# Mobile Web 2.0 Browser for Collaborative Social Networking

Yuri Natchetoi Viktor Kaufman Louenas Hamdi Albina Shapiro

SAP Research
{yuri.natchetoi,viktor.kaufman,louenas.hamdi,albina.shapiro}@sap.com

**Abstract.** The popularity of Social Networking applications on the Web as well as the need for ubiquitous internet-based collaboration tools, leads to a transformation of our understanding of what the Web browser should be capable of. The Internet is increasingly seen as a global composite application platform and a Web 2.0 workspace. To cope with this evolution of the Web perception, there is a need to extend the notion of the next generation Web browser. This paper describes a Web 2.0 and Web Service-based approach to the Mobile Web Browser 2.0 for Social Networking. In the domain of collaborative mobile applications, we are supporting the preliminary standard of Social Networking Web Services defined by ECOSPACE [1] project. These services can be seen as building blocks for a BPEL [2] composition scenario. In addition, work is underway on the implementation of Web 2.0 style interaction, which could be supported by the next generation of Mobile Web browsers available on almost all mobile devices, including mass-market cell phones.

## Introduction

In today's world, the popularity of Web 2.0 based social networking applications is rapidly increasing. In the Web 2.0 arena, the social networking website MySpace is soon to reach 200 million members. Many young people use Social Networking services in their everyday life to exchange messages, share files, and expand their social circle. Upon joining the enterprise workforce, they will expect similar communication tools facilitating teamwork and business networking.

Mobile devices can provide a convenient means for accessing Web 2.0 applications due to a recent progress in the mobile phone technologies. There is a big potential in using cell phones to access the Social Networking and collaboration services [3]. Unfortunately, not all of the Web 2.0-based Social Networking applications can currently be used on mobile phones because of the following limitations: the mobile devices are still limited in bandwidth, have little memory, a small screen size and a tiny keypad. For example, the average page size of the Social networking Web site such as Facebook or MySpace exceeds 1 Mb. Even the most advanced smart phones can not render pages of that size.

On the other hand, modern cell phones are equipped with camera, GPS, Bluetooth, etc. These features, as well as voice communication, could be an added value for Social Networking applications. However, standard Web browsers do not support any of these input modalities. The popularity of mobile devices and Social Networking services could potentially lead to a redefinition of a Web browser concept. In order to carry out Social Networking communication, the new generation of Mobile Web 2.0 browsers should be asynchronous, support two-way communication, multi-modal input including voice, video and various Bluetooth-connected sensors. They should be able to automatically scale down the complexity of web pages or delegate this function to some proxy servers. They should also support Web services that perform Social Networking and collaboration activities.

# State of the art

In this paper, we would like to emphasize the need to redefine the notion of the Web browser for mobile devices. On the one hand, there is a demand for the standard support of collaborative and Social Networking features. On the other hand, the usability and performance of handheld devices strongly depends on the Web browser capabilities.

Facebook is a representative example of an increasingly popular social networking site. It started as an internal network for Harvard students, opened its doors to the Internet community, and it has now reached 34 million members. The users can not only create their profiles, exchange photos, messages, join groups of friends, but also take advantage of special features like “News Feeds”, which displays all the recent activities of one’s friends, or the “Wall” in one’s profile, where other users can leave short messages and “gifts”. There is also a marketplace for Facebookers, features to attract each other’s attention, to communicate one’s status or plans. Owing to the recently launched Facebook Platform, it is even possible to develop applications and mashups interacting with the core Facebook services. Mobile users can use many Facebook features by navigating to a dedicated URL (m.facebook.com).

Another interesting example is the social networking startup Twango[4], recently acquired by Nokia. Its social networking site allows the users to upload multimedia files from fixed and mobile devices. Twango has made a point of publishing its APIs for the third-party developers.

