Adding Location-Based Services Support to the Mobicents Service Delivery Platform

Ndakunda Shange-Ishiwa, Madeleine Wright and Alfredo Terzoli
Department of Computer Science
Rhodes University, Grahamstown, South Africa
Tel: +27 46 603 8291, Fax: +27 46 6361915
email:G05n5057@campus.ru.ac.za, {a.terzoli, m.wright}@ru.ac.za

Abstract- Research has indicated that there has been a steady increase in the number of location-aware devices and applications and that this trend is likely to continue well into the future. The private and secure transmission of geographical data over SIP/IP core networks is therefore seen as adding significant value to traditional VOIP services. Towards this end, standardization efforts by the Internet Engineering Task Force (IETF) and the Open Mobile Alliance (OMA), amongst others, have resulted in the development of several specifications and protocols to guide the implementation of location-based services on the internet. Despite this fact there is still a lack of tools to support developers in implementing these services for SIP/IP core networks, especially within Service Delivery Platforms (SDPs) like Mobicents (which is an open source VoIP platform). Such tools can provide abstract environments that mask the underlying complexity of the services while still exposing the necessary interfaces to service developers. In this paper we discuss and critique the specifications and protocols proposed by the above-mentioned standardization bodies. Thereafter, we propose and discuss the implementation of location-based services architecture for the Mobicents SDP.

Index Terms—Location-based Services, Mobicents, GEOPRIV, HELD, LOCSIP

I. INTRODUCTION

As market research by Gartner has shown, location-based services and GPS-aware mobile devices became more commonplace from 2008 and this is a continuing upward trend [8, 13]. More and more users want applications that connect the dots between their location and their activities while still preserving their privacy over the internet. Although no one knows what the next *killer application* might be, it most probably will have location-based elements. Clearly, then, it is vital to integrate and incorporate support for location information processing in next-generation network development environments to give a geographical dimension to converged network applications. Location data can be utilized and processed in existing SIP/IP core networks to enrich the experience of consumers while in the process adding more value [4]. This will essentially result in building services that are aware not only of traditional user status and availability but also of their geographical context.

Standardization efforts by the Internet Engineering Task Force (IETF), in tandem with the child working group known as the Geographical Location/Privacy (GEOPRIV), have resulted in the release of specifications that detail the data formats and protocols for implementing location-based services on SIP/IP core networks. From the work of this group has emerged an XML document format that bears location information as it is exchanged between Location LG, Location Servers (LS) and Location watchers (LW). This document format reuses the Presence Information Document Format (PIDF) which is used to carry presence data in SIP presence systems [1]. At the application level, any protocol that can carry XML mime types may carry the modified PIDF document. IETF mainly proposes the use of the Session Initiation Protocol (SIP) for this purpose, while the GEOPRIV specifications suggest the Hyper Text Transfer Protocol (HTTP). When using SIP, IETF does not indicate the use of any event package to manage the interaction between entities exchanging these messages. For this project, however, we investigate the usage of the SIP presence event package for the carriage of location data.

In addition to the IETF and GEOPRIV standards, the Open Mobile Alliance (OMA) has also released a specification which is complementary to the IETF specifications [4]. It also incorporates elements from other specifications produced by The 3rd Generation Partnership Project (3GPP) standardization body. It has a more practical market-oriented focus on aspects related to Quality of Service (QoS), billing and security. The OMA Location for SIP/IP Core Networks (LCOCSIP) specification, as it is called, also utilizes the same document format as IETF and similarly uses SIP as the *using* protocol. The major difference between this specification and that of the IETF is that the OMA LOCSIP proposes the use of a new event package that is similar to the SIP presence event package but is solely used for location.

