and impossible to redeem, even as a device to spur a constructive dialog. The analytic categories are simply too jumbled to be salvageable.

Here is my very nascent take on a similar problem of teasing out the trade offs between centralized, federated, and p2p architectures (within the single domain of messaging):
https://leap.se/en/technology/infosec

Partially, I am of course coming from a different perspective than the authors. I know the capacity for surveillance and the historical/structural forces at work expanding that capacity. So I am simply disinterested by any analysis that does not take surveillance seriously.

-elijah

From: Benjamin Greschbach bgre@kth.se
Subject: Re: distributed systems
Date: July 2, 2012 2:27:38 PM GMT+02:00
To: Seda Guerses sguerses@esat.kuleuven.be

Hej Seda,

I really liked the paper, as it not only gives a systematic overview of DOSN approaches but also a very interesting analysis of their drawbacks. I can agree with many of the raised issues, some I don't see as absolute or severe as the authors did, others I don't agree with (e.g. the cognitive factors they list). You find some thoughts about some of the concrete items below.

Best regards,
Benjamin

== Sec 4, technical disadvantages ==

• functionality ("computations that are hard or impossible without a unified view of the data").

Not sure, if "impossible" really holds for any operation. Isn't theoretically every computation possible in a distributed manner and it's only the question at what cost? The interesting questions here are probably which parties get to see what fraction of the data and which trust assumptions one has for these parties.

If there is a functionality, that can be implemented in a way, that two parties do the computation on half of the data each (revealing only the result of the computation to each other), than I would see this as an advantage over a centralized, completely integrated system (at least if the two parties are administratively/organizational separated). Furthermore, this issue can be treated feature-wise (rather than it dictates a global design decision). For example search could be outsourced to a central provider (leaving the users in control over what data they expose to the central search-directory server), while the rest of the SNS is implemented in a decentralized way.

* unreliability ("inherently higher network unreliability")

Depends on the definition of reliability and the baseline for the comparison (twitter "fail whale"). E.g. censorship resilience might be better in a distributed system and scalability can at least theoretically profit from a peer-to-peer approach, where each new member does not only consume but also contribute more resources to the network. Furthermore, centralized systems already trade consistency for performance (I remember something about best-effort approaches for rendering a Facebook wall which yielded different views of the same wall for different members), and this does not seem to bother users too much.

• standardization/innovation

The observations might be true for federated systems, but if development of a DOSN is in one administrative domain, I could imagine that they are equally innovation friendly. In this context, Mozilla software or Ubuntu are examples for

- working "business models" not based on collections of user-data,
- deploying security critical software products (having silent auto-updates),
- based on community reviewed open sources,
- and being innovation friendly (new features can be rolled out immediately).

== Sec 5, cognitive factors ==

• software installation

- there are examples for applications, where users are willing to install software (Spotify, DropBox, Wuala)
- on smartphone devices, standalone apps are more common than browser-based apps

• decision complexity

I don't see that "more control" leads in any way to "more decision options". I understood the "more control" goal in all approaches I have seen so far in improving the enforcement of the user-defined access rights. Not in allowing more fine-grained access right settings (can it be more fine-grained/complex than the Google+ model, at all?)

• user lack of experience

Again, I don't see that a DOSN necessarily has a more complex configuration space on the user-interface level. This might be true for many of the academic, proof-of-concept, alpha implementations, but not an inherent consequence of the decentralization. Furthermore, centralized, web-based services face the same challenge for maintaining a secure browser.

- users inability to verify privacy guarantees

If you let the crypto people explain it to the user, this might be an issue ;-) Otherwise, I don't think that users have to understand the math in order to appreciate a privacy guarantee. There's so many examples of very complex systems (e.g. cars) where users without a deep understanding of the technical details are still able to make meaningful decisions on a more abstract level.

== Sec 4.1, control over personal data ==

I agree that control over hosting does not immediately lead to privacy, but did anyone claim that at all? In most DOSN approaches, control over hosting is only one building block, where encryption is another and then only the combination of both is claimed to guarantee stronger privacy properties. Another motivation for provider independent hosting can be the possibility to choose a different host-provider, without being locked-out from all personal data.

