

# ◆ Enhanced Active Phone Book Services: Blended Lifestyle Services Made Real!

Ramesh Pattabhiraman, Musa Unmehopa, and Kumar Vemuri

*Typical address book software provides for personalization of the end user service experience and introduces a level of stickiness to a service offering. There are tremendous opportunities to extend this experience into a true blended lifestyle service by enhancing the basic functionality with dynamic and context-sensitive features. Enhanced active phone book services introduce an innovation that offers value-add through cleverly blending various next-generation service enablers and network capabilities into a dynamic address book for the user, spanning wireless and converged networks. This paper outlines the enhanced active phone book value proposition for service providers, paints the standards landscape underpinning the service, and offers design choices for a possible platform architecture that supports the service.*

© 2007 Alcatel-Lucent.

## Introduction

There was a time when users' communications needs were sufficiently met by the use of a single device or a single application. But communications technologies today are evolving at Internet speeds (since they rely on, and, in many cases, tend to leverage advances being made in the Internet domain). So communication savvy end users have come to expect service bundles (groups of services or applications marketed and priced together) from which to choose the application or set of applications that best fits their dynamic communication requirements.

But the possibilities do not end with offering services in bundles. The opportunity to blend services, which are traditionally independent, within such a bundle takes the user experience to the next level. An attractive example of such a blended service is an enhanced active phone book service, which introduces an innovation that offers value-add through cleverly integrating various next-generation service enablers

and network capabilities into a dynamic address book for the user, spanning wireless and converged networks and Internet communities. Enhanced active phone books are the focus of this paper.

This section introduces the address book application as it is used today, and the technology that underpins it. We will then build upon that introduction to outline the enhanced active phone book service value proposition in later sections.

## Address Books Today

As modern consumers in a society thriving on ubiquitous technology, we are all increasingly dependent on electronic communication means to maintain our daily contacts. Address books that store user contacts electronically are becoming a commodity. In fact, many of us maintain and use more than one address book, as typically these are associated with a particular device such as a mobile phone,

### Panel 1. Acronyms, Abbreviations, and Terms

3G—Third generation	PC—Personal computer
3GPP—3rd Generation Partnership Project (GSM evolution)	PDA—Personal digital assistant
3GPP2—3rd Generation Partnership Project 2 (CDMA evolution)	PoC—PTT over cellular
APB—Active phone book	PNA—Presence network agent
AOL—America Online	PS—Present server
API—Application programming interface	PTT—Push-to-talk
BDS—BREW distribution system	RLS—Resource list server
BREW—Binary Runtime Environment for Wireless	SCF—Service capability feature
CDMA—Code division multiple access	SIMPLE—SIP for instant messaging and presence leveraging extensions
DM—Device management	SIP—Session Initiation Protocol
DS—Data synchronization	SMTP—Simple Mail Transport Protocol
DTD—Document type definition	SyncML—A synchronization protocol from the SyncML consortium (now OMA)
GSM—Global System for Mobile Communications	UE—User equipment
HTTP—Hypertext Transfer Protocol	UMTS—Universal Mobile Telecommunications System
IETF—Internet Engineering Task Force	URI—Unified resource identifier
IM—Instant messaging	VoIP—Voice over IP
IMS—IP Multimedia Subsystem	WebDAV—WWW distributed authoring and versioning
IMPS—Instant messaging and presence server	WG—Working group
IP—Internet Protocol	WSP—Wireless Session Protocol
ISC—IMS Service Control Protocol	WV—Wireless Village
J2ME—Java 2 Micro Edition	WWW—World Wide Web
MIC—Mobile IMS extensible client	XCAP—XML Configuration Access Protocol
MSN—Microsoft Network	XDMC—XML document management client
OMA—Open Mobile Alliance	XDMS—XML document management server
OSA—Open service access	XML—Extensible markup language
OTA—Over the air	XMPP—Extensible Messaging and Presence Protocol
PAG—Presence and Availability Group	XPath—XML path language
PAM—Presence and availability management	

a personal digital assistant (PDA), or a laptop computer. With these tools becoming a pervasive part of both our professional and our personal lives, the multiplicity of address books introduces new headaches. Anyone who has ever tried to call someone whose phone number was stored only in the address book on his or her laptop, or who has had to send someone's phone details in an e-mail but first needed to get that number off the phone, or has had to find someone's current e-mail address on the PDA, just to find that it changed recently and that it had only been updated it in the e-mail application, can relate to the challenges multiple disparate address books can pose.

Storing address information in multiple locations, either locally on devices or separately per application, also introduces the issue of data inconsistency. The approach common in the world of database management is to store the information centrally on a network-hosted server. All address information is maintained and stored on the network. Individual devices or applications may access the centralized address information when required. Each device may also keep a local copy of the data, both for performance reasons, as well as for situations when the device is not connected to the network or the centralized store is inaccessible. Changes to the local copy of the address book on a particular device need to be reflected in the centralized

address book. Similarly, changes to the centralized address book need to be propagated to all devices maintaining a local copy. The process of ensuring that local copies remain consistent with the centralized data store is called synchronization. This process guarantees that address information is accurate and up-to-date across wireless devices, voice systems, personal computer (PC) clients, and Web browsers.

Apart from data consistency, synchronization with centralized address books provides additional benefits to the end user. It enhances the end user experience through the convenient management and maintenance of potentially large volumes of data. Convenient desktop user interfaces can be provided, obviating the need for the user to key in contact data on small phone keypads or with PDA styluses. Synchronization offers the principle of “Write once, publish to many,” further adding to the end user experience.