Even though there are all these specifications and standards for privately conveying location information, there is still a lack of reusable tools and extensible services that partially or fully implement them in convergence environments. The specifications also require a clear and concise understanding of different protocols and technologies such as HTTP, SIP, XML Messaging and Presence Protocol (XMPP) and XML Application Configuration Protocol (XCAP), amongst many others. There is a lack of location servers and interfaces that expose geographical location to SIP/IP core services, and subsequently to customers.

In this paper we discuss and expose the details around the afore-mentioned standards and their integration into the Mobicents JAIN-SLEE (Java Advanced Intelligent Networks Initiative - Service Logic Execution Environment) Service Delivery platform. JAIN-SLEE is a java standard that supports the development of telecommunications

1This work was undertaken in the Distributed Multimedia Center of Excellence at Rhodes University, with financial support from Telkom SA, Comverse, Stortech, Tellabs, Mars Technologies,Bright Ideas 39 and THRIP

applications. Based on the analysis, we resort to SIP and HTTP as partner protocols for carrying location information, while utilizing the presence event package for the management of interactions between entities. Thereafter we propose an architecture and discuss its implementation in the SDP of choice - Mobicents.

A. The Presence Information Document Format

To carry geographical information on the internet, an XML document format that reuses the Presence Information Document Format (PIDF) is specified by the GEOPRIV society. This follows on from the fact that the secure transport and communication of location data over the internet is essentially the same as that of the secure transport and communication of presence information [1]. Presence provides real-time information about the presence or availability status of a presence entity (also known as *presentity*). Similarly, location entities need a document format that is suitable for carrying real-time geographical and security information to other entities in the network. This document format can therefore be extended to either carry location information only or to convey both location and presence information simultaneously. Following IETF terminology, the geographical data is encapsulated within a *location object*.

The PIDF document constitutes the *presentity's* Uniform Resource Identifier (URI) and an optional contact URI to provide means of contacting the *presentity*, usually using a SIP address [3]. The *presentity's* URI is utilized in location-based systems to identify the LG that composed the document. As with the XML elements, the document contains an optional *timestamp* element for indicating when the document was composed, as shown in figure 1. This particular element is, of course, reusable for time-sensitive location-based services. The actual presence information is contained within the *tuple* element which has an *id* tag to identify a particular *tuple* element from other *tuple* elements that might be contained within the same document (in the diagram this is set to *sg89ae*). This comes in handy in situations where a *presentity* or an LG has many sets of presence or location information to share.

Within the *tuple* element is a *status* element that holds the real-time information about the availability of the *presentity*. This is not shown in figure 1, as it is tailored for location only. The *basic* child element contains, in very broad terms, the presentity's presence disposition. It is either *open*, in which case the presentity is available for communication, or *closed*, which means it is unavailable [1, 3, 14]. The *status* element, by design, is extensible and it is through this property that other more informative presence document formats like Rich Presence Information Document (RPID) are added. The *location object*, in like fashion, can be added to the PIDF document to give the *status* element location-bearing capabilities [1]. The new document format is then called the PIDF-LO document to reflect this addition.

II. THE LOCATION OBJECT, THE GEOGRAPHICAL MARKUP LANGUAGE AND CIVIC LOCATION

The *location object* is the XML document format consisting of elements that hold geographical information and the privacy constraints that govern its disclosure. The overarching element of this object is the *geopriv* element which must contain one *location-info* element, as shown in the document in the figure. This is the actual location information container [1]. The real-time geographical information within it is either represented in the Geographical Mark-up Language (GML), or in the Civic Location format. GML is for geodetic data and is expressed using a coordinate-based system. The Civic Location format presents the data with human readable physical addresses. To elaborate more, the location can be expressed by listing the country, the region or province, the city, the building, the floor and so on. However, the document in figure 1 only carries geodetic information.

GML, simply put, is a mark-up language for representing the geographical and geometrical information about physical objects [4]. The full GML schemas are large in size and most of the time support features and geometries that are not required by applications using them. For this reason, GML is originally structured to be used within *application schemas*. These are essentially schemas orchestrated for a particular application. They both limit GML to what is applicable and provide application-specific data types. For this particular problem of conveying location information within a *location object* of a PIDF document, a schema is defined as well.

