A Location-based Push Architecture using SIP

Günther Pospischil[1,2], *Johannes Stadler*[1,3], *Igor Miladinovic*[1,3]

[1)] Forschungszentrum Telekommunikation Wien (FTW),
Tech Gate Vienna, Donau City Straße 1, 1220 Vienna, Austria
http://www.ftw.at

[2)] Institut für Nachrichtentechnik und Hochfrequenztechnik,
Technische Universität Wien, Gußhausstraße 25/389, 1040 Vienna, Austria
guenther.pospischil@nt.tuwien.ac.at, http://www.nt.tuwien.ac.at

[3)] Institut für Kommunikationsnetze,
Technische Universität Wien, Favoritenstraße 9/388, 1040 Vienna, Austria
{johannes.stadler, igor.miladinovic}@tuwien.ac.at, http://www.ikn.tuwien.ac.at

Abstract

Service developers and standardization bodies have focused on push concepts for the mobile Internet recently. Because of the convergence of IP and mobile networks a common architecture for push scenarios is possible.

We discuss several use-cases for push concepts and their technological implications. Finally we present and evaluate a unified push service architecture, based on SIP.

Key words

Push service architecture, SIP, location based UMTS services, mobile Internet.

1. Introduction

Recently push functionality for the mobile Internet has gained much attraction. It allows to establish a connection to a user terminal without his/her action and to deliver content as soon as it is available. Especially the "always online" paradigm and the limited air interface bandwidth make push features an important component of mobile Internet services/applications. Using the telecommunications nomenclature, we speak about applications if we refer to the implementation of dedicated function blocks, e.g. a call control application or a specific push application. If a general, abstract functionality is considered, we use the term service.

The basics for push services in mobile networks are currently being standardized in 3GPP (UMTS service architecture [1], Multimedia Messaging [2]) and WAP-Forum (WAP push [3]). Packet switched UMTS sessions will be handled via IETF's Session Initiation Protocol (SIP [4]). With the extensions defined in [5][6][7], SIP can also be used for messaging and asynchronous notifications.

The convergence of IP networks (Internet) and SS7-based mobile networks towards a common "mobile Internet" makes a unified service architecture for push scenarios possible. However, to our knowledge there is no "global view" on push concepts yet, i.e. what use-cases are possible and what are the technological implications. Therefore our paper starts with a section defining various push scenarios and deduces the key aspects a push architecture should provide. The second part of the paper gives an overview on possible technical solutions with SMS, WAP, and SIP, describing a proposed SIP-based solution in detail. The last part of the manuscript provides an assessment of our solution regarding two major use cases.

2. Push for mobile networks

Push services are relevant for mobile networks because they use the limited bandwidth efficiently, i.e. communication only occurs if there is information available. In pull-based communications, the user performs periodic checks for information - even if no new content is available. This includes Packet Data Profile (PDP) context activation, which consumes network resources (air interface bandwidth, buffers, IP-addresses) and terminal resources (transmit power).

Additionally, push services are suitable for the notification of asynchronous events. This may happen frequently in mobile networks, because users can select relevant information a priori, e.g. subscribe to traffic information channels, and receive information directly ("always connected" paradigm). A special case of notification are location based events, they are unique for mobile networks (see section 3).

Mobile networks are early adopters of push features. It started with the Short Message Service (SMS) of GSM. Currently, an extension called Multimedia Message Service (MMS [2]) is being standardized. MMS uses Wireless Application Protocol (WAP) or Mobile Execution Environment Applications (MExE, i.e. JAVA+TCP/IP), a message may consist of text, pictures, audio, and video. SIP integrates all these types of content and offers possibilities for real-time sessions, like streaming, voice over IP (VoIP). Additionally, SIP User Agents (UAs) will be available in all UMTS terminals, supporting bearer establishment directly. As there are also SIP User Agents for conventional PCs, "service portability" is automatically given, i.e. a user may access a service with any kind of terminal.

