White Paper on Mobile Social Network Work Item Investigation

Approved – 16 May 2011

Open Mobile Alliance

OMA-WP-Mobile_Social_Network-20110516-A

Contents

Figures

Tables

1. Scope

The aim of this document is to investigate the Mobile Social Networks topic, such as which features are commonly used in Social Network services available on the Internet, which entities are involved, with the final goal of exploring which role OMA may play and which OMA Enablers may be reused, enhanced or newly created to fill in the gap in the Mobile Social Networks context.

2. References

[OMADICT] "Dictionary for OMA Specifications", Version 2.8, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_8, URL:http://www.openmobilealliance.org/

[ACTIVITYSTREAMS] URL: http://activitystrea.ms/

[ATOMPUB] http://www.ietf.org/rfc/rfc5023.txt

[CMIS] Content Management Interoperability Services (CMIS) Version 1.0

URL: http://docs.oasis-open.org/cmis/CMIS/v1.0/cmis-spec-v1.0.html

[FOAF] The Friend of a Friend (FOAF) project. URL: http://www.foaf-project.org/

[HCARD] hCard 1.0. URL: http://microformats.org/wiki/hcard

[MagicSig] Panzer, J., Laurie, B., and D. Balfanz, "Magic Signatures."

URL: http://salmon-protocol.googlecode.com/svn/trunk/draft-panzer-magicsig-01.html

[OAUTH] OAuth 2. URL: http://oauth.net/2/

[OEXCHANGE] URL: http://www.oexchange.org/

[OMA- Autho4API] "Authorization Framework for Network APIs (1.0)", Open Mobile Alliance™, URL:http://www.openmobilealliance.org/

[OMACAB] "Converged Address Book", Open Mobile Alliance™, OMA-ERP-CAB-V1_0, URL:http://www.openmobilealliance.org/

[OMACAB_APIs] "Converged Address Book APIs", Open Mobile Alliance™, OMA-WID_0215-CAB_APIs-V1_0, URL:http://www.openmobilealliance.org/

[OMACAB1.1] "Converged Address Book v.1.1", Open Mobile Alliance™, OMA-WID_0217-CAB-V1_1-20110301-A, URL:http://www.openmobilealliance.org/

[OMACMI] "Content Management Interface", Open Mobile Alliance™, OMA-ERP-CMI-V1_0, URL:http://www.openmobilealliance.org/

[OMACPM] "Converged IP Messaging", Open Mobile Alliance™, OMA-ERP-CPM-V1_0, URL:http://www.openmobilealliance.org/

[OMACSEA] "Client-Side Enabler API", Open Mobile Alliance™, OMA-RRP-CSEA-V1_0, URL:http://www.openmobilealliance.org/

[OMADCD] "Dynamic Content Delivery", Open Mobile Alliance™, OMA-ERP-DCD-V1_0, URL:http://www.openmobilealliance.org/

[OMADPE] "Device Profiles Evolution", Open Mobile Alliance™, OMA-ERP-DPE-V1_0, URL:http://www.openmobilealliance.org/

[OMAMSF] "Mobile Search Framework", Open Mobile Alliance™, OMA-ERP-MSrchFramework-V1_0, URL:http://www.openmobilealliance.org/

[OMANGSI] "Next Generation Service Interfaces", Open Mobile Alliance™, OMA-ERP-NGSI-V1_0, URL:http://www.openmobilealliance.org/

[OMANGSI-S] "SOAP Binding for NGSI V1.0", Open Mobile Alliance™, OMA-ERP-NGSI_S-V1_0, URL:http://www.openmobilealliance.org/

[OMAParlayREST2] "RESTful bindings for Parlay X Web Services v.2.0", Open Mobile Alliance™, OMA-ERP-ParlayREST-V2_0, URL:http://www.openmobilealliance.org/

[OMAPOC] "Push to talk over Cellular", Open Mobile Alliance™, OMA-ERP-PoC-V2_1, URL:http://www.openmobilealliance.org/

[OMA-Privacy] "Privacy Requirements for Mobile Services", Open Mobile Alliance™, OMA-RRP-Privacy-V1_0, URL:http://www.openmobilealliance.org/

[OMAPSA] "Parlay Service Access", Open Mobile Alliance™, OMA-RRP-PSA-V1_0, URL:http://www.openmobilealliance.org/

[OMAPUSH] "Push Over The Air", Open Mobile Alliance™, OMA-TS-PushOTA-V2_3,

	URL:http://www.openmobilealliance.org/
[OMARC_APIs]	"APIs for Rich Communications", Open Mobile Alliance™, OMA-WID_0213-RCS_APIs-V1_0, URL:http://www.openmobilealliance.org/
[OMASCAB]	"CAB architecture simplification with new market requirements", Open Mobile Alliance™, OMA-WID_0214-SimplifiedCAB-V1_0-20101221-A, URL:http://www.openmobilealliance.org/
[OMA-SIMPLE-IM]	"Instant Messaging", Open Mobile Alliance™, OMA-ERP-SIMPLE_IM-V1_0, URL: http://www.openmobilealliance.org/
[OMASUPM]	"Service User Profile Management", Open Mobile Alliance™, OMA-ERP-Service_User_Profile_Management-V1_0, URL: http://www.openmobilealliance.org/
[OMAWRAPI]	"Web Runtime API", Open Mobile Alliance™, OMA-WID_0210-WRAPI-V1_0, URL:http://www.openmobilealliance.org/
[OMAXDM]	"XML Document Management", Open Mobile Alliance™, OMA-ERP-XDM-V2_1, URL:http://www.openmobilealliance.org/
[ONESOCIALWEB]	URL: http://onesocialweb.org
[OPENID]	OpenID Foundation website. URL: http://openid.net/
[OPENID-AX]	OpenID Attribute Exchange 1.0 – Final. URL: http://openid.net/specs/openid-attribute-exchange-1_0.html
[OPENLIKE]	http://openlike.org/
[OPENSOCIAL]	OpenSocial. URL: http://www.opensocial.org/
[OSTATUS]	URL: http://ostatus.org/
[POCO]	Portable Contacts. URL: http://portablecontacts.net/
[PuSH]	URL: http://pubsubhubbub.googlecode.com/svn/trunk/pubsubhubbub-core-0.3.html
[MSRP]	"The Message Session Relay Protocol (MSRP)", B. Campbell et al, September 2007, URL: http://www.ietf.org/rfc/rfc4975.txt
[SALMON]	Salmon Protocol, http://www.salmon-protocol.org/
[SIMPLE]	SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE), http://datatracker.ietf.org/wg/simple/
[SIOC]	URL: http://sioc-project.org/
[SIP]	"SIP: Session Initiation Protocol", J. Rosenberg et al, June 2002, URL: http://www.ietf.org/rfc/rfc3261.txt
[SWAT0]	http://www.w3.org/2005/Incubator/federatedsocialweb/wiki/SWAT0
[VCARD]	"vCard MIME Directory Profile", http://www.ietf.org/rfc/rfc2426.txt
[W3C-FSW]	W3C Federated Social Web Incubator Group URL: http://www.w3.org/2005/Incubator/federatedsocialweb/
[WEBFINGER]	URL: http://webfinger.org
[WebFingerProtocol]	"WebFinger Protocol", http://code.google.com/p/webfinger/wiki/WebFingerProtocol
[WEBID]	WebID 1.0. URL: http://payswarm.com/webid/
[XFN]	XHtml Friends Network. URL: http://gmpg.org/xfn/
[XMPP]	XMPP Standards Foundation. URL: http://xmpp.org/
[XRD]	Extensible Resource Descriptor (XRD) Version 1.0. URL: http://docs.oasis-open.org/xri/xrd/v1.0/xrd-1.0.html

3. Terminology and Conventions

3.1 Conventions

This is an informative document, which is not intended to provide testable requirements to implementations.

3.2 Definitions

Aggregation	This is defined as the ability to connect to multiple Social networks (OMA Compliant SN and not-OMA Compliant SN) using the SN service specific credentials to access information related to friends (e.g. status, contact details, activities) via the exposed interfaces. Aggregation in SNS is related to only in-bound interaction.
Cross-posting	This is defined as the ability to connect to multiple Social networks (OMA Compliant SN and not-OMA Compliant SN) using the SN service specific credentials to share information (e.g. status, media) via the exposed interfaces. Cross-posting in SNS is related to only outbound interaction.
External Social Network	A social network that is not-OMA Compliant and made available through either proprietary or non-proprietary mechanism and/or interfaces
Federated Social Networks	This is defined by a set of social networks (OMA Compliant SN and not-OMA Compliant SN) that share some level of trust or set of rules, data formats and protocols, while, each social network retains its own administrative control and structure. The trust or set of rules binding the social networks govern the information that can be shared/searched/exchanged among users who are part of the respective social network.
Inbound/outbound interactions	Inbound interactions relate to the concept of aggregation of activities and media from external networks, making thus possible to allow users that own accounts on multiple external SNs to access aggregated information about their friends (e.g. contact information, activities) on these networks. Outbound interactions, on the opposite, relate to the ability of cross-posting activities and/or media to multiple external SNs. In this way, users could potentially share their activities over all their external SNs at once.
OMA-Compliant Social Network	This is defined as a social network that conforms to the standards defined by OMA for social networking, essentially to ensure seamless interoperability and satisfaction of various actors in a social network

3.3 Abbreviations

API	Application Programming Interface
JSON	JavaScript Object Notation
MSISDN	Mobile Subscriber ISDN Number
MSN	Mobile Social Network
OMA	Open Mobile Alliance
RCS	Rich Communication Suite
RDF	Resource Description Framework
RSS	Really Simple Syndication
SDO	Standards Development Organization
SMTP	Simple Mail Transfer Protocol
SN	Social Network
SNS	Social Network Service
SSL	Secure Sockets Layer
TLS	Transport Layer Security
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol

4. Introduction

Thanks to Web 2.0 technologies, the World Wide Web has turned into a social space, moving from document links to people links. Individuals and organizations are now linked and leverage on user-generated content, communities, networking and social interaction.

Whilst the Web is becoming increasingly social, social networking itself is heavily fragmented due to the multitude of disparate services that are popular among users.

The existing social network sites available on the Internet are all based on centralised isolated systems, and usually each of them is run by a single company. Users on one Social Network (SN) cannot (easily) interact with users on another Social Network and people will often have to sign up for an account on multiple SNs to keep in touch with different groups of friends.

Despite the fact that there are hundreds of other social network sites on the Web, almost every single SN works standalone, ignoring other siblings.

