3^{rd} Party Geolocation Messaging: A Positioning Enabler Middleware for Realizing Context-aware Polling

Mohamed Salem
Service-centric Networking
Telekom Innovation Laboratories
Berlin, Germany
Email: mohamed.salem@telekom.de

Bersant Deva
Service-centric Networking
Telekom Innovation Laboratories - TU Berlin
Berlin, Germany
Email: bersant.deva@tu-berlin.de

***Abstract*—Next generation Location-based Services (LBSs) are coined by a proactive user-interaction as well as a cross-referencing target relationship. In a proactive user-interaction, a service continuously keeps track of a user's locations and performs arbitrary actions upon entry or exit of a certain region. A cross-referencing target relationship implies that the LBS user and the target are not identical, which is the case, for instance, when a 3^{rd} party subscribes for certain location events on behalf of the tracked user, i.e. in a *Location-based Advertising* scenario. In order to efficiently realize sophisticated proactive, cross-referencing LBSs, a number of challenges need to be surmounted. Unfortunately, existing research within this respect is still rudimentary and a huge potential still remains untapped. Proactive LBSs involve continuous background tracking to know the location whereabouts of a user as well as process it in order to detect useful information in the vicinity. However, this results in severe battery drainage, hence shaping the main barrier for the realization of such services. Moreover, continuously tracking a user's location raises major privacy concerns, especially in case of cross-referencing LBSs. In this paper, we present the *Positioning Enabler* service middleware platform which provides a set of functionalities and APIs for enabling battery-efficient background tracking of single and multi-targets in a proactive manner, enabling the realization of sophisticated LBSs based on continuous background tracking and geofencing. Furthermore, it supports 3^{rd} party *Geolocation Messaging* while implementing certain location privacy measures for not invading user's privacy. The service middleware is used in the implementation of a *Context-aware Polling* mobile application for smart cities, *FlashPoll*.**

Keywords*— proactive LBS, location-aware middleware, geolocation messaging, positioning enabler, background tracking.

I. Introduction

Location data is permeating the entire mobile space and the expansion of Location-based Services is outstripping expectations. With the mobile industry being invaded with an enormous number of apps incorporating location-enabled features, among which are *Foursquare*, *Gowalla*, *Yelp*, *Instagram*, *Groupon*, *Loopt*, and *Shopkick*, it is indisputable that Location-based Services (LBSs) have entered the mass-market. LBSs are services which utilize the knowledge of geographic locations of a mobile device or several devices which are in the matter of fact users (usually referred to as *targets*), either to enhance existing applications, such as context-aware applications where location is one of the main context pillars, or to provide novel application services based on location information [1]. According to [2], LBSs can be classified into different categories depending on their *user or service interaction model*, *user and target relationship*, *plurality*, *infrastructure*, and *environment* they are used in. *Service or user interaction* describes whether the LBS is used in a *reactive manner*, pulling information as soon as a user or service is actively requesting it, or if the information is *proactively* pushed to the service as soon as certain conditions are met. The *user or target relationship* is called *self-referencing* if the service is relying only on the location of the requesting user itself, and *cross-referencing* if the LBS is based on locations of other users. Concerning the *plurality*, it can make use of the location of a single user *(single-target)* or of many users' locations *(multi-target)*. LBS are said to have a *central infrastructure* when the whole location information is being collected and computed on a central infrastructure component, or *peer–to–peer* if the service communicates directly with each participating node to compute location information.