== 5. Recommendations, 1., 2. ==

In general I agree with the importance of the "business model" and "users's values" issue. On the positive side there I would see that I understood the user-studies literature about "the value of privacy" not as clearly pointing to "nobody cares about privacy". People might even be willing to pay a small amount of money (I remember a user study about an email service, where the authors pinned it down to a couple of dollars, can't find the reference unfortunately, but related to this topic are e.g. http://dx.doi.org/10.1007/1-4020-8090-5_10 or
http://www.guanotronic.com/~serge/papers/isr10.pdf).

From: Michael Herrmann Michael.Herrmann@esat.kuleuven.be

Hi,

in general I like the idea of the paper and it definitely has its reason. For quite some time I think about how well decentralized networks actually work and if they are not being proposed to be used too darn often. Sometimes I think, that especially we in academia seem to avoid the real controversy when using decentralized P2P networks.

I am not familiar with most of the works they cite, especially in Section 2. I am a little bit surprised, that their main argument seems to be: Every software that has tried to get a job in a decentralized way is either very niche or (mostly) doesn't work and therefore, decentralization doesn't work. I would disagree with this statement and when reading the paper, I get the feeling they want to argue against decentralized networks.

I think the paper gives until Section 4.1 nice examples and fairly good arguments. Especially the two classifications seem to be interesting and I would like to read something about that in a full paper. However, from Section 4.1 until the end, I think the authors try too much to argue against decentralization. For example:

Section 4.1:

2nd paragraph: Counterargument -> We can store data encrypted into the cloud.

3rd paragraph: There are solutions to overcome the NAT problem.

4th paragraph: Don't see their point here at all. I think that's a good example to see, that they want to argue against decentralized/open source systems.

I think Section 5 has good and bad points:

[...]

1. One could address this point also by saying: Although this software is not heavily in usage, it still gives users the possibility to use it. For example to send encrypted messages via Facebook.

2. More elaboration on that one would be very nice.

With the other points I can more or less agree.

So, to summarize: Although I am not super familiar with all their cited work and I don't like some parts of the paper (especially not Table 3), I think that in general the criticism is appropriate. Definitely, I am not happy about how they tackle the problem of judging about decentralized networks. I would rather see technical measurements to justify their claims.

If you have any question, please just let me know.

Best,
Michael

From: Jaap Kuipers (Id Network) jaap.kuipers@id-network.nl
Subject: 2012-08-07-idn-PDEC
Date: August 7, 2012 5:31:19 PM GMT+02:00
To: Seda Guerses sguerses@esat.kuleuven.be

Hi Seda,

Some remarks…

ad. 3. Classification

Table 1.
I agree with the idea of a spectrum between self-hosted and outsourced.
I suggest the classification gives some extra examples

- Physical location of data
  - on a users device (under control of the user)
  - distributed (network of indexes, like eHealth dossier in the Netherlands)
  - at a trusted third party (this is a more or less legal concept)
  - remote
  - "opaque", unknown : we do not know where the data are

Test question: can I ask for the removal of my data out of all the backups? Does the provider give a warranty for long time archival (say 30 years?)

- Legal location of data
  - under which jurisdiction
    - US
    - US+ safe harbor
    - EU
    - Asia
  - (the power to gain control over data for instance after death of the owner, in a law case)

- "Ownership" of the metadata
  - Who are allowed to use the (meta) data
    - for statistics

        - for "improvement of user experience",
        - ananymous for profiling

- Governance over data storage
    - best effort (Google)
    - under a service level
    - audited

Test question: can the provider be held liable for the misuse of my data under its custody?

I miss more ideas about mechanism for giving (delegating) access to others (friends, family, colleagues, groups, attribute owners, medical staff etc) What will the mechanism for sharing be. Real personal data that does not leave my own personal domain is perhaps not the most important issue. It is about sharing, rights management. Perhaps a policy definition language.

ad. 3 point 4 Data portability

Data portability is a good classifier. Are users able to move their own data to prevent vendor lockin.

ad. 4.2 Open standards and Interoperability

The statement about StatusNet is new for me. It does not point in the direction of work done by the Identity Commons, Kantara, OIX foundation.

ad 5. Point 7. Work with regulators

I agree. Banks know a lot about personal data we trust(ed?) banks because the are regulated. We need regulation and law to protect the interests of individuals, groups. If ICT will be an important part of society the ict use will becom regulated like the use of automobiles, it will take some time.