Unfortunately, or fortunately, there is no single social graph (or even multiple which interoperate) that is comprehensive and decentralized. Rather, several disperse social graphs exist; many of them implement a “walled garden” approach and/or are operated by small and unproven companies. This inconvenience ultimately results in a few very large networks with an inordinate amount of control over peoples’ most personal data and a lack of choice and privacy for users.

In this regard, such walled gardens are data silos where user data can easily be inserted, but only accessed and manipulated via proprietary interfaces for humans and machines, therefore preventing the user from moving easily from one social network platform provider to another one by creating a “wall” around their social data that cannot be shared across networks.

This current situation is analogous to the early days of hypertext before the World Wide Web, where various systems stored hypertext in proprietary and incompatible formats without the ability to use globally unique links and access hypertext data across systems, a situation solved by the creation of URIs and HTML. The same happened, for example, at early proprietary email systems which were "federated" by the open Internet SMTP. Internet email and the World Wide Web have created immense wealth and social well-being for the people who used them – vastly more than the monolithic, isolated networks that preceded them.

The goal of this Whitepaper is to investigate how this road may apply to the Mobile Social Networks world.

In this vision, this Whitepaper aims mainly at investigating on enabling client-server communication in a standard way between mobile devices and SN servers, as well as between servers from different service providers. Connections with external SNs are expected (through gateways implementing proprietary interfaces) but the definition of which external Social Network will be interconnected is out-of-scope of this activity.

In addition, the idea of APIs, both device APIs and network APIs will be evaluated within this activity.

Finally, this Whitepaper will study the peculiarities of the Mobile Social Networks, both at protocol and application level that may lead into reusing existing OMA enablers to overcome limitations or provide further functionalities or information.

5. Background

5.1 Industry

Social Network Service (SNS) has a reach of 66% of Internet, and is a rising phenomenon in Mobile Internet. The mobile phone is a ubiquitous personal device and is hence becoming the best medium for accessing social network communities as updates can be sent and received on the move. When accessing the Internet sitting in front of a PC, a user is accessing information about some past events and not in real time. In contrast, using his mobile phone a user can take part of a live event, capture it, provide comments and share all this with other users. Mobile phones therefore provide a richer social interaction. Mobile phone users are increased tremendously in the last five years. After reaching around 4.6 billion mobile cellular subscriptions by the end of 2009, at the beginning of 2010 ITU has foreseen the number of mobile cellular subscriptions globally to exceed five billion at the end of 2010. With the current growth rates, the number of people accessing web on the move — via laptops and smart mobile devices – is likely to exceed the number of people accessing web from desktop computers within the next five years. eMarketer has estimated that the mobile social network users would exceed 800 million by 2012. Visiongain believes that revenue from mobile social networks and user generated contents will grow to around $60 billion in 2012.

MOTOBLUR™ [1] and Vodafone 360[2] were launched at almost the same time in Q3 2009. From the end of 2009 to 2010, a lot of mobile devices with operating systems supporting social network functions have been launched into the market: MOTOBLUR™, Windows Mobile 7[3] and Android 2.2[4]. It is expected that more and more mobile social network services will be available in the market in the next a couple of years.

Current mobile social network services, e.g. MOTOBLUR™ and Vodafone 360, are primarily based on the aggregation of popular Internet social network sites such as Facebook[5], MySpace[6] and Twitter[7], plus other RCS[8]-like features and infotainment services. More and more mobile social network services tend to provide the aggregation functions.

Mobile operators have been proposing their own social network service to their users:

- Vodafone 360 uses the mobile phone as the starting point, enabling users to exchange messages and content with anyone in their mobile phone address book.
- Orange Pikeo[9] is a social network service that allows user to store, organize and share the favourite photos online.
- O2 Bluebook[10] enables a user to back up the names and numbers in the phone book, store texts and photos online.

From the end user perspective, the biggest challenge is interoperability, for example:

- Can a Moto CLIQ user access Vodafone 360?

---

[1] http://www.motorola.com/

[2] http://vodafone360.com/

[3] http://www.microsoft.com/windowsphone

[4] http://www.android.com/

[5] www.facebook.com

[6] www.myspace.com

[7] twitter.com/

[8] http://www.gsmworld.com/our-work/mobile_lifestyle/rcs/gsma_rcs_project.htm

[9] http://www.pikeo.com/

[10] https://www.o2.co.uk/bluebook_login

- Can a Moto CLIQ user share his content with his friend that uses H1[11] / Vodafone 360?

From the operator perspective, it should be considered how to provide added value to the users communicating with each other across social network services.

5.2 Social Network Features and Initiatives

Federated systems have proven over time to have several advantages: they are incredibly robust, they encourage technical innovation, and they are more secure.

At present, any individual, company, or organization can own a Web site or email server and be part of the World Wide Web. Any such entity should hence be able to bring its own identity to any SN, or run its own.

The importance of distributed social networks can be outlined in the advantage that the user gains: freedom to choose among any compatible service providers without losing access to their contacts, and privacy control.

However, a critical problem in realizing this vision of a distributed and secure SN is the fact that any "distributed" social network will become yet another walled garden unless it is based on open standards. Via open standards, multiple social network platforms ranging from large vendors to simple personal websites will be able to interoperate with each other.

It is reasonable to assume that a federated social web will at least do what monolithic social network applications do today.

5.2.1 Social Network Features

In this section a list of functionalities are briefly described; these functionalities are commonly used by Social Network services currently available and are to be addressed in order to define a common approach to (distributed) social network.

5.2.1.1 Identity

Identity is the unique identifier of an entity or resource. An important part of identity is addressability – having a machine readable address that computers and people can use to find a resource uniquely. Users can have more than one identity in a social network, e.g. with respect to email, where many people have work and home addresses, but they are not typically linked in any way, despite some proprietary initiatives.

Because identity is important for remote login and security, it is probably the most crucial part of this system.

There are competing schools of thought on identity and addressing when coming to web identity: OpenID [OPENID] borrows the URL format from the Web (HTTP URI), whilst WebFinger [WEBFINGER] on the other hand borrows the address format from email.

Whatever address format is considered, it is likely to be hierarchical, including an organizational part that belongs to the reference network, and an individual part that is unique within that network.

5.2.1.2 Profile

Profile data can contain any information about an entity: name, avatar, postal address, phone number, favourite colour, religion, political orientation, past jobs, education history, etc. Almost all SNs provide a single profile Web page for individuals with all their profile data; some SNs restrict access to sensitive data to a subset of viewers.

A number of standards exist currently for profile and relationship information on the Web. One distinction among them is in which data format (plaintext, XML, RDFa) the profile is structured and whether those data formats are easily extensible or not.

Even more importantly, there are differences in how, given a digital identity, any particular application can discover and access profile data and other capabilities that the digital identity may implement.

[11] http://www.samsung.com/uk/mobile/pressView.do?ptype=press&page=1&news_seq=22647

The following is a non-exhaustive list of standard that enable profile discovery and/or representation:

- XRD [XRD] is a XML file format for discovering what capabilities a particular profile provider may have.
- hCard [HCARD] is a microformat for publishing the contact details (which might be no more than the name) of people, companies, organizations, and places, in(X)HTML, Atom, RSS, or arbitrary XML.
- FOAF [FOAF] provides an extensible approach to modelling information about people, groups, organizations and associated entities, and is designed to be used alongside other descriptive RDF (Resource Description Framework) vocabularies.
- PortableContacts [POCO] is derived from vCard [VCARD], and is serialized as XML or, more commonly, JSON. It contains a vast amount of profile attributes. More than a profile standard, the PortableContacts profile scheme is designed to give users a secure way to permit applications to access their contacts.
- OpenSocial [OPENSOCIAL] defines a common API for social applications across multiple websites. With standard JavaScript and HTML constructs, developers can create applications that access friends and update feeds of a SN.
- OpenID Attribute Exchange extension [OPENID-AX] defines a set of messages to exchange (access, insert) user profile information with an OpenID provider
- OMA CAB [OMACAB] provides an extensible list of elements related to people, groups and organizations.

Current profile-related standards provide a way to exchange data between networks, but keeping them synchronized and up-to-date is still to be addressed.

5.2.1.3 Privacy

Users need be able to define who can access their data and under what conditions.

- OAuth [OAUTH] is a popular standard for granting data authorization to third parties, allowing users to grant access to private resources after authenticating themselves via their online identity.
- WebId [WEBID] (a.k.a FOAF+SSL) uses TLS and client-side certificates for identification and authentication.

5.2.1.4 Relationships

Declaring relationships to friends and colleagues is the lifeblood of any SN.

Some SNs allow the users to define the nature of the relationship. Some SNs require that both related people approve the relationship (“friends”); others support a one-way approach (“fan” or “follow”). In either case, some or all people on the network can navigate a user’s list of friends.

Reading lists of friends or colleagues on the Web is easy. Restricting access to those lists has similar problems to restricting access to profile information as mentioned above.

Machine-readable relationship format is one of the best-developed features of federation, with FOAF and XFN [XFN] (which embeds its own social contact relationships directly into HTML links using the “rel” attribute) leading the way.

5.2.1.5 Content sharing

The Social Web has evolved beyond connections between people. Sharing photos, audio, links, video and rich text with friends is an important part of SNs. Some networks specialize in only one medium like photos or video; others cover the entire spectrum.

Sharing office files like word-processing documents and spreadsheets is important for corporate networks, too.

Navigating or consuming media uploaded by another user is usually easily supported by syndication standards, but present the same issues such as setting permissions based on relationships (e.g. “only for family and friends”).

On the other hand, uploading media remotely isn't well-specified and is typically pushed using proprietary interfaces or as attachments for syndicated activities (Atom and RSS). However, OExchange [OEXCHANGE] is an open specification for sharing rich contents over the Web using URIs between social sites, which may improve this process. It defines a protocol that supports the offering of URIs to other services in a standardized way (with authentication capabilities) and further allows sites to advertise their ability to receive data using XRD and WebFinger. This proposal has gained support from many players in the Social Web.

5.2.1.6 Activities

The most distinguishing feature of the Social Web over the previous hypertext Web is the increasing focus on sharing information in real-time. As opposed to pulling information on an as-needed basis, users desire to have information that may be of interest, pushed to them immediately. The social interactions of user and resources, including other users, represent the activities of the user.

Each activity, such as changing status, making new connections, creating a blog post, uploading a picture, or attending events can be considered an update in an activity. The total of all activities of a user is named “the stream” of that user.

This is where the most effort is focusing today in the federated social web, both at data representation level and at protocol level.

Generating and sharing feeds, with RSS and Atom, is a forte of the open web; its latest development is ActivityStreams [ACTIVITYSTREAMS], a popular set of Atom extensions, which encodes machine readable information about social activities like posting a video or joining a group.

