Social Personal Data Stores: the Nuclei of Decentralised Social Machines

Social Personal Data Stores: the Nuclei of Decentralised Social Machines

Max Van Kleek
Web and Internet Science
University of Southampton, UK
emax@ecs.soton.ac.uk

Daniel A. Smith
Web and Internet Science
University of Southampton, UK
ds@ecs.soton.ac.uk

Dave Murray-Rust
School of Informatics
University of Edinburgh, UK
d.murray-rust@ed.ac.uk

Amy Guy
School of Informatics
University of Edinburgh, UK
Amy.Guy@ed.ac.uk

Kieron O'Hara
Web and Internet Science
University of Southampton, UK
kmo@ecs.soton.ac.uk

Laura Dragan
Web and Internet Science
University of Southampton, UK
lcd@ecs.soton.ac.uk

Nigel R. Shadbolt
Web and Internet Science
University of Southampton, UK
nrs@ecs.soton.ac.uk

ABSTRACT

Personal Data Stores are among the many efforts that are currently underway to try to re-decentralise the Web, and to bring more control and data management and storage capability under the control of the user. Few of these architectures, however, have considered the needs of supporting decentralised social software from the user's perspective. In this short paper, we present the results of our design exercise, focusing on two key design needs for building decentralised social machines: that of supporting heterogeneous social apps and multiple, separable user identities. We then present the technical design of a prototype social machine platform, INDX, which realises both of these requirements, and a prototype heterogeneous microblogging application which demonstrates its capabilities.

Keywords

Decentralising the Web, Social Machines, Software Architectures, Privacy

1. INTRODUCTION

The increasing centralisation of the Web remains the greatest threat to its continued existence as a democratic and ubiquitous shared medium of communication [3]. Although originally designed as a decentralised system to ensure longevity and sustained fair and equal access to all that use it, today the Web has become a highly centralised environment, dominated by very large institutions that each control all of the traffic that flows within their respective borders. The result is that such institutions harbour a disproportionately large percentage of Web traffic, and, in turn, exercise an unprecedented degree of control over its governance and operation. Moreover, this control extends not just to the operation of the sites themselves, but also over the personal data that people are voluntarily pouring into them, whether they pertain to one's online social network activities, or, increasingly, the contents of one's personal information collections shifting from desktops into "the cloud".

The centralisation has made service providers the *de facto* locus of control for both user data and interaction. Whenever a service provider wishes to change some functionality of its service, such as to roll out a new feature, it is within their power to simply roll out changes, and the users of the platform essentially are usually quite powerless to do anything about it regardless of their preferences or needs. This is not good for users for multiple reasons; first, it disenfranchises people from the choice to have the features they want, or in the ways that they have grown accustomed to. Secondly, the sudden and unexpected roll out of new designs and features often results in people perceiving changes in more negative a light, ultimately slowing and stagnating innovation.

This relates to social machines because the Web itself has proven the most profoundly transformative social machine in the history of humanity, fundamentally enabling currently over 60% of the world's population to work, collaborate, meet more effectively, interact socially and enjoy new forms of recreation. Within the Web, studies of differences among individual "sub" social machines, such as social networking platforms, online communities and the like, have shown that subtle differences among each have greatly shaped the resulting community around them and the ways people use and interact through them [14]. The implications of these observations is that even relatively minor details of next-generation Web based social machines will likely have a tremendous impact on the ways people will interact with and through them. As a result, we feel that it is very critical that requirements and technical design decisions be driven by an understanding of what makes today's Web based social machines effective, sustain, as well as of where they fall short or how they fail.

In this paper, we extend a line of work we have introduced previously ([19, 17, 18]) pertaining to investigating the fundamental idea of giving end-users of the Web more sophisticated

SOCM2015 2015, Florence, Italy

data management capabilities "at the edge", ie.. on the physical computing device(s) that they own and/or control. In particular, we focus on one very specific aspect of personal data store work pertaining to social machines: the problem of building social platforms in fully decentralised Web environments.

2. BACKGROUND

Before the rise of the massive social platforms, decentralised operation was essentially the norm; operating systems like UNIX supported a number of utilities for connecting with other hosts in ways that could be either seen as primitive social software (e.g. UNIX Talk), or used to build social software (e.g. telnet for MUDs and MOOs). Even after the growth of massive providers, being able to connect multiple disparate platforms was seen as a reason to build protocols like Jabber/XMPP[15] or to use a common social protocol such as IRC. Sadly, the use of such standard protocols seems to be on the decline, with Google and Facebook both terminating support for XMPP on their Talk and Facebook chat services, respectively, essentially, for the moment, eliminating the possibility of bridging these platforms.