The schema limits the GML to only a few shapes and custom elements that might be truly useful in this context. These include plain points (as the one used in the document in figure 1); two-dimensional polygons, circles, ellipses and arc bands; and three-dimensional spheres, ellipsoids and prisms. All of these shapes can, of course, be used to demarcate areas of interest or location communicated by location entities. They can also be used for location privacy purposes such as reducing the granularity of provided information and spatial conditions that apply when an LG is in a certain area. As mentioned earlier, however, location privacy is not within the scope of this paper, even if it is part of the overall research project.

```
<?xml version="1.0" encoding="UTF-8"?>
 <presence xmlns="urn:ietf:params:xml:ns:pidf"
   xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
   xmlns:gml="urn:opengis:specification:gml:schema-xsd:feature:v3.0"
   entity="pres:geotarget@example.com">
 <tuple id="sg89ae">
  <status>
   <gp:geopriv>
    <gp:location-info>
     <gml:location>
      <gml:Point gml:id="point1" srsName="epsg:4326">
       <gml:coordinates>33:46:30N   142:25:11W</gml:coordinates>
      </gml:Point>
     </gml:location>
    </gp:location-info>
   </gp:geopriv>
 </status>
<timestamp>2003-06-22T20:57:29Z</timestamp>
</tuple>
</presence>
```

Figure 1. The PIDF-LO Document with GML information.

III. REUSING THE SIP PRESENCE PACKAGE FOR LOCATION CONVEYANCE

If the PIDF-LO document were a vehicle then it would need a highway to transport in the traffic system. This is what is described in the introduction as the *using* protocol. To reiterate a little, it is an application level protocol which is capable of carrying XML documents between network entities. In this case a XML payload is the PIDF-LO document. As mentioned in the introduction, the IETF specifications propose the SIP protocol as the *using* protocol [1]. The specifications do not specify the use of any event package for the interchange of location-based information. Here we reuse the SIP presence event package, which means that location can extend existing SIP presence services. By definition, an IETF event package is essentially an extensible framework by which SIP nodes can request notification from servers when specific events have occurred [2]. Geographical and presence information can be communicated between network entities in the same PIDF-LO document depending on availability of either data. This, obviously, could be more bandwidth efficient than implementing a whole new system for location sharing.

The presence information in SIP-based systems is communicated by the presentity to the Presence Server (PS) by using a SIP *PUBLISH* method [3]. The payload of this message is the PIDF document. A 200 OK is sent in response if the format of the document and the SIP message itself satisfy the constraints of the server. The PS caches the status of the *presentity*. The status is then refreshed at the server before the time specified in the SIP header expires. For location-based information, the message flow happens between a LG, the PS and the LW. A location object is added to the PIDF document at the LG and it is sent off to the server. It, however, is not discussed in the specification on how location and presence should be contained within the same document.

Moving on further, the SIP *SUBSCRIBE* method is sent by the Presence Watcher (PW) to the PS. The header of this message contains the SIP address of the presentity of interest. The PS, in response, sends a 200 OK with *tags* for maintaining a dialog. Assuming there is no privacy involved, the cached presence status is sent to the PW right away or after it has been updated. The PIDF information is sent within a SIP *NOTIFY* message. If this PIDF contains a location object, then it is sent to the Presence/location Watcher in the same way. Assuming there are no errors, the watcher has to acknowledge the receipt with a 200 *OK* response, otherwise the PS will keep resending until there is a timeout.

All the above-mentioned messages have the *event-type* or *application* header set to presence and if the *content-type* is set, it is set to *xml/pidf*. This means that the Presence Event Package is being used unmodified, for location purposes. As to whether the *location object* could be carried in the same document that contains a *basic* element, it is not clearly specified. Close scrutiny of the design of the PIDF-LO document format, however, shows that it is valid. All the other specifications of the Presence Event package and Resource-Lists like presence *watcher-info* and *buddy-lists* are left unaffected by this addition. This means that buddy requests and additions are done the same way, except for aspects relating to privacy policies.