Obviously, there are many different proprietary APIs with similar functionality and their integration is practically not supported at all for mobile devices. The missing interoperability between different Social Networking sites is disadvantageous for the end users. Many of them have to maintain accounts with different providers in order to stay in touch with their friends; having to take care of synchronization and other issues.

Mobile users can communicate, collaborate and socialize with each other by means of an integrated groupware and other software, most of which is available on the modern mobile phones, smart phones such as Blackberry and iPhone, and PDAs. Applications such as e-mail or instant messaging client, as well as such office tools as the calendar, notes and personal contacts are nowadays common. Modern mobile phones also feature the use of multimedia and provide an mp3-player, a radio and a camera. Smart phones can even imitate some functionality of desktop computers, i.e., the more advanced ones can display attachments in Adobe PDF or Microsoft DOC format.

Given the variety of mobile devices, it is not surprising that the existing mobile browsers differ significantly in their functionality, performance, accessibility and maturity. For example, the optimized Blackberry browser performs pretty well, even though it uses a proprietary expensive server infrastructure and has other drawbacks. The existence of proprietary mobile operating systems is an additional obstacle to the providers of the mobile content. The end user experience suffers from poor readability of content which was developed for desktop computers and not optimized for mobile users.

Modern mobile browsers mature slowly [5]. Their engine usually supports HTM, XHTML, links in Image-Maps, SSL and UTF8/UTF16. JavaScript 2.0 and CSS 2 support is often deficient. Only several browsers support iFrames, RSS 1.0, DOM-manipulation and Cookies. Finally, the features and standards like AJAX, animated GIFs, transparent PNGs, SVG, auto-completion, password management, multiple tabs, viewing PDF, Flash, Windows Media, Real Player, QuickTime and Java are not supported by most browsers.

Safari Web browser [6] is one of the most advanced “traditional” Web browsers for mobile phones. It can render many complex Web pages, which are using JavaScript. Nokia Symbian-based smart phones and Apple iPhone use the Safari browser. Unfortunately, iPhones can only use AT&T infrastructure, since it is part of their security concept. Apple has published a set of restrictive guidelines [7] for mobile content providers. Although Apple and many others believe in the One

Web approach, where the mobile browsers are capable of rendering regular Web pages by means of an alternative style sheet customized for the smaller screen, only a small number of content providers optimize their sites for mobile devices. This once again highlights the limitations of current mobile browsers.

In the mobile Web browser development domain, Opera Mini [8] browser is probably the most innovative existing browser and a first serious "deviation" from the traditional browser model. It uses a server-based Transcoder tool that transforms HTML pages into an OBML format which better fits the mobile environment with a limited bandwidth and a variety of phones. It also offers a photo sharing feature, which is the first step to a multi-modal input. As a result of such innovative approach, Opera Mini can not only work on smart phones, but also on almost all J2ME enabled mobile phones. However, accessing the enterprise content through the proxy server outside of the company's security perimeter is an unacceptable security threat for many companies. Additionally, for example, AJAX doesn't work in Opera Mini. Opera Mobile is a substantial improvement in comparison with Opera Mini, but it requires Windows Mobile or Symbian OS. Both Opera Mini and Safari do not support file downloads or uploads (except for the Opera Mini's camera image blogging) and there is no plug-in support such as Flash.

An interesting approach is taken by Thumbspeed, which helps Mobile collaboration beyond SMS and into the realm of mobile social networking. The company develops mobile email, instant messaging and a product that allows users to post camera phone pictures to blogs, which operate across a variety of operating platforms. This approach meets the communication needs of many users, but it is not flexible enough to be adopted by the global community. A much more promising approach is using Web Services.

Mobile applications have been making use of Web Services for quite some time now. Companies like IBM, Nokia and Microsoft have made software available for enabling mobile devices to act as Web Service clients [9]. There exist several development toolkits for Web Services on mobile devices. For example, gSOAP [10] is a platform independent C++ toolkit for Web Services. On the J2ME platform (mostly for Symbian based devices), JSR-172 is a widely used set of Web Services APIs [11]. The .NET Compact Framework for smart phone architectures also supports the use of synchronous and asynchronous invocation of Web Services [12].