Meanwhile, this ever-increasing centralisation has also had the positive effect of inspiring both academic and private sector efforts at building decentralised versions of popular social platforms. Such efforts themselves have now become so distributed and widespread that a number of different indexes exist on the web to track these efforts, ranging from Wikipedia[1] to Alternative-Internet[2] and the IndieWeb movement[3].

These systems can be roughly divided into three classes corresponding to their degree of decentralised operation. The first category, containing possibly the largest number of independent web projects, are those that are simply standalone servers intended for operation and use by the end-user. In this basic category, services are in essence entirely centralised, but are made to be run by the end-user (on his or her own computational substrate) instead of by a third-party, thus providing for entirely private operation, maintenance and control. The second category extends the first with interoperability through the use of common, standardised data representations and APIs to enable interchange among heterogeneous systems. Such social protocols include **Activitystreams**[4], which are a syndicate format for feeds from social networking services, as well as **XMPP**, **OpenID**, **OAuth**, and **WebID**, among other standard formats for identity and data exchange. A third category, in contrast to the first two, implements social applications in a distributed, rather than federated, fashion over an open set of participating nodes, typically end-users themselves or volunteers. This is the philosophy of applications such as **Bittorrent**, **Freenet**, **IPP**, **Tor**, **Twister**[5], and **Bitcoin**; in some, such as Bittorrent, Freenet and Twister, participation requires contribution of the computational, network and storage resources required to keep the service running; while in others, such as Tor and Bitcoin, one can use a service without contributing, i.e. serving as a Tor relay/exit node, or Bitcoin Miner.

A second dimension on which these systems differ is the extent to which they are bespoke to a particular application. Many have 'baked in' functionality specific to emulating the biggest social applications today, roughly comprising interfaces for social networking (e.g. activity status update and photo/item sharing), microblogging, blogging, forums, and file/data item storage and hosting. Others, such as Tor, are purely generic, providing generic functionality for building social applications, such as anonymous data routing and connectivity. Still other examples exist that are the creation of new social applications by co-opting others; an example of this is the a proof-of-existence service[6] that uses the Bitcoin blockchain to prove that particular information was known by a party at a particular relative time.

Finally, perhaps the the most well known distributed social networking project is Diaspora[7], once hailed as the 'decentralised Facebook-killer' [8]. Its architecture is most relevant for the discussion in this paper, because it, like ours, is more decentralised than distributed; anyone with appropriate network and computational resources can host their own "pod" server, and engage in Facebook-like social networking activities with others either within the same pod or those hosted on others. Unlike Diaspora, however, which has a bespoke social networking application baked into it, our architecture is designed to be a generic substrate for building decentralised social applications, as we describe next.

3. DESIGNING A PLATFORM FOR DECENTRALISED SOCIAL MACHINES

Given the varied and abundant efforts at decentralising the Web just described, we wished to know whether the capabilities envisioned by such efforts would align well with the potential design requirements for future social machines [7]. To identify such requirements, we undertook a brainstorming activity with two experts on personal data architectures and one on Web-based social machines. Out of the twelve design requirements we derived, six remained unfulfilled by any of the proposed decentralised projects; out of these, we focus on two which are the most relevant to social aspects of future social machines.

3.1 Heterogeneous Social Software

One of the key goals of the efforts at decentralising the web is to put people "in control" of their social software. But the implications of such a simple statement could be many; one, for example, might be to "control" data generated in social interactions (e.g., one's tweets, status updates, instagrams), which we interpret to be able to access, use, appropriate, such content as one wishes, indefinitely. At this level, several of the existing personal data stores already fulfil this need well; many PDSes interface with existing social platforms on the Web through APIs, to make and store copies of the individual's own content with the objective of storing this content for safekeeping.