IV. SIP Location Event Package For Geographical Information Conveyance, the OMA Way

The Open Mobile Alliance (OMA) reuses the ideas from the IETF specification discussed in the previous section. Instead of reusing the Presence Event Package, however, it specifies a whole new event package for the carriage and management of location-based information [7]. The core framework of event packages is based upon the publish/subscribe architecture [2]. The Presence Event Package, and every other SIP event package, extends this architecture to share application-specific events. The OMA location event package extends this architecture to support events in the location domain. This clearly translates to the fact that, unlike the approach discussed in the previous section; presence and location in OMA do not use the same documents and event packages. A PIDF and a PIDF-LO document with an added location object are separate from the OMA point of view. In addition to this, OMA also adopts standards from IETF regarding *Quality of Service* (QoS), and 3GPP for authentication, billing and charging. These three aspects are out of the scope of this paper, however.

The LS in the context of the OMA specification is called the LIS, which stands for the Location Information Server. When a request is sent from the Location Client it must contain a *feature* tag to provide the server with means to distinguish between location and presence requests. Depending on the event package, presence or location, the request is routed to the Home Subscription Agent (HSA) or to the LIS. Most of the aspects related to location publications and subscriptions, from the SIP perspective, are performed in the same way as in the location utilizing Presence Event Package.

V. GEOPRIV HTTP Enabled Location Delivery (HELD)

The GEOPRIV working group also embarked on a standardization effort dealing with the *geolocation* and privacy information over the web. The *using* protocol of choice here is, of course, HTTP [5, 6]. The customized protocol for the carriage of location that resulted thereafter is known as HELD, short for HTTP Enabled Location Delivery. Moreover, the components defined for this specification include the Location Information Server (LIS), and web clients which also assume the role of Location Recipients (LR). This LIS contains a static database of addresses that can be matched with an IP address to provide location. It also relies on lower level network protocols and standards like Domain Host Configuration Protocol (DHCP), to determine the location of the clients. The basic assumption here is that the client's location is stored somewhere on the network (possibly manually put in the LIS), or that the LIS can determine the requesting clients' location. These are assumptions that might, obviously, not always be true.

In short, the HELD protocol provides the means for the retrieval of the LR's location in the form of a PIDF-LO document and/or Location URI(s) from a LIS. This is done through the three specified messages namely *locationRequest, locationResponse* and *error* which are defined in XML [14]. When the client wants to determine its location from the LIS, it sends a *locationRequest.* This is essentially an HTTP *GET* or *POST* request with the *content-type* of *application/held+xml*, and an *accept* header with the same value as the *content-type*. The content is an XML document that specifies the format of the information to be returned-*geodetic* or *civic*. The *locationResponse* is then sent back by the LIS in the form of an OK request with the *content-type* of *application/held+xml*. If an error occurs, then the *error* response is given.

VI. DESIGN CONSIDERATIONS AND ARCHITECTURE

A. The Development Environment

Mobicents is a Java-based service delivery platform for the quick and easy development, deployment and management of next-generation network applications [11]. It is JAIN-SLEE 1.0 compliant, and it adopts data from various IP protocols for use as Java objects. This functionality is achieved through JAIN-SLEE components called Resource Adaptors (RA) [12]. Mobicents has SIP, HTTP and Diameter resource RAs, to mention a few. For this implementation the SIP and HTTP RA are particularly important. For the composition and building of services, the Mobicents uses JAIN-SLEE units known as the Service Building Blocks (SBBs). These SBBs provide interfaces that facilitate their extension and reuse by other services. Therefore, service developers can reuse them with ease. It is suitable for this implementation because:

1) it is open-source,
2) it has a large developer community and has the potential to reach many Java programmers
3) through the RAs it supports multiple IP protocols
4) it provides SIP-based services including presence
5) through its event model and architecture, it presents an easy and fast way of developing and extending converged services.