Inspired by the aforementioned LBSs classification, the next generation of LBSs is based upon *proactive user interaction*, proactive LBSs, which persistently keep track of the users' locations in an unobtrusive manner and proactively send a notification about potentially useful information in the vicinity according to location events pre-subscribed for [3]. Such location events are spatial regions defined by *geofences*, which are in the matter of fact virtual perimeters denoting real-world geographic areas (i.e. circular or polygon shaped). Another pillar of next generation LBSs is 3^{rd} party tracking, i.e. a *cross-referencing target relationship*, which implies that an LBS user's own position is not processed but rather that of another target. This is in contrast to traditional *self-referencing LBSs* where the user and target are identical. The key applications driving the LBS market according to [4], e.g. Location-based advertising, social networking etc., are all categorized as next generation LBS. In order to achieve such services, i.e. *proactive* and *cross-referencing LBSs*, a number of challenges need to be overcome. A successful LBS running on a mobile device should not drain the phone's battery, otherwise it would harm the LBS user rather than provide an added-value. An LBS thus must be designed to minimize power consumption, especially if the service is to run persistently in the back-

MOBILWARE 2013, November 11-12, Bologna, Italy

DOI 10.4108/icst.mobilware.2013.254141

ground. Hence, the realization of proactive LBSs which runs for hours or days, implies particular consideration of battery consumption. Continuous tracking of users' locations raises major implications with respect to privacy, which in the matter of fact plays a crucial role in the applicability and acceptability of an LBS application, especially in case of 3^{rd} party tracking. However, users expect that new LBS applications will improve their lives without invading their privacy, which is hence an inevitable requirement for ensuring long-term success of next generation LBSs. The third major challenge in the realization of efficient LBSs is the accuracy with respect to positioning of users for *Geolocation Messaging*. *Geolocation Messaging* allows the delivery of messages to a plurality of mobile device users at a particular geographic location. These messages can be delivered either upon entering, leaving or after a defined length of stay at a certain location. The major challenges with respect to *Geolocation Messaging* lie in the aspects of message addressing, message matching and delivery. This involves the design of a suitable messaging approach which enables the detection of entry or exit of a user to a geofence where a notification would be necessary.

In this paper we discuss the design, implementation and deployment of a *Context-aware Polling* mobile application for smart cities; *FlashPoll*[1]. This tool enables public municipalities to leverage citizen involvement in the decision-making process in urban development of their cities by sending them contextual matching polls. While we present *FlashPoll* as an application scenario, the general aim of this work is to tackle the three aforementioned challenges associated with the realization of next generation LBSs. The *Positioning Enabler Platform*, presented by the authors in previous work [5][1], is a middleware service platform that provides a set of functionalities and APIs for enabling battery-efficient positioning of single and multi-targets in both a reactive as well as a proactive manner, enabling achievement of simple LBSs as well as more complex ones based on background tracking and geofencing. In this work, the functionality of the *Positioning Enabler Platform* is enhanced by more intelligent methods for realizing highly efficient continuous background tracking and geofencing. Furthermore, the users' privacy with respect to their sensitive location information is incorporated. Ultimately, the 3^{rd} party *Geolocation Messaging* functionality along with corresponding interfaces is realized.

This paper is structured in six sections including the foregoing introduction. Section II presents the existing work in the area of proactive LBSs with respect to battery efficiency, user privacy and *Geolocation Messaging* . The architecture of the *Positioning Enabler* is described in Section III, along with its functionality with respect to background tracking, position management and *Geolocation Messaging* . As the actual and potential markets for LBSs grow, so too does the need to address the implications for consumer privacy. The user privacy in LBSs are discussed in Section IV. The Context-aware Polling application scenario, *FlashPoll*, is presented in Section V, whereas Section VI concludes the paper.

II. Related Work

The presented Context-aware Polling application scenario serves as a core example of proactive LBSs which involves 3^{rd} party *Geolocation Messaging* . In this scenario the public municipality sends polls to location areas, these polls are received by citizens who enter or leave that area. For the realization of such an application, it is necessary to permanently track the locations of citizens and correlate their position fixes with respect to the geofences. In this section, the relevant work on background tracking, user privacy and *Geolocation Messaging* is presented.

