# GeoFence services

Dmitry Namiot

*Abstract*— **This paper describes a geo-fence approach. Paper analyzes the basic elements of the service, the various approaches for the applied services and proposes a way for integrating geo fence API's into mobile OS directly. Also we describe operator's API for geo fencing. By our opinion, this area is one of directions where telecom operators can use their advantages over the Internet companies.**

*Keywords*— **LBS, geo services, geofence, messaging API.**

## I. Introduction

Geo-fence is a virtual perimeter for a real-world geographic area. Virtual perimeters could be static (pre-defined) or dynamic. A typical example of dynamically generated geo-fence is a radius around a store or point location. The statical geofence can be a predefined set of boundaries, like school attendance zones or neighborhood boundaries. Custom digitized geo-fences are also in use. When the location-aware device of a location-based service (LBS) user enters or exits a geo-fence, the device receives a generated notification. This notification might contain information about the location of the device. The geo-fence notice might be sent to a mobile phone for example as well raise any another form of actions [1].

We can describe geo fencing as proactive LBS. In the many cases proactive systems are much more convenient than reactive ones, where the user has to explicitly request for location-based data. There are different kinds of location events the GPS position fix can be tested for, for example, whether the user is in close proximity to a point of interest (POI) or to another user.

Why it is interesting? It looks like geo-fences become one of the hottest areas in Location Based Services. Right now the original development for LBS is more or less completed. We can see some common standards and applications (think about the various implementations of Places services like Foursquare, Twitter places, Google places, Facebook places, etc.) All of them let you either share location info or get some things nearby. And geo-fence opens the door to some personalization. You can get some custom tailored data right on the place, especially when both data stream and place (area) for receiving it are dynamically generated.

At the second, geo-fence will be in the nearest future tight integrated with sensors and M2M applications. City sensors in the various "Smart City" projects will define geofences and notification messages. Another reason why geo-fences are hot is the complexity of the implementation. There are simply much more "places" - based LBS applications comparing with geo-fences systems. Geo-fences being conceptually simple are actually not so easy to implement. Being more precisely, it is not so easy to create an efficient implementation. And geo-fences could be connected also with another hot LBS area – indoor positioning. From the practical point of view both systems mostly interested due to high commercial promising – deliver commercial offering here and right now.

Manuscript received November 21, 2013

Dmitry Namiot is senior scientist with Faculty of Computantional Mathematics and Cybernetics of Lomonosov Moscow State University, Moscow, Russia (e-mail: dnamiot@gmail.com).

## II. Implementations

Now let us see what we have from the practical point of view. Conceptually, geo-fences approach is quite transparent. The principles and model are simple. We have to have some definitions for the areas, e.g. as two pairs of geo-coordinates (latitude, longitude) values. They are defining some square (Nord West – South East). For this square (squares) we can define some notification message (messages). So, as soon as our mobile user (subscriber or user enabled our service is in (or out, or in/out – depends on the rules) he will get that message. As we can see, the basic problems here are how to check user's location against some predefined geo boundaries and where this checking should be performed.

The location of the user can be easily derived by various positioning technologies like GPS or Cell. We can determine the location of a user while he is active in a service session, or we can organize some form of continous monitoring. For latter form the user needs to be continuously tracked in the background, even when the mobile device is idle or executes other applications [2].

One method for positioning is often not enough. So, in the most cases mobile phones can use some combinations, e.g. Assited GPS (A-GPS). A-GPS uses assistance data received from the network to obtain a faster location calculation compared with GPS alone. The positioning data can be exchanged between the phone and the network over either the control channel (control plane) or the user channel (user plane) plane. A control plane implementation uses a dedicated control channel. This approach could be used used for emergency services (e.g., 911 in the US). For non-critical location-based application user plane could be used. The main difference is the the significant network overhead in case of the dedicated control channel [3]. It is so called Secure User Plane (SUPL). With SUPL positioning data is sent over the user's traffic channel using a secure IP connection between the smartphone (in standard is is called SET - SUPL enabled Terminal) and SUPL Location

Platform (SPL) on the network side. It was developed by the Open Mobile Alliance (OMA) [4]. The objective of the SUPL enabler is to provide an industry standard framework for positioning over the User Plane as an alternative to existing control plane solutions, which are bandwidth-constrained and limited to access types that are part of the control plane system. User Plane may comprise IP and SMS bearers in the mobile networks environment and IP bearers in the WLAN/Internet environment. This mechanism could be implemented in a wide range of contexts (i.e. a controlled mobile network operator's environment or an open Internet/WLAN environment).