B. The Using Protocol

A decision has to be made on the protocol to use for the carriage of the PIDF-LO document for this implementation. The argument is that, in the Mobicents environment, using the SIP protocol is closer to other SIP core services and protocols than using HTTP. If we pick the SIP easily integrate location functionality into the Mobicents SIP Presence Service, if we choose to go the OMA way and decide on using an entirely new event package, SIP is still a much better alternative. This is because it becomes easier to implement and design due to its resemblance to the presence service. We can also, with ease, incorporate it into other applications that include voice calling and conferencing. Furthermore, from the service implementation and management perspective, the protocol is probably much more familiar to VOIP developers than HTTP or HELD, notwithstanding the similarities between HTTP and SIP.

On the other hand, however, we can argue that the solution is aimed for use in converged environments, and Mobicents supports SIP, HTTP and other IP protocols. HTTP also happens to be the predominant Internet Protocol with the potential of publishing and dispensing location information between numerous web-based clients and Mobicents. This extends the customer base from just SIP-based users to Internet and web users. If we choose HTTP as a *using* protocol, the approach presents an opportunity and a guideline to extend interfaces and information from Mobicents to clients and services using other IP protocols, like XMPP, as long as they have location elements. XMPP is one of the most popular instant messaging protocols on the internet.

With these arguments in mind, it holds much promise not to choose between the two *using* protocols but rather to use them both with an architecture that fosters interoperability and consistency. Privacy is not in the scope of this paper but, even from that perspective, it is sensible to implement both *using* protocols and share a central privacy server. This architecture is described and discussed in sub-section IV.

C. The Event Package

In the OMA specifications a new event package is proposed for the SIP *using* protocol. For this project we have also suggested the reuse of the presence event package because it might be a much more suitable alternative. A decision has to be made as to which one of the two to implement in the Mobicents platform. On the one hand, we discussed the reuse of the presence service package using the IETF specifications. On the other, the OMA specifies a new event package for location altogether. The advantages of reusing the presence service are that it is more bandwidth-efficient and than the alternative which is are independent event packages for location and presence. A client that supports both presence and location can easily alternate between the two by changing the content of the PIDF document. Presence and location can also be sent over the network in the same document with the same overhead for one. Using an entirely independent event package means that the client and the server need to send different SIP messages and maintain separate dialogs for both presence and location. For this reason, the reuse of the presence event package of the Mobicents Presence Service is chosen.

D. Proposed Architecture

By putting together the specifications, the chosen implementation environment and the design considerations, the following architecture is proposed (figure 2):