A. Location background tracking

The significance of battery-efficient LBSs depends on the usage patterns. In this respect, an important parameter is the duration a service is supported to be running on a mobile device. The power consumption for different types of LBSs are classified in [6] and a factor indicating the impact on the battery life time compared to a stand-by battery consumption power is presented. According to the author, *Proactive location-based search* services are designed to run for hours or days and hence consumption of such services is medium to high. Therefore, it is essential that they consume a minimal amount of power. Background tracking is essential for realizing proactive LBSs. In practice, it is the continuous tracking of a target as a background process on a multitasking-enabled mobile device, either when this device is idle or while other applications run in the foreground. This implies that the target's current position is always known. Positioning being the fundamental prerequisite for realizing continuous background tracking, heavily impacts the battery consumption levels. Among the different available positioning methods, the most commonly used is GPS. It serves as the most accurate method for positioning, though it is accused of its poor functionality indoors and in dense urban areas. Moreover, GPS suffers from a long acquisition time, Time-to-First-Fix (TTFF), and most importantly it has considerable power consumption making it not the most favourable method for background tracking. As the aforementioned drawbacks present serious barriers for the success of proactive LBSs, different variants of positioning technologies have been created, which use a combination of the common positioning methods on smartphones, i.e., GPS, WiFi and Cell-Id positioning. These positioning methods have different performance against the attributes of positioning systems for ubiquitous computing investigated by Hightower et al. [7], which includes accuracy, availability, power, precision and TTFF. An overview of the performance of different positioning methods with respect to the aforementioned attributes is presented in [8] [9].

The location background tracking strategy presented in this paper exploits the combined information from several positioning technologies (i.e. GPS, Cell-Id and WiFi), utilizing their different characteristics in terms of energy consumption, accuracy, precision and availability to energy-efficiently realize continuous location tracking without diminishing accuracy. Our approach aims at using the positioning method with the least energy consumption by dynamically switching between different available methods and incorporating dynamic user context-information, such as the distance from the tracked user to the target geofence as well as his real-time movement activity (i.e. being in a vehicle, walking or standing still).

[1] http://www.flashpoll.eu

B. User privacy in LBS

The privacy concerns with respect to the tracked targets within *proactive cross-referencing* LBSs, especially in 3^{rd} party tracking, puts forward another major limitation to the growth of such kinds of applications. An important question is how much privacy protection is necessary. In practice, full privacy is clearly impossible as long as communication takes place. In the literature, there exists several approaches to protect the location of users, among which are *privacy policies* and *anonymization* approaches. *Privacy Policies* imply that targets specify the way their location data can be processed. On the other hand, *anonymization* approaches provide computational counter measures to hide a target's true identity with respect to disclosed location information. Within this scope, it can be distinguished between techniques of data or identifier abstraction. However, they all have in common the goal of minimizing the information disclosed as well as avoiding disclosure of unnecessary information. An extensive overview about location privacy is presented in [10]. The authors in [11] propose a mechanism called *cloaking*, which conceals a user within a group of *k* people. A user is considered to be *k-anonymous* if he is indistinguishable from at least *k - 1* other users. This method, however, has a negative impact on the spatial accuracy, as any of the users within the disclosed area could have been the user. Furthermore, they have considered reducing the accuracy of disclosure timestamps, reducing temporal accuracy. The drawback of this approach lies in its reduced spatial and temporal accuracy, which heavily affects the QoS and hence disqualifies it for our scenario. Another approach is the *mix zones* [12], where the infrastructure provides an anonymity service by delaying and reordering messages from subscribers within a mix zone to confuse an observer. Anonymization is realized by associating pseudonyms with the location information to protect target identities. The authors in [13], present a model which involves path segmentation and suppression for anonymization of users' tracking. In contrast to anonymization techniques, which have the objective of concealing target identities, the authors in [14] present a model for *obfuscating* location information. In this approach, the target identity is known, however the location accuracy is reduced as far as the LBS application requirements can handle.