AtomPub [ATOMPUB] is another alternative based on HTTP and is used for publishing and posting on the Web. The AtomPub together with the Atom Syndication Format (ASF) provides interaction with content, especially in blogs and RSS.

In addition, PubSubHubbub [PuSH] is a real-time push protocol that avoids the typical polling of RSS feeds and optimizes the exchange and update of information. In the OMA context, this may relate to OMA PUSH [OMAPUSH].

5.2.1.7 Follow-up actions

In several existing SNs, users can take actions on activities performed by other users, such as comment on media, “favourite” or “like” it, or re-share the media to their own social network.

OpenLike [OPENLIKE] is an open protocol to allow sharing what people like in a simple and common way between web applications (e.g. “like” button). This proposal however doesn’t seem to have seen wide-scale deployment.

In addition, the Salmon protocol [SALMON] is another open standard way to update and integrate back further comments attached to an update to their original source.

5.2.1.8 Private messages

Over the past years the volume of direct messages on social network services reached and surpassed the number of emails. Private, direct messages are readable only by sender and recipient and can often contain embedded media or links. Some services further provide an empty message feature, called a “poke” or “nudge”.

This is a part of the federated social web that hasn't yet received much attention. Some work is currently ongoing to implement remote private messaging using PubSubHubbub-enabled streams.

5.2.1.9 Groups

Grouping people together is an important part of most SNs. Users can join or leave a group (sometimes requiring the group owners’ approval), send private messages to the group (which are distributed to all members), and upload images, video, or other media that relate to that group. Group administrators can also announce events or provide a profile for the group.

Most of the issues for individuals apply for groups, and the standards used for communications and relationships between individuals will largely work for groups. For example, joining a group can be seen as establishing a relationship with it.

5.2.1.10 Search

Users on SNs can usually search for other people by name, location, interests, or profile information like age and gender. They can often restrict these searches to their own friends or friends-of-friends. Some SNs further provide the ability to search for media or groups by keyword, with an emphasis or exclusive filter on your friends.

It is unclear so far which role search will play in the federated social web: public media and groups may be accessible to Web Search Engines, but searching media that has privacy constraints is not been addressed so far.

5.2.1.11 Client Web API

Some SNs provide a Web-based API that third-party developers can use to create desktop and mobile web applications that access the network. Typical functionalities include establishing relationships, browsing an activity feed, and uploading or viewing media.

Defining a standard Client Web API that developers could use to access any social web site would significantly help exploiting federated social web services. OpenSocial is an initiative that has started to address this area.

5.2.1.12 Data Portability

Data portability is one of the intrinsic features required for a federated social web that is usually not implemented on existing SNs. Traditionally, users need to re-register their personal information and contacts on every new SN. Through data portability, users would be able to seamlessly import their profile, history, media and connections to new social applications and platforms. Moreover, authorization policies and privacy rules should be seamlessly portable across SNs.

Nowadays, social network sites encourage users to put their data into their own proprietary platform and sometimes tend to restrict the portability of the user's own data to another site or even their home computer.

A user should be able to move easily from a social network to a different one, bringing with him/her all his/her histories and data from the previous social network, such as identity, profile, shared contents, social graph. The moving from a social network to a different one shall be managed in an automatic way (with minimum human actions) and, most importantly, keeping authorization and privacy policies already specified by the user in the previous social network.

OpenID addresses identity and profile portability, whilst Portable Contacts or FOAF are focused on contact information portability. In addition, ActivityStreams or SIOC [SIOC] can help porting user activities from one SN to another.

5.2.2 Main Social Network Federation Initiatives

In a federated Social Web vision, the core architecture of an activity stream presumes the ability to send content (status updates, messages, and other content) in near real-time. Whilst many initiatives exist at the moment to create and operate their own SN, two main architectures and related initiatives raise above the others. While the underlying protocol is different, the core functionalities and part of the data representations are similar.

The first proposal, the OneSocialWeb [ONESOCIALWEB] is based on XMPP [XMPP], where the XMPP messaging framework natively provides an XML “envelope” for data to be sent in real-time with updates. XMPP in its simplest form can be regarded as an asynchronous protocol for exchanging XML fragments, which defines its own methodology for identity authentication and extensibility.

As one of the main concerns of the Social Web in general is to provide status updates and messages in near real-time, XMPP is a natural fit for federated social networks, natively supporting user profile and relationships as well. However, XMPP is not built on HTTP transport and as such provides a whole set of issues when dealing with interconnecting sites remotely.

OneSocialWeb further leverages ActivityStreams format for activity sharing as well as XFN-inspired XMPP extensions for relationship definition.

The second proposal is named OStatus [OSTATUS], an alternative architecture rapidly emerging. It relies on HTTP as baseline transport, further overriding its traditional “pull” architecture with a “push” architecture based on PubSubHubbub.

In general, OStatus can be seen as a “meta-specification”, umbrella or design-pattern for relating and sending status updates to people in a federated Social Web. It collects together a number of previously mentioned specifications (PubSubHubbub, ActivityStreams, Salmon, Portable Contacts, and WebFinger) to enable distributed social networking.

Overall, this specification provides a service to the Social Web community by providing an HTTP-based meta-architecture that defines the baseline functionality needed in a distributed social application based on activities.

In 2010 a number of projects have started to build federated Social Web platforms, which allow users to run their own SN, keeping control of their own data while still interacting with the rest of the Social Web. Several initiatives (e.g. Identi.ca[12], Status.net[13], OpenMicroBlogger[14], Dijit![15], CouchAppSpora[16], Project Danube[17], Project Nori[18]) already claim compliance with OStatus specifications. Moreover, other players (Google Buzz[19], Wordpress[20], Drupal[21], LiveJournal[22], Tumblr[23]) have already implemented some of the protocols involved in the OStatus suite (e.g. Pubsubhubbub, ActivityStreams) and are planning to fully support it.

Finally, it has to be remarked that W3C has recently set up an incubator working group named “Federated Social Web Incubator Group” [W3C-FSW] whose mission is “to provide a set of community-driven specifications and a test-case suite for a federated social web”.

5.3 Value Chain

In the ecosystem of mobile social networks, the interoperability will effectively connect all dots together and benefit all stakeholders in the value chain. The aspects of the Mobile Social Network (MSN) value chain are provided in Figure 1.

[12] http://identi.ca/

[13] http://status.net/

[14] http://openmicroblogger.org/

[15] http://www.dijit.com/

[16] https://github.com/maxogden/couchappspora

[17] http://projectdanube.org/

[18] http://www.projectnori.org/

[19] http://www.google.com/buzz

[20] http://wordpress.org/

[21] http://drupal.org/

[22] http://www.livejournal.com/

[23] http://tumblr.com/

Figure 1: Mobile Social Network Value Chain

Subscribers / end users will have more options and freedom to use the mobile social network services which they feel more valuable without having to switch operators or mobile phones. End users and their friends will be able to interact with each other more conveniently across mobile social networks.

Operators (MSN provider) will have more customer loyalty and will be able to retain customers even if subscribers use a different social network service. Furthermore, operators will also be able to increase their customer base and improve ARPU by offering unique, differentiated value of their mobile social network service to subscribers.

Manufacturers will be able to provide mobile phones that can connect to all mobile social network services irrespective of the operators. Thus, the manufacturers will benefit from customer loyalty, lowering the development cost, shortening the time to market, and increasing the market share by outreaching the broader market.

Authorized Service Providers and Application Providers will be able to access end users' personal data (such as identities, email address) from the mobile social network, thus enabling them to provide richer services and applications to attract the user and thereby generate revenue.

Developers will also benefit this value chain by providing richer applications or widgets according to the mobile social network APIs. It will be for example possible to arrange an application store combined with mobile social network for expanding the usage of the applications and widgets.

6. Social Network Service Scenarios

6.1 Gateway Scenario

6.1.1 Short Description and Flows

This scenario aims at enabling users to interconnect with external SNs on which they already have an account (e.g. Facebook, Twitter, others) using the proprietary interfaces of such networks. This scenario introduces the concept of direction in interacting between an OMA-compliant SN and one or more external SNs.

In particular, inbound interactions relate to the concept of *aggregation* of activities and media from external networks, making thus possible to allow users that own accounts on multiple external SNs to access aggregated information about their friends (e.g. contact information, activities) on these networks. Outbound interactions, on the opposite, relate to the ability of *cross-posting* activities and/or media to multiple external SNs. In this way, users could potentially share their activities over all their external SNs at once.

Step 1: User A updates status and gets notification based on comments from User B to the status update:

1. user A associates her identity on external SNs (SN_i with i=1,2, …) with its identity on an OMA-compliant SN using the proprietary procedures required by the external SNs
2. At a later stage, user A wants to update her status
3. user A selects to which external SN her status update will be posted and posts it (at least she selects SN 2) (*cross-posting*)
4. user B2 is a friend of user A2 (other identity of user A) on external SN 2, and sees user's A2 new status
5. user B2 comments the new status of user A2 on external SN 2 (out-of-scope)
6. comment from user B2 gets notified to user A on her OMA-compliant SN (*aggregation*)

Step 2: User B updates status and User A gets notification based the update:

7. user B2 updates her status on external SN 2, which is seen by user A2
8. user A gets notified of user B2 status update on her OMA-compliant SN (*aggregation*)

This scenario aims at defining the requirements needed to enable gateway functionalities in a generic way, not addressing the peculiarities of the single external SN.

Furthermore, the concept of activity in this scenario is very generic. Whilst the scenario itself focuses on a status update, media upload and check-in activities are intended as additional valid examples.

6.1.2 High-level Requirements

Hence in general this scenario introduces the following high-level requirements:

- Ability to update status or perform check-in at the same time on multiple external SNs through an OMA-compliant SN.
- Ability to select external SNs for outbound interactions from an OMA-compliant SN. This function may be performed contextually with the activity post/update, or a previously set as a preference. In this latter case, the scenario would require the ability to manage preferences for interactions with external SNs.
- Ability to configure external SNs accounts through an OMA-compliant SN. This function may be an implementation issue, unless the functionality is intended to be implemented on the mobile device.
- (optional) Ability to notify users in an OMA-compliant SN of activities performed on external SNs (e.g. receiving a comment, reading a friend's update)

Note that this scenario may be further enhanced with privacy-related features, allowing users to define the level of privacy of their activity, thus replicating this privacy level on an external SN "at best".

6.1.3 Consumer benefits

Mobile operators can provide the social network gateway function in a standalone way or in addition to their own mobile social network. As such it can attract more users and provide competitive services. Users would be attracted by a social network which can provide the gateway function for communication with friends in multiple SNs.

