



Vision

In our vision there is only one social web consisting of social web nodes. A social web that would be open and universal so that anyone can join or create a social web node that in turn can communicate with all other social web nodes without constraints from either corporations or governments very similar to how email or instant messaging work today. We think to everyone's benefit, this would greatly increase freedom, flexibility, the rate of innovation and overall stability.

Current state of affairs

Quite a few share our vision and have developed initiatives to empower the social web: make it possible to reuse a single identity (e.g. OpenID), share your activities (e.g. ActivityStrea.ms), take your contacts along (Portable Contacts) or allow services to do things on your behalf (OAuth). Most of these initiatives have been presented and discussed on the SWxG calls and mailing-list. In our view there however is not a single holistic initiative that brings together and harmonizes all these efforts and fills in the gaps. We strive to create a holistic solution addressing all the issues by building upon these initiatives and other relevant standards.

What XMPP has to offer

We believe that XMPP, the engine best known for its instant messaging capabilities, can serve as a solid foundation for the new social web. There are several similarities that can be leveraged:

- A unique identifier called jabber ID or JID e.g. alice@wonderland.lit in a familiar e-mail like format
- A profile with basic information about you (in the shape of the upcoming vcard xml spec & avatar)
- A list of friends (the so called roster)

- A means to share events of individuals (called Personal Eventing Protocol or PEP)

A tremendous advantage of XMPP is that in its very nature it tackles security issues that are not straight forward to solve in the web world. Since all traffic is routed through the server and the identity of that server can be validated with signatures issued by Certified Authorities, end users do not need to worry about complex things like signatures to establish secure communication. Like in the email world the trust is at the server level: the server is expected to behave according to several rules.

Another advantage is that xmpp has been designed to send all messages in real-time using a very efficient push mechanism. Existing web or polling based mechanisms are often making many unnecessary requests introducing network load and not real-time.

Finally XMPP is already a distributed social web in its own right for the purpose of chatting. With several subtle modifications it could very quickly become a full fledged social web with substantial critical mass.

What XMPP is still lacking

To realise our vision the XMPP foundations would need to be extended and complemented with the following elements:

- Activities: the core elements shared on social networks. We are using the ActivityStrea.ms standard for this purpose. This is the de-facto standard and a 1.0 release is expected soon.
- Fine grained access control: we want to enable both public oriented Twitter like social nodes as well as more private oriented Facebook like social nodes. For this we think it is essential to include fine grained

access control at a protocol level. With fine-grained we mean that end users have precise control on who can see or do what: for activities on a per activity basis, for profiles on a field per field basis. This is not only about viewing rights, you could also delegate editing or updating to third parties e.g. you could allow a location provider to update your location.

- Subscriptions: we allow anyone to subscribe to another's activity, profile or relation updates again with fine grained control e.g. maybe you only want to get photo's or blog posts from someone. This to keep the user in control and avoid information overload. Subscriptions can be asynchronous, can be setup without friending and respect relevant access control rules. You only receive what you are allowed to see.
- Collaboration: this is a different flavour related to fine grained access control. You can allow others to edit or work together on projects or tasks e.g. a shared photo album.
- Relations: you can define the nature of the relationships of your friends e.g. colleague or fiancé. Relations can be confirmed or declined. We add a mechanism for anyone to validate a claimed relationship.
- Offline message store: if a user is offline, normal chat messages are dropped. We need to add an offline message store that stores messages when you are not online. We will effectively put all messages you get in a database so end users can navigate backwards in time.

Activities, profile and relations would have to be extensible to accommodate unforeseen future use cases. All the existing standards we selected are employing extensible xml formats using name spaces.

The role of the web

Does this mean there is no role to play for the web? We believe that the web will remain a very important place to discover, consume and publish social activities as well as

people's profiles. Moreover we think that http based interfacing (e.g. via REST) for now will remain the de facto way of communicating between social web clients and servers. That is why we are including a web client for OneSocialWeb and eventually will include a REST API for client to server communication.

We foresee that in the future you can selectively expose activities and parts of your profile transparently through your browser when you are browsing the web. You could for instance authorize Amazon when you visit their website to read your book reviews and recommend relevant books to you without having to login.

Practice makes perfect

We strongly believe, we have to build a reference implementation to understand how this will all work in practice. This means several distributed servers with different clients (i.e. web, desktop and mobile) to replicate experiences people are used to seeing on existing social network sites and clients. Recently we open sourced an OpenFire server plugin and plan to release the web and mobile clients soon. We aim to evolve our project with the help of the social web and developer communities.

XMPP on the mobile

OneSocialWeb enabled clients can work on mobile phones just like existing social network or chat clients on the market place, either using web based APIs or native XMPP. To offer real-time notification, it is important that the mobile platform supports background threads (otherwise you always need to have the app activated) and that the client uses native XMPP. Since XMPP has really different characteristics, we are implementing an Android client to understand connectivity issues when switching bearers or reconnecting (2G/3G/WIFI/No network), network traffic implications (XMPP by default broadcasts presence changes of your complete roster, but is otherwise highly efficient) and overall impact on battery life.

Authors



Alard Weisscher

Vodafone Group R&D
Maastricht, Netherlands

+31 6 212 96 199

alard.weisscher@vodafone.com



Laurent Eschenauer

Vodafone Group R&D
Maastricht, Netherlands

+31 6 119 54 411

laurent.eschenauer@vodafone.com