This enhanced end user experience, in turn, stimulates the use and uptake of new services, which may share the use of the centralized address book, effectively using it as a “launching pad” for other applications. Service providers benefit through increased subscriber loyalty and hence reduced subscriber churn. Centralized address books create opportunities for additional value-added services on existing network infrastructure, also contributing directly to new revenue streams from both the existing and the growing (new) subscriber segments.

Synchronization thus is key to centralized address books. Several synchronization technologies are in use to date. The next sections introduce the three most prevalent among them: SyncML\*, WebDAV, and XCAP.

**WebDAV.** Web-based distributed authoring and versioning (WebDAV) [18] is defined by the Internet Engineering Task Force (IETF) as a set of extensions to the Hypertext Transfer Protocol (HTTP) that allows users collaboratively to edit and manage files on remote Web servers [9]. WebDAV adds a number of new verbs to the HTTP protocol vocabulary beyond the GET and POST methods. For example, LOCK and UNLOCK allow users to make exclusive edits on a shared file. WebDAV started out as a protocol for remote collaborative authoring of generic documents

in a Web environment. Of course, one such document could be a centralized address book. Extensions to the WebDAV protocol to support vCard data stored as address books on the server are being proposed in [6]. A major advantage of WebDAV is the fact that it builds on HTTP, and hence support for WebDAV can be built into platforms already supporting HTTP and already hooked up to the Web. Many operating systems include support for WebDAV.

One of the disadvantages is the need to support additional verbs in a generic, lean, and widely adopted and deployed protocol. Also, there are indications that this protocol is growing in complexity as it evolves to support more functionality.

**SyncML.** Where WebDAV was developed as a generic means for collaborative authoring, other initiatives came to light focusing specifically on the issue of synchronization. SyncML, initially developed in the SyncML Forum (which is now subsumed into the Open Mobile Alliance\* [5]), defines a client-server architecture for synchronization. An extensible markup language (XML) representation of the data is shared by a SyncML server and a SyncML client. Several transport bindings are supported, of which HTTP is one. When opting for the HTTP transport binding, the SyncML commands are carried using the normal HTTP GET and POST verbs. That is, no new verbs are required, overcoming one of the drawbacks of WebDAV. Other bindings include the Wireless Session Protocol (WSP), Simple Mail Transport Protocol (SMTP), and local connectivity protocols like Bluetooth\*.

SyncML defines two protocols. The SyncML Representation Protocol specifies extensible markup language (XML) document type definitions (DTDs) for protocol primitives, called commands, and the structure of the data that can be carried [14]. The SyncML synchronization protocol defines the message sequences for a number of synchronization procedures, including two-way synchronization and refresh synchronization from the server [15].

**XCAP.** The SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) Working Group in the IETF has defined the XML Configuration Access Protocol, or XCAP [17]. XCAP allows a client to read, write, and modify application configuration data stored

on a server. As with SyncML, the data are stored in XML format. XCAP selection is based on XML path language (XPath), which defines how to point at specific pieces of an XML document. XCAP then maps XML document sub-trees and element attributes to HTTP universal resource identifiers (URIs), so that these components can be directly accessed by HTTP.

### **Dynamic, Context-Sensitive Features**

As seen earlier, a centralized, network-hosted address book already blends various communications means together. This can be taken even further, by blending in dynamic, context sensitive features (this is, after all, what makes the “active phone book” really “active”). Presence technology can be studied as one example of such a feature.

Presence technology is applied to communications in order to yield process and productivity improvements. Presence aims to eliminate “phone tag,” allowing people to reach one another on their first try. It allows people to hook up promptly with their colleagues, whether on the road, in meetings, or working from remote locations, via a multitude of communication means, enabling them to use the most suitable of those means given the characteristics of their communication context at that time. Presence also helps simplify the often tedious and time-consuming task of scheduling conference calls or team sessions. It allows users more efficiently to choose when and how to communicate, increasing the probability that the person they are trying to reach will respond favorably to a communication attempt.

### **Enhanced Active Phone Book Services**

So far we have talked about service bundles, a multiplicity of devices, centralized and network-hosted address books, and dynamic, context sensitive features like presence. Taken together, these features can all be implemented as an enhanced active phone book—an innovative blended lifestyle service. They add context sensitivity, i.e., presence, to the regular phone books in daily use today. The enhanced active phone book is thus a single address book for multiple devices, multiple applications, and multiple communication means that dynamically adapts to the presence of the end user. This addition of context sensitivity now enables us to incorporate application

launch into the address book. That is, we could enable phone book capabilities that let users select one or more buddies, establish a “session context,” and then chat, or click-to-dial, or invoke other applications. The value proposition of having ubiquitous access to a unified address book that shows buddy groupings, incorporates real-time presence, shows the device capabilities for buddies, allows users to personalize their own presence and availability settings, and on top of that embeds automatic application launch of click-to-call or click-to-instant messaging (IM), is extremely powerful indeed. Such seamless service blending results in message-waiting indications across all user applications. The enhanced active phone book becomes “communications central” across all user devices.