Another intepretation is that end-users should control the look, feel and functionality of their social software. One way that this has been achieved in the past is the use of third-party apps that provide different interfaces and, in some cases, add functionality, to the big Web social platforms. However, nearly all of the major social platforms today are actively shutting down the ability for third party developers to do so for the purpose of exerting more control (although often done under the auspices of protecting platform users – a separate issue entirely) [11]. As a result, in the foreseeable future it seems that the major social

[1]Comparison of software for distributed social networking `en.wikipedia.org/wiki/Comparison_of_software_and_protocols_for_distributed_social_networking`

[2]Alternative Internet: `github.com/redecentralize/alternative-internet`

[3]IndieWeb `indiewebcamp.com/`

[4]Activitystreams specification `http://activitystrea.ms/`

[5]Twister, A Fully Decentralised Microblogging Client - `http://twister.net.co/`

[6]Proof of Existence - `www.proofofexistence.com`

[7]Diaspora - `joindiaspora.com`

platform providers will mandate use of their apps and clients only, eliminating user choice in interface or features.

In a decentralised setting, however, such mandating use of a single client is not only unlikely but also potentially harmful to widespread adoption. Therefore, we believe that we will see a rise of *heterogeneous social software*, consisting of what might be thought of as an ecosystem of "apps" today that support social interaction despite their not being the same.

The standard method for allowing heterogeneous clients to interact is establishing a common protocol for interaction, such as XMPP, Activitystreams, as mentioned earlier. Indeed, the many aforementioned systems already do this. However, this is a conservative approach that slows innovation, because the vocabulary has to be established essentially *a priori* to creating applications, and, applications must fully comply with the protocol(s) in order for systems to be able to interact. Thus, in such scenarios protocol designers have incredible power by essentially dictating how a social machine is to work, yet, based on previous examples, are rarely involved in the process of designing apps themselves.

An alternative approach is to provide support for more organic interoperability by allowing app designers to extend or forge new data representations themselves, and then, to promote interoperability, support others' use of their representations though representational alignment. We propose that such support could easily be provided by the social machine platform framework, and describe this functionality in the next section.

With this approach, barriers between the activities supported by individual social machines can be arbitrarily blurred by social app designers; if an app maker adds a new kind of social action to an existing data type, for example, to support 'retweeting' instagrams, they are free to do so; and any new apps can choose to support this action or ignore it.

3.2 Identities, Pseudonyms and Personas

We have used the term *personal* data store extensively, but without fully exploring what personal really means. As they live their lives, individuals tend to carry out a variety of different tasks and activities, for different reasons, with different groups of people. This is as true online as it is offline, but with some differences. In a recent study, we discovered that a majority of people who use the Web perform some kind of identity separation online, in an effort to control the information they choose to make available about themselves in particular contexts [9]. Our study also found that people become frustrated when they unwittingly lose control of their personal data as it is propagated throughout systems by algorithms and architectures of which they are not aware or do not fully understand. Examples of this include changes to Facebook privacy settings which suddenly expose sensitive status updates to work colleagues, or forced linking of Google+ and YouTube accounts which reveals real names and locations of users to aggressive commenter.

Unintentional spreading of personal information from one group, community or network to another can result in collapsing of social contexts [10], which potentially has serious implications for the individual involved. People take steps to mitigate against this in many different ways, as described in [9], including lying about personal data when they feel it is not required; keeping distinct social media accounts to separate work and personal affairs, or to post different types of content to; clustering sets of accounts together under a single pseudonym to maintain consistency without exposing their offline identity; exaggerating or omitting aspects of themselves to perform for or better integrate with a particular community; keeping 'safe' social media accounts for their friends and family whilst expressing their true feelings from alternative 'secret' accounts.

Yet, service providers continue to develop more sophisticated methods to counteract this behaviour, in an effort to more accurately track individuals for advertising purposes, and to enforce real-name policy routines. Click-stream profiling and Deep Packet Inspection (DPI) have become increasingly commonplace [4], which make it extremely difficult for end-users to avoid being re-identified.

Existing work on Personal Data Stores largely fails to account for the need for multiple identities, effective separation of roles and anonymity, and to prevent unwanted tracking and clickstream profiling. Some of the most prominent PDSes, such as Mydex, for example, aim to serve as 'trusted Identity Brokers which certify that people are who they "truly claim to be" (fulfilling a Versign-like role) [6], which is completely contrary to current Web users' most frequent needs when protecting themselves online. Even systems like Diaspora and Buddycloud, which support standard distributed ID protocols such as OpenID, fundamentally assume that the individual has and maintains a single profile and identity which they disclose to all services equally; a practice that does not reflect the reality of what people do, nor the various reasons for doing so.