From: Nicolas Maleve copy.cult@constantvzw.org
Subject: Re: August meet
Date: August 10, 2012 9:23:59 PM GMT+02:00
To: Seda Guerses sguerses@esat.kuleuven.be
Cc: […]

[…]

It is very difficult to really respond to the text. I have the feeling that the scope is very wide and doesn't really help to go beyond a series of general considerations over a series of elements. What I lack in the article is a coherent vision of their articulation. They are described to me as juxtaposed problems, but if they are not related to a political vision and project for society, it can only lead to a boring conclusion of balancing interests. And as we know the balancing of interests in the democracies are rarely more than a validation of the relationships of power already in place. I am very interested in your response, again sorry to be late.

,n

Van: Markus Sabadello markus.sabadello@gmail.com
Verzonden: woensdag 11 juli 2012 23:28
Aan: Jaap Kuipers (Id Network)
Onderwerp: Re: 2012-07-11-idn FW: request for comments

Hello Jaap, nice to hear from you,

I did indeed meet Seda in Amsterdam, and also before at the second Federated Social Web Summit in Berlin. I am a big fan of her work, for example her talks about the "responsibilization" of users.

[…] and I both know that paper, and it is actually referenced in our third (April) issue of the Personal Data Journal, to which I contributed an article titled Personal Data in Decentralized Network Architectures" (find it attached)

The paper in turn mentions several members of the PDEC Startup Circle.

Here are a few quick thoughts:

- One issue is that people mean very different things when they talk about "decentralization" and "peer-to-peer". There are many variations and hybrid forms. Therefore section 3 of the paper is an okay approach for some basic classification, but its view is also overly simplified.
- The paper is perfectly correct about the fact that decentralized architectures also have many disadvantages, people sometimes forget that.
- One funny thought that keeps popping up more and more is that the reason why decentralized network architectures are not successful is because they contradict a capitalist economic system and therefore have little chance of getting funded. For more on this, see e.g. Dmitry Kleiner's work (The Telekommunist Manifesto), or this publication.
- It is obvious that all of the current decentralization projects are just too complicated to use for anyone other than geeks. Unfortunately, many grassroots communities working on such projects are building them primarily for themselves, rather than for a general audience.

- The EU Commission has allocated some money to fund anti-censorship projects. I don't know too much about that, but perhaps there is potential to support not only anti-censorship, but also more generic decentralized networking projects.
- Maybe another good person for Seda to talk to would be Heather Marsh of The Global Square. She has done a few great interviews, e.g. here.

My overall assessment of the current state of decentralized networking alternatives would be that awareness for this is growing rapidly, and the only reason why they haven't taken off yet is because they're not good enough, but could be very soon.

I'd love to catch up with Seda at some point. Maybe you could keep us updated with her research, or maybe it would be possible to participate in her online debate?

all the best,
Markus
--
Project Danube: http://projectdanube.org
Personal Data Ecosystem Consortium: http://personaldataecosystem.org/

From: Antonio Tapiador antonio.tapiador@guest.kuleuven.be
Subject: Re: [SPION-RESPONSE] article by narayanan et al.
Date: July 5, 2012 4:09:29 PM GMT+02:00
To: […]
Cc: Seda Guerses sguerses@esat.kuleuven.be

In my point of view, the technological issues mentioned in the article are not so important. It is true that there are a lot of issues to be solved, starting from choosing one protocol for federation (OStatus vs XMPP, as we saw last year in Berlin).

(I am choosing federation over distribution as I see it more plausible and reliable, in the same way email works now.)

However, the main issue for me is who will drive the effort. Who are the social actors that are interested in federating? Certainly not the main social network sites, as now they enjoy the privilege from walled gardens. But I believe that they will finally join, if federation occurs. The most difficult phase is obtaining critical mass in the network, so it is attractive enough and more people is willing to join.

Who is willing to make the initial effort?

From: Hellekin Wolf

Here we go...