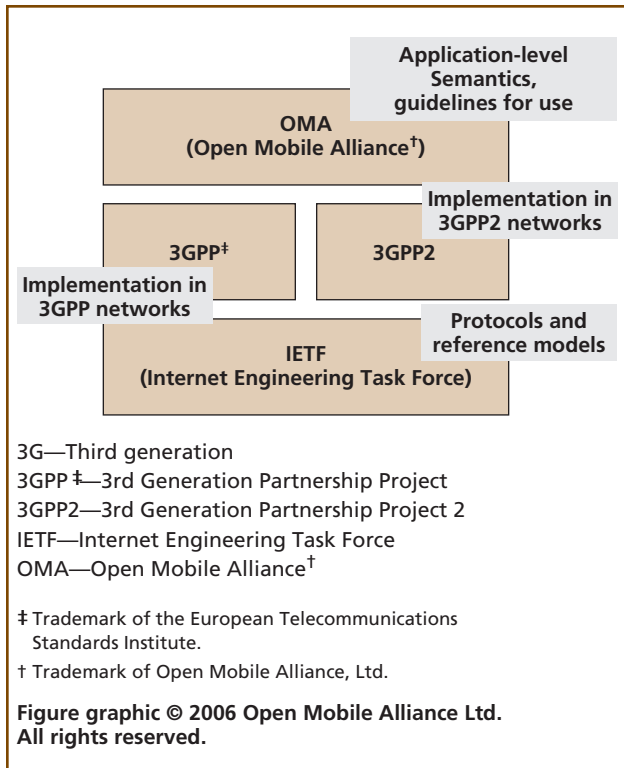
### **Standards Landscape**

At the time of this writing, there is significant interest in presence technology from the perspective of both wired and wireless networks. This is largely due to the widespread usage of services such as instant messaging and push-to-talk or Voice over Internet Protocol (VoIP) chat that have been popularized by companies like Yahoo!\*, AOL\*, and MSN\*. Together, these three companies have over 100 million users worldwide. Though the communities are largely disjoint, a movement is afoot to permit users to interact across these relatively “artificial” technology and domain boundaries so truly seamless communication is possible.

Several standards initiatives are extant in the area of presence; these include specifications published by the Internet Engineering Task Force (IETF) [10], 3rd Generation Partnership Project (3GPP\*) [1], 3rd Generation Partnership Project 2 (3GPP2) [4], and Open Mobile Alliance\* (OMA) [11]. Typically, presence standards tend to focus on three areas:

- The collection and dissemination of presence information and associated processing (including security and related aspects),
- List management, be it group lists or authorization control lists,
- A sample application such as instant messaging (IM) or push-to-talk (PTT).

The Open Mobile Alliance presents a comprehensive view of how the various standards relate to each other. To do so, they use a simple model, which we



**Figure 1.**  
**Standards and their inter-relationships.**

term the *three-layer brick model* because of its structure and appearance. This is illustrated in **Figure 1**.

As can be seen, the lowest layer of this model depicts the IETF standards. Generic requirements for presence and instant messaging protocols have been standardized by the IETF in RFCs 2778 and 2779 [7, 8]. Subsequently, Session Initiation Protocol (SIP) constructs and reference models to fulfill these requirements have been developed in the SIMPLE Working Group (WG). In particular, the SIMPLE WG explains how:

- IETF RFC 3265-defined primitives for event subscription and notification can be used in the presence context, and designs the various types of event packages that can be used effectively to support presence-related services [16].
- A new protocol called XML Configuration Access Protocol (XCAP), a variant of XPath, can be utilized to support list management capabilities.

Similarly, given the prevalence of Jabber\*-based implementations in use today, attempts are being made to standardize and extend that popular protocol

within the Extensible Messaging and Presence Protocol (XMPP) WG of the IETF, again trying to fulfill the requirements laid out by the two RFCs mentioned in the process. XMPP is typically viewed as supporting pre-Internet Protocol Multimedia Subsystem (IMS) implementations.

As can be seen from the preceding discussion, these IETF protocols and reference models form a fertile basis for implementation-related work to be carried out in particular network contexts. Most importantly, the IETF standards define basic vocabulary including the terms *presentity*, an entity whose presence information is of interest to other entities that subscribe to the service, and *watcher*, an entity that watches presentities for their presence updates.

The 3GPP and 3GPP2 components constitute the next layer of the standards model. These standards bodies respectively define how the IETF standards can be leveraged to build conformant implementations in GSM\*- and CDMA-evolved 3G networks. Each of these bodies has defined particular requirements and constructs for how presence can be supported within IMS architectures common to both. Also, 3GPP2 has agreed to adopt 3GPP references with minor variations to its own architectural context, thereby facilitating reasonable reuse across implementations. The fact that 3GPP and 3GPP2 both adopted IMS service control protocol (ISC), a variant of SIP, as their signaling protocol of choice only makes the IETF SIMPLE work even more relevant within the 3GPP/3GPP2 context.

While talking about 3GPP and 3GPP2 definitions of presence, passing mention must also be made of the open service access (OSA) specifications defined within the umbrella charter of those bodies. These OSA application programming interfaces (APIs) include a service capability feature (SCF) for presence and availability management (PAM) in section 14 of 3GPP specification TS 29.198-14 [3]. These APIs provide programmatic constructs for application developers to build applications that interface to IMS elements such as Parlay/OSA gateways in seamless ways, thereby leveraging the confluence of Internet and telecommunications technologies for more rapid and profitable service creation and deployment.

Finally, the uppermost layer in our three-layer brick model is the Open Mobile Alliance (OMA).