In order to support people's needs surrounding their privacy, we sought to design a system to actively support the kinds of practices people already perform when interacting with the many large platforms on the Web. As we describe later, INDX aims to actively support the creation and effective management of as many separate identities as the person desires, and to permit fluid switching of identities as they switch from provider to provider. More critically, we sought to help protect users from unwanted context collapse through the use of counter-surveillance techniques that are even able to defeat most forms of clickstream and DPI. These methods remain an area of active research for us, and are described in greater detail in a separate paper [12].

4. INDX: A DECENTRALISED SOCIAL MACHINE PLATFORM

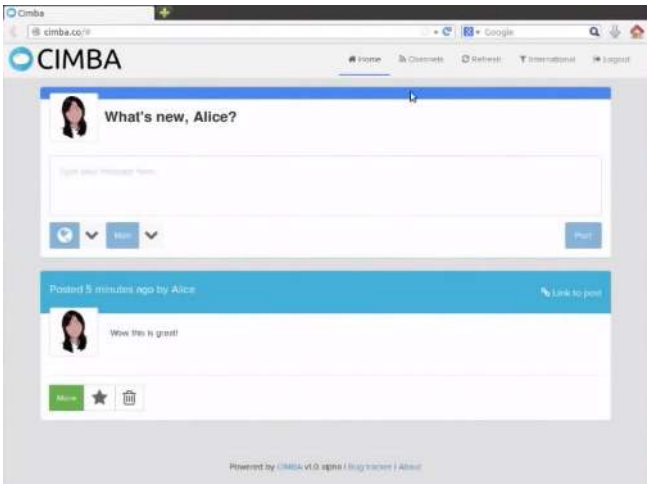
In this section, we describe our progress with a prototype Social Personal Data Store (SPDS) platform in which we first sought to realise the functionality pertaining to the aspects just described, as they were most germane to building social machines. To do this, we extended the INDX Personal Data Store [18] which was a fairly traditional centralised PDS that end-users of the Web could run on their personal compute hardware and use to securely store and archive content from all over the Web.

From this as a starting point, we added two capabilities: heterogeneous schema management, for the purposes of allowing multiple social apps to easily exchange information with one another despite having different data schemas and representations, and, second, multiple identity management capabilities in support of the aforementioned perceived need to help users manage multiple identities and personas online. We next describe the technical details of each.

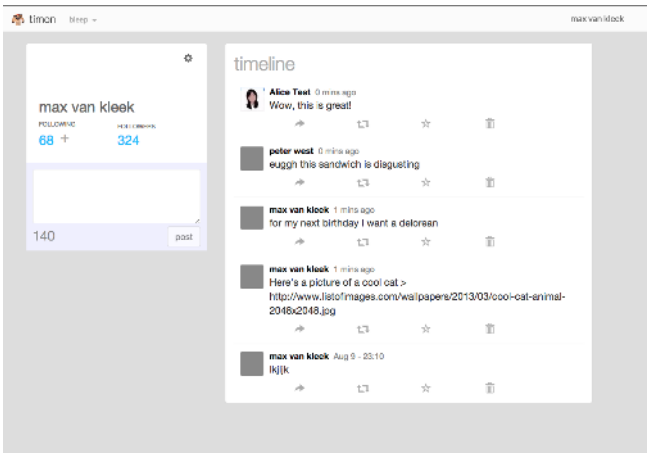
4.1 Heterogeneous Social App Data Mapping

As mentioned earlier, unlike in centralised Web settings where social platforms can fully establish and dictate the ways end-user clients interact with them, in an open, decentralised setting, there will be, in general, no such ability to fully control the constituents in the network. Therefore, supporting effective exchange among the heterogeneous constituents is a priority.

Figure 1: Transformer engine loaded with CIMBA-TIMON microblogging bidirectional transform rule from INDX representation to CIMBA.



(a) CIMBA



(b) TIMON

Figure 2: Original CIMBA client, top, running on the Linked Data Platform, communicating transparently with INDX's TIMON (bottom) via the schema-translation rule engine.

Fortunately, the Web has standardised, at the protocol level (e.g. HTTP, WebSockets, etc) how information should be exchanged; however, it does not dictate the forms and representations of the data exchanged.