> “Decentralized social networking has been a largely parallel, sometimes overlapping line of development with similar motivations. We subdivide such social networks into federated (ecosystem of interoperable implementations in the client-server model) and distributed (peer-to-peer). The term distributed social networking is frequently but incorrectly used to describe all decentralized social networks.”

What a pleasure to read this paragraph. Finally, someone is using the right terms for "federated" and "distributed".

Unfortunately the "Representative Survey" does not mention GNUnet and Briar approaches, both distributed and specific. GNUnet started as a peer-to-peer file-sharing system, to evolve into a full-blown overlay network spanning from local WLAN on OSI layer 2 to several application protocols. Briar does not rely only on "on-line networks" but also on "off-line" exchange of data and secrets, making it the first hybrid digital/analog P2P network, if you except the deadswap project of Dmytri Kleiner. Another big absent in this survey is Delft University's P2P project Tribler.

The "Classification" raises an eyebrow: although it warns about the fact it's rough, it insists more on the vertical axis of the cut than on the horizontal axis: obviously, "federated" systems can be self-hosted, unless self-hosted excludes devices like the FreedomBox. Note that I still consider hosted-in-a-datacenter as a case of outsourcing.

"Open standards vs. proprietary" and "Open vs. closed-source implementations" sound controversial to me: usually, we talk about "open standards" and "free software" on the one hand, and of "proprietary standards" and "privative software" on the other hand, if one has an incentive to distinguish between various degrees of deprivation: an open standard with a closed-source implementation is as privative as a proprietary standard with an open-source implementation. In any case, separating standards from their implementations seem dubious to me, especially in the perspective of rising free hardware--such a distinction simply cannot be made there.

From there on, I'm looking at the paper from a distance. I'm waiting for the "liberation phase", or phrase, that will tell me it's not entirely against decentralization.

I read: "The benefits and costs of standardization are a prominent socio-technical factor." What is "socio-technical"? Isn't it "technological"? Why use a different term...

> “Many decentralized systems depend on multiple interoperating pieces of software, which requires standardization of technical protocols, design decisions, etc.”

That is exactly what makes UNIX superior to other approaches of OSes: one tool, that does one thing, and does it well. All software requires design decision, actually, all technology is about making design decisions. standardization of technical protocols happen in two ways: top-down, via, e.g., the ITU, the IETF, or the W3C, and they're usually clumsy, (see the evolution of HTML) or bottom-up, via massive adoption: Bittorrent is the most beautiful example of that approach. Interoperability is not a technical issue, but a political one. "Many decentralized systems" AND ALL TECHNICAL SYSTEMS. Moreover, one cannot talk of "an ecosystem" regarding a set of corporate and institutional processes engulfing millions of EURO of functioning costs every year. That is an artificial construction, not an ecosystem.

> “A related point in the context of social networks: we hypothesize that the network effect is stronger for centralized systems due to tighter integration.”

I assume we're talking here about "social network services" and not about the more generic term of "social networks": e.g., my neighborhood, or: the sum of human relationships I maintain in my neighborhood. The above quote sounds like an endorsement of fascism.

When talking about "historical accident" regarding "Path Dependence", and citing 2 lines later that "commercial applications" remained "unsurprisingly centralized" is beyond my understanding. What historical accident is there that a commercial company, centralized, efficient because inherently tensed toward accumulation and profit maximization, would generate a centralized application reproducing just the same? Is that an historical accident that we chose profit over social wealth?

> “However, it is not fully clear why many types of intermediaries have taken hold in many other markets—employment agents, goods appraisers, etc—but not in the market for personal data.”

Guy Debord might hold an answer to that question. At some point, people stop not caring. Rape is not of those situations.

> “A variety of cognitive factors hinder adoption of decentralized systems as well. First, the fact that decentralized systems typically require software installation is a significant barrier. Second, more control over personal data almost inevitably translates to more decisions, which leads to cognitive overload. Third, since users lack expertise in software configuration, security vulnerabilities may result. A related point is that users may be unable to meaningfully verify privacy guarantees provided through cryptography.”