This body defines specifications in its Presence and Availability Group (PAG) that outline user scenarios and usage models for protocols, reference models, and architectures from lower layers. The OMA encourages the use of XCAP for list management and SIMPLE for presence collection and delivery. However, in addition to these components referenced within the PAG Working Group, OMA has technical affiliate standards with some functional overlap—inherited for the most part from other bodies that were later subsumed under the OMA umbrella. This sometimes leads to confusion in stating OMA compliance, though products typically address this by indicating the particular standards to which they build. In particular, these “supplemental” OMA presence references include:

- *SyncML*. This protocol, discussed earlier as an example of a synchronization protocol widely used today, is defined to support two reference points in the OMA model, namely, data synchronization (DS) and device management (DM). The former function can also be achieved equally well through the use of the XCAP protocol—an overlap that is not particularly well addressed in the OMA standards. However, market realities dictate that vendors support both mechanisms for synchronization for at least the near future. SyncML has been in use for some time now, though outside the presence context. In addition, SyncML clients already are preintegrated with most handset software today.
- *IMPS/WV*. The Wireless Village\* instant messaging and presence service is in wide use today primarily in pre-IMS deployments of presence solutions in cellular networks. This is akin to IETF XMPP in scope and the way it relates to the overall SIMPLE-based presence architecture. The next section offers design choices with passing references as to how these pre-IMS technologies can be integrated into address book solutions.

### Design Choices and Platform Architecture

While the previous section provided an overview of the standards that are relevant to presence, design decisions for the presence solution architecture also merit some consideration and they are addressed in the following.

### The OMA Presence Model

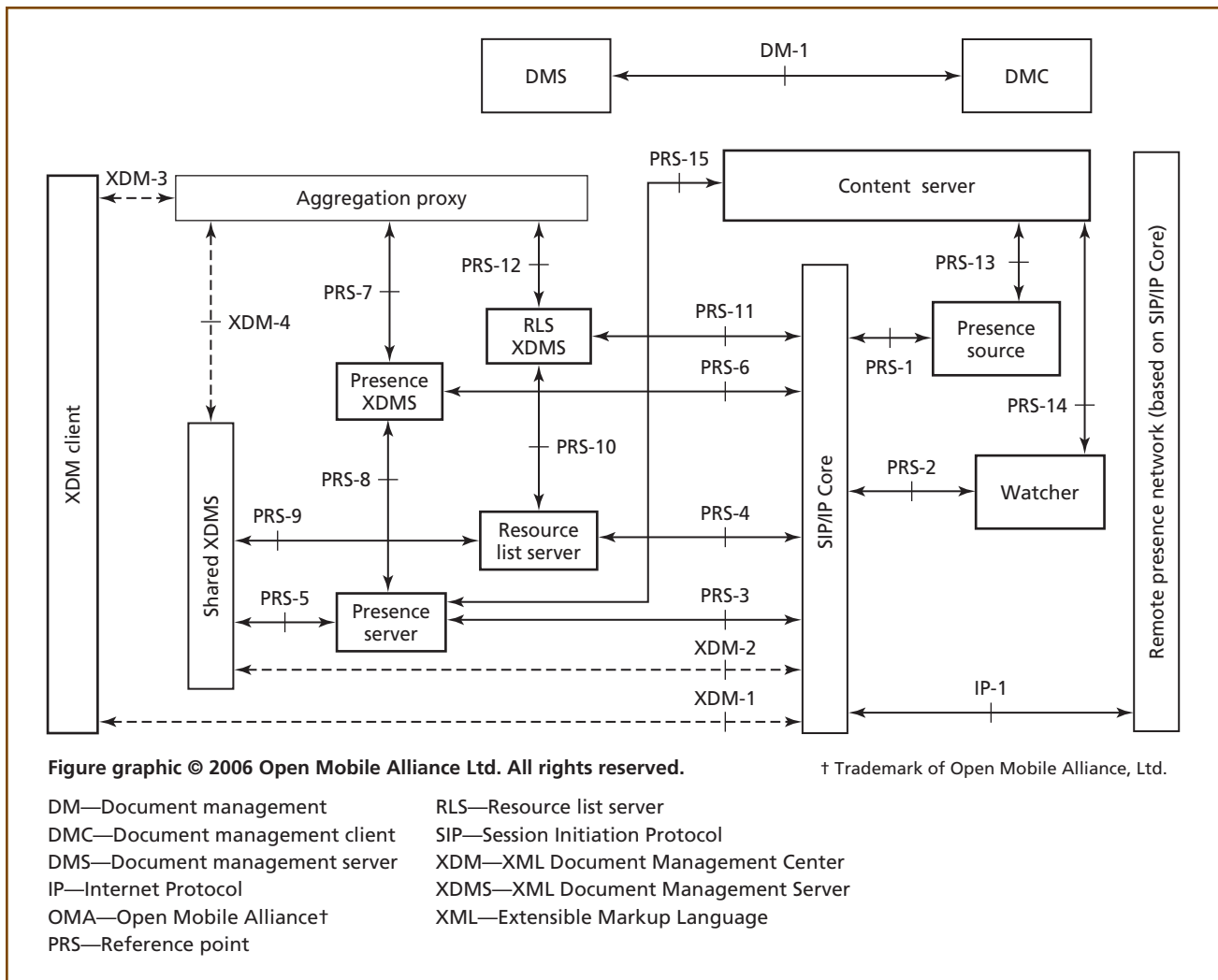
The OMA presence model [12] is depicted in **Figure 2**. As can be seen from this model, there are several important functional elements could be a list:

The XML document management server (XDMS) and XML document management client (XDMS), respectively, constitute the XML document management server and the XML document management client. The former receives client queries and updates relating to list management operations and maintains up-to-date versions of user data (such as buddy lists or presence authorization rules [19]) and provides appropriate responses to the client in each case. The XCAP protocol is used to support this communication, and the standard borrows mechanisms such as etags and conditional headers from HTTP to facilitate and optimize these interactions.