In this paper we build on the idea of the Ecospace project [1] founded by EU, which aims at providing a standard set of Web Services for the social networking tasks. We incorporate this approach into our Mobile Browser 2.0 concept, which enables us to potentially bring the Web2.0 style and P2P paradigm into mobile applications and significantly enrich the Social Networking features. In the next section we describe our vision of the implementation of a mobile Web browser.

We share our understanding of the next generation of Web browsers for mobile devices, which both leverage unique Social Networking and collaboration potentials of mobile devices and follow the One Web approach.

# Mobile Browser 2.0

.Unlike the early Internet which used to consist of mostly hypertext, the today's Internet includes multimedia content, various services, communication and collaboration support and is rapidly becoming a global composite application platform. Mobile devices stand for the same idea of the ubiquitous connection as the Internet and are more appropriate than laptops for many activities. Many Social Networking sites such as twitter.com and others supporting emergent communication are increasingly being accessed through mobile handhelds. The next generation mobile browser has to therefore be able to cope with the evolution of the Internet.

In order to meet the requirements imposed by the mobile collaboration and social networking services, the next generation of Web browsers should have a number of features currently missing in a traditional Web browser. Specifically, the missing features are: two-way communication, direct access to Web services, standardized Web services APIs and standardized browser APIs for collaboration, client-side workflow, multi-modal input, Scalable Vector Graphics (SVG), XForms [13] and Widgets.

Interestingly, the first browser WorldWideWeb [15] supported two-way communication and let users edit the WWW content. Currently the traditional browsers only allow client to server communication and in some cases provide a very limited push mechanism. However, many social networking applications, such as, for example, instant messaging, require server to client communication. In order to enable efficient collaboration, the Web Browser 2.0 must therefore support two-way communication. Contrary to the Wiki–style editing of content, we propose standardized means for a two-way communication with the browser.

Secondly, the Mobile Browser 2.0 has to provide some ways of communicating with it using Web services. In particular, for any given session such as, for example, an instant messaging session, the browser should offer a Web service to asynchronously push an existing or generated site content. The browser should also provide a standard Web service, which would return a list of Web services, supported by the client. Information and application sharing, which are essential to Social Networking, could in turn be implemented as standardized Web services performed by the mobile client through the proxy server.

The Ecospace project [1] is aiming to provide a standard set of Web services for the Social Networking and collaboration tasks. The Web Browser 2.0 could

potentially be a client application for those services. Some examples of such services are: Instant Messaging, E-mail, Application Sharing, Video Conference, etc. Instead of having separate clients for those services it would be beneficial to have only one client implementing standardized interfaces to all the above functionalities.

Apart from the emerging standardized Web service APIs, many existing Web 2.0 applications are evolving as Web services. The collaboration Web services are especially important for the user-centric web browsers. The traditional AJAX applications can't invoke Web services directly. Since Web services have become a standard way of integrating applications on the Web, the Browser 2.0 must support direct Web service invocation.

In a world of Web services, BPEL[2] is becoming the most popular language for the scripting of composite scenarios using core Web services as the building blocks. The client-side implementation of a mobile-friendly subset of BPEL in addition to JavaScript would therefore be a logical step in the development of the next generation user-centric Web browser. In addition, the Browser 2.0 should also support BPEL4People [16] standard, which is an extension of the BPEL-based scenarios, used for the description of the human-performed tasks in the Business Process workflow. By implementing this standard, the core Social Networking tasks performed by people during the collaboration process could be easily scripted as a part of more complex Web Services, thus combining machine- and human-performed activities in a uniform way.

