Functional breakdown of decentralised social networks

Student: Wouter Miltenburg
Supervisor: Michiel Leenaars (NLnet)
Research Question

❖ What current implementation of a social decentralised network could be considered as an alternative to the current centralised social networks and could be offered as a service by hosting providers?

Fig. 1—(a) Centralised, (b) Decentralized, (c) Distributed networks.
Research Questions

❖ Which functionalities exist in the typical social networks that we know nowadays?

❖ Which alternative open source projects are available that are mature enough and which provide these functionalities in a decentralised model?

❖ How do these different alternative open source projects differ from each other in a practical sense (e.g. security, standardisation, ID re-use, and scalability)?

❖ Which implementation is most suited to create a decentralised social network that can be provided by hosting providers as a service?
Related work

- D. Sandler and D. S. Wallach, Birds of a FETHR: open, decentralized micropublishing.

Related work


❖ Thiel et al. A Requirements-Driven Approach Towards Decentralized Social Networks.
Approach and methods

- Analyse existing centralised social networks
- List their features and make a basic set of features
- Make an inventory of existing decentralised social networks
- Only analyse the solutions that meet requirements
- Analyse its features and inner working
First, why do people use Facebook?

Based on the existing literature, we propose a dual-factor model of FB use. According to this model, FB use is primarily motivated by two basic social needs: (1) the need to belong, and (2) the need for self-presentation.

– A. Nadkarni and S. G. Hofmann, Why do people use Facebook?
Facebook is also used

- For bridging (keeping in touch with persons far away)
- People post pictures to create their ideal image
Features

❖ Posting social updates
❖ (re-)sharing these updates
❖ Commenting on updates
❖ Like an update

❖ Favourite an update
❖ Favourite a comment
❖ Sending notifications
❖ Privacy
<table>
<thead>
<tr>
<th>Out of scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>masques</td>
</tr>
<tr>
<td>Maidsafe</td>
</tr>
<tr>
<td>Ethereum</td>
</tr>
<tr>
<td>Trsst</td>
</tr>
<tr>
<td>NXTmemo</td>
</tr>
<tr>
<td>Bitmessage</td>
</tr>
<tr>
<td>Pond</td>
</tr>
<tr>
<td>Kune</td>
</tr>
<tr>
<td>Pixepark</td>
</tr>
<tr>
<td>Avatar</td>
</tr>
<tr>
<td>Tonika</td>
</tr>
<tr>
<td>Phoenix</td>
</tr>
<tr>
<td>Meomni</td>
</tr>
<tr>
<td>Sone</td>
</tr>
<tr>
<td>Secushare</td>
</tr>
<tr>
<td>OpenAutonomy</td>
</tr>
<tr>
<td>Jappix</td>
</tr>
<tr>
<td>Elgg</td>
</tr>
<tr>
<td>Noosefero</td>
</tr>
<tr>
<td>Buddypress</td>
</tr>
<tr>
<td>Tent.io</td>
</tr>
<tr>
<td>duuit</td>
</tr>
<tr>
<td>Higgins</td>
</tr>
<tr>
<td>ODS</td>
</tr>
<tr>
<td>Lorea</td>
</tr>
<tr>
<td>Themineproject</td>
</tr>
<tr>
<td>Kopal</td>
</tr>
<tr>
<td>Helloworld</td>
</tr>
<tr>
<td>Buddycloud</td>
</tr>
<tr>
<td>Libertree</td>
</tr>
</tbody>
</table>
Reasons

❖ Can not be used in a production environment
❖ Not broadly accessible
❖ Abandoned projects
❖ Other philosophy
❖ Missing cross-server message exchange
Implementations

- pump.io
- Friendica
- IndieWebCamp
- Diaspora*
- GNU social
- RedMatrix
- Movim
- rstat.us
Advanced privacy settings

- Offered by RedMatrix and Friendica
  - RedMatrix provides 18 options
- Diaspora*
  - Only has aspects
- GNU social seems buggy
- pump.io not really advanced
Identities

❖ Form of identity
  ❖ All use: username@host.com

❖ Proof of identity
  ❖ Friendica no signature
  ❖ pump.io OAuth signature does not cover body
  ❖ Others use Salmon Magic Envelope, HMAC or own system

❖ Nomadic identity
Encryption

- Only RedMatrix stores encrypted data
- Messages between servers are encrypted with
  - RedMatrix, Diaspora*
  - Friendica (if RINO enabled)
- End-to-end encryption only offered by RedMatrix
Messaging

- Message distribution
- Message consistency
  - All implementations have consistency issues
  - No message queue in: pump.io
- Message relay
  - Not implemented in: pump.io, seems broken with GNU social
Administering, searching, and blocking

- SPAM
  - A real issue with pump.io and GNU social
  - Diaspora, users can be blocked
  - Advanced options to protect yourself available in Friendica and RedMatrix
- Reputation system
  - Only available in RedMatrix
- Directory server
  - Friendica and RedMatrix
Hidden contacts

❖ Not everybody needs to know who you friends are
❖ Possible with Friendica, RedMatrix, and Diaspora*
Public poll

- RedMatrix: zotfeed
- pump.io: firehose
- Friendica and Diaspora*: Feed per user
- GNU social: public feed
Something different
IndieWebCamp

- Movement/community
- Guided by principles, one important one: users own their data
- Data is syndicated to silos
  - POSSE, PESOS, PESETAS
- Red Wind and Known
- IndieAuth
- Webmention
Standardisation
Standardisation

The nice thing about standards is that you have so many to choose from.

—Andrew S. Tanenbaum
Standardisation

- Almost no interoperability, unless one uses plugins
- There are standards but used or implemented slightly different
Protocols

❖ DFRN
❖ Zot2
❖ OStatus (stack)
❖ WebFinger
❖ Salmon
❖ PubSubHubbub
❖ Webmention
❖ Tent
❖ Libertree
❖ DSNP
❖ OpenBook
❖ Activity Streams
❖ Portable Contacts
Conclusion

❖ A variety of reasons why people use social networks
❖ Comment, like, favourite, and post
❖ Looked at GNU social, Diaspora*, Friendica, pump.io, and RedMatrix
❖ RedMatrix is most suited to be provided as an alternative
Recommendations

- Permanent usernames
  - Have two usernames, lookup performed by WebFinger
- Message distribution
  - Let friends share one’s data, use session key
Future work

❖ Deadlock
❖ Security
❖ Benchmark
❖ Stale data and accounts
❖ Proof of concept of suggestions
The End
Questions?

wouter.miltenburg@os3.nl
Credits

[1]: http://www.rand.org/content/dam/rand/pubs/research_memoranda/2006/RM3420.pdf