The standard encourages the support of multiple XDMSs, with each potentially tailored to a specific function. Accordingly, there are XDMSs that may be shared, e.g., those that hold presence data (presence XDMS) or push to talk over cellular (PoC XDMS). The reference architecture is extensible; that is why provision is made for an “enabler XDMS” that supports a generic “enabler server.” In order to provide for routing capabilities such that the XCAP messages reach the appropriate XDMS in each case, a component called the aggregation proxy is included. This element performs common authentication, routing, and potentially billing functions for XCAP transactions.

Obviously, collection and dissemination of presence information require a presence server, which is indicated by the reference architecture as well. This forms the central component and may be extended to interface to various network sources of presence in addition to supporting user sources or other external sources (e.g., sources derived from other partner network components). The architecture is rather flexible and open in this regard.

Next, there is a resource list server (RLS) component which essentially enables a user to subscribe to multiple “resources” (read buddies) through a single subscription and then receive updates on their presence all together in collated notifications whenever appropriate. The RLS factors in privacy and authorization



**Figure 2.**  
**OMA presence architecture.**

rules as it delivers information to end users. In addition, the RLS may support advanced features like sophisticated presence composition rules, watcher filters, rate limiting, and traffic throttling to regulate the flow of information to interested watchers in each case.

Finally, there is a content indirection server. The reference architecture intends for this to be a repository of changed lists, especially if users can update their lists from multiple devices. In such situations, the notification of list updates can be sent to all the user's subscribed devices, with a pointer to the updated content. The user's device has the option (based on either manual intervention or programmed logic) to download the updated lists for future use.

The OMA reference model is undergoing some minor changes as of this writing, but the functions described previously continue to be supported.

### Solution Architecture Considerations

There are a few critical architectural considerations for building a conformant implementation of an active phone book service. Such an analysis needs to be looked at from two perspectives—a server aspect and a client aspect. A server could encapsulate all of the network functions, namely, the RLS, XDMSs, the aggregation proxy, the content (indirection) server, and the presence server. Similarly, the client might implement multiple functions including the XDMC,

various presence-enabled applications, and perhaps even a common framework that forms the basis of interactions with network components that provide all the support functions. In this section, we consider only the questions of importance; solution details and answers are not presented because these tend to be competitive differentiators and closely guarded secrets, and some decisions depend on previous ones made within the architectural context.

From a server perspective, the most important considerations for an efficient implementation need to address:

- *Support for particular event packages.* This could be one or more of the following: the presence event package (required) for dissemination of presence information, the winfo event package to report changes to logged-in watchers for each presentity, the SIP profile event package used to report changes in user lists across devices, or the regulate publish event package that regulates the frequency of publications to offer better efficiencies of scale in communications.
- *Potential security and privacy issues.* People tend to be rather wary of watchers and worry whether their address book information can be stolen or misused. The standards provide for four dispensations users can use to control access to their presence content. These include allow, deny, polite-block, and confirm. Allow and deny dispositions are self-explanatory. Polite-block enables a presentity to block a watcher while appearing to have authorized the watcher but presenting only the unavailable state to that watcher. Confirm is a construct whereby a presentity is notified of a new watcher's attempting to watch the presentity's presence information and has the ability proactively to authorize that watcher at that time.
- *Scalability and availability.* A centralized solution designed for a network as a whole or even one that spans networks is relatively useless if parts fail repeatedly, thereby either denying service to a subset of the population or making presence of some subset of entities unavailable to interested subscribed watchers.

- *Performance and optimizations.* This is of particular reference to presence implementations, especially so given that the tension between the accuracy of reported presence information, the necessity to provide more frequent updates, and the necessity of keeping network traffic within reasonable bounds must be resolved.

From a client-side view, the most important considerations include:

- *Client platform.* Identification and selection of one that is most prevalent to maximize the take rate and market penetration of the deployed service. Several handset platforms are prevalent today—these include J2ME\*, BREW\*, Windows\* Mobile PocketPC, and Symbian\* among others.
- *Support for content and application distribution infrastructure.* This is important because it determines how handset software may be provisioned, and whether this needs to be done at the point of sale or over the air (OTA provisioning). This later architecture is inherently supported by the BREW distribution system (BDS) in the BREW context and through the Java\* vending virtual machine architecture for J2ME terminals. Depending on the overall architecture considerations, this may drive the choice of handset platform as well.
- *Completeness of support for client capabilities.* This incorporates data on the number of server supported capabilities that can be used and the ways they can be used. Typically, to ensure reusability of these clients across network architectures, client and server capabilities could be insulated to the extent possible.

## Deployment Architectures in Various Network Contexts

So far, the paper has addressed the standards landscape as well as design and architecture factors for active phone book services. The next section will cover deployment considerations for various network environments.