OMA SUPL Configuration Service provides a mobile operator with the ability to host a SUPL configuration and provisioning service that handsets may query to obtain SUPL configuration information. The configuration information may include the following elements:

An H-SLP or E-SLP for each roaming partner of a mobile operator

An H-SLP for the use of the handset OEM

An H-SLP for operating system vendors such as Google

D-SLP information for a given region

H-SLP to utilize for an individual application

Application Identifier information to be used when using the SUPL 2.0 or above location protocols

OMA Positioning Protocol Extensions (LPPe) include support Image Recognition Based Positioning (IRBP) as new positioning method. IRBP uses image feature analysis in the target device or on the location server to determine, or help determine, the target device's position. LPPe enhances positioning accuracy and availability in WLAN environments. LPPe supports Pedestrian Dead Reckoning (PDR) as new positioning method. PDR enables pedestrian borne target devices to calculate their current position by extrapolating from previously known positions using target device sensor input and server provided step length models and environment information (e.g., building information, floor plans, etc.) [4].

Where can we perform our geo checking for geo-fencing? Obviously the following three options cover all the cases:

a) client side checking
b) some external server
c) on the operator's size

Let us start with client side checking. It should be a background application that constantly works on the phone and compares its current location with the pre-defined (or preloaded) areas. As soon as our phone is in the area covered by the trigger we should raise an appropriate alarm. We can note at least two main problems with this approach. At the first hand it is battery consumption. For example, GPS, an important enabling technology for mobile location, is a battery hog, typically drawing 300-350 mA on a modern smart phone device (as reference, typical smart phone batteries have a capacity in the 1,200-1,500 mA hour range.) [5]. The second problem is our background tracking application itself. Developers need the contant motivation for potential customers. Why do they need to run this particular application? Many services will face the classical chiken-eggs problem. The motivation for application usage depeneds on the already exting customers, and nobody register there bacase there are no customers.

What could be done here right now? As seems to us the key factor for client side based geo-fencing is a mobile OS. Any applied application (external source for the platform) will always loose in terms of hardware/feature access to the native OS. It is simply yet another layer above the OS. So the application could make things worse, but could not improve the performance/deployment characteristics, etc. Technically, server-side geo processing should win. The above mentioned SUPL solves the problem with background monitoring. At least, developers do not need to provide a special application for the monitoring. And mobile customers do not need to run special applications. Collected location information could be shared between several applications. It is another big problem with client based monitoring, where each application collectes data independently. But here we can highlight one important question. Who will own data collected with server side monitoring? Technically, SUPL is some TCP/IP based communication with the server. And this server is not neccesory to be installed at operator's site. It is simply the more logically to associate server with mobile OS, by the way. It means, that finally the market for geofencing will be own by the mobile OS vendors (Google, Apple).

Lets us see the model, offered by SUPL. At he first hand, it is Standard Location Immediate Service. This service facilitates the location retrieval of the handset on a one-shot basis. The sequence of messages is illustrated on Figure 1.

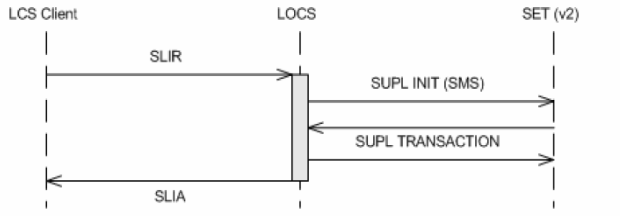


Figure 1. Standard Location Service [6].

SET here is the smartphone in question. The next service is so called Triggered Location Reporting Service. This service facilitates the periodic location or event-based reports retrieval from the handset. The message sequence is illustrated on Figure 2.

It is obvious, the the logipal place for location server (LOCS on the pictures) is the infrastructure for mobile OS. What is a difference here from push messages, provided by Android or iOS [7]? Practically, it could be just another element of the infrastructure. There is simply no place for telecom operators.

## III. GEOFENCING AND TELECOM OPERATORS

In the same time, mobile phone himself (without the geo fencing and any another LBS service) exists in the mobile

network. The phone itself has got information about the network and current cells. So, without the access to GPS the phone itself could be located (yes, the precision could be varied) relatively the current cell.