6.2 Federated Scenario

6.2.1 Short Description and Flows

The federated Social Web community has defined a simple social federation scenario named "Social Web Acid Test - Level 0" [SWAT0] as an integration use case for the federated social web.

This scenario aims at enabling users to interact in a standard way across social networks run by service providers compliant with a set of interoperable specifications.

The generalized flavour of this scenario is copied here for convenience, where "different services" has to be understood as "different services on different Social Networks":

"

1. *user A takes a photo of B from their phone and posts it*
2. *user A explicitly tags the photo with B*
3. *B gets notified that they are in a photo*
4. *C who follows A gets the photo*
5. *C makes a comment on the photo*
6. *A and B get notified of the comment*

*Where users are on at least 2 (ideally 3) different services […], and the users only need to have *one* account, on the specific service of their choice (requiring the users to have an account on each service that is interacting misses the point of Federation)"*

6.2.2 High-level Requirements

This scenario introduces the following high-level requirements:

- Ability for the user to have only *one* account in one of the Federated SNs and be able to interact with users on the other Federated SNs
- Ability to post/upload media (pictures, audio, video)
- Ability to attach tags to media, either contextually to the upload process, or at a later stage. Such tags may represent plain keywords, users or other entities to be defined.
- Ability to interconnect Federated SNs (servers)
- Ability to notify users of an activity that they are related to (e.g. being tagged in a picture, receiving a comment) , including users of other Federated SNs
- Ability to "follow" other users, including users of other Federated SNs.
- Ability to access other users' activities and media, including users of other Federated SNs.

- Ability to take action on (e.g. comment) a user's activity/media, including users of other Federated SNs.

6.2.3 Consumer benefits

Social Network Federation enables users to only have *one* account for SN, whilst still giving them the ability to interact with users on different Federated SNs. It is thus possible to enable a Federated SN as the anchor of mobile operator's service and other services.

The mobile operator can thus operate its own SN thus attracting its own subscribers to an interoperable SN service that allow them to communicate and interact freely with subscribers of other mobile operators.

6.3 Multiple devices support

6.3.1 Short Description and Flows

This scenario aims at enabling user using different mobile phones or devices to access his mobile social network services. User would like to access the mobile social network from different devices, such as using PC while at home or using a mobile phone while on the road. A user may also use multiple mobile phones to connect the social network sites. The mobile social network services need to be provided as other mobile services which are independent of the devices. Some of the operations that users can carry out in multiple devices scenario are:

1. user A accesses the mobile social network service by PC.
2. user A writes a new blog post.
3. user B accesses the mobile social network service through his mobile phone.
4. user B follows user A and get the notification to the mobile phone that user A has a new blog post.
5. user A accesses the mobile social network service using his mobile phone.
6. user B switches device, e.g. accesses the mobile social network service using his PC.
7. user A updates his status through his mobile phone.
8. user B get the notification to the PC that user A has updated his status.

6.3.2 High-level Requirements

In general this scenario introduces the following high-level requirements:

- Ability to update status or perform check-in through different access and devices.
- Ability to execute basic social network service through different access and devices.

6.3.3 Consumer benefits

The mobile operator would set mobile social network as one major entry in the mobile services if it is device independent.

The subscribers / end users will have more options and freedom to switch mobile phones and enjoy the mobile social network with any device.

The manufacturers will be able to make the mobile phone that can connect to all mobile social network services of all operators. Thus the manufacturers will benefit from lowering the development cost, shortening the time to market, and increasing the market share by outreaching to broader markets.

6.4 Multiple device applications support

6.4.1 Short Description and Flows

This scenario aims at enabling multiple applications running in a device to benefit from the interworking with SNs in an optimized fashion. In fact there may be multiple applications in a device that require exchanging information with one or several SNs. Instead of managing this interworking with the SNs on their own, the applications may rely on a specialized entity in the device, i.e. the MSN Client, which relieves the applications from this task by centralizing the communication with the user's OMA Compliant Mobile Social Network, which in turn interworks with other OMA or non-OMA Compliant Mobile Social Networks. The centralization of this interworking through the MSN client results also in an optimization of the use of air data traffic.

Device applications may have different needs in terms of interworking with SNs, for instance:

- Application 1: Social Network Aggregation Client, that allows a real-time interaction with multiple SNs (A, B and C):
    - Receive push notification of updates from SNs A, B and C
    - Update status and post media to SNs A, B and C
- Application 2: Multimedia Gallery application that stores all pictures from SNs A and C in which the user is tagged
    - Receive push notifications about the user having been tagged in pictures from SNs A and C.
- Application 3, Address Book application, that enriches the contacts with information from SN B
    - On demand access to contact information from SN B

As a first step the different applications have to register to the MSN Client their needs in terms of interaction with the SNs. The user needs to grant permission to each application to access the requested resources and/or perform the requested operations.

Provided that the user grants access, the applications are able from that point of time to interwork with the SNs as requested, that is, the applications are provided with the information from the SNs that match the filters indicated by the application at the registration phase and are also able to post information towards the selected SNs.

A special case occurs when an application posts towards the SNs a piece of information that another application running in the same device is intended to receive. In this case the MSN Client may cache this information to make it available to any other interested applications on the same device, in addition to posting it to the network. In this case there is no need to get this piece of information delivered back from the network to the MSN Client, as it was generated at the device and the MSN Client has already cached it locally.

6.4.2 High-level Requirements

In general this scenario introduces the following high-level requirements:

- Ability to centralize and optimize the interworking of multiple applications running in a device with the SNs through a specialized element residing in the device, i.e. the MSN Client, which centralizes the bidirectional communication with the user's OMA Compliant Mobile Social Network (which in turn may provide interworking with other OMA or non-OMA Compliant Mobile Social Networks).

- Ability for the different device applications requiring interworking with the SNs to register their needs (i.e. resources, operations, time constraints, etc) to the MSN Client, and subject to user permission, ability to perform the required interworking from that point of time through the MSN Client.

- Ability for the MSN Client to cache locally in the device information uploaded to the OMA Compliant Mobile Social Network so that a local device application can receive a SN update generated at another application running in the same device locally, without the need to have it sent back from the network to the device; access to the cache is subject to the policies relevant for the specific applications.

- Ability for the MSN Client to cache locally in the device information downloaded from the OMA Compliant Mobile Social Network so that a local device applications can receive a SN update already downloaded by another application running in the same device, without the need to have it downloaded again from the network to the device; access to the cache is subject to the policies relevant for the specific applications.

6.4.3 Consumer benefits

Application developers will be able to develop applications that require interaction with SNs more easily, lowering the development cost, shortening the time to market and thus increasing the application portfolio.

The end users will have a wider offer of applications to access the SNs and will get other type of applications enriched with information from the SNs.

Air data traffic will be minimized as every piece of information will be exchanged with the SNs just once even if there are multiple applications on a device that require exchanging that same piece of information with the SNs.

7. Architectural Aspects

7.1 Reference Model

For supporting the scenarios listed in Section 6, the following figure depicts the Functional Components and Interfaces to enable the MSN functionality.

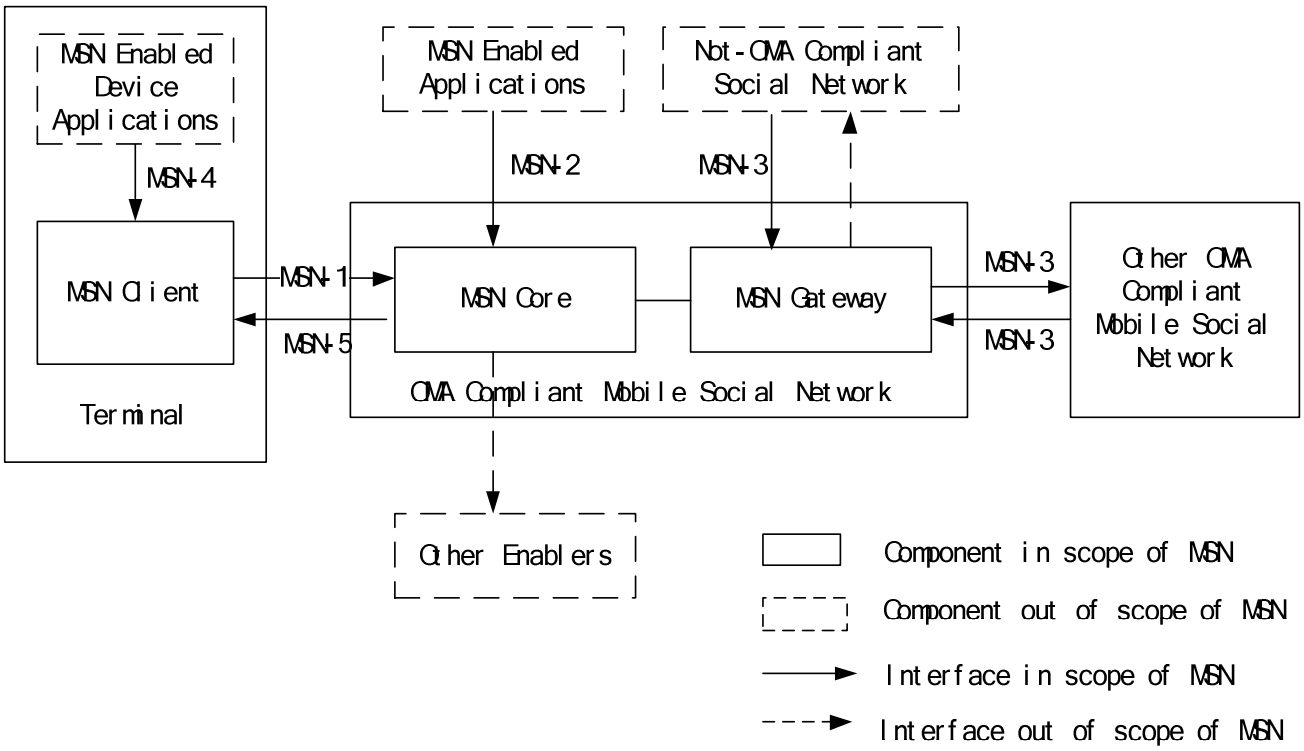

Figure 2: Mobile Social Network reference model

The components identified at this stage are:

- MSN Core is responsible for providing mobile social network service to the end user. As user data centre, it provides user profile management, social relationship management, user generated data and content management. At the same time it could open the user data to the third party based on user permission. It works as an aggregation to provide unified identity management, combined information and contents. The MSN Core can interact with other existing enablers, exchange information with social network services and provide unified user data/application management to the end user.
- MSN Gateway is responsible for interacting with other social networks (OMA Compliant and not-OMA Compliant). It supports inbound interactions to aggregate activities and media from external networks. Support outbound interactions for the ability of cross-posting activities and/or media to multiple external SNs.
- MSN Client has the responsibility for interacting with MSN Server to exchange social network service information. It enables the user to access aggregated information about their friends (e.g. contact information, activities) through the terminal side (e.g. through MSN Enabled Device Applications), set his account for Federated SNs and interact with users on other Federated SNs.