### 3G Evolution

Third generation networks offer higher capacity, higher data rates, and new service capabilities for

mobile operators when compared to second generation networks. 3G networks leverage IP-based protocols such as SIP to offer an “always on” user experience. The 3GPP presence specification [2] in IMS leverages the SIP/SIMPLE specifications for presence, group list management, and instant messaging architecture. These specifications define different roles and functional entities such as the resource list server (RLS)—a presence list server that accepts SIP SIMPLE-based subscriptions based on resource lists and provides notifications to interested watcher applications of the state of resources in the resource list. RLS serves subscriptions to presence, winfo, and SIP profile event packages. The resource lists themselves are manipulated via XCAP servers which are used to store and modify groups and resource lists, and specify authorization policies such as allow, deny, confirm, and polite-blocking. In the IMS architecture, the presence server, RLS, and presence network agents (PNAs) may be implemented and deployed as one or more application servers. As OMA, 3GPP, and 3GPP2 each utilize the same underlying IETF SIMPLE-based protocols (XCAP and SIP), these application servers and XCAP servers can be adapted to function as OMA enabler components.

### Deployment Contexts

Three high-level deployment contexts can be considered for a centralized network-based active phone book solution.

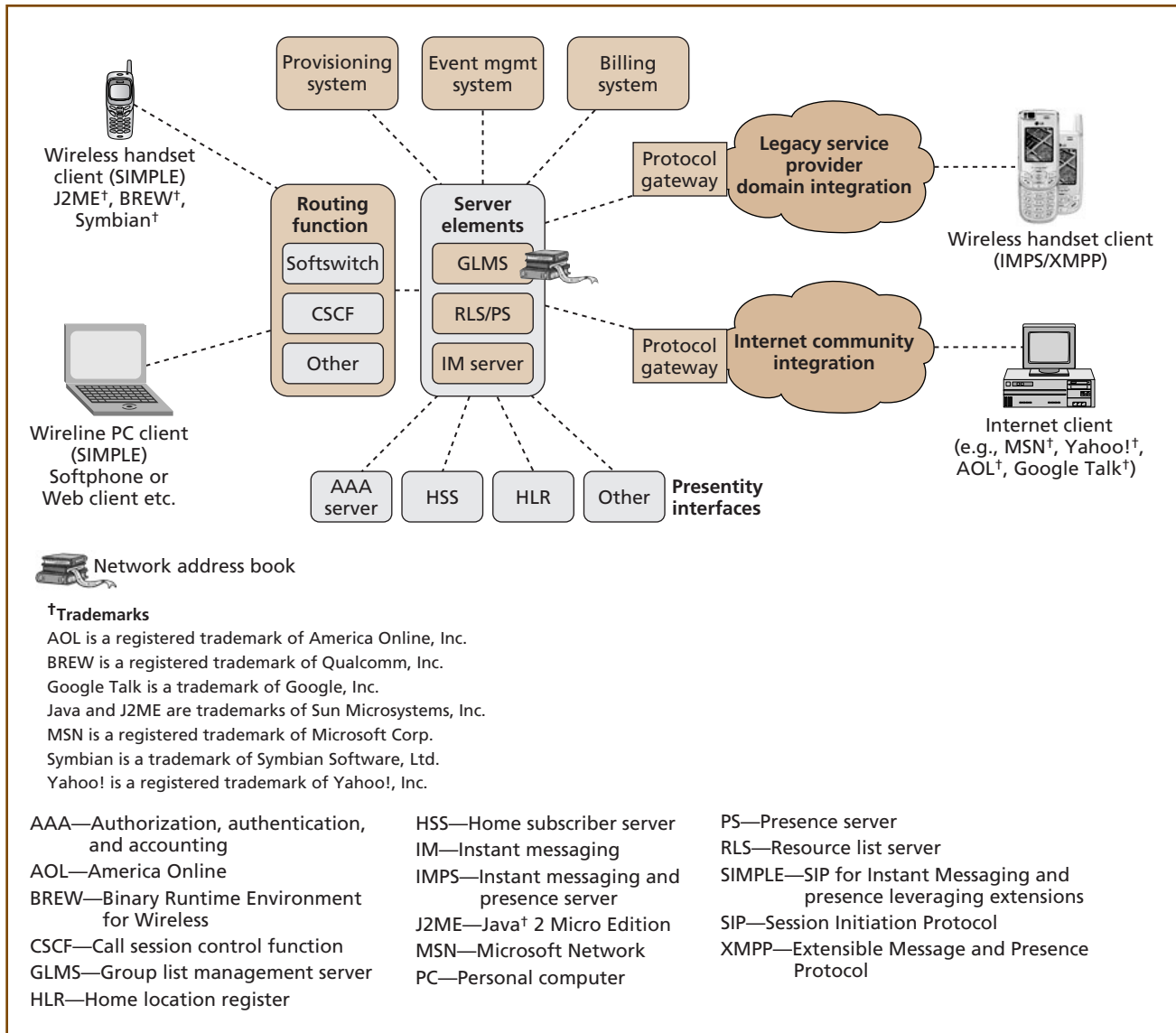
**Pre-IMS Solution.** These are pre-3G wireless networks or segments, such as those prior to the integration of IMS components. XMPP and Wireless Village are two prevalent presence, list management, and instant messaging technologies available in this network. Supporting pre-IMS presence is often an important consideration for operators who are many times interested in blending together user experiences for subscriber segments using legacy solutions with those subscribing to their newer IMS-based infrastructure and services. With network evolution to IMS, challenges in the areas of architecture integration, interoperation, and migration (if subscribers want to change segments) need to be addressed. Interoperation is largely achieved through the use of

gateways that convert from the pre-IMS legacy protocol to SIMPLE in IMS and vice-versa. **Figure 3** shows a network view of an enhanced address book solution deployment

**Non-IMS Solution.** Wireline VoIP network contexts typically include softswitches, session border controllers, and media gateways. The IETF SIMPLE technologies provide the foundations for offering centralized presence-enabled address book solutions in non-IMS networks as well. Here, the challenges faced while deploying a centralized presence solution are slightly different. The predominant issues here include disparities and differences in contextual architectures. Each wireline network may be set up slightly differently, with the differences being significant enough to warrant changes in presence deployment architectures around such aspects as message routing, security mechanisms in use, identity management, addressing and aliasing support, registration management, and maintenance of device, user, identity and service relationships.