To which extent an anonymization mechanism is useful strongly depends on the type of LBSs. Cloaking of location data is sufficient for reactive self-referencing LBS with low or medium requirements on data accuracy. On the other end of the spectrum, *mix zones* can be suitable for proactive self-referencing LBSs with high accuracy requirements and associated with a bounded application area. Location information *obfuscating* can be suitable in case the LBS application can handle medium location accuracy.

C. Geolocation Messaging

It enables sending of messages to targets when they are present in a certain geolocation and works by setting up geofences around locations of interest. Various messaging systems have evolved during recent years, which usually provide some sort of server architecture and clients that send and receive messages and connect to these servers. The *Publish/-Subscribe* is a well known event-based messaging paradigm for an asynchronous communication between publishers and subscribers. In practice, publishers send notifications to the notification system, which in return will be issued to the matching subscribers, interacting with each other via an event notification system with message buffering. The approach by Chen et al. [15] adds location information to publish/subscribe systems by introducing the concept of spatial events. In addition, it shows a spatial subscription model allowing two types of predicates, the *within* and the *distance* predicate. While the former evaluates if a user is inside a predefined zone, the latter evaluates if users are close to each other. In order to achieve high performance for the subscriptions filtering, a *spatial matching engine* that uses data structures and algorithms that are optimized for spatial indexing and search, e.g. R-trees, is described. However, this approach is not applicable in our scenario as it relies on only two types of predicates which limits the accuracy for matching of users to geolocation events. Egenhofer et al. [16] presents categorization of binary topological relations between regions, lines and points. 8 predicates are used to describe all binary spatial predicates between two regions: A *disjoint* B, A *covers* B, A *isCovered-by* B, A *equals* B, A *touches* B, A *coversAndTouches* B, A *isCoveredAndTouched* B, and A *intersects* B. The message matching and delivery approach realized by the *Positioning Enabler* is inspired by the approach presented by Chen et al. and Egenhofer et al.

III. Positioning Enabler Middleware for Context-aware Messaging

The *Positioning Enabler* is a middleware service platform that provides a set of functionalities and standardized APIs that can be utilized for realizing a comprehensive range of LBS applications. The main functionality of this platform is two-fold; primarily it tracks the positions of targets by implementing a power-efficient background location tracking approach for the realization of *proactive* and *cross-referencing* LBSs. Secondly, it analyzes the collected location information in real-time providing services to targets based on their locations and pertinent to the specific LBS application subscribed for. In addition to its major functionalities, the middleware significantly contemplates various user privacy measures, considering *privacy policies* and *anonymization approaches*, so as to ensure that the users' privacy is not invaded through continuous location tracking. Furthermore, it provides interfaces for 3^{rd} party entities, hence enabling the achievements of 3^{rd} party *Geolocation Messaging* . In this work, the *Positioning Enabler* is used to realize a Context-aware Polling scenario, which is implemented using the functionalities and APIs provided by the platform. In this scenario, the purpose of the platform is to enable a 3^{rd} party to send messages to registered and subscribed mobile clients that are present in a certain geographic area at a certain time. This geolocation is specified along with the message when using the 3^{rd} party interface, which may be a circle, an ellipse, a rectangle, or a polygon.

A. Architecture

Figure 1 highlights the functional entities of the *Positioning Enabler* middleware service platform. As it can be shown, the platform comprises a set of fundamental functionalities and APIs required for realizing *proactive* and *cross-referencing*