Thus, for INDX, we focused on the problem of achieving effective heterogeneous structured data exchange. To do this, we adopted the pragmatic approach: a rule engine that is capable of transforming any simple structured data representation to another, through the use of modular rulesets. Modularity ensured that, as new social apps were introduced, or old ones were modified or improved, the data representations could be easily and appopriately made compatible simply by updating the corresponding ruleset.

Rule-based data transformation is a technique that has been applied to systems for decades; our contribution is simply, first, to determine whether this technique would afford sufficient flexibility to permit heterogeneous applications to be made to speak to each other, and second, whether the complexity of managing modular rulesets could be made sufficiently manageable for end-users. With respect to technical implementation, we considered a number of existing rule langauges and engines, including RDFS and OWL reasoners; for the first prototype, we opted for a simpler (less expressive) rule language based on INDX's query language, so that queries could be both succinctly expressed in terms of a INDX's native data format, and so that the satisfaction of rules could be performed directly within INDX's core Postgres engine itself.

Figure 1 displays a patching rule for INDX's *TIMON* adaptive microblogging application to communicate with the CIMBA linked-data microblogging platform [16, 13]. In the figure, the left-hand query represents a native INDX query which specifies the domain of the transform; the right-hand side represents the equivalent transformed representation that CIMBA expects. Note that CIMBA uses RDF, while INDX uses a JSON-like object format; yet, because JSON can express a superset of RDF, the rule language can effectively generate RDF. Note that, in its simple initial implementation, rules are uni-directional; therefore two symmetric rules are required for bi-directional integration. We are intending to make the rule-engine able to run rules in reverse (by using the output transformation syntax as a query for reverse transforms) in the next iteration of the prototype. The current versions of CIMBA and TIMON interoperating are visible in Figure 2.

The advantage of this approach over simply adopting a common vocabulary (such as done by Buddycloud or Diaspora) is flexibility; while a particular format or ontology might be standard today, a new application may include new information that poorly suits the schema, thus outgrowing it. As can be seen in the example, CIMBA itself uses two standard vocabularies intertwined, FOAF [1] and SIOC [2]. Not only can our approach directly support the particular peculiar use of these two standard vocabularies, but it can also easily be bridged, via a separate ruleset, to communicate with Diaspora as well, which uses Activitystreams, OStatus and other vocabularies.

4.2 Multiple Identities and Counter-surveillance

Fundamentally, we aim to have INDX support end-users' desires in keeping identities as separate as they wish, with the properties they please. The first feature we implemented was to support the creation and maintenance of multiple, separate identities and social networks; INDX facilitates creation of as many identities as the user wants. Each of these identities can be adapted to popular distributed ID representations (at this point,

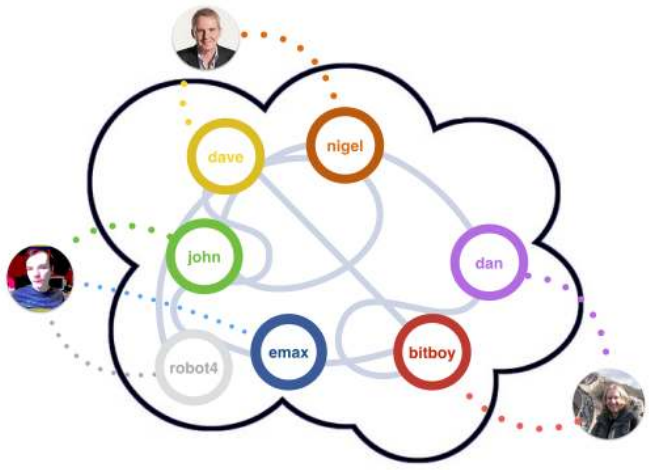


Figure 3: Cartoon illustration of INDX endpoints work in Tor; a single user (indicated as bubbles outside) can have as many identities as he or she wants; each identity creates a hidden service endpoint within the Tor network. When endpoints interact, they have no notion of which endpoints correspond to which physical INDX hosts or owners because all exchanges are conducted through the anonymising onion network.

comprising only OpenID or WebID, although more are planned) to facilitate their use with third parties. In addition to supporting distributed ID protocols, each identity can be associated with sessions/session cookies and principals created on closed Web-based platform identities. What explicit association between identities and principals does, thus, is allow the platform to help manage switching among identities at appropriate times, to aid in preventing accidental context overlap. For example, by storing credentials separately and by third party service, INDX can allow an individual to choose which identity to present whenever interacting with a third-party service. Eventually, we wish to add predictive algorithms to make this selection easier or even automatic.