2.1. Types of push

- Confirmed/unconfirmed: For important data or specific billing models (pay per message), confirmed push should be used. If data is not so important, e.g. because it is valid only a short time, unconfirmed push can be used. Examples for important and time-critical data are stock news, while location information may be less important data because it is replaced with a new location estimate after a few minutes.
- Visible/hidden: Visible push includes user notification, while hidden push delivers new information to a specific application (application related push). Examples are multimedia messages or text/html-based advertisements (visible push) and location updates within a navigation service (hidden push). For application related push, an application addressing scheme is necessary.
- Unicast/multicast/broadcast: Unicast data is typically related to specific applications (hidden push), e.g. delivery of location estimates to a navigation application, or a user-to-user data transfer, like MMS or SMS. Multicast and broadcast data can either be application related (e.g. delivery of traffic information to all users of a navigation application) or visible (e.g. advertising to all users who have subscribed to the "advertising channel").
- Global/regional (location based): Current messaging systems offer global service, i.e. content is not related to the user's location. UMTS provides the additional potential to create location based push services, e.g. location based advertising or traffic information.
- Connection oriented/connection less [3]: Connection oriented push is just a network initiated session establishment, which allows the transfer of any kind of information afterwards. Connection less push is the delivery of a single piece of information without establishing a connection for subsequent data transfers.

2.2. Push scenarios

We have identified three kinds of push services: location/ navigation, information distribution/advertising, extended Internet applications.

2.2.1. Location/navigation push service

Location based services will often need updates on the user's position. As user movements are not deterministic, a pull based approach is inefficient, push communication is more adequate. Depending on the service, there may be an active connection between the mobile and the network, or not. If the connection is available, any remote communication concept, like JAVA remote method invocation (RMI), CORBA, or socket connections, may be used. However, if no link is established, we need an architecture which supports bearer establishment as well. SIP provides all required functionality for this unconfirmed, hidden, unicast push.

2.2.2. Broadcasting push service

This group of push services consists of information distribution, e.g. traffic news, stock exchange data, and advertising. Optionally it may incorporate a location based component, i.e. contact only users that are in a specified region. Thus it is possible to inform only potentially affected users about a traffic jam or present special offers in a user's vicinity. Here we find unconfirmed (or confirmed), visible, multicast, connection less, regional push.

This type of services offers new revenue models for operators and subscribers, including sponsoring of air time. This scenario is already popular in the Internet, where many services are free of charge for the user because they are financed via advertising.

2.2.3. Extended Internet applications

Basically, these are e-commerce applications as already available in the Internet. They can be extended for m-commerce, e.g. by using mobile payment solutions. These services use push for messaging/notification, e.g. deliver new mail without polling, notify a user about transaction status. All types of push may occur, depending on the application.

2.3. Security for push

This topic is only addressed briefly in our paper. There are three basic security concerns: protecting data traffic, ensuring authenticity of sender and receiver, and maintaining privacy. The privacy issue is discussed in chapter 4, special focus is put on the location information of a user. The other aspects, protecting data traffic and authenticity, are not specific to push, they are relevant for all SIP-based communications.

The simplest solution is to use a trusted network, which is state of the art for mobile networks. However, in the Internet world we face exactly the opposite, a network without any guarantees. Therefore various concepts have been developed for authentication and content protection (e.g. PGP [8]) as well as protecting signaling. SIP is based on UDP/IP, hence any IP layer security concept, like IPsec [9] can be used for SIP sessions. In UMTS mobile networks, additional security mechanisms are possible [10]. Users are identified with their USIM card (which holds authentication data), transport security is provided via UMTS AKA protocol [11].

A specific problem of application related push is that a push initiator could probe which applications a user has installed. This problem can be solved with SIP quite easily as presented in chapter 3.3.7.

3. Technical aspects

Push services rely on a concept for bearer establishment and a well defined data distribution architecture. In section 3.1 we give an overview on UMTS bearer establishment with SIP, section 3.2 deals with the SIP-based push architecture, the chapter ends with a section on push data management.

3.1. Bearer establishment

Bearer establishment in the packet switched UMTS domain is called PDP context activation, i.e. set up an IP bearer as described in [12]. The Gateway GPRS Support Node (GGSN) starts a network initiated PDP context activation as soon as it receives a PDP protocol data unit (PDU).

Current GSM/GPRS networks do not support network initiated PDP context activation, therefore real push is not possible with GPRS. Work arounds are the use of GSM circuit switched data connections or SMS in conjunction with a "call back" application in the terminal (which "pulls" the

information after a SMS push notification). This limits the applicability of SIP push in current networks because *a) a supporting call back application is needed* in the terminal and *b) additional bearer level messages* are exchanged, causing a fairly large protocol overhead (data and delay).