Another important feature to be added to the mobile Browser 2.0 is the multi-modal input. Traditional Web browsers only support keyboard and mouse as their input modes. Mobile devices typically do not have a mouse and have a tiny keypad, which makes input inconvenient. However, mobile phones have a microphone and many also have a camera, Bluetooth and GPS. These features can be used for inputting information. For example, a user can talk into the microphone instead of typing, or use the camera for barcode scanning. Various sensors connected through the Bluetooth interface can be further used for different types of input, such as measuring temperature, getting directions, etc. All these input modalities should be accessible in the Browser 2.0.

The traditional Web browsers are also missing the scalable vector graphics. They support only raster formats such as GIF and JPEG, which do not scale down well for mobile devices. Scalable Vector Graphics (SVG) is an emerging standard that allows delivering much more compact images to the mobile devices and would be an important asset to Browser 2.0. Also, the browser performance can be further improved by replacing the traditional HTML forms by XForms, which is an emerging XML format used to update structured XML data asynchronously. XForms can be further enhanced by multi-modal input modes to allow input using speech recognition, bar code reading and GPS positioning. XForms could

therefore provide a convenient user interface for a multi-modal input, offering a compact and user-friendly alternative to extending html forms with the JavaScript.

Finally, using widgets can further improve user experience with Web Browser 2.0. All recent versions of the operating systems including Windows Vista, Apple OS X and Symbian S60 support widgets. Web Browser 2.0 should therefore be a standard environment for supporting widgets.

# Mobile Architecture overview

Today most of the popular Social Networking tools and services (such as email and Instant Messaging) are using the Web 2.0 technologies. However, many collaboration tools still require the power of desktop application platforms. Unfortunately, Web 2.0 applications are not compatible with such features as TCP/IP sockets, voice, 3D graphics and video input. In addition, the JavaScript applications are much slower than C++ or Java programs, less secure and are lacking the standard API. Furthermore, the AJAX frameworks work differently in various browsers such as IE, Firefox and Opera, while many mobile versions of the Web browsers don't support AJAX capabilities at all. Nevertheless, AJAX applications have an advantage that overweighs many drawbacks, especially for the text-oriented applications. The AJAX frameworks use the Web browser as a runtime platform, because it is already installed on many computers. OpenAJAX project [17] is an effort towards the inter-operability between browsers and unification of AJAX API.

When the true AJAX inter-operability becomes a reality, it would become a perfect platform for mashup applications - an approach to building composite applications using Web 2.0. An alternative promising approach to building Web 2.0 composite applications as a Web Service orchestration using BPEL. This architecture organizes the Web 2.0 application as a multi-tier construction using Web Services as the building blocks.

# The building blocks of Social Networks

The goal of the Ecospace [1] project is the definition of SOA-based building blocks for Social Networks and the development of an efficient service orchestration mechanism for creating complex mashup applications using these building blocks. The many years of positive experience of using BPEL for building composite SOA applications makes BPEL an obvious candidate tool for creating Social Networking mashups.

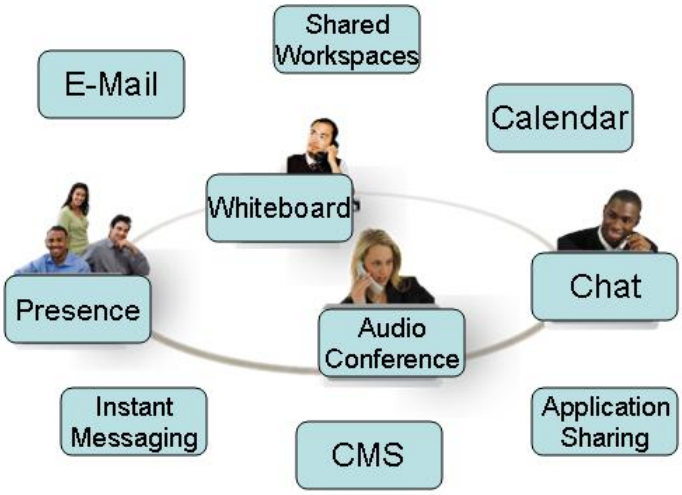


Figure 1. Social Networking web service framework

The core Web services for social networking are grouped into functional modules. Each module defines a number of Web Services related to a specific type of communication or source of information. The list of these Social Networking oriented modules is presented in Table 1. The second column of the table indicates the modality of the Service group: text, rich text, voice or video, and the third column describes the service module in terms of the number of participants involved in some action.