**IMS Solution.** The IMS deployment context is based on well-defined access-independent specifications offering interoperability between various components of a presence-enabled XDMS and instant messaging solution. IMS solutions have various flavors: GSM/UMTS\*, CDMA-based, or wireline IMS. As standardized presence and instant messaging IMS services (specified by 3GPP/3GPP2) are integrated within OMA, a single presence solution built to these standards can be deployed transparently in most IMS network contexts. In IMS, XCAP/HTTP is specified for the  $U_i$  reference point and is used as the group list management interface, while ISC (SIP with IMS extensions) is used for presence publication, refresh, and notification functions. For security, user equipment is required to register with the IMS network prior to utilizing any of the address book, resource list, and presence services.

As observed earlier, service providers desire solutions that apply equally well to, or somehow blur, the user experience across IMS and non-IMS presence contexts with convergence of presence across wireline, wireless, and Internet domains. A seamless integration of contact lists, address books, presence,



**Figure 3.**  
**A network view of an enhanced address book solution deployment.**

and availability across these environments is required, along with the integration of Internet presence and instant messaging communities such as Yahoo!, MSN, and AOL. (This is commonly referred to as community integration.)

**Alcatel-Lucent Active Phone Book Solution**

The Alcatel-Lucent Presence Solution includes the XDMS, RLS, and presence server components. These components in concert with the Alcatel-Lucent Mobile IMS Extensible Client (MIC) provide a complete end-to-end centralized address book and resource list

management solution. The MiLife<sup>®</sup> Presence Solution also offers a centralized event state composition and PNA functions that support a variety of presentity and watcher interfaces. These components of the Alcatel-Lucent Presence Solution are poised to address non-IMS, pre-IMS, and IMS deployment contexts.

**Conclusions**

As mobile communications gain more prevalence, and as VoIP and other technologies more seamlessly tie into the communications fabric, the integration of presence (previously relegated to the Internet

domain) and the support for a single unified address book across devices and services is inevitable. This demand for a technical solution from users who want more convenience gives operators a means to tap into a new revenue stream while incorporating presence into address book technology. Standards drive interoperable implementations.

### Acknowledgements

The authors would like to thank the technical reviewers for their valuable comments and input.

### \*Trademarks

3GPP is a trademark of the European Telecommunications Standards Institute.

AOL is a registered trademark of America Online, Inc.

Bluetooth is a registered trademark of the Bluetooth Special Interest Group.

BREW is a registered trademark of Qualcomm, Inc.

GSM and Globe System for Mobile Communication are registered trademarks of the GSM Association.

Jabber is a registered trademark of Jabber, Inc.

J2ME and Java are trademarks of Sun Microsystems, Inc.

MSN is a registered trademark of Microsoft Corp.

Open Mobile Alliance is a trademark of Open Mobile Alliance, Ltd.

Symbian is a trademark of Symbian Software, Ltd.

SyncML is a trademark of Open Mobile Alliance, Ltd.

UMTS is a trademark of the European Telecommunications Standards Institute (ETSI) under auspices of the ITU.

Windows is a registered trademark of the Microsoft Corporation.

Wireless Village is a trademark of Open Mobile Alliance, Ltd.

Yahoo! is a registered trademark of Yahoo!, Inc.