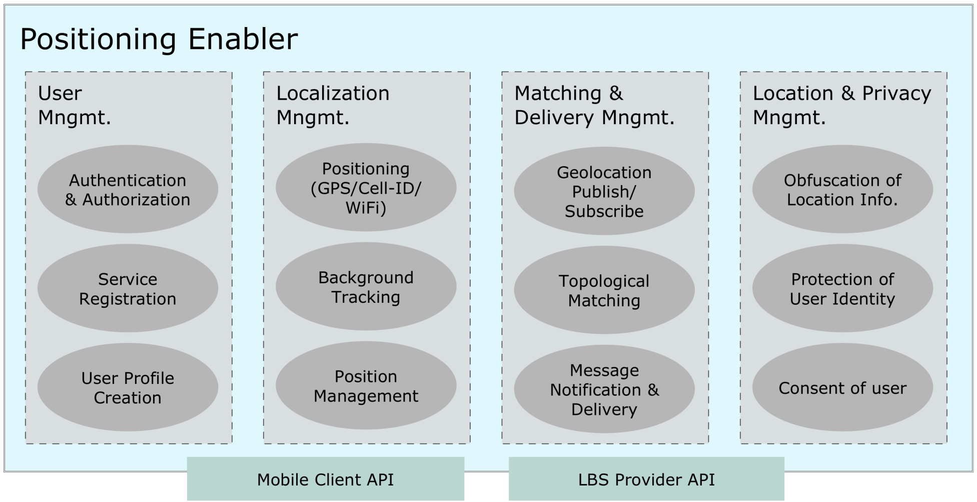

Positioning Enabler
User Mngmt.
Authentication & Authorization
Service Registration
User Profile Creation
Localization Mngmt.
Positioning (GPS/Cell-ID/ WiFi)
Background Tracking
Position Management
Matching & Delivery Mngmt.
Geolocation Publish/ Subscribe
Topological Matching
Message Notification & Delivery
Location & Privacy Mngmt.
Obfuscation of Location Info.
Protection of User Identity
Consent of user
Mobile Client API
LBS Provider API

Fig. 1. Positioning Enabler Middleware Architecture

LBS. From a functional perspective, the service platform is based on four major functional components:

- **The User Management Component** is mainly responsible for the mobile application users accessing the platform to subscribe for a certain supported LBS application.
- **The Localization Management Component** is basically responsible for the efficient positioning of users as well as tracking them using different positioning methods.
- **The Matching & Delivery Management Component** is apparently the *broker* of a publish/subscribe system utilizing real-time collected location information and sending notifications after matching of users and geolocations.
- **The Location & Privacy Management Component** is responsible for ensuring the privacy of users' locations.

The *User Management* component consists of the essential functionalities that are necessary for managing different users with respect to the offered services, and acts as the first entry point for mobile application users. The *User Management* is crucial for authorizing and authenticating the users prior to the subscription to an LBS application service; *Authentication & Authorization*. Other functionalities of this component include the *Service Registration* and *User Profile Creation* which enable users to register to specific LBSs supported by the platform, via *Apps* on their mobile clients, and hence creating user profiles. Different applications developed for this middleware are distinguished by unique service ID namespaces to achieve multi-tenancy.

The *Localization Management* component is indispensable for determining the positions of tracked users by relying on the *Positioning* functionality which supports multiple positioning methods and technologies, i.e. GPS, Cell-Id and WiFi Positioning. The *Background Tracking* functionality controls and decides for the most efficient positioning method to be used, it strives to provide a high QoS LBS while not draining battery power. In order to achieve this a number of context parameters are taken into consideration, more detailed information about the background tracking logic is explained in the following. In addition, the *Position Management* functionality is tightly coupled with the background tracking and is another key concept for enabling proactive LBSs. It is essential for wisely deciding upon when to send a location update, as sending updates too often costs communication and computational overheads, while sending updates too rarely might cost a location event not to be detected in due time.

The *Matching & Delivery* component is in charge of aggregating, processing and interlinking geographic positions of target locations and subscriptions of 3^{rd} parties via the *Geolocation Publish/subscribe* and the *Topological Matching* functionalities. The latter incorporates spatial matching of predicates and is explained in more details later in this paper. Acquainting users or sending them messages in case a predefined location event has been fulfilled is achieved by the *Message Notification & Delivery*.

The *Location & Privacy* component enables the management of user tracking permissions against 3^{rd} party entities, *Consent of user*, as well as for protecting users' privacy. This is realized by the anonymization of user identities, *Protection of User Identity*, or through pseudonymization and concealment of users' locations; *Obfuscation of Location Information*.