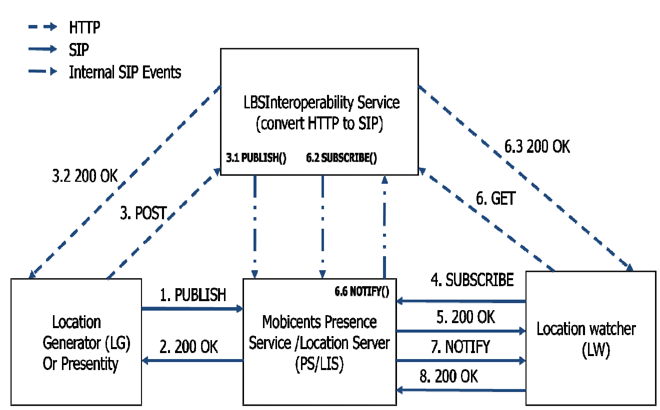

Figure 2. The Mobicents location-based services architecture

E. Publishing Location Information

At the LG or Presence Entity (*presentity*) shown in figure 2, the geographical information and the current presence are determined by the local software and hardware. The information is then packaged into a PIDF-LO document. If location information is not available, for some reason, then a location object is not augmented to the PIDF document. Also if this is a refresh request and the status of the presentity has not changed the *basic* element is not added to this document. A SIP PUBLISH method with the SIP address of the presentity and the *content-type* header set to *pidf/xml* is composed. The PIDF or PIDF-LO document is then appended to the body of this message and is sent off to the server (labeled *1.* in figure 2). The server sends a 200 OK message to acknowledge the receipt of the message (labeled *2.* in figure 2). The publication can then be refreshed as is done in normal presence systems. If the LG supports HTTP only, however, the PIDF-LO document is composed and placed into the body of an HTTP POST message and sent off to the server (labeled 3. in figure 2).

At the Mobicents server, the publication message is channeled through the SIP RA en-route to the SIP Presence Service, if it is in SIP format. The SIP presence Service then caches it as it does for presence information and responds with a 200 OK, if all is well (labeled 2). Otherwise, if the request is in HTTP format, then it is received through the HTTP RA. After this, the request is forwarded to the LBSInteroperability Service.

The LBSInteroperability Service, briefly put, is a Mobicents service developed by us that maps an HTTP request to a SIP publication and emulates, to some extent, the logic of a *presentity* on the server. At the one end, it receives an HTTP message with a PIDF-LO payload and, at the other, a SIP message with the exact same PIDF-LO payload. To elaborate this further, when an HTTP *POST* request is received, a presence agent within the LBSInteroperability service takes the SIP address, the PIDF-LO document and other information. With this it makes a SIP PUBLISH request against the presence server and awaits the response (labeled 3.1). The request is treated at the presence server as if it were a genuine SIP PUBLISH request. Once received, the SIP 200 OK response is converted to a 200 OK response HTTP response and is sent off to the client (labeled 3.2).

F. *Subscribing to Location Information*

At the Location/Presence Watcher, a SIP *SUBSCRIBE* request is composed. Amongst other information, it contains the SIP address of the *presentity* of interest. It is sent to the Presence Server (labeled 4) which has the cached presence and location information about the presentity. On Mobicents, the request goes through the SIP Resource Adaptor (RA) and it is directed to the SIP Presence service where a SIP 200 OK message is generated in response (labeled 5). If the Location/Presence Watcher is HTTP-based, however, an HTTP GET request is composed at the client. This request should contain the URI of the presentity of interest.

Once at the server, the request goes through the HTTP resource adaptor to the LBSInteroperability Service (labeled 6). This time around, the presence agent takes the SIP address and other relevant information and makes a SIP SUBSCRIBE request against the presence server (labeled 6.2). Upon the receipt of the 200 OK SIP response, a conversion to a 200 OK HTTP response is undertaken. The response is then relayed to the web client (LW) involved (labeled 6.3). To refresh subscriptions and publications, the HTTP-based client has to send POST and GET requests with new expiry times and the presence agent does the same conversion as usual. The SIP client has to do the same.

For notifications to presence and location information, a SIP NOTIFY request is composed by the presence server. It should contain the address of the watcher and the PIDF-LO document carrying the location and presence of the presentity of interest. The request is sent to the address of the watcher through the SIP RA if the *using* protocol is SIP (labeled 7). A 200 OK message should be received from the client in response (labeled 8). Otherwise, if the *using* protocol is HTTP, the request is routed to the LBSInteroperability Service. The major differences between HTTP and SIP lie within the asymmetry of their architectures. An HTTP endpoint is either a client or a server but never both. In contrast a SIP user agent usually has client and server functionality. Therefore, the LBSInteroperability Service cannot make a request or push anything to the HTTP client without the client having explicitly requested it. The service rather caches the PIDF-LO document and the client polls for this information with HTTP GET requests. With that said, we move on to the implementation-related issues.