Table 1. Standard Social Networking Service types

| ***Web Service*** | ***Modality*** | ***Typical usage*** | ***Type*** |
|---|---|---|---|
| ApplicationSharing | Video | group | sync |
| AudioConference | voice | group | sync |
| Blog | text | mass | async |
| Calendar | text | individ.,group | async |
| Chat | text | mass | sync |
| ContentManagement | rich text | ind.group, mass | async |
| ContextManagement | rich text | individ., group | async |
| ControlVersion | N/A | individ.,group | async |
| DataBase | text | group,mass | async |
| DistributionList | text | group | async |
| Email | text | group | async |
| Forum | text | group,mass | async |
| GroupManagement | text | group | async |
| InstantMessaging | text | group | sync |
| MultimediaConference | video | group | sync |
| News | text | mass | async |
| NotificationAndAlert | text | group | (a)sync |
| Phone | voice | group | sync |
| PollsAndSurveys | text | group,mass | async |
| PresenceAndAvailability | text | group | sync |
| ResourceManagement | N/A | ind.,group,mass | async |
| RightsManagement | N/A | group,mass | async |
| SemanticAnnotation | rich text | ind.,group,mass | async |

| *Web Service* | *Modality* | *Typical usage* | *Type* |
|---|---|---|---|
| SharedWorkspaces | text | group,mass | async |
| Statistical | text | ind.,group,mass | (a)sync |
| SynchrMultiUserEditing | rich text | group | sync |
| Syndication | text | group,mass | (a)sync |
| TaskManagement | N/A | individ.,group | async |
| UserManagement | N/A | group,mass | async |
| VideoConference | video | group | sync |
| Voting/Rating | text | group,mass | async |
| Whiteboarding | video | individ.,group | sync |
| Wiki | rich text | ind.,group,mass | async |
| Workflow | N/A | group | async |

A complex business collaboration scenario that uses multiple Web Services defined in the Social Networking SOA standard is a good example of building a composite Web 2.0 application. Figure 2 describes a simplified workflow of the online document editing process. In order to allow synchronous updates of the shared document, the collaboration organizer must perform the following tasks: verify that all participants have user accounts, check user availability, set up a Shared Workspace, notify all participants when the process starts, set up voice and multimedia conference services, use Document Sharing and Instant Messaging for communication and, finally, publish the resulting document using Content Management service.