However, it should be noticed that the described problems for bearer establishment are inherent to the existing GPRS networks. They are not SIP specific, WAP push for GPRS networks has the same difficulties. It is very likely that SIP will overcome these problems in UMTS networks because SIP will be used to establish VoIP sessions, thus enabling a combined bearer- and session establishment.

3.2. Application layer signaling

There are several application layer signaling protocols available in mobile networks and the Internet. Most of them are tailored to specific session types, e.g. SMS for short text "sessions" or WAP for simple text based mobile Internet content. The remainder of this section gives a brief overview on SIP, SMS, WAP and H.323 protocols.

3.2.1. Session Initiation Protocol (SIP)

SIP is an application layer control protocol developed in the MMUSIC working group of the IETF [4]. SIP can establish, modify and terminate all types of sessions. With some extensions SIP can also be used for presence and instant messaging applications. SIP is a text-based, HTTP-like protocol, with two types of messages – requests and responses. A SIP message can transport any MIME-type content. For VoIP applications and multimedia conferencing applications the Session Description Protocol (SDP) [13] is commonly used. Multimedia transmission usually adopts the Real Time transport Protocol (RTP) [14].

There are two types of entities in SIP, called SIP User Agents (UA, in the terminal) and SIP network servers. Network servers are used for call routing and maintaining call states. A user has to register by sending a register message to the corresponding Register Server every time he/she wants to be reachable. A register message contains the current network address of the user (e.g. IP address), which is stored in a data structure of the Register Server. Validity of a register message is always time limited. A user can also unregister by sending a register message with a time limit of zero.

As mentioned above there are SIP extensions for presence and instant messaging [5][6][7]. Three new request methods are proposed: SUBSCRIBE and NOTIFY for presence and MESSAGE for instant messaging. In our opinion these methods can be used for push services, too. For instance, a user subscribes for a push channel (e.g. traffic alerting) with the SUBSCRIBE method. When an event occurs, it is pushed to all interested users with the NOTIFY method. This replaces the periodic queries for new information (pull scenario) with a single subscribe message. The user can also unsubscribe a channel, if he/she is not interested anymore.

Concluding, we see the following benefits of using SIP for push services: available in all future mobile terminals and in the Internet, flexible content (MIME types), easy to implement protocol (text based), open internet standard, de-centralized and scalable architecture.

The disadvantages of SIP for push are that SIP is not developed for push services (therefore it needs the mentioned extensions), and that SIP is not widely used yet.

3.2.2. Short Message Service (SMS)

SMS is an integrated element of GSM and UMTS. It is suitable for push delivery of short text messages. The SMS architecture consists of a SMS User Agent which is integrated into the mobile phone and a SMS Center (SMS-C) which stores and forwards the messages. Advantages of SMS are that the system is available and proven in all GSM networks, scalability and roaming is also given. But GMS/UMTS short messages have major disadvantages, too. Interworking with the Internet is not possible, SMS content is limited to 160 characters, application addressing is not possible, and there are no hard delivery guarantees.

3.2.3. Wireless Application Protocol (WAP)

The WAP push architecture is a candidate for the UMTS Multimedia Message Service (MMS). It is quite similar to the SIP architecture, including a WAP Browser (similar to the SIP UA) in the terminal and a WAP Push Proxy Gateway (PPG). The WAP PPG is similar to the SIP Proxy. It accepts push request from a push initiator in the Internet, using the Push Access Protocol (PAP). The PPG converts the request into the push Over The Air protocol (OTA) format and forwards the request to the terminal.

WAP push addresses many of the goals that we have identified for our push architecture. It includes security mechanisms, allows application addressing, supports connection oriented and connection less push. Some limited multimedia support (e.g. to transfer images) is also available, but streaming or other real-time multimedia features are not possible. WAP lacks Internet integration, i.e. usually there are no WAP browsers in Internet terminals. Currently, there is no support for advanced push concepts, like location based push, and WAP push is not directly integrated into UMTS Open Service Architecture (OSA). As stated above, the lack of network initiated bearer establishment is also a problem for WAP push. The WAP push architecture is asymmetric, needing a specific WAP push initiator application. Our proposed SIP push architecture is symmetric, where (in principle) every SIP User Agent may issue a push request.