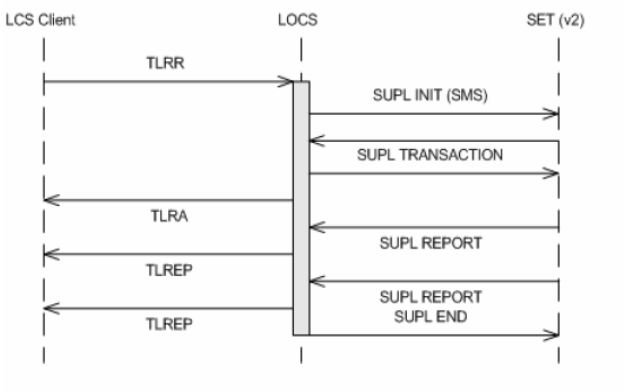


Figure 2. Triggered Service

We mean here the well know technology – Cell ID [8]. Locations for cells are actually known. Even without the operator, third party providers (like opencellid.org) could provide geo information for cells. Phone's Cell ID info exists on the SIM Card and of course it is available for the mobile OS. Yes, the precision could be less than in case of GPS based approach, for example. But in the same time there is a lot of services, where geo fence boundaries could be presented in terms of Cell ID. For example, the proximity based services like SpotEx [9] could work with Cell ID. The same is true for proximity messaging [10]. And the biggest plus here that Cell ID info is associated with SIM-cards. This information is managed and updated by telecom provider. It means, that technically telecom operators can provide some restricted version of SUPL services, based on the Cell ID only. As a base for such service we can suggest the offering from Fi-WARE [11]. The Location Generic Enabler (GE) in FI-WARE targets any third-party application (GEs in FI-WARE, or any complementary platform enabler) that aims to retrieve mobile device positions and area events. For example, let us see the data flow for location subscription. This type of query is used to retrieve either periodic location reports or area entry/leaving/inside/outside type events from a target terminal. The message flow is on Figure 3.

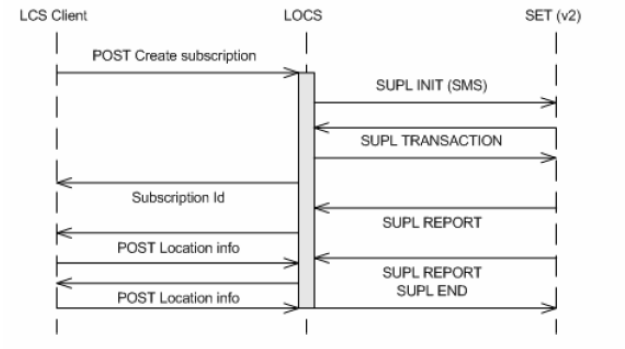


Figure 3. Subscription API

What we suggest, actually, is to replace SUPL related transactions with Cell ID requests. Of course, telecom operator can do that without opening TCP/IP cannection. For example, some providers of SIM card for travelers can report (via Web site) their location [12]. Some SIM-card applet could be responsible for answering positiong requests. Actually, we just offer to simulate SUPL here. It let us use the rest of the services (e.g. described by FI-WARE) with geo data based on the cell's location. LCS client of this figure is some third party application. All such application will work throug operator's site. So, it could be a potential new revenue source for telecom.

## IV. Offloading Cell ID info

Here we can suggest a new approach for using Cell ID data in geofencing (we have not seen such implementations yet).

Typically, Cell ID processing means getting Cell ID info from the phone, converting that info into geo coordinates (geocoding) and performing some actions against obtained geo data. Let us try to reverse this task. What if operator will have a front end where developers can define geo areas (boundaries) in the geo terms (on the map, as we are doing it usually for example), but instead of saving our data as geo data (in the natural form) our front end will translate them into (approximates them with) Cell ID data. Simply, present our (latitude, longitude) as an ID (ID's) for the nearest cell (cells).

Why do we need that? Developers can pack this information and load it into mobile phone as a database for own geo service. The process for getting location data (events) could be presented as obtaing current Cell ID data from SIM card and mapping them against preloaded database with cellid-based boundaries. It could be much more battery-friendly process. Here we borrow also the idea from mobile maps. What mobile maps developers do when you need mobile map in the roaming or the network is very limited (unavailable at all)? They are preparing the downloadable maps. So the end user can save bandwidth/money etc. We can do the same. Such a "map" could be a part of application. Data on SIM card (current Cell ID info) will be updated asynchronously. And third part application is responsible just for checking alarm rules.

Of course, the linked database could be updated too. And of course, operator has got the advatage here over any third party provider of cell-id info.