Note: the MSN Gateway may have additional functionalities for translating with not-OMA Compliant SNs; how this will be addressed and represented in the architecture diagram is for further study.

The interfaces identified at this stage are:

- MSN-1 is an interface exposed by MSN Core. MSN Client could use this interface to interact with MSN Core for unified identity management, user authentication and authorization, data exchanging, user content management and so on.
- MSN-2 is an interface exposed by MSN Core which is responsible for providing Mobile Social Network functions to the network-side MSN Enabled Applications; as such, MSN-2 represents a network API.
- MSN-3 is an interface exposed by MSN Gateway for realizing the interaction between OMA Compliant Mobile Social Networks. It enables the data, activity, feeds and media exchange between two different mobile social networks.
- MSN-4 is an interface exposed by the MSN Client to MSN Enabled Device Applications running on the device, providing them with Mobile Social Network functions according to the specifics needs indicated by each MSN Enabled Device Application (e.g. updating status at a social network, receiving updates from a social network); as such, MSN-4 represents a device API.
- MSN-5 is an interface exposed by the MSN Client and can be used to push notification/information (e.g. content, user status, feeds) for the MSN user.

8. Mobile Social Networks in OMA

8.1 Gap Analysis

This section provides the gap analysis with respect to the features related to MSN by identifying the available technologies/initiatives and reflecting relevant OMA-enablers along with the possible recommendations.

For each feature, a short explanation is further provided to illustrate concretely the current situation within the OMA and provide recommendation.

The term "recommended" used in this whitepaper does not imply any action to any OMA working groups. The recommendations are the results of the gap analysis in order to define what is missing to address a Mobile Social Network service from an OMA point of view. How to contribute and fill in this gap is an individual company decision

8.1.1 Identity

A user only needs one account for Federated social networks, and it could be used to interact with other social networks. OpenID can support this mechanism, while URL format is not good readable for a user. How to enable unified identity by supporting multiple human readable identity formats (e.g. email, telephone number, user name) need further study.

User identity in the OMA has various meanings, representations and formats depending on the enablers and in some cases on the underlying infrastructure they are build upon (e.g. SIP-based OMA enablers). Over the last years, a consistent approach has been undertaken to rationalize user addressing throughout OMA enablers, which has resulted in leveraging both email-base formats (user@domain) and user international phone numbers. Whenever useful, URL schemes are used for representation of these formats.

In the context of Internet-based services, many types of networks (email, SIP, XMPP and many current Social networks) rely on the user@domain format for its wide use and understanding amongst users, besides its easy addressability, resolution and routing. The approach of addressing users per username only (without the domain part) is decaying rapidly.

It is therefore recommended to adopt MSISDN or user@domain (where the “user” part may be alphanumeric or the MSISDN) as primary user identification format. The user@domain format can further easily be integrated with the WebFinger profile discovery protocol. Other types of user identities (e.g. OpenID) may be associated with that identity as part of the user profile and possibly used as search criteria for user lookup.

8.1.2 Profile

The information used to generate the profile of a SN user differs widely depending on the scope and interest of the SNS (e.g., academic SNs, professional SNs, video SNs, gaming SNs), thus making it difficult to identify and recommend one or more existing initiatives that can completely satisfy all the requirements.

There exist various initiatives like OMA-CAB, OMA Profile XDM, OMA SUPM [OMASUPM], POCO, hCard and FOAF associated with profile. To better address the variegated requirements associated with profile, the gap analysis and recommendations are discussed under the following categories:

1. Discovery of user profile based on user identity;
2. Data format representations of user profile to represent profile information;
3. Management interfaces for user profile.

Discovery of user profile is the key to the discovery of multiple types of profile data, repositories or interfaces corresponding to a given user identity. OpenID and WebFinger are the two common initiatives associated with the discovery based on user identity.

OpenID uses unique URL to identify the users and allow a requesting party to access profile related information from the identity provider. Although the use of URLs offers great readability and discoverability, they suffer from a not user friendly experience and User Interface design issues.

On the other hand, WebFinger uses email address as a unique identifier. It is based on the WebFingerProtocol [WebFingerProtocol] which assists to obtain an XRD XML file describing how to find a user's public meta-data (XRD is a XML file format for discovering what capabilities a particular profile provider has). This public meta-data may be associated with the user's public profile data, pointer to identity provider (such as OpenID server), public key or profile data (e.g. nickname, full name).

OMA Enablers like MSF, XDM and CAB allow generic search/discovery including user profile and more details are discussed in Section 8.1.10 of this White paper.

Regarding data formats, data portability is a major concern. From the user's perspective, providing same set of data to multiple SNs is a drawback. Semantic web formats like FOAF and SIOC are the initiatives that model user information and user-generated content in machine readable format and facilitate portable data. Both FOAF and SIOC adopt RDF and may be written in XML syntax or any other of the syntaxes of RDF such as RDFa.

JSON is a contender to XML in data formats. It is considered to be simpler than XML with its smaller grammar and considered to be faster too. However, due to the wide spread availability of XML it is a more commonly used/returned format from the APIs available to extract data from Social Web.

Furthermore, initiatives like OpenSocial handle data portability by providing APIs to let applications (widgets) access data and networks from multiple SNs.

Regarding management interfaces for user profile, management interfaces are related to creation, deletion and updating of user profiles. OMA Enablers like CAB, Profile XDM and SUPM address aspects related to management of user profiles.

The Profile XDM Specification of OMA defines naming conventions, data semantics, and validation constrains related to a User Profile. The Profile XDM facilitates basic operations like search, subscription and restore on the user data.

The OMA SUPM enabler allows an authorized principal (i.e. applications and/or enablers) to manipulate (i.e. create/ read/ update/ delete) Services User Profile data (they may be static and dynamic), i.e. any element or group of element belonging to a Service Provider managed set of information related to a User that may be used to create personalized and contextualised services. To do so, the SUPM enabler has the ability to transform the format of data between that (either supplied or received) on the SUPM interface and the underlying data sources and to expose an aggregated view of the Service User Profile.

OMA CAB enabler provides detailed set of elements and sub-elements related to a User's profile. Furthermore, the availability of contact views in CAB allows a MSN user to easily create profiles suitable to different SN sites unlike in other initiatives like POCO and hCard. In addition, OMA-CAB has provision for the use of elements from other namespaces for the purpose of extensibility.

In conclusion, WebFinger is recommended as a mechanism to identify the specific type of user profile. XML is a suitable data format representation for Social Web.

Furthermore, it may be noted that although there are parts of OMA Enablers like CAB and Profile-XDM that can be used directly; there is a lack of support to create or define relationships which is an important features in MSN.

Based on this observation, it is recommended that a standard based approach should evolve while reusing the available features and functionalities in OMA Enablers to facilitate creation and management of a profile from MSN perspective.

8.1.3 Privacy

Informational privacy is about data protection, and the user's right to determine how, when and to what extent information about his/her is communicated to other parties, and the execution of this right might be based on his/her knowledge about what the other party's intention is.

OMA has identified a list of common technical requirements on service enablers that may process personal data on individual data subjects [OMA-Privacy]. In addition to general privacy requirements, some services or service enablers may have specific privacy-related requirements, which are addressed in respective specifications.

IETF OAuth is a popular standard for granting data authorization to third parties, allowing users to grant access to private resources after authenticating themselves via their online identity.

OMA Autho4API (Authorization Framework for network APIs, [OMA- Autho4API]) is an on-going Work Item based on OAuth v2.0 that will enable a user owning network resources exposed by an OMA RESTful API to authorize third-party applications (desktop, mobile and web applications) to access these resources via that API on the user's behalf. Although OMA Autho4API is at this time mainly focused on REST APIs, it may be considered to extend its principles to other types of OMA interfaces, and MSN activity might provide some inputs in that sense.

Based on this analysis, it is thus recommended to consider the IETF OAuth, and OMA Autho4API Work Item to the best possible extent, to address the privacy requirements in the context of MSN so that users may define policies to control access to their data and the related conditions.

8.1.4 Relationships

The *"network"* part of a social network is built around the "relationships" linking the users in the SNs.

FOAF, XFN, OMA CAB, OMA XDM and ActivityStreams are some of the initiatives related to relationships in a SNS environment.

FOAF and XFN are the main initiatives that focus on the representation of relationships between users in a creating the links between the users of a SNs.

FOAF helps machines understand the relationships that connect people through the use of simple semantics to establish acquaintances. FOAF is built on top of W3C's RDF technology. This facilitates FOAF to bring-in specialized inter-personal relationships as well as representing any kind of relationship needed in a social environment.

XFN is another initiative using XHMTL attribute "rel" to describe relationships (or nature of link) between two people (with primary focus on the blogging community). It facilitates defining relationships as friend, acquaintance, contact, co-worker, family to mention a few. Also, there are attempts to express XFN in RDF to allow its inclusion in FOAF.

Currently[24] there is no OMA specification for representing relationships among the SNS users. However, OMA enablers like CAB, Profile-XDM and Group-XDM have provision to populate information related to a user and a user's contacts. They largely cover aspects related to management of users' data like adding, deleting users and user related information. In addition, information related to organizations, groups and links to weblogs is available. This information may be used create a meaningful graph or association among the various users. For example, when user A and User B have organization as "XYZ", the association or relationship between them can be established as "colleagues".

ActivityStreams could be useful to publish/notify/share information about relations that are established among users in a social network.

Based on this analysis, it is recommended:

- to use FOAF for representing relationships between users;
- to enhance the OMA Enablers like CAB and XDM to manage available user data and build social graphs;
- to use ActivityStreams to track changes in the relationships among users.

8.1.5 Content sharing

Content sharing in SNs can be divided into several concepts:

- The act of uploading or posting content (multimedia, file, or text) from its owner to his SN, also associating permissions (e.g. for reading or modifying)

[24] OMA CAB v.1.1 [OMACAB1.1] and OMA SimplifiedCAB v.1.0 [OMASCAB] are two Work Items on working in OMA that have in the scope to incorporate new market requirements (e.g. RCS, and new social networking features).