3.2.4. H.323

H.323 is an ITU-T series of protocol recommendations for multimedia communications over packet switched networks, including Local Area Networks and Internet. H.323 consists of a group of protocols for control (H.245), connection establishment (H.255.0), large size conferences (H.332), security (H.235), interoperability with circuit-switched services (H.246), supplementary services (H.450.0–H.450.2), video (H.26x) and audio codecs (G.7xx).

There are four types of components in H.323: terminals (equivalent to SIP UA), gatekeeper (equivalent to SIP Proxy and Register Server), gateway and multipoint control unit.

The advantage of H.323 is that there are some implementations of H.323 components already. For transport of push data (e.g. a HTML file) T.120 recommendation can be used. T.120 is a part of H.323 and provides optional capabilities in H.323 such as application sharing, whiteboard sharing, file transfer, fax transmission and instant messaging.

However, since H.323 does not support a system like SUBSCRIBE and NOTIFY in SIP yet, it is not easy to use it for push applications with subscribing services. Furthermore,

H.323 is more complex than SIP and 3GPP has decided to use SIP for signaling in UMTS. Because of all this reasons, we have decided to use SIP for push applications.

3.3. Architecture components for SIP-based push

3.3.1. Overview

The proposed SIP based push architecture enables location based push services and conventional content provider push (e.g. advertising).

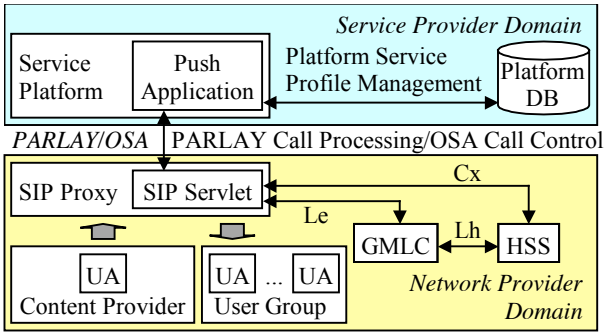

Figure 1: SIP push architecture

As shown in Figure 1, there are two domains, the *Network Provider Domain* and the *Service Provider Domain*.

The network provider runs a SIP-enabled communication network, e.g. a UMTS network. Call management is performed via the SIP Proxy, which is called Call State Control Function (CSCF) in UMTS. It is used to forward SIP Requests or Responses to their destinations and to run some applications. It may need user preferences and location information, therefore it interacts with the Home Subscriber Server (HSS; the central user data base) and the Gateway Mobile Location Center (GMLC; used to determine a user's location) using MAP/Cx and MAP/Le interfaces [15].

The signaling in the terminals is done by SIP User Agents, making no difference if the end user is a content provider or not. Although SIP is designed for call control it has the capabilities to transport everything in the body of a message which fits into the MIME format, e.g. a HTML page. This circumstance offers the possibility of HTML pushing behavior instead of the usual HTML pull concept.

The service provider operates a service platform, which acts as a framework for the creation, operation and maintenance of applications. These applications can be offered to one or more network providers. A standardized interface, like PARLAY, between service provider and network provider helps to ensure service portability and to reduce implementation effort and time to market.

A detailed description of the architecture components is given in the following sections.

3.3.2. Push application

The push service consists of two parts. The data delivery part runs in the SIP Proxy and is described in the next section. The second part resides on the service platform of the service provider. It is responsible for the management of push subscriptions and interaction with other services. End users may sign in for push services either explicitly, e.g. on a web form or with a SIP SUBSCRIBE message, or implicitly. An example for the latter is a subscription to an "advertising tariff", where push advertising is mandatory. All users who have signed in for a certain push service (push channel) are included in a user group which has a specific SIP address.

The platform database (Platform DB in Fig. 1) holds the list of subscribers for a push channel, i.e. it performs the mapping between the channel address (SIP or PARLAY user address) and the SIP addresses of its members.

The second task is the communication with the SIP Proxy of the network provider via the PARLAY API. This is necessary to receive push requests and return the list of subscribers after accessing the platform database.

If users subscribe to a push channel during a pending push request, i.e. before push request expiration, they should be added dynamically to the list of receivers. In this case the application has to contact the SIP Proxy again to initiate the data transfer to the additional users.