VII. IMPLEMENTATION

A. *Implementing Location-Object Support*

On the Mobicents JAIN-SLEE service delivery platform, we start by building location support into the presence service. This service already processes PIDF documents for the presence-based information; we just need to add support for the *location object*. This is achieved, first of all, by adding objects that can read and write the data from and to *location object* XML documents. We also have to implement the necessary logic to ensure that, once the data is on the server, it is processed according to the design constraints. We then implement the LBSInteroprabilty Service for HTTP-based clients. After all the additions are complete, the server should still maintain its compliance with systems that are exclusively presence-based.

To add Java objects to perform the reading and writing of the *location object* XML document, we use the Java Architecture for XML Binding (JAXB) platform. JAXB is a Java tool that, amongst other things, can be used to translate XML schemas to corresponding *Plain Old Java Objects* (POJOs) [10]. This tool takes the given XML schema and converts all the elements and attributes to a consistent tree of objects and variables with the necessary *getters* and *setters*. At run time, a process called *marshalling* takes the supplied Java information and organizes it into the generated Java objects and then writes it out to an XML document that is consistent with the supplied schema. In constrast, *unmarshalling* takes in an XML document, validates it with the schema, and collapses it into an accessible tree of Java objects. The *location object* schema is acquired, and the document is then converted to corresponding Java Objects using this platform.

As mentioned in the document format section, the *location object* needs to be extended with the customized GML schema for *geodetic* coordinates' functionality, and with the *civic location* schema to support physical or human-readable addresses [1]. These schemas are also taken as they are and are converted into *POJOs* with JAXB. At this point, when a PIDF document containing a location object is sent to the server the generated JAXB *status* object is checked for presence information as well as for any other data in the extensible section. If there is data there, it is matched to the *location object's geopriv* object. If the data is in fact that of a location, then it is validated according to the schema and collapsed into Java objects. The GML and *civil location* data is processed in the same way as well.

Once in the memory, the Location information is cached by putting it into a persistence management component known as a CMP field [11]. The data belonging to the publisher or *presentity* can be accessed from the CMP and sent to subscribers during a notification requests processing. Using conditional systems in the publication and subscription services, the information is *marshalled* and *unmarshalled* according to its availability. If no location object is found in the data, then the server processes the presence information in the usual way by bypassing the location processing logic.

B. Implementing the LBSInteroperability Service

As mentioned in the design section, the LBSInteroperability Service contains the logic for converting requests from an HTTP format to the SIP format so that location and presence data can be exposed via HTTP-based interfaces. From the implementation perspective, the most important thing to do here is to make sure the service receives HTTP POST and GET requests. It should also call the internal SIP publication and subscription service interfaces to fire SIP messages against the server and to get back responses. The JAIN-SLEE event model has a specific way of relating services and protocol *resource adaptors* [12]. Within the service descriptors, we specify that the new service is interested in HTTP messages by referencing the HTTP resource adaptor. This means that incoming HTTP messages are forwarded to this particular service as well as to all the other interested applications.

Within the service's *Service Building Block* (SBB) we specify the handler methods for HTTP POST and GET requests. Within this SBB, we add the functionality to process requests as they are received. We extract the SIP addresses of the LG and the LW, the expiry time for refreshing the requests and the body of the request from the POST message. We then call the publication interface and provide it with the information. A PUBLISH request is made against the server, and a 200 OK response is received. From this, we take the important data and transfer it 200 OK HTTP response. In the HTTP GET method handler we take the SIP address and the expiry time and we provide the internal subscription interface with the data. We then process the OK message is in the same way.

Finally when it is time to notify the subscriber, we take the PIDF-LO document from the body of the SIP NOTIFY request and cache it in the services persistence fields for an amount of time a little more than that specified in the expiry headers. When the client polls for this information with a GET request, we use the addresses to get the correct PIDF-LO document and related information and reply with a HTTP 200 OK.