- The act of consuming content (upon permission), as owner or other user
- The act of notifying the availability of content by 'sharing' someone else's content indirectly, thus attiring attention on it without claiming its ownership

In the area of posting or remote publication of content the OMA CMI enabler has defined interfaces to manage content remotely, including content transfer, both self-contained and decoupled, and purchase. However, such enabler is primarily intended for server-to-server communication through a set of dedicated programmatic APIs that enable the association of metadata to content by the mean of "packages".

OASIS CMIS [CMIS] is another specification that addresses content management in general through remote interfaces, giving a strong emphasis on permission management and providing SQL-like query features. Although such functionalities are promising in the context of MSN, it is unlikely that the full specification be easily supported on a mobile device.

In the area of Social Media, IETF AtomPub has been used as a simple remote protocol (including on mobile devices) focused on SN content and its metadata, thus enabling the publication of both content itself and its corresponding Atom entry. This solution natively fits with ActivityStreams representation of activities over SNs, being content publication ('post') a type of activity.

Content consumption is the most widely developed feature in SN as it enables the access to someone else's content and activities, mainly through syndication techniques and feeds. In particular, files and multimedia content can easily be referenced in ActivityStreams feeds as URLs related to a related activity (e.g. the 'post' itself or a 'share' action on it)

With respect to content notification aka "share" functionality, no relevant OMA enabler has been identified so far. Instead, the OExchange initiative is focused on providing a simple framework, based on XRD that allows discovery of such sharing functionality, further providing a common specification for the actual sharing action.

Note that this functionality mainly relates in MSNs willing to offer this capability to external web sites in order to receive notifications about external content. When used by an MSN client, the "share" functionality can be achieved by sending a "share" activity entry (defined in ActivityStreams) over AtomPub.

It is thus recommended to endorse AtomPub protocol as reference protocol for actual content publication (file or multimedia content) from client to server within MSN, as well as for the "share" functionality. Such protocol should seamlessly integrate support for setting content permissions that integrate with privacy mechanisms at consumption time from other users. OASIS CMIS' principles on permissions should be evaluated when defining content permissions such as ACLs in the context of MSN.

ActivityStreams is further recommended as basic data format for referencing content into activities (such as "post" or "share") and consume it through feeds, both from clients and when exchanging content with other MSNs.

Depending on service provider deployment, OExchange specification is recommended for MSNs willing to open to external web sites.

Depending on service provider deployment, OMA CMI can be considered a valuable enabler to be implemented by MSN server (acting as Service Provider) to enable back-end upload of content.

8.1.6 Activities

Activities in the context of the OMA may relate to multiple enablers, such as for example a change of presence status, an instant message sent to a group, etc. However, OMA is lacking from enablers that allow representation and sharing of activities in a consistent and generic way.

On the Internet, activities and related user streams are becoming widely represented throughout current SNs using the ActivityStreams specification. Such specification is also being endorsed by several popular Federated Social Network initiatives, such as OStatus and OneSocialWeb.

Activity publication can be divided in 2 steps:

- content publication: This step is responsible for the actual remote publication of content from a client to its target server. Content can be any activity (e.g. posting a photo, joining a group, start following a user, commenting) and can embed multimedia content as well as textual content. AtomPub is a specification designed for this purpose as

protocol for client-server interoperability, which can carry any Atom content, including ActivityStreams formatted content.

- publication sharing and notification: This step is responsible for providing a (list of) published activity(ies) to an interested consumer. Access to activities can be performed either in "pull" or "push" mode. For "push" mode notifications, solutions exists based on XMPP, on HTTP (PubSubHubbub) as well as in the OMA (PUSH enabler). In particular, XMPP push allows for direct notifications to users, whilst PubSubHubbub uses a central hub for managing subscriptions and related polling requests (not real notifications). OMA PUSH relies on PAP and OTA protocols to notify mobile users, and could be used directly from the original server or through a PuSH hub.

Based on this analysis, it is thus recommended to endorse the ActivityStreams specification within OMA for activity and stream representation. It is further recommended to:

- Analyse and consider the endorsement of AtomPub protocol for activity publication to an OMA-compliant SN. Such analysis should consider publication of activities from the MSN Client, as well as provide means for external enablers (or applications) to insert activities on behalf of a user within a OMA-compliant SN, that would correspond to a specific action on these enablers (or applications).

- Analyse and consider extensions to the ActivityStreams specification whenever needed, to be developed within the OMA or discussed with the ActivityStreams initiative.

- Analyse and consider the reuse of OMA PUSH enabler as part of a push notification mechanism towards the MSN Client (acting as Push Application on the device).

8.1.7 Follow-up actions

Follow-up actions (e.g. comment, reply, "I like", favourite) can be treated as "activities" in a similar way than original actions, and as such posted and shared using similar mechanisms.

However, their intrinsic characteristic of relating to a previous activity implies their notification to the author of the original activity, and its subsequent propagation to its subscribers. As such, a specific mechanism is needed to guarantee that this follow-up action is notified to its source, even if performed on another SN, and integrity protected, prior to be propagated to its subscribers.

The OMA has no specific enabler addressing the propagation and assertion of messages conveying user follow-up actions. Although email or IM-based protocols, coupled with security techniques, can be considered for addressing this issue, the Salmon protocol is a popular open initiative that provides such features in the Social Web arena. Relying on HTTP, WebFinger and specific signatures [MagicSig] to verify the message and its author, Salmon traditionally carries ActivityStreams messages.

Based on this observation and on its importance as a core functionality of SNs in general, it is recommended to investigate a standard-based approach to this feature within OMA in the context of MSN. In particular, it is further recommended to analyse and consider the endorsement of the Salmon protocol for follow-up activity notification and assertion, and ActivityStreams for representing follow-up actions.

8.1.8 Private messages

Private Messaging is a way to communicate with other members inside social networks. It enables to send messages privately between members, and is somehow similar to e-mail service. Private messages cannot be read by any other member except by the person it was sent to. As such, private messages are always only available for registered users. From the conversation perspective, we can classify the private messages into two kinds of messages: one-shot messages and conversation messages. One-shot messages refer to messages sent to another user not related to a conversation session. On the other side, conversation messages enable users to exchange messages in a conversation session.

From the domain perspective, private messages can be classified into intra-domain message and cross-domain message:

- Intra-domain private messages: here refer to send a private message to another registered user in the same social network. Although it could use any format and methods depending on the social network specific deployment, it could benefit from a standard approach to allow interoperability between client and server on the same SN.

- Cross-domain messages: refer to send private message to a user on another social network. For this type of interaction, standard cross domain protocols are to be used to enable interworking.

Different technologies may be used for one-shot messages and conversation messages:

- One-shot message: HTTP POST method is an easy way to realize this based on existing IP connection. A format like XML can facilitate different SNs when parsing the message. Between client and server or between servers, one-shot message can be considered by posting a message in the receiver's inbox over AtomPub. The use of Salmon needs to be investigated both for client-server and server-to-server communication. It is also possible to use other social network APIs such as Opensocial API for one-shot message. Opensocial java script API defines local API that can be exposed by SN Client. Opensocial REST API is possible to be used for server-to-server or server-to-application. OMA PUSH can enable message delivery to the receiving user via various communication methods or protocols from server to client to complete the overall delivery of the message between sender and receiver. Even if the user is not a subscriber of the social network, it is also possible to push a message through short message to that user's client using OMA PUSH.

- Conversation message: The IETF Extensible Messaging and Presence Protocol (XMPP) is an open technology for real-time communication, which covers a wide range of applications including instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middleware, content syndication, and generalized routing of XML data. Unlike most instant messaging protocols, XMPP uses an open systems approach of development and application, by which anyone may implement an XMPP service and interoperate with other organizations' implementations. Alternatively, IETF MSRP [MSRP] (Message Session Relay Protocol) is a protocol for transmitting a series of related instant messages in the context of a session, and it is the basic protocol that can be used for conversation message. OMA SIMPLE IM [OMA-SIMPLE-IM] is a set of capabilities allowing exchange of instant messages between users in near real-time, it based on the IETF SIP [SIP] protocol with SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions) extensions [SIMPLE]. OMA CPM [OMACPM] is the Converged IP Messaging and provides the convergence of multi-media communication services while leveraging standardized service functionalities from existing OMA communication enablers like OMA SIMPLE IM or OMA PoC [OMAPOC]. IETF SIMPLE, OMA SIMPLE IM and OMA CPM are all based on SIP technology.

The use of protocols based on IETF SIP may introduce a level of complexity for MSN scenarios, and are to be evaluated if it might be appropriate.

According to previous analysis, it is recommended to:

- For one-shot message, consider private messages using AtomPub or Salmon between OMA-compliant SNs and between the MSN Client and the OMA-compliant SN. Further consider OMA PUSH for delivering of the message from OMA-compliant SN to MSN Client.

- For conversation message: IETF XMPP and MSRP, and protocols specified in OMA SIMPLE IM and OMA CPM are potential candidate protocols for cross-domain messages, i.e. between OMA-compliant SNs. It is for further study if they can be suitable to delivering conversation message between clients and the OMA-compliant SN.

- Security mechanisms are to be taken into account to keep messages private and preserve integrity, especially across SN domains.

8.1.9 Groups

Groups in a SN ecosystem are key to socialize and establish new connections, partake information or views, carry out collaborative tasks (e.g. polling) and share multimedia contents (such as images, video, audio) among other activities.

Traditionally groups have been in use for asynchronous communication, wherein, registered or non-registered users can start new threads. The replies or comments to the original post are made by other users at their leisure. This is in contrast to the groups from MSN perspective, which are generally active or dynamic in nature requiring near real-time interactions among members of the group (e.g. group chat in Facebook). This requires groups in a SNs environment to be a standalone entity.

OMA Group XDM, ActivityStreams, OMA IM, OMA PoC and OMA CAB are some of the initiatives that deal with subset of aspects pertaining to group feature.

OMA Group XDM specification has the data format pertaining to a group. It captures all the basic elements like group identifier, group name, participants, private messages, and join-handling associated with groups. OMA Group XDM facilitates other OMA Enablers (e.g. OMA IM, OMA PoC) to support "groups" feature and perform various management operations like creation/deletion of groups, adding/inviting/deleting members and history management. ActivityStreams provide mechanisms to represent a subset of the management aspects like joining or leaving groups in a social web environment.

OMA IM and PoC Enablers facilitate mobile subscribers to engage in group-wide chat communication using the respective services. This, typically, has at least two participants communicating over a chat session. This is in contrast to group communication or functionalities as perceived from social network perspective, wherein, a single user can post messages to the group to be reviewed or commented by other members of the group at a later point of time in addition to real-time chat. The chat functionality within a group provided by OMA IM may, however, be useful in the context of group chat sessions among users in a social network environment.