3.3.3. Push Gateway

The push gateway consists of a SIP Proxy with an additional SIP push servlet. The SIP servlet API is based on the HTTP servlet API [16]. If a request (or response) comes in, the appropriate servlet is activated. The message is forwarded to this servlet, which executes some code to handle the request.

Consider a push servlet, which gets a request, containing push related information. The push information may consist of a HTML push content, the location information for location based scenarios and the expiration time. The latter indicates how long the push is valid in time. We use the expire header of the SIP request for this purpose. The push content and the location information are transported as multipart message in the SIP body. The location information is given in some user and system readable format, preferably XML as proposed in [17].

All this information has to be identified and stored until the push request is completed. After receiving the list of push subscribers from the push application, the servlet determines the push settings and location of every subscriber. The servlet contacts the Home Subscriber Server (HSS) to identify the current push settings of the desired user, the coordinates are supplied by the Gateway Mobile Location Center (GMLC). If the user is in the desired region and has push enabled, the servlet forwards the push request to the user's SIP UA.

A detailed description of this behavior is presented in section 4.1.

3.3.4. Home Subscriber Server/Home Location Register

The HSS/HLR is the central user database of a UMTS/GSM mobile network. It is used to store general subscription information (e.g. user IMSI number, tariff model), and service specific data. In our case two parameters are relevant: push settings and localization settings.

Via the push settings of the HSS, a user may quickly disable or enable all push services he/she is subscribed to. This avoids that a user has to contact all service platforms and modify the subscription profile there. Optionally the HSS may contain lists of content providers or push applications that are allowed to perform push, it is also possible to block certain push originators.

We have developed this concept in analogy to the localization security exception settings of the HSS [18]. A user may decide who is allowed to know his/her location and

enter appropriate settings in the HSS. Again it is possible to enter some "global" settings, i.e. allow or block all localization requests, and to enter "individual" settings, specifying all allowed (or blocked) requestors separately.

For obvious reasons, a convenient access to the HSS via the mobile terminal is needed. Therefore a html/wml interface to the HSS should be provided by the mobile operator. A special challenge in this context is authorization, but the SIM-based security mechanisms can be used for this purpose, just as it is currently done for accessing the mobile voice mailbox.

3.3.5. Gateway Mobile Location Center (GMLC)

The GMLC is used to determine the user's location for location based push requests. There are three possible situations: a) the user has blocked localization, b) a user is in the region, c) a user is outside the push region.

If a user has blocked localization, no location based push data will be delivered, it is assumed that the user is permanently outside the target region. If localization indicates that a user is in the push region, the push content is immediately forwarded to the user's terminal. In the last case, a trigger is set within the GMLC. It includes the desired push region and the expiration time of the push request. If the user enters the push region before the timer expires, the trigger "fires", i.e. it notifies the SIP push servlet which forwards the push request to the user upon receiving the notification.

3.3.6. Content Provider

If the content provider wants to initiate a push message, it generates a SIP request including the HTML content and, in the case of location based push, the location parameters. This feature has to be supported by the SIP UA of the content provider, i.e. it must be possible to attach MIME content to the INVITE/NOTIFY message. Since SIP User Agents are available for virtually all Internet terminals, any Internet user may originate a push message. The SIP proxy and the push application may decide to process the request or discard it if the originator is not allowed to initiate push requests.

3.3.7. Terminal application/user

The SIP UA of the end user receives the push message. It extracts the HTML page and presents it to the user. Similarly, any other MIME content can be attached, it may be presented via appropriate external programs. This enables application oriented push, the only requirement is that the application is registered with the related MIME type in the SIP UA. There is no feedback if an application for a specific MIME type is available or not, therefore there is also no privacy problem regarding application probing.

4. Realization

4.1. Content provider push and location based push

As mentioned above we distinguish between two different scenarios of push, which we call (generic) content provider push and location based push. Both are feasible with the architecture proposed in chapter 3.

The following section describes in detail a location based push and outlines the differences to the content provider push. Figure 2 gives an overview of the scenario. The marker /n indicates that an operation is done for every user. Without this marker, the operation is performed once, i.e. it is representative for the whole group of push subscribers.