VIII. CONCLUSION

This paper has highlighted the growth and the importance of the inclusion of location-based elements in converged services. This, of course, will add value to the SIP/IP core applications and will allow for customization from a geographic point of view. We then discuss the specifications by the IETF and the OMA and highlight the fundamental aspects around them. The result is an architecture that combines both the SIP and HTTP *using* protocols and reuses the Mobicents SIP Presence Service. Finally, we present the implementation issues surrounding the proposed architecture. In future location-based applications like friend finders, location-based call controllers and trackers can be built using the ideas presented in this paper. Before this can be done, however, a privacy-aware infrastructure has to be put together to govern the disclosure of the location information. This way such information will be protected against potential misuse by unauthorized parties.

REFERENCES

[1] IETF, "rfc4119: A Presence-based GEOPRIV Location Object Format," December 2005, http://www.rfc-editor.org/rfc/rfc4119.txt

[2]IETF "rfc3265: Session Initiation Protocol (SIP)-Specific Event Notification," June 2002, http://www.ietf.org/rfc/rfc3265.txt

[3]IETF "rfc3863: Presence Information Document Format (PIDF)," August 2004, http://tools.ietf.org/html/rfc3863

[4] IETF, " GML 3.1.1 PIDF-LO Shape Application Schema for use by the Internet Engineering Task Force (IETF)," April 2007, http://www.rfc-archive.org/getrfc.php?rfc=5491.

[5] IETF, "HTTP Enabled Location Delivery-Draft," March 2007, http://tools.ietf.org/html/draft-winterbottom-http-location-delivery-05.

[6] GEOPRIV, "Location Services @ IETF72 ," http://sites.google.com/site/geopriv/.

[7] Open Mobile Alliance, "TS 24.141: Location in SIP/IP core - Candidate Version 1.0," 18 Aug 2009, http://www.openmobilealliance.org/Technical/release_program/locsip_v1_0.aspx

[8] Gartner, "Gartner Forecasts Worldwide Location-Based Services to Grow Nearly 170 Per Cent in 2008," February 2008, http://www.gartner.com/it/page.jsp?id=600011.

[10] ORACLE, "Java Architecture For XML Binding," March 2003, http://java.sun.com/developer/technicalArticles/WebServices/jaxb/.

[11] Redhat, "Mobicents SIP Presence Service User Guide," 2009, http://hudson.jboss.org/hudson/job/MobicentsBooks/lastSuccessfulBuild/artifact/sip-presence/index.html.

[12]SUN Microsystems "JAIN-SLEE 1.0 Specification,Final Release", January 2004. http://phoenix.labri.fr/documentation/sip/Documentation/Papers/Programming_SIP/Paper_Publication_and_Draft/Java_SIP/jain_slee-1_0-fr-spec.pdf

[13]PTOLEMUS, "Positioning Technologies: Evolving Landscape and Opportunities", July, 2010. http://www.ptolemus.com/content/uploads/2010/07/PTOLEMUS-FREE-LOCATION-STUDY-201012.pdf

[14] Gonzalo Camarillo and Miguel Garcia-Martin. The 3G IP Multimedia Subsystem (IMS). John Wiley and Sons, Ltd, third edition, 2008.

Ndakunda Shange-Ishiwa received his undergraduate degree in 2007 from Rhodes University and is presently studying towards his Master of Science degree at the same institution. His research interests include location-based services and privacy, presence-based systems, network convergence and mobile phone services.

Madeleine Wright is currently a Java lecturer, Computer Science Department at Rhodes University. Her research interests include web technologies and mobile phone applications development amongst others.

Alfredo Terzoli is Professor of Computer Science at Rhodes University, where he heads the Telkom Centre of Excellence in Distributed Multimedia. He is also Research Director of the Telkom Centre of Excellence in ICT for Development at the University of Fort Hare. His main areas of academic interest are converged telecommunication networks and ICT for development.