### References

- [1] 3rd Generation Partnership Project, <<http://www.3gpp.org>>.
- [2] 3rd Generation Partnership Project, "Presence Service; Architecture and Functional Description; Stage 2," 3GPP TS 23.141, <<http://www.3gpp.org/ftp/Specs/html-info/23141.htm>>.
- [3] 3rd Generation Partnership Project, "Part 14: Presence and Availability Management (PAM) Service Capability Feature (SCF), (Release 6)," 3GPP TS 29.198-14, V6.3.1 (2006-07), Jul. 17, 2006, <<http://www.3gpp.org/ftp/Specs/html-info/29198-14.htm>>.
- [4] 3rd Generation Partnership Project 2, <<http://www.3gpp2.org>>.
- [5] M. R. Brenner, M. L. F. Grech, M. Torabi, and M. R. Unmehopa, "The Open Mobile Alliance and Trends in Supporting the Mobile Services Industry," *Bell Labs Tech. J.*, 10:1 (2005), 59-75.
- [6] C. Daboo, "vCard Extensions to WebDAV (CardDAV)," IETF Internet Draft, Oct. 16, 2005, <<http://cidr-report.org/ietf/all-ids/draft-daboo-carddav-00.txt>>.
- [7] M. Day, S. Aggarwal, G. Mohr, and J. Vincent, "Instant Messaging/Presence Protocol Requirements," IETF RFC 2779, Feb. 2000, <<http://www.ietf.org/rfc/rfc2779.txt?number=2779>>.
- [8] M. Day, J. Rosenberg, and H. Sugano, "A Model for Presence and Instant Messaging," IETF RFC 2778, Feb. 2000, <<http://www.ietf.org/rfc/rfc2778.txt?number=2778>>.
- [9] Y. Goland, E. Whitehead, A. Faizi, S. Carter, and D. Jensen, "HTTP Extensions for Distributed Authoring—WEBDAV," IETF RFC 2518, Feb. 1999, <<http://www.ietf.org/rfc/rfc2518.txt?number=2518>>.
- [10] Internet Engineering Task Force, <<http://www.ietf.org>>.
- [11] Open Mobile Alliance, <<http://www.openmobilealliance.org>>.
- [12] Open Mobile Alliance, "Presence SIMPLE," Architecture Document, Candidate Version 1.0, OMA-AD-Presence\_SIMPLE-V1\_0-20060110-C, Jan. 10, 2006, <[http://www.openmobilealliance.org/release\\_program/docs/PresenceSIMPLE/V1\\_0-20060110-C/OMA-AD-Presence\\_SIMPLE-V1\\_0-20060110-C.pdf](http://www.openmobilealliance.org/release_program/docs/PresenceSIMPLE/V1_0-20060110-C/OMA-AD-Presence_SIMPLE-V1_0-20060110-C.pdf)>.
- [13] Open Mobile Alliance, "Presence SIMPLE Requirements," Approved Version 1.0, OMA-RD-Presence\_SIMPLE-V1\_0-20060725-A, Jul. 25, 2006, <[http://www.openmobilealliance.org/release\\_program/docs/PresenceSIMPLE/V1\\_0-20060725-A/OMA-RD-Presence\\_SIMPLE-V1\\_0-20060725-A.pdf](http://www.openmobilealliance.org/release_program/docs/PresenceSIMPLE/V1_0-20060725-A/OMA-RD-Presence_SIMPLE-V1_0-20060725-A.pdf)>.
- [14] Open Mobile Alliance, "SyncML Representation Protocol, Data Synchronization Usage," Candidate Version 1.2, OMA-SyncML-DataSyncRep-V1\_2-20060316-C, Mar. 16, 2006, <[http://www.openmobilealliance.org/release\\_program/docs/DS/V1\\_2-20060316-C/OMA-TS-SyncML-DataSyncRep-V1\\_2-20060316-C.pdf](http://www.openmobilealliance.org/release_program/docs/DS/V1_2-20060316-C/OMA-TS-SyncML-DataSyncRep-V1_2-20060316-C.pdf)>.
- [15] Open Mobile Alliance, "DS Protocol," Candidate Version 1.2, SyncML Data Sync Protocol, OMA-TS-DS\_Protocol-V1\_2-20060316-C," Mar. 16,

2006, <[http://www.openmobilealliance.org/release\\_program/docs/DS/V1\\_2-20060316-C/OMA-TS-DS\\_Protocol-V1\\_2-20060316-C.pdf](http://www.openmobilealliance.org/release_program/docs/DS/V1_2-20060316-C/OMA-TS-DS_Protocol-V1_2-20060316-C.pdf)>.

- [16] A. B. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification," IETF RFC 3265, June 2002, <<http://www.ietf.org/rfc/rfc3265.txt?number=3265>>.
- [17] J. Rosenberg, "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)," IETF Internet Draft, May 5, 2006, <<http://www.ietf.org/internet-drafts/draft-ietf-simple-xcap-11.txt>>.
- [18] WWW Distributed Authoring and Versioning (WebDAV), <<http://www.ietf.org/html.charters/webdav-charter.html>>.
- [19] J. Rosenberg "Presence Authorization Rules," IETF draft Simple Presence Rules, Oct. 2006, <<http://www.ietf.org/internet-drafts/draft-ietf-simple-presence-rules-08.txt>>

*(Manuscript approved August 2006)*

*RAMESH V. PATTABHIRAMAN is a member of technical*



*staff in the Convergent Business Group at Alcatel-Lucent in Columbus, Ohio. He was recently responsible for the architecture of Presence and GLMS solution and is currently responsible for the architecture of a service interaction solution. His interests include service mediation gateways, Web services, Session Initiation Protocol (SIP), and open-programmable communication networks. He has co-authored several papers and pending patents. Mr. Pattabhiraman holds an M.S. degree in pure mathematics from Kent State University in Ohio.*

*MUSA UNMEHOPA is a distinguished member of*



*technical staff within the office of the CTO of Alcatel-Lucent's Convergence Business Group. He has held various standards leadership positions, and currently serves as the chairman of the Architecture Working Group of the Open Mobile Alliance (OMA). He has published various papers in technical journals and conferences, and has co-authored the book "Parlay/OSA: From Standards to Reality". M. Unmehopa has one patent and a dozen patents pending in the area of service mediation and service delivery. He holds an M.Sc. degree in computer science from the Technical University of Twente in The Netherlands.*

*KUMAR V. VEMURI is a member of technical staff in*



*the Convergence Business Group at Alcatel-Lucent in Naperville, Illinois. Mr. Vemuri currently consults as an architecture subject matter expert for several projects, and most recently he has been actively involved in the architecture and design of the Alcatel-Lucent Presence and GLMS solutions. He has authored and/or co-authored several papers for journals and conferences and a technical book and has several patents pending in the areas of Internet technologies and telecommunications. Mr. Vemuri holds an M.S. degree in computer science from the University of Cincinnati in Ohio. ♦*