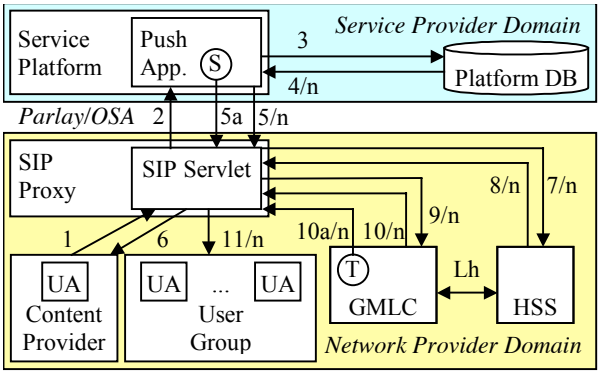

Figure 2: Location based push scenario

At first the content provider sends a SIP INVITE request to the SIP Proxy (1). The body of the request contains the location parameters and the push content. The content may be a HTML page, any MIME content, or multimedia session information (described with SDP). The header of the SIP request contains the expiration time as well as the channel identifier. The servlet engine of the proxy activates the corresponding push servlet and forwards the request.

The push servlet has to maintain the state of the push process as well as to store important information, like the push content, the location parameters and the expiration value. The servlet forwards the push request, including the channel identifier, to the push application on the service platform over the open PARLAY interface (2).

Now the application retrieves the list of subscribers for the given push channel from the platform database (3,4). Afterwards the application returns the list of push subscribers to the push servlet (5). In parallel, it initiates a status report to the content provider (6). If the push channel is valid and contains at least one subscriber, a positive SIP response is sent to the content provider. Otherwise, or if any other problem occurs, an error report is sent.

A user may subscribe to the push channel while the push request is pending, i.e. before its expiration time. This situation is shown with the encircled S in Fig. 2. In this case, the push application adds the new user to the channel and notifies the push servlet (5a).

Now it is checked for every user if push is allowed. This information and the current address of the user are retrieved from the HSS (7,8). The servlet now sets up a list of users who will receive the push content. This is necessary for managing the push delivery, e.g. retries if a user cannot be reached immediately. It is also used for billing purposes, i.e. to report the number of successfully contacted users to the push application after completing the push request.

In the case of a content provider push, data delivery is performed now by forwarding the SIP request, including the push content, to the identified end users (11).

In the location based case, the servlet has to contact the GMLC to find out if the requested user is present in the specified geographical area (9). The GMLC responds with the user location (10). If the user is not in the area, the servlet sets a trigger T for this user. It fires when the user enters the push

area (10a). Upon a positive response from the GMLC, the push servlet forwards the SIP request to the SIP stack of the proxy, including the address of the user terminal. Now the proxy tries to deliver the push message to the SIP UA of the specified terminal (11). If it succeeds, the servlet updates its internal user list accordingly. After expiration time of the push request, all triggers are disabled in the GMLC. Then the push servlet sends the delivery statistics, i.e. number of reached subscribers, and billing data, to the push application.

Note that this architecture ensures a clear separation between service providers, who provide applications, and network providers, who are responsible for the transport infrastructure. Furthermore, dealing with security issues is simple and effective in this architecture. Authentication is only required between pairs of entities, e.g. content provider and push application or push application and SIP servlet. This can be done with state-of-the-art methods like PGP and IPsec. Privacy of end users is automatically ensured because only statistical data is reported to the push application and the content provider. No individual user data (preferences, location) are sent outside the network provider domain. This also helps to minimize traffic over the PARLAY interface.

4.2. Application related push for LoL@

LoL@, the *Local Location Assistant* is a prototype of a UMTS location based service which is being developed at Forschungszentrum Telekommunikation Wien (FTW, http://www.ftw.at). It provides a tourist guide for the City of Vienna. LoL@ includes navigation features, i.e. the user is guided from his/her current position to a selected destination. This feature uses maps where the current position and the next part of the route are shown. To update the map display, i.e. to show a new user location and/or route segment, an application related push concept is used. In this case the architecture is much simpler than in example 1.

The content provider requests periodic updates of the user location information from the GMLC, using the PARLAY Mobility Service. If the user has moved, the content provider initiates a push data transfer to the terminal, including the new coordinates to be shown in the map. In this example we have an unconfirmed, connectionless, unicast push request which is not location based, since the user location is now the content of the request.