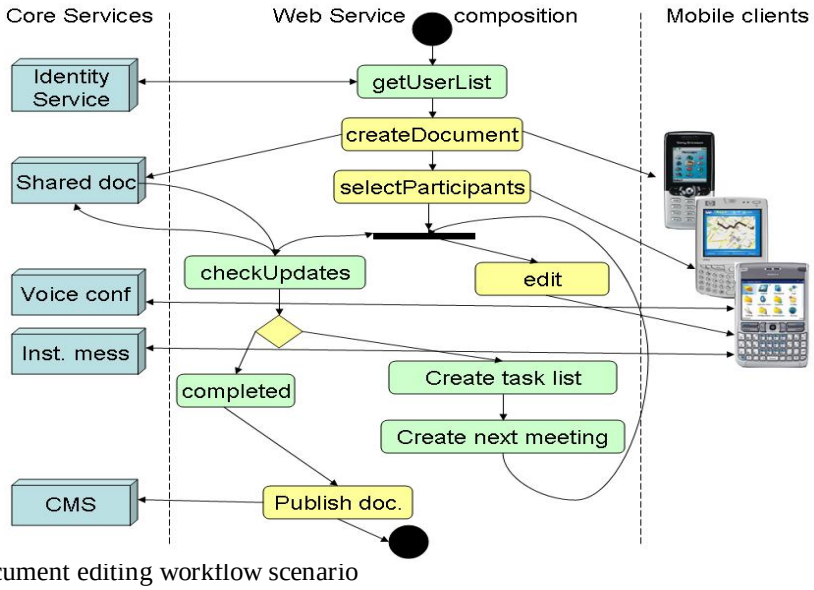

Figure 2. Shared document editing workflow scenario

# Rich Web 2.0 client for Mobile Devices

In this section we suggest an approach to Web service-based Social Networking, taking into consideration various issues discussed in the previous sections. The goal of our research project is to enable SOA-based Social Networking services not only on desktop computers with high-speed internet connection, but also on mobile devices including smart and mass-market cell phones. Unfortunately, we have found that traditional AJAX-based Web 2.0 techniques that decently work on desktop browsers don't work well in mobile environment because of the following limitations: 1) limited connection bandwidth; 2) small screen unable to render full HTML page and inconvenient tiny keypad; 3) limited memory and CPU capacity making interpreting of large JavaScript files extremely slow even on the smart phones; 4) lack of JavaScript support on the majority of cell phones and very limited support on top-level smart phones; 5) verbose nature of XML, SOAP and JavaScript that requires high speed communication bandwidth and significant amount of memory to store large files containing many redundancies; 6) lack of support of such standard cell phone input features as a phone camera, voice, GPS and various Bluetooth-connected sensors by the AJAX-based applications.

In order to overcome all aforementioned limitations, we propose a number of solutions that enable access to complex Web 2.0 applications from a wide range of mobile phones. An efficient context-based XML compression algorithm has been proposed in [18] to reduce the redundancy of XML documents and especially SOAP messages. In addition, it provides encryption of the XML messages compliant with the enterprise security standards. In order to enable rich and Web

2.0 UI on a small screen of the mobile device, we have also proposed an automatic UI reduction solution based on XML tagging and UI pattern transformation.

Unlike the traditional Web, Web 2.0 intensively uses asynchronous communication. AJAX applications use an asynchronous XML message exchange to provide a better user experience. In our proposed architecture the Connection Manager exchanges messages with the server asynchronously, using compressed XML files. To further improve the user experience in online and especially offline modes, the Connection Manager provides an efficient mechanism of proactive caching, where the server sends data to the client before then client actually requests it. This approach allows the client application to function even during continuous disconnected periods. Finally, the most important proposed enhancement for the mobile Web 2.0 front-end client is enabling the multi-modal input forms, using the phone features such as microphone, camera, Bluetooth, etc.

In order to test the suggested methodology, we have implemented and compared three different approaches to the Web 2.0 client framework architecture for the mobile devices. The first approach is a classic AJAX client. We used an OpenAJAX framework and a Safari browser to test the AJAX-based rich client.