The second built-in set of functionality we see as integral to supporting effective identity separation in INDX pertains to helping users to avoid being tracked without their consent, particularly using DPI, user agent profile and clickstream profiling. Such methods could cause service providers to infer a person's other identities, for example, defeating their ability to keep identities separate. Towards this end, thus, INDX makes default all connections to third parties to use Tor [5] to ensure that connection and packet-based profiling remains impossible. As a second measure, INDX integrates via a browser-plugin to provide coordinated user-agent randomisation for third parties corresponding to particular identities; for example, when interacting with Facebook under Identity 1, INDX might set the browser user agent to present itself as Safari running on OS X, while, with Identity 2 IE11 on Windows 8.1. While such counter-surveillance measures are still rudimentary compared to the level of sophistication being applied by service providers to track their users, they represent a first step in a programme to shift user control back to the user.

4.3 Practical Considerations w/ Connectivity

While not among the major themes of this work, we nonetheless wish to mention an important detail about INDX's implementation pertaining to connectivity that has fundamentally thwarted other "run-at-home" personal data store efforts. Although many such platforms (in particular, FreedomBox[8], BuddyCloud[9], CozyCloud[10], ownCloud[11] tout their suitability to be run on simple devices at home, this is, in practice, difficult for using standard Internet Service Providers for at least two reasons: addressability and reachability. Typically, ISPs use dynamic addressing to allocate addresses which can change spuriously (for some, with every home router reboot); therefore, decentralised services must employ sophisticated mechanisms to detect and re-discover a client's location every time the address changes. The second, reachability, is even worse - most home routers act as network address translators (NATs) that effectively partition off home computers from being able to be reached by the outside world (without elaborate mechanisms to get around this, such as employed by Bittorrent).

INDX hopes to set a precedent with a very simple solution, which is to use tor as a Virtual Overlay Network. The very specific way that this is done is that INDX creates tor hidden services to create stable endpoints. Since a hidden service's address remains the same regardless of network endpoint, this solves the addressability problem. Second, since tor pierces most NATs and firewalls, this also simultaneously solves reachability. Finally, because tor has anonymising properties, it even solves the anonymity-in-decentralised-environments problem; since tor allows services to easily create new hidden service endpoints, INDX creates one per public identity and onion routing ensures that it cannot be ascertained by the originator that both endpoints exist on the same logical host. Figure 3 illustrates this approach to preventing identity collapse.

5. CONCLUSION

Personal Data Stores face a significant uphill battle in the information marketplace, where the personal data economy is powered by the use and exploitation of personal data. Service providers are incentivised to use whatever means necessary to harvest as much accurate data from users as possible, so that this may be applied to marketing and advertising, and to keep users coming back. However, this desire for control may ultimately turn out to be an advantage for PDSes, by putting significant restrictions on what people can do on their platforms, and creating a feeling of personal exploitation.

Pushing personal data store efforts towards applications that achieve the Web ideal of true openness through purely participatory, decentralised operation is one that will take significant time, but also one that has the most potential for the future. People will have to get used to the once again being part of a community, contributing computational and network resources in exchange for making the community stronger. But in exchange, the potential to build entirely new kinds of social machines that both respect individual privacy (using strong guarantees nonetheless) and that continually keeps individuals in control of the things most important to them, we believe, will ultimately prevail.

We believe that removing constraints pertaining to centralised operation, and honouring the identity and privacy needs of individuals opens up a very large design space for future social

[8]FreedomBox - `freedomboxfoundation.org`
[9]
[10]CozyCloud - `cozy.io`
[11]ownCloud - `owncloud.org`

machines. Even within this initial work of designing the first ever decentralised social app for INDX, TIMON, we faced making essentially a large number best-guess decisions about how much control and functionality people would want. Even though TIMON looks like Twitter, under the covers, it supports a much more varied set of ways that it can work for the user. For one, TIMON can be used to microblog any structured content whatsoever, not just text; two, it can support an unlimited different number of channels, rather than one single stream. Channels can be propagated directly to followers, or bubbled throughout the network. The list of potential variations goes on indefinitely - even within the simple domain of microblogging.