For geofencing application on the smartphone there are no external requests at all. Everything is locked within the phone. Rule checking is not complex anymore than checking for new incoming SMS or some like that. We can expect that such process would be a much more battery friendly than the existing ones. What is the trade off?

Obviously, it is a precision. Getting location via Cell ID is not such a precise as getting the same data via GPS. And of course translating true geo data into Cell's location could be done with the some approximation only. But the reasons here are very transparent – for the most of services geo fencing does not require high precision. But in the same time it solves the battery drain problems.

We can see here some paralles with the above mentioned SpotEx service. Geo fencing could be defined in the terms of

network nodes proximity [13]. Actually, the third party application there depends on the list of network nodes. Wi-Fi access identification plays a role of cell id. And base of Wi-Fi nodes is used as a foundation of geofencing.

As per existing geofencing services from mobile operators, we can mention Sprint GeoFencing [14]. It allows the developer to define an area to determine if a Sprint device is inside or outside the defined area.

And we think that geo-fence API for telecom is the most promising area for such kind of services. They are just starting. For example Sprint Geofence API allows the developer to define an area to determine if a Sprint device is inside or outside the defined area. [6]. By our opinion, operators need join forces and create some common standards in this area. Existing standards like Parlay do not cover this set of possible services. But services are in hot demand and especially now, when operators are loosing market to Internet companies. Geofence is actually an area where operators can gain their natural advantages.

## V. Conclusion

The purpose of this article is to find models for telecom operators to enable them to compete with Internet companies. In this paper we analyze several approaches for geofence related services implementations. This paper introduces a new way for sharing geofence data and approximates them with Cell ID data. Also we introduce a special form of SUPL simulation, especially oriented to operators. Proposed models let telecom operators replace internet companies in LBS applications. Also this article highlights the serious potential advantages of operator's based API for geofence tasks.

## References


[1] Munson, Jonathan P., and Vineet K. Gupta. "Location-based notification as a general-purpose service." Proceedings of the 2nd international workshop on Mobile commerce. ACM, 2002.

[2] Küpper, Axel, Ulrich Bareth, and Behrend Freese. "Geofencing and Background Tracking–The Next Features in LBSs." Proceedings of the 41th Annual Conference of the Gesellschaft für Informatik eV. 2011.

[3] Goze, T., Bayrak, O., Barut, M., & Sunay, M. O. (2008, August). Secure user-plane location (SUPL) architecture for assisted GPS (A-GPS). In Advanced Satellite Mobile Systems, 2008. ASMS 2008. 4th (pp. 229-234). IEEE.

[4] http://openmobilealliance.org/about-oma/work-program/location/ Retrieved: Nov, 2013.

[5] Sayed, Ali H., Alireza Tarighat, and Nima Khajehnouri. "Network-based wireless location: challenges faced in developing techniques for accurate wireless location information." Signal Processing Magazine, IEEE 22.4 (2005): 24-40.

[6] Engine, Mobile Simulation. "10.6 Main Interactions." Private Public Partnership Project (PPP): 135.

[7] Sneps-Sneppe, M., & Namiot, D. (2013). Smart cities software: customized messages for mobile subscribers. In Wireless Access Flexibility (pp. 25-36). Springer Berlin Heidelberg.

[8] Trevisani, E., & Vitaletti, A. (2004, December). Cell-ID location technique, limits and benefits: an experimental study. In Mobile Computing Systems and Applications, 2004. WMCSA 2004. Sixth IEEE Workshop on (pp. 51-60). IEEE.

[9] Namiot, D., & Sneps-Sneppe, M. (2013, March). Wireless Networks Sensors and Social Streams. In Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on (pp. 413-418). IEEE.

[10] D. Namiot and M.Sneps-Sneppe."Local messages for smartphones". In Future Internet Communications (CFIC), 2013 Conference on, pages 1–6. IEEE Conference Publishing, 2013 DOI: 10.1109/CFIC.2013.6566322

[11] Specifications, WARE Open. "11 Location Server Open RESTful API Specification." Private Public Partnership Project (PPP) (2012): 157.

[12] http://www.travelsim.lt/en/cellid-telemetric-service Retrived: Nov, 2013

[13] D. Namiot and M. Sneps-Sneppe. "Geofence and Network Proximity", Internet of Things, Smart Spaces, and Next Generation Networking Lecture Notes in Computer Science Volume 8121, 2013, pp 117-127, DOI: 10.1007/978-3-642-40316-3_11

[14] Sprint Geofencing http://developer.sprint.com/dynamicContent/geofence/ Retrived: Nov, 2013