The major distinction between groups in an IM environment and in a social network environment is that groups in a SN can be standalone entities while the same is not true in an IM environment. OMA CAB has provision to add "group" as a standalone entity. However, it does not support features like sharing comments, multimedia and other collaborative tasks common to SNs.

Based on this analysis, it is recommended that a standards based approach is needed to define groups feature from SNs perspective while enhancing or reusing the capabilities and functionalities in the available OMA initiatives like OMA Group XDM, IM and CAB depending on whether or not the protocols and mechanisms therein are suitable for MSN environment. In particular, the following features should be considered with regard to the group feature from SNS perspective:

- Group management capabilities from OMA Group XDM;
- OMA IM functionality to allow chat among users in a social network environment;
- OMA CAB's provision to create group as an independent entity.

8.1.10 Search

Users' search based on profile characteristics (e.g. by name, location, interests, or profile information like age and gender) is a common feature of existing SNs. Further, content or group search is also a common functionality. In the OMA, search has been investigated by MSF [OMAMSF] as well as XDM enablers.

OMA XDM [OMAXDM] in general allows for searching locally on XML documents within an XDM server. In particular, Profile XDM allows for generic search based on user profile information, executable over all user profile documents. XDM further supports authorization policies and change notifications through SIP, and is accessible by XDM Principals. OMA CAB supports contact search including Personal Contact Card (PCC) search, Address Book search and External Directories search based extension XDM search. Group XDMS is the logical repository for Group Documents. OMA CMI [OMACMI] enabler provides mechanisms for the management of contents, defines the APIs between Content Provider and Service Provider (e.g. Operator) for managing contents.

OMA MSF is an enabler dedicated to mobile search, relying upon Atom and OpenSearch specifications to provide an interoperable framework for search, including search engine registration, inter-MSF server search, as well as personalized search and recommendations, and subscribe-push functionality. OMA MSF is natively extensible to include several Search Engines that can register with the MSF. Unless using personalization feature, no user identity is required although possible. Searches within MSF rely on "domains" (and related criteria) for which relevant search engines can register their availability.

In the area of Social Web, the role of search is still unclear and so the related initiatives for providing profile-based search or content search in general. WebFinger is an initiative that provides user identity and profile discovery, although it requires the a priori knowledge of a user@domain identity and does not provide any feature for profile-base search.

Such current lack of standardization does not allow to find users (and thus communicate with them) without knowing their identity and/or searching for them on specific SNs in a proprietary fashion. The same applies to any content search on such SNs.

It is thus recommended to provide a standard-based approach within OMA for allowing transversal search within MSN. In particular, it is recommended to reuse MSF enabler to support any type of search relevant to MSN by the means of dedicated Search Domains and associated domain specific parameters to retrieve search results and recommendations. In particular in the context of profile-based user search a "user" Search Domain would enable the retrieval of user identity candidates and possibly provide user (friend) recommendations.

Although no recommendation is given on the actual repository(ies) to be used as MSF Search Engine(s) in the context of MSN search capability, it is recommended to identify possible OMA enablers with search capabilities that are relevant to MSN (e.g. Profile XDM, Group XDMS, CAB) and provide guidelines to register and integrate the related server as an MSF Search Engine within the MSF enabler. Such servers may be used as reference backend systems for MSN search as a Service Provider deployment option.

Search functionality should at least enable to search for user identities (according to Section 8.1.1 recommendations) based on a set of profile information. Such user identities may be used as input for WebFinger discovery of profile and activities information.

8.1.11 Open API

Open APIs provided by a SN enables third-party web applications, applications residing on the device, and other SNs to access its functions such as: browsing an activity feed, uploading content, finding a friend. API is usually distinguished in network API and device API. Network API is provided by the MSN Server. While device API is provided by the MSN Client and is used by other applications installed and running on the same device as the MSN Client.

Most popular Social Networks provide their own proprietary Web API: Facebook Graph API enables application to get data for an "object" concept and get the relationships of these "objects"; Twitter API has also made available many functions to the third parties, e.g. member list, friendship, social graph, notification and so on.

There are also some standard APIs that relate to SNs:

- OpenSocial is a set of APIs for getting information about the person, group, activity information and so on; both JavaScript API and REST API are supported.
- OMA provides many kinds of APIs and some of these are designed to expose functionalities of specific OMA enablers. Examples of OMA network APIs that may be related with social network functions are:
    - The address list management API of OMA Parlay Service Access V1.0 (PSA) [OMAPSA] and OMA ParlayREST V2.0 [OMAParlayREST2] allows an application to manage groups (address lists, or group URI's) and their associated properties.
    - OMA PSA Content Manangement API enables content handling with submit, read, modify and query operations; support content notification, storing and managing the meta-data associated with content elements.
    - OMA Next Generation Service Interfaces V1.0 (NGSI) ([OMANGSI] and [OMANGSI-S]) has Identity Resolution API and Identity Management API for identity control that allows an application to manage the Identifiers of an Identity.
    - OMA RC APIs [OMARC_APIs] will provide RESTful API about social presence information to share and publish presence information. It also provides Network Address Book API to get contact information and receive updates on contact information.
    - OMA CAB APIs [OMACAB_APIs] will provide applications with access to and management capabilities of the rich personal data stored in the Converged Address Book and Personal Contact Card as specified by OMA CAB 1.0
- Example of OMA device APIs that may be related with social network functions is:
    - The OMA Client-Side Enabler API [OMACSEA] defines requirements for API's as provided to client-side applications executing in Web runtime environments, e.g. in Web browsers or widget engines, using Javascript as the primary procedural language. In its initial release, CSEA defines API requirements for

OMA Dynamic Content Delivery (DCD) [OMADCD], OMA PUSH, and OMA Dynamic Profile Evolution (DPE) [OMADPE].

- Moreover, OMA has produced an OMA Guidelines for Network REST API specification intended to provide the guidelines for defining REST interfaces in OMA, and guidelines for device API are on working as part of the OMA Web Runtime API (WRAPI) [OMAWRAPI].

Based on this analysis, OpenSocial APIs seem to be promising candidate to be used for providing MSN functions to the third party applications. However it is not possible to provide a more detailed recommendation before to have clearly identified the functionalities that are to be exposed from a MSN. Once these functionalities are identified it will be possible to determine which already existing API may be re-used as they are or with appropriate extensions.

It is further recommended that the design of network APIs and device APIs toward applications and 3rd parties follow OMA guidelines.

8.1.12 Data Portability

Without user re-register, data portability allow to get or import the user personal information, contacts, profile, history, media and content from the original social network. Two aspects need to be considered: permissions for getting the portable data and how to import the data.

- Permissions for getting the portable data: authorization policies and privacy rules should be portable across SNs. OpenID enables identity portability, allow the user using one account for Federated SNs. In order to ensure an application or a social network has permissions to fetch user data, the user original social network may support an authorization mechanism.

- Import or exchange the data: a user may want to bring many kinds of data from the original social network to another SN or application. Standard format is the foundation for SNs or applications to parse the data. Different data has different formats, e.g. profile referenced by Webfinger, contact information based vCard or Hcard, content description on ATOM. There are also some particular mechanisms or protocols for fetching different kind of user data.

    - Contact: Portable Contacts parse contact information from atom or json and made the contact portable under user control, the Portable Contacts API is wire-compatible with OpenSocial RESTful Protocol and can be used for requesting address book, profile, and friends-list information from the original social network. OMA CAB is capable of share contacts information between users, but it can't provide the contact information to other applications currently[25].

    - Profile and relationship: FOAF is one choice about information and relationship. FOAF files are just text documents adopt the conventions of the Resource Description Framework (RDF). It allows social network to generate profile pages automatically from a database, post RDF/XML files and link them from an HTML homepage. SIOC allows representing rich data from the Social Web in RDF, it is commonly used in conjunction with the FOAF vocabulary for expressing personal profile and social network information.

    - Contents, media resource: AtomPub uses HTTP methods to create, edit, and delete media resources. An application or social network can use GET to retrieve a representation of a known resource from the original social network. Social web service can use OpenSocial API to retrieve application data on behalf of a given user from the original social network.

    - Activities: using ActivityStreams for descibing the activities and syndicate the activities from one SN to another.

Based on this analysis, it is thus recommended to:

[25] OMA CAB API is a Work Item still on working that will define API to expose CAB functionalities to third parties.

- For the Federated SNs, analyse and consider the reuse of AtomPub to support data exchange such as contents and activities. How to endorse different data formats and how to exchange other types of data need to be further analysed;
- In the case of an application or social web services fetching data, using API is a good way to address this, for example to retrieve person, group information, application data or activities on behalf of a given user (see Section 8.1.11).

Regarding recommendations on permissions see Section 8.1.3 on Privacy.

8.2 Wrap-up and Conclusion

The following table is a summary of the gap analysis detailed in Section 8.1.

Feature	Most relevant OMA enablers or initiatives	Other most relevant initiatives	Recommendation
Identity	-	OpenID, WebFinger	Adopt MSISDN or user@domain as primary identity; reuse OpenID and WebFinger as basis to support one identity for Federated SNs.
Profile	CAB, Profile XDM, SUPM	XRD, WebFinger, HCard, FOAF, POCO, OpenSocial, OpenID	WebFinger can be used to discovery the type of user profile. XML based initiatives like FOAF are recommended data format for Mobile Social Networks. OMA CAB can be used for profile, however, it needs to be evolved to support SNS features like relationships
Privacy	Privacy Requirements for Mobile Services, Autho4API	IETF OAuth v2.0	Consider IETF OAuth v2.0 and OMA Autho4API Work Item
Relationships	CAB, XDM	FOAF, XFN, ActivityStreams	Consider using FOAF for representing relationships between users Consider to enhance OMA enablers like CAB and XDM to support relationships Consider to use ActivityStreams to track changes in relationships
Content sharing	CMI	AtomPub, OASIS CMIS, OExchange	Consider use of AtomPub for content sharing and ActivityStream for referencing Consider OExchange for

			sharing from external sites with MSN
Activities	PUSH (notifications)	ActivityStreams AtomPub XMPP PubSubHubbub	Consider use of ActivityStreams for representation and AtomPub for publication Consider use of OMA PUSH for push notifications
Follow-up actions	-	Salmon, ActivityStreams	Consider use of Salmon for follow-up action propagation and assertion, and ActivityStreams for representing follow-up actions.
Private messages	PUSH, SIMPLE IM, CPM	AtomPub, Salmon, IETF XMPP	Consider use AtomPub or Salmonfor one-shot message and OMA PUSH for message delivery. Consider use MSRP or XMPP for conversation message.
Groups	IM, PoC, XDM Group, CAB	ActivityStreams	Consider evolving an appropriate method to address the deficiencies in the available initiatives. Furthermore, consider: - OMA Group XDM for group management functionalities - OMA IM for chat functionality in a group - OMA CAB's capability to have group as an independent entity
Search	MSF, Profile XDM, Group XDM, CAB, CMI		Use MSF as basis for overall search. Allow for back-end interconnection with specific OMA enablers for vertical search functionality
Open API	PSA, Parlay REST V2_0, NGSI, RC APIs, CAB APIs	OpenSocial APIs	A recommendation may take place only when MSN functionalities to be exposed have been identified. Follow OMA guidelines for network API and device

			API.
Data Portability	CAB	AtomPub, ActivityStreams, FOAF, OpenSocial, SIOC	Consider use of AtomPub for data portability between SNs. Use OpenSocial API to provide data for applications.