Thank you for this paragraph. First point is, again, not a technical matter, but a political one: all software has to be installed at some point, and if your "OS of choice" comes loaded with decentralized software, that is a non-issue. Second point seems to me a question of usability rather than a consequence of decentralized systems: the best illustration I can find is the Microsoft Windows thing that constantly pops up stupid questions that are most of the time entirely irrelevant, and the astounding effort of Facebook to design an impossibly boring interface to one's privacy settings, both cases leading to cognitive overload, as a wanted way to prevent the user from taking control of the application; I don't see more decisions to be made with a software based on its centralized or decentralized nature, and I can imagine properly designed and configured software that gets out of the way. The third point is not an issue of centralized vs. decentralized systems, once again: you can have a centralized system (e.g. GMail) and use encryption in it, leaving Google AdBot clueless; it's a political question and a design issue, as well as an educational one: before cars, nobody knew how to drive, before SMS, nobody knew how to send them, etc. This whole paragraph talks about PERCEIVED barriers that are IRRELEVANT of the scope where they're considered.

> “Finally, we find that decentralized social networking systems in particular fare poorly in terms of mapping the norms of information flow. Access control provides a very limited conception of privacy. We provide several examples. First is the idea of "degrees of publicness." For example, on Facebook”

Facebook is not decentralized. What kind of example is that?

> “Second, in current social networks privacy is achieved not only through technical defenses but also through "nudges". When there are multiple software implementations, users cannot rely on their friends' software providing these nudges.”

Yes, but that's not a problem of decentralized software. That's an issue of interoperability development. When you have a single entity doing everything (e.g. Facebook), it's obvious that it will be compatible with itself. In the case of decentralized approaches, many different projects with various kinds of fundings, have to confront their ideas. It will always take more time, but in the long run, the free software has demonstrated that it's superior to the centralized approach, didn't it?

> “Third, distributed social networks reveal users' content consumption to their peers who host the content”

Not always...

> “(unless they have a "push" architecture where users always download accessible content, whether they view it or not, which is highly inefficient.)”

You obviously don't grab the concept of "push", and how it is way more powerful, scalable, and efficient than the existing infrastructure. Have a look at Secushare's multicast contexts to get an idea.

> "Finally, decentralized social networks make reputation management and "privacy through obscurity" (in the sense of [26]) harder, due to factors such as the difficulty of preventing public, federated data from showing up in search results."

You are conflating two concepts: public social networking activity, which results in public contents, analyzable, search-able, etc., and private social networking activity, that relates friends to friends, and won't end up in search engines nor in reputation systems.

So far, all the flaws you describe about decentralized systems are either not relevant to decentralized systems per se, or transient, due to the current situation, or misconceptions of the scope of the applications. You're offering some interesting insights on what can get better, but overall, you're missing the point.

4.1 On Control of Personal Data

Here you give a good synthesis of the state of affairs. Interestingly enough, you conclude that "absolute control is impossible in practice". Obviously, "absolute" anything is impossible in practice, but you do not mention that a FreedomBox could work in the case you describe, given that it runs on free hardware designs. Again, you make the false assumption that things are here to stay.

> "Further, it suggests that control over information is probably not the right conceptualization of privacy, if privacy is the end goal."

Here you raise an interesting question...

4.2 Open Standards and Interoperability

Lorea, was first to implement all OStatus protocols. But we're still hitting a few walls, here a fundamental design issue in the OpenID librabry implementation, there a lack of standard definition of OStatus beyond basic microblogging, etc. And that cannot be solved because people cannot actually work together by lack of funding. Meanwhile, big corporations sit in four-star-hotels to chat about the next top-down centralization-improving standard. WTF?

Also, I don't think that approach will be successful because it requires an ever-expanding vocabulary that will become too large (oh, great specialized jobs creation for more people to get busy doing shit nobody needs!) to handle.

Regarding federated authorization, it does not pose any more problems that local authorization. It's even implemented with OAuth. The real problem is that OAuth was designed as a temporary authorization mechanism for a very specific action, but the big companies use it as a permanent, do-it-all mechanism, actually stepping over its original intent, and allowing any application to suck anything it wants from the users without any regard for their privacy. There's no historical accident here, but a purpose.

> “We conclude that while federated social networks have the potential to converge on a reasonably interoperable collection of software -- subject to the caveats of differing feature sets and parameters -- it is not simply a matter of making some technical decisions, but instead needs serious developer commitment as well as the involvement of standards bodies with significant authority.”