Moving forward, we have a number of directions of planned work. Our current priority is to work on practical issues of making INDX useful and usable and make it available to the maker/hacker community. The second priority is to experiment with extending the rule engine with potential other automatic approaches - might automatic schema alignment algorithms be suitable to be used in place of having to write explicit matching code? There are also significant challenges with applications and security and trust - in a decentralised environment, there is no established common method by which one can assure that a piece of code, such as a social machine application, can be trusted to access private sensitive data stored within a personal data store. Therefore, we think that a social machines approach, e.g., the use of a community to audit and review application code might be a good approach, especially in combination with code signing mechanisms which will ensure that people are getting the same version that was audited. Finally, we also wish to extend the work on automatic and user-directed counter-surveillance mechanisms, including work on assessing the legality and ethics of the use of such methods when interacting with particular kinds of third parties, including governmental, medical, and other public and private entities.

6. ACKNOWLEDGEMENTS

This project was supported by the *Theory and Practice of Social Machines* project, funded by the EPSRC under grant EP/J017728/1. We would like to thank Sandro Hawke, Andrei Sambra, and Sir Tim Berners-Lee for their time, effort, advice and input on both INDX and facilitating integration with the CIMBA platform.

7. REFERENCES

[1] D. Brickley and L. Miller. Foaf vocabulary specification 0.98. *Namespace document*, 9, 2012.

[2] P. A. Champin and A. Passant. Sioc in action representing the dynamics of online communities. In *Proceedings of the 6th International Conference on Semantic Systems*, page 12. ACM, 2010.

[3] L. Clark. Tim berners-lee: we need to re-decentralise the web, 2014.

[4] R. T. G. Collins. Privacy implications of deep packet inspection technology: Why the next wave in online advertising shouldn't rock the self-regulatory boat, the. *Ga. L. Rev.*, 44:545, 2009.

[5] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. Technical report, DTIC Document, 2004.

[6] W. Heath, D. Alexander, and P. Booth. Digital enlightenment, mydex, and restoring control over personal data to the individual. In *Digital Enlightenment Forum Yearbook 2013: The Value of Personal Data*, pages 253–269, 2013.

[7] J. Hendler and T. Berners-Lee. From the semantic web to social machines: A research challenge for ai on the world wide web. *Artificial Intelligence*, 174(2):156–161, 2010.

[8] B. Kersey. The troubled history behind diaspora, the $200,000 facebook killer launched on kickstarter, 2012.

[9] M. V. Kleek, D. Murray-Rust, A. Guy, D. A. Smith, and N. Shadbolt. Self curation, social partitioning, escaping from prejudice and harassment: the many dimensions of lying online. Jan 2015.

[10] A. E. Marwick et al. I tweet honestly, i tweet passionately: Twitter users, context collapse, and the imagined audience. *New media & society*, 13(1):114–133, 2011.

[11] N. Mott. The era of third-party apps is ending, as security risks prompt whatsapp and snapchat to shut down their apis, 01 2015.

[12] D. Murray-Rust, M. Van Kleek, L. Dragan, and N. Shadbolt. Social palimpsests–clouding the lens of the personal panopticon. 2014.

[13] J. J. W. Presbrey. *Linked data platform for web applications*. PhD thesis, Massachusetts Institute of Technology, 2014.

[14] L. Ross and R. E. Nisbett. The person and the situation. *T. Nadelhoffer, E. Nahmias, & S. Nichols, Moral psychology: Historical and contemporary readings*, pages 187–196, 2010.

[15] P. Saint-Andre. Streaming xml with jabber/xmpp. *Internet Computing, IEEE*, 9(5):82–89, 2005.

[16] A. V. Sambra, S. Hawke, T. Berners-Lee, L. Kagal, and A. Aboulnaga. Cimba-client-integrated microblogging architecture.

[17] M. Van Kleek and K. OHara. The future of social is personal: The potential of the personal data store. In *Social Collective Intelligence*, pages 125–158. Springer, 2014.

[18] M. Van Kleek, D. A. Smith, N. Shadbolt, et al. A decentralized architecture for consolidating personal information ecosystems: The webbox. 2012.

[19] M. Van Kleek, D. A. Smith, R. Tinati, K. O'Hara, W. Hall, and N. R. Shadbolt. 7 billion home telescopes: observing social machines through personal data stores. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 915–920. International World Wide Web Conferences Steering Committee, 2014.