The *Positioning Enabler* provides two APIs; the *Mobile Client API* to mobile application users and the *LBS Provider API* to 3^{rd} party entities (LBS providers). This concept has been driven by the fact that the LBS market has developed into a long-tail market with various niche applications and services making use of location information. Moreover, advancements in the Mobile Internet market, e.g., smartphones, Web 2.0 paradigm, concept of mashups and App Stores, have facilitated entry of single developers, start-ups and larger companies into the LBS supply chain in addition to the already existing huge user base. The following APIs are supported:

- **Mobile Client API**: The *Positioning Enabler* offers a REST interface through which it can send notifications and messages to subscribed clients. The platform sends users notifications when a location event is fulfilled or sends a message that a 3^{rd} party has published onto the system. For the realization of background tracking and position management the platform sends push notifications, using this API, to the mobile client to switch between different positioning methods as well as different update strategies. On the other end of the spectrum, the LBS mobile application users are authenticated, authorized and registered to an LBS service on the platform using this API. Furthermore, mobile clients send their position fixes using the *Mobile Client API*. The sent location information of clients is solely used by the middleware platform to match users with geolocations, and is never exposed to any other application nor to the LBS provider.

- **LBS Provider API**: One of the radical reasons for the failure of the 1^{st} LBS generation is owed to mobile network operators not incorporating 3^{rd} parties into the LBS chain. As a result, the emergence of open and competitive long-tail markets for LBS were obviated. In order to cope with advancements in the LBS market, the *Positioning Enabler* provides an API to 3^{rd} parties allowing them into the LBS supply chain, i.e. Municipalities in Location-based E-Government Applications, Retail Shops in Location-based Advertising Applications as well as other users in Location-based Social Networking or Child Tracking Applications. This API supports realization of 3^{rd} party *Geolocation Messaging*, by enabling parties to subscribe for certain location events on behalf of targets so that a notification alert or message is delivered to targets as soon as the pre-subscribed location event is fulfilled. The geolocation area is specified along with the message, and it may vary between a circle, a polyline or a polygon.

B. Background Tracking & Position Management

In order to enable services like *Geolocation Messaging* an efficient background tracking approach is required. The *Positioning Enabler* exploits several positioning methods based on a number of context parameters in order to determine the position of a user efficiently. Furthermore, the requirements on accuracy, TTFF or battery efficiency of the 3^{rd} party service are taken into consideration. Additionally, low level position management is essential for wisely exchanging position fixes between two or more entities. It is basically an efficient way of keeping location information up-to-date, irrespective of using it locally, transmitting it to a central server or sharing it among different peers. There exist different methods for exchanging position fixes between the mobile device and the Positioning Enabler Platform: *Polling*, *Time-based*, *Distance-based*, *Zone-based*, *Activity-based*. When polling the Positioning Enabler requests the current position fix of a mobile device. It can be used on a periodic basis, according to a caching strategy or when an immediate location update is required. In a time-based approach the device sends a position update if a pre-defined time interval, i.e. update interval, has elapsed since the last position update. Time-based location update strategies tend to be inefficient if a target is rather stationary whereas it tends to become inaccurate when the target is moving at higher velocities, because the location information becomes outdated at a fast pace. Another method describes is the distance-based where the mobile device sends a position update if the distance between the last reported position and the current position exceeds a pre-defined threshold. While not having the disadvantages of the periodic strategy, the dilemma is that to determine the distance from the last update, usually its location has to be determined from time to time. The zone-based approach, also known as geofencing, initializes a position update as soon as a user leaves a pre-defined geographical area, called zone. A zone could be described by a circle with center coordinates and a radius as well as by other geographical shapes, e.g. polygons. Finally, the activity-based approach sends a location update as soon as the mobile device of the user's recognizes a change of activity. This approach has the advantage that unnecessary location updates are avoided when the device is not used and in still mode. However, when a user changes his activity very often, redundant location updates are sent.