When was the last time "standards bodies with significant authority" brought anything good to the Internet? No, I can't remember either.

5. Recommendations

5.1 Economic incentive does not work in social networks. They work in markets, and the market of "centralized social network services" is marketing over users' profiles. The "economic incentive" of Facebook users is to "be connected" because "everybody else is there". That is a characteristic of the private network to show an abundance of users, but when people realized that real abundance lies in the public network, and the tools there are good enough, they will switch.

5.2 Yes

5.3 I guess that is what FreedomBox provides, a server within the scope of the privacy of your home. the GNU project, EDRI, LQDN, etc. are also looking at the legalese... I guess the CIRS should do it that way too.

5.4 You want to lookup Briar again.

5.5 Yes

5.6 AFAIK Lorea provides no less features than Facebook does. They're just less well integrated--we don't have billions of dollars, and thousands of developers. And we're not about replacing Facebook either: FB is a surveillance machine, we're a community-amplifier.

5.7 Interesting. Thank you for the tip. One day we talk about the CIRS project.

6. Conclusion

I don't agree that you "brought fresh perspective to the question of why they have largely floundered." I think you have a couple of interesting hints, but overall the vision is restricted to an economic market environment that fails to reflect to profoundly social and affective foundations of social networking in general, and their online representation in particular. For example you failed to analyze the economic incentive behind centralization of the services, where we had a decentralized Internet before- that you mentioned at the beginning of the paper.

I appreciate you end your conclusion with "market equilibrium", which makes absolutely no sense in my world, even if I know what it means from my previous studies of Economics. It's just that Economics now is irrelevant to study something that encompasses Politics, Sociology, Psychology, Economics, etc. Looking for market equilibrium is such a context is wishing for the centralized services to win the battle for controlling group thinking. I fear they already lost, but nobody knows it yet. It's flawed at the root: it's not an ecosystem.

Nevertheless, I'm looking forward for more research in the field.

Regards,
==
hk

From: […]
Subject: critical look at decentralization
Date: […]
To: Seda Guerses Seda.Guerses@esat.kuleuven.be

Hi Seda,

I received the mail pasted below from […], and here my quick comments:

«The term distributed social networking is frequently but incorrectly used to describe all decentralized social networks.»

I completely agree and I really like you mention it.

«Diaspora are a hybrid between distributed and federated.»

I'm not sure if finally it's distributed. Recently looking at the code the only I could find is that they're using PubSubHubbub

«OStatus, being coordinated by the W3C, represents an interesting approach to standardization for federated microblogging: it references a suite of existing protocols rather than developing them from scratch.»

Though recently is almost died and have very little activity.

Maybe some other two main approaches for federation are, both using the publish/subscribe model, aka push technology:

- Google PubSubHubbub (PuSH): a research, non end-user ready example using it is SMOB, and wrongly using the name distributed, BTW
- XMPP PubSub: an example using it is Buddycloud

«First is functionality: there are several types of computations that are hard or impossible without a unified view of the data.»

With unified view of data you mean data in the same format or in the same place?

I think that this is true for computations like social graph analysis and graph transversal (data needs to be in the same format/some machine) but AFAIK, some P2P networks can perform computations like recommendation system (Tribler). But I should document myself.

«Detection of fraud and spam, search, collaborative filtering, identification of trending topics and other types of analytics are all examples»

Again I think that in some P2P networks are possible these analytics

«data duplication»

I don't see this as an inconvenient, but rather an advantage

«hypothesize that the network effect is stronger for centralized systems due to tighter integration»

I don' see why. What I see is that when a social network reaches the critical mass (maybe through marketing) there's this social effect in which people join a network because all their friends are there.

I'm sorry I didn't finish reading the paper and I can't comment anything about the privative/closed software and standards because I don't know them.

In a simplistic and general way, I'd say that the "failure" of decentralized social networks more as a political issue than a technological one.

As a ridiculous example, in 3 different flats here in […] they've asked me if I do "bad" things with internet, just because they hear that I do something with P2P and here they're educated (or to say it better, scared) with this Hadopi law...

Good look with the paper, and sorry I didn't have more time for a more scientific critical.

[…]