Table 1: Gap Analysis and Recommendation

Many initiatives exist currently in the Social Web arena, and some of them are clearly becoming more popular. However, many of them are still evolving, having different degrees of maturity and adoption by the major SN players. In addition, most of the current players and specifications do not specifically focus on mobileaspects, thus potentially leading in difficulties when adopting such solutions over mobile networks scenarios.

Based on the (unavoidably not exhaustive) analysis conducted in this document and with specific reference to OMA, a first conclusion suggests that the main "gap" missing for MSN within OMA relates to mechanisms to inform other users about their actions. In other terms, it is needed that users have an (easily discoverable) identity and, based on this identity a user is able to follow someone else's activities (whatever they may be) and share his/her owns. This topic has not been addressed yet within OMA and would benefit dedicated standardization effort.

A further conclusion of this analysis is that several OMA enablers have been identified as relevant to the context of MSN for some features. In particular, some enablers can be reused "as-is" as facilities (e.g. OMA PUSH for notifications), whilst others may be enhanced to address MSN-specific requirements (e.g. OMA CAB for relationships management).

Some further OMA enablers may be considered as side-enablers in the context of MSN and can be used to implement extra functionalities (as deployment options by service providers). For example, as mentioned throughout Section 8.1, some OMA enablers can be considered to address the actual communication between MSN users (e.g. chat). No specific need has been identified so far to mandate or define new enablers to perform the actual communication between users that are already widely covered by standards (including in the OMA). However some study may be needed for their use in the context of MSN.

Based on the above conclusions, the needed common set of core features for MSN interoperability within OMA is considered limited (positive aspect).

In parallel, other features can be initially covered through "guidelines" to optionally reuse relevant OMA enablers within a single MSN (intra-domain). Depending on the adoption of such features and the maturity of relevant enablers or initiatives, such features may progressively be introduced as part of OMA MSN to enrich interoperability between SNs.

9. Recommendation

9.1 Pursue standardization activity on MSN within OMA

It is recommended to pursue the study of an MSN specification within OMA through one or more Work Items to embrace the Social Web activity in the large, closely monitoring mobile aspects and possibly leveraging OMA enablers. In particular, a new Work item may be dedicated to the development of a core specification for MSN, whilst others may focus on the evolution of other OMA enablers for use in conjunction with MSN (e.g. CAB).

9.2 Scope MSN activity

It is further recommended to identify a coherent subset of functionalities specific to MSN that can be rapidly specified as a core specification for MSN within OMA, thus allowing large-scale deployments and interoperability of MSN Clients and MSN Servers in a timely manner.

In this sense, it is envisioned that some additional features related to MSN may or may not be provided by the service providers, and if provided they may or may not reuse OMA enablers (i.e. may be provided through proprietary means). To the best possible extent, guidelines may be given to rely on existing OMA enablers also for such additional features. In particular, it is recommended to include in the scope of a ***new dedicated OMA MSN Work Item*** the following items:

- define the interface between the MSN Client and the MSN Server, and the interface between MSN Servers to enable federation of OMA-compliant SNs, supporting at least features such as:
    - profile discovery;
    - publication and sharing of contents, activities and follow-up actions;
- consider the endorsement of OStatus-related specifications (WebFinger, ActivityStreams, Salmon, PubsubHubbub), typically for server-to-server interactions
- consider the endorsement of OpenSocial REST protocol and/or AtomPub, typically for client-server interactions;
- consider OMA PUSH enabler to support notifications to MSN Client (follow-up actions delivery, private message delivery, user status notification, etc);
- consider exposing device APIs (e.g. OpenSocial JavaScript) and network APIs, as well as an appropriate privacy framework, to easily integrate OMA-compliant SN with external applications;

In addition, it is recommended to evaluate ***further Work Items in relation to existing OMA enablers*** to benefit MSN activity, such as:

- consider the evolution of OMA CAB Enabler for MSN user profile, groups and relationship management[26];
- create guidelines to reuse existing OMA enablers for additional MSN features (e.g. profile search using OMA MSF).

9.3 Liaise with other SDOs

Finally, it is recommended to liaise with the W3C group “Federated Social Web Incubator Group” to clarify roles and commonly address the standardization of Social Networks both from a Web and Mobile perspectives, thus speeding up the delivery of interoperable solutions.

In general, due to the vast amount of specification work being undertaken in the context of the Social Web, it is recommended to monitor existing or new SDOs embracing this context and evaluate future liaisons.

[26] OMA CAB v.1.1 and OMA SimplifiedCAB v.1.0 are two Work Items on working in OMA that have in the scope to incorporate new market requirements (e.g. RCS, and new social networking features).

Appendix A. Change History (Informative)

Document Identifier	Date	Sections	Description
Draft Versions OMA-WP-SocialREST	16 Nov 2010	all	initial version of WP as permanent doc
	24 Nov 2010	3.3, 4	Update accordingly to INP discussed in Seoul (R&A): - OMA-REQ-2010-0121R01-INP_Mobile_Social_Network_WP_Introduction
	30 Nov 2010	5.1, 2, 5.2	Update accordingly to INP discussed in REQ CC 20101130: - OMA-REQ-2010-0118R02-INP_mobile_social_network_industry - OMA-REQ-2010-0123R01-INP_Mobile_Social_Network_WP_State_of_the_Art
	02 Dec 2010	2, 6.1, 6.2, 6.3	Update accordingly to INP discussed in REQ CC 20101207: - OMA-REQ-2010-0122R03-INP_Mobile_Social_Network_WP_Scenarios - OMA-REQ-2010-0119R02-INP_MSN_multiple_terminal_scenario
	11 Jan 2011	5.3, 7.1	Update accordingly to INP discussed in REQ CC 20110111: - OMA-REQ-2010-0139R02-INP_MSN_value_chain - OMA-REQ-2010-0140R03-INP_MSN_Architecture_Aspects
	21 Jan 2011	3.2	Update accordingly to INP discussed in REQ CC 20110111 and following R&A: - OMA-REQ-2011-0004R02-INP_MSN_Definitions
(renamed) OMA-WP-Mobile_Social_Network	25 Jan 2011	all	Update accordingly to INP discussed in REQ CC 20110125: - OMA-REQ-2011-0008R01-INP_MSN_WP_Review_for_consistency
	28 Jan 2011	8	Update accordingly to INP discussed in REQ CC 20110125: - OMA-REQ-2011-0010-INP_MSN_WP_Gap_analysis_template
	16 Feb 2011	6.4, 8.1, 8.1.10, 2, 8.1.6, 8.1.1	Update accordingly to INP discussed in Honolulu(R&A): - OMA-REQ-2011-0018R01-INP_MSN_WP_Device_Applications - OMA-REQ-2011-0019R02-CR_Mobile_Social_Network_WP_Gap_Analysis_Search. - OMA-REQ-2011-0021R01-CR_MSN_WP_Gap_Analysis_Activities - OMA-REQ-2011-0024R02-CR_MSN_WP_Gap_Analysis_Identity
	28 Feb 2011	8.1, 8.1.12, 8.1.3, 2, 8.1.5,	Update accordingly to INP discussed in REQ CC 20110222 (in R&A): - OMA-REQ-2011-0025R01-CR_MSN_WP_gap_analysis_data_portability - OMA-REQ-2011-0028R02-CR_MSN_WP_Gap_Analysis_Privacy - OMA-REQ-2011-0029-CR_MSN_WP_Gap_Analysis_Content_sharing
	01 Mar 2011	8.1, 8.1.2	Update accordingly to INP agreed in REQ CC 20110301 (in R&A): - OMA-REQ-2011-0031R03-INP__INP_MSN_WP_gap_analysis_Profile
	11 Mar 2011	2, 8.1, 8.1.7, 8.1.9	Update accordingly to INP discussed in REQ CC 20110308 (agreed in R&A): - OMA-REQ-2011-0040R01-CR_MSN_WP_Gap_Analysis_Followup - OMA-REQ-2011-0041R02-CR_MSN_WP_Gap_Analysis_Groups - removal of sections 6.5 and 8.2 (agreed in the call) - minor editorial changes
	15 Mar 2011	2, 8.1, 8.1.8	Update accordingly to INP agreed in REQ CC 20110315 (agreed in R&A): - OMA-REQ-2011-0042R03-CR_MSN_WP_Gap_Analysis_Private_messages
	18 Mar 2011	2, 8.1, 8.1.11, 8.1.4,	Update accordingly to INP discussed in REQ CC 20110315 (agreed in R&A): - OMA-REQ-2011-0048R02-CR_MSN_Gap_Analysis_Client_Web_API - OMA-REQ-2011-0053R01-CR_MSN_WP_Gap_Analysis_Relationships
	24 Mar 2011	5.2.2, 8.2, 9, 2	Update accordingly to INP discussed in REQ CC 20110324: - OMA-REQ-2011-0047R02-CR_MSN_WP_Recommendations

Document Identifier	Date	Sections	Description
	29 Mar 2011	All, in particular up to section 5	Update accordingly to INP discussed in REQ CC 20110329: - OMA-REQ-2011-0071R01-INP_MSN_Editorials_up_to_Section_5 - editorial changes
	11 Apr 2011	All	Update accordingly to INP discussed in Sorrento 20110411: - OMA-REQ-2011-0083R01-INP_MSN_Whitepaper_Final_Review - Editorial clean.
Approved Version OMA-WP-Mobile_Social_Network	26 Apr 2011	All	Status changed to Approved by #: OMA-TP-2011-0145- INP_Approval_for_MSN_Whitepaper_public_and_published
	16 May 2011	All	Editorial Changes after legal checks