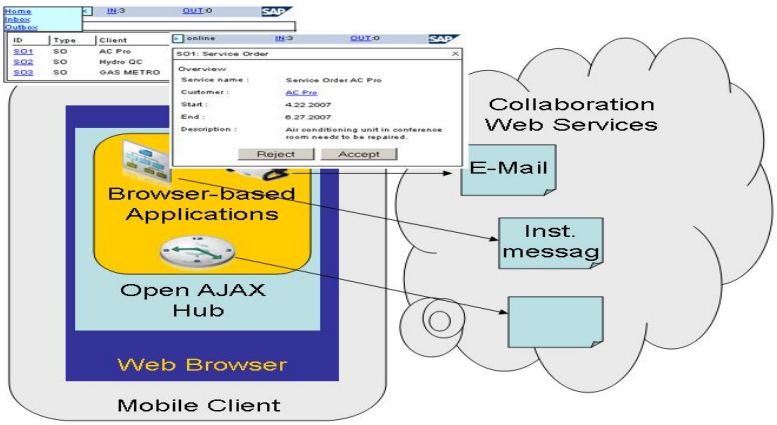


Figure 3. Mobile AJAX client

The main advantage of this approach is that Web 2.0 applications can run on the mobile client without any deployment. The disadvantages include being slow and being suitable only for the high-end smart phones such as Nokia N95. In addition, this approach does not allow for the rendering of complex Web pages. In order to access a complex Web application UI on a small screen, the Web pages must be reduced manually.

The second approach is a J2ME client that runs in the lightweight OSGi container [19]. The client can interpret compressed XML files containing XForms [13] as well as some other popular XML formats. The advantage of this approach is that it works on almost every mobile phone, supporting two-way communication and all of the phone features.

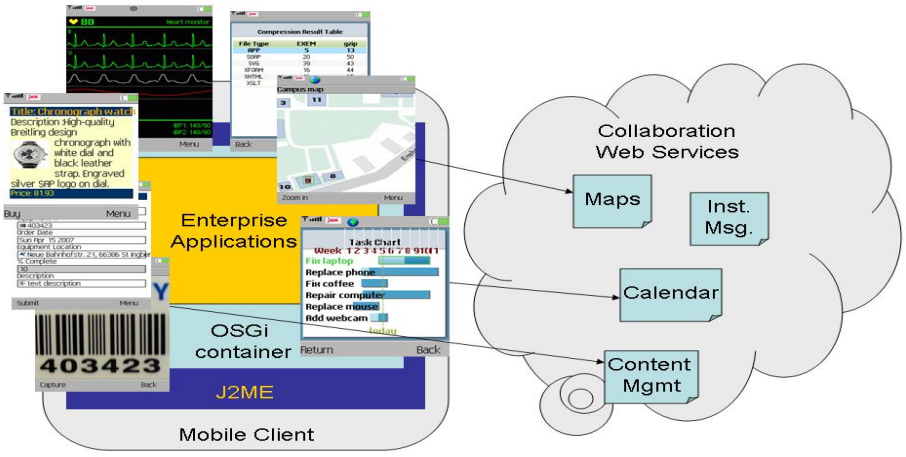


Figure 4. J2ME approach

Finally, the third approach is a combination of the first two. The configuration is based on an OSGi module that contains a Web server. The OSGi module manages the local cache and uses the compressed XML format for the communication with the server. The decompressed and decrypted Web 2.0 pages will then be received by the browser from the local Web server significantly faster.

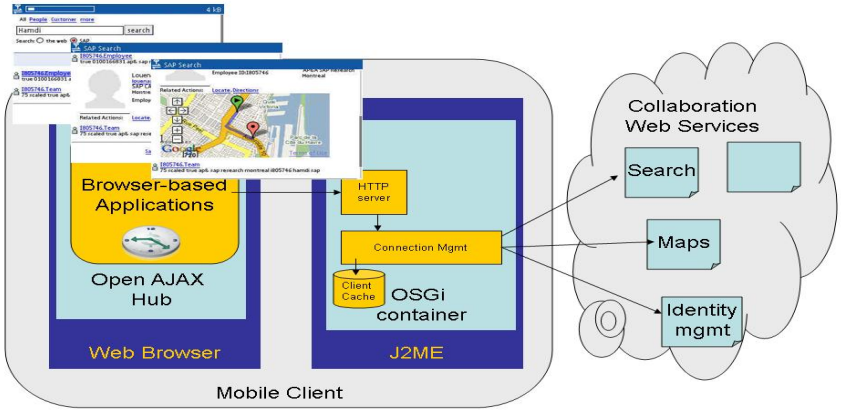


Figure 5. Combined approach

# Conclusion

The typical perception of the Web 2.0 applications is that they should be based on AJAX and online Web technologies. However, in the world of mobile devices this is not always the case. The main differentiation of Web 2.0 from traditional Web is that the Internet is becoming a true runtime platform for applications. Web 2.0 clients use Web Services and XML to communicate with each other. Core Web Service interfaces can have multiple different implementations and comprise composite Web Services, which would in turn implement complex scenarios using BPEL workflow description language.