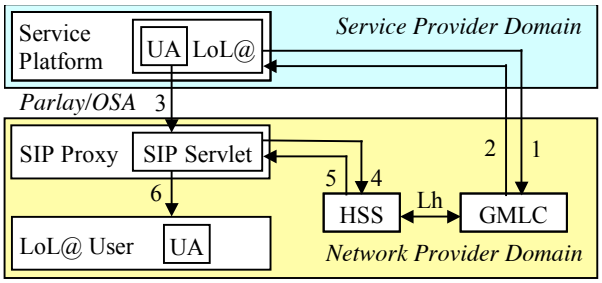

Figure 3: LoL@ push

The control flow is as follows: The user logs into LoL@, which activates the periodic location requests (1). If a new location estimate is available, the GMLC notifies the LoL@ application (2). The application determines if the movement was big enough to send a notification to the LoL@ client running in the user terminal. If the decision is positive, a SIP push request is issued (3). The SIP proxy identifies the request and activates the push servlet. The servlet now checks the push settings in the HSS (4,5). If push is allowed, the push request is sent to the SIP user agent (UA) running in the terminal (6). The user agent is responsible to forward the request to the local LoL@ applet. At the first glance, this situations looks like a standard SIP session activation, but we have some differences: *a) checking the push preferences in the HSS, b) the SIP session is automatically accepted and closed after data delivery, c) session data is directly sent to the appropriate application.*

5. Conclusions

We have described a SIP-based push architecture with special considerations for location based push services in UMTS. A comparison with other push concepts and two show-cases have been presented. It was shown that a function split between network and service provider is useful for UMTS push services. It ensures a clear assignment of tasks while minimizing data traffic over the OSA/PARLAY interface.

6. References

[1] 3G TSG Services and System Aspects, "3G TS 22.101: Service Aspects; Service Principles (Release 5)", 3GPP (http://www.3gpp.org), Jan-2001.

[2] 3G TSG Terminals, "Multimedia Messaging Service; Functional Descr., Stage 2 (Release 4)", 3GPP, Jan-2001.

[3] WAP-165, "WAP Push Architectural Overview", WAP-Forum (http://www.wapforum.org), Nov-1999.

[4] Handley M., et al., "SIP: Session Initiation Protocol", RFC 2543, IETF (http://www.ietf.org), Mar-1999.

[5] Roach A., "Event Notification in SIP", Internet Draft, IETF (http://www.ietf.org), Feb-2001.

[6] Rosenberg, J., et al., "SIP Extensions for Presence", Internet Draft, IETF (http://www.ietf.org), Mar-2001.

[7] Rosenberg, J., et al., "SIP Extens. for Instant Messaging", Internet Draft, IETF (http://www.ietf.org), Apr-2001.

[8] Elkins, M., et al., "MIME Security with PGP", RFC 2015, IETF (http://www.ietf.org), Oct-1996.

[9] Kent, S., et al., "Security Architecture for IP", RFC 2401, IETF (http://www.ietf.org), Nov-1998.

[10] Kroeselberg, D., "SIP security requirements from 3G wireless networks", Internet Draft, IETF, Jan-2001.

[11] 3G TSG Services and System Aspects, "TS 33.102: 3G Security; Security Architecture (Release 1999)", 3GPP (http://www.3gpp.org), Dec-2000.

[12] 3G TSG Services and System Aspects, "TS 23.060: GPRS Service Description; Stage 2 (Release 1999)", 3GPP (http://www.3gpp.org), Jan-2001.

[13] Handley M. and Jacobson V., "SDP: Session Description Prot.", RFC 2327, IETF (http://www.ietf.org), Apr-1998.

[14] Schulzrinne H., et al., "RTP: A Transport Protocol for Real-Time Applications", RFC 2543, IETF, Jan-1996.

[15] 3G TSG Services and System Aspects, "TS 23.002: Network Architecture (Release 4)", 3GPP, Dec-2000.

[16] Davidson, J., et al., "JAVA Servlet Specification V.2.3", Sun Microsystems (http://www.sun.com), Oct-2000.

[17] Korkea, M., et al., "A Common Spatial Loc. Data Set", Internet Draft, IETF (http://www.ietf.org), May-2001.

[18] 3G TSG Services and System Aspects, "TS 23.271: Funct. Stage 2 descr. of LCS (Rel. 4)", 3GPP, Jan-2001.