The are mainly three positioning methods used by the *Positioning Enabler* which are GPS- , WiFi- and Cellular Network positioning. GPS is a highly accurate device-based positioning method which computes the user's position by receiving the broadcast signal of at least four or more GPS-satellites. WiFi-based positioning relies on hardware MAC-addresses of WiFi access points (APs). The position of these APs is related to a reference position which is provided for example by using GPS. This position is stored along with the MAC address of the access point. Whenever a device senses this MAC address the WiFi-based positioning method is able to detect the user's position. Cellular Network-based positioning (often referred as Cell-Id positioning) is similar to the WiFi-based method. Instead of WiFi access points cell identifiers (Cell-Id) are used. Perceptibly, none of the shown approaches provides a single best solution which requires the combination of them to provide a reasonable background tracking for 3^{rd} party services.

The combination of the low level position management alongside the smart selection of the most appropriate positioning method is crucial for providing an efficient background tracking approach. Furthermore, the decision-making on which positioning method and location update strategy should be used is chosen by incorporating additional context parameters. These context parameters include the distance to the target, the current state of the user's environment (indoor, outdoor) and the state of the user's activity (still, moving slow, moving fast). In order to use these context parameters further computation is required from the mobile client device.

The distance to the closest target is crucial for the background tracking approach. If a target is far away from the user the background tracking does not require an accurate position of the user which would lead to rather using Cell-Id or WiFi positioning instead of GPS. Depending on adaptable distance thresholds, the mechanism switches between different positioning methods. When a user is very close to the target, GPS positioning is switched on which allows a more precise matching between the target and the user. If the user is still far away a zone-based location update strategy is used as it allows the user to move independently within a defined secure zone without further communication needed. If the user comes close to the vicinity of the target the positioning method can be switched to a more accurate one. In this case the usage of the time-based approach is required where the time interval between location updates is short and the position needs to be up-to-date. The user's environment is also crucial for choosing the appropriate positioning method. By detecting if the user is in an indoor or outdoor environment unreasonable decisions can be avoided. For example, when a user is indoors the use of GPS is pointless and Cellular Network positioning would not be recommended. Therefore, WiFi positioning should be taken into consideration. Additionally, the user's direct activity information can be used for switching between positioning methods and choosing the fitting location update strategy. With the capabilities of modern smart devices, it is able to detect whether a user is moving fast (in a vehicle or by bicycle), moving slow (walking) or not moving at all (still). Obviously this is of relevance for the background tracking mechanism which leads to several implications that can be made by using this context information which are shown in Table I.

TABLE I. ACTIVITY-BASED RECOMMENDATIONS

Activity State	**Recommended Positioning Method**	**Recommended Location Update Strategy**
Not Moving / still	—	Time-based
Moving slow / walking	WiFi/GPS	Zone-based
Moving fast / vehicle or bicycle	Cellular Network	Zone-based Distance-based

For example, when the user is not moving tracking is not required and thus no positioning method is recommended. However, a location update should be send from time to time with a larger value for the time interval. On the contrary, when the user is moving fast in a vehicle a positioning method with a coarse grained accuracy and with a low TTFF might be sufficient and can be realized by using cellular network positioning. In this case a zone-based or distance-based approach are recommended.

Finally, the position method and location update strategy heavily depends on the requirements of the 3^{rd} party service requirements. For example, if a service demands a high accuracy at all times, the above mentioned mechanisms can be omitted and GPS is used. If battery efficiency is of highest importance for a service then the background tracking can solely rely on cellular network positioning.

C. 3rd Party Geolocation Messaging

The *Geolocation Messaging* entity allows 3^{rd} parties to send a various form of messages, e.g. advertisements, instant messages or polls, to multiple users at a pre-defined geographic area. This area is matched against the user's location which is tracked by the *Positioning Enabler*. The message is delivered upon a match between these two geographic locations. This process faces three major challenges. Compared to traditional messaging services such as Email or SMS