In this paper we have proposed a mobile application framework based on the Web 2.0 technology, which implements core Web Services for Social Networking functions. Also, we have described several possible architectural solutions for the client side. We have introduced a concept of the Mobile Browser for Web 2.0 that supports two-way communication, information sharing, XML and HTML compression, multi-modal input as well as Social Networking Web Service invocation and composition. Depending on the class of a mobile device and the type of the task, the client application platform can be implemented using either pure AJAX technique or J2ME, or a combination of both. Regardless of the technique used, the framework provides the main advantages of Web 2.0: the distributed connectivity using core and composite Web Services, asynchronous information exchange using XML format, and application-agnostic client platform that interprets applications downloaded from the Web.

The advanced input technologies currently available on mobile devices, such as video, audio and wireless sensor-based input, as well as compatibility with the novel XML standards such as XForms, BPEL, SVG and SOAP, allow for the extension of the traditional definition of a Web browser. The re-defined Browser 2.0 as a platform for mobile Web 2.0 applications will take advantage of the audio, video and sensor features of the phone. It will also support direct invocation of Social Networking Web Services and BPEL4People scripting, in addition to the HTML and JavaScript. Using SOA-based Web 2.0 for building Social Networking applications on the server side and the re-defined Web Browser 2.0 on the client side could potentially enable a new paradigm of collaboration and significantly improve the quality of communication through the Net.

# References


[1] Ecospace IP - eProfessional Collaborative Workspace. http://www.ip-ecospace.org/.

[2] OASIS Open: OASIS web services business process execution language (WSBPEL)TC.

[3] Nokia Corp. Myths of Mobility: New report examines common misperceptions of mobile technology. Press Release Dec. 2005. http://press.nokia.com/PR/200512/1025827_5.html.

[4] Ray Le Maistre, Nokia: It Takes Web 2.0 to Twango. http://www.lightreading.com/document.asp?doc_id=129798.

[5] Urs Mansmann. Unterwegs ins Web. Browser für Handys and Smartphones. C't 13/2007,p86.

[6] http://www.apple.com/safari/.

[7] http://developer.apple.com/iphone/designingcontent.html.

[8] http://www.operamini.com/.

[9] A. Woodruff and P.M. Aoki. Media Affordances of a Mobile Push-To-Talk Communication Service. Computing Research Repository (CoRR) Technical Report cs.HC/0309001, 8/2003.

[10] http://www.cs.fsu.edu/~engelen/soap.html .

[11] Sun Microsystems. J2ME Web Services Technical White Paper, July 2004.

[12] Microsoft Developer Network (MSDN). Consuming Web Services with the Microsoft .NET Compact Framework, March 2003.
[13] http://www.w3.org/TR/xforms/.
[14] http://my.opera.com/dstorey/blog/index.dml/tag/Handheld, http://www.w3.org/TR/REC-CSS2/media.html .
[15] Tim Brners-Lee, The WorldWideWeb browser http://www.w3.org/People/Berners-Lee/WorldWideWeb.html .
[16] WS-BPEL Extension for People (BPEL4People), Version 1.0. http://www.ibm.com/developerworks/library/ specification/ws-bpel4people .
[17] http://www.openajax.org/.
[18] Y. Natchetoi, H. Wu, and G. Babin. A context-dependent XML compression approach to enable business applications on mobile devices. In Proc.of Euro-par 2007.
[19] OSGi, last visit in 2007. http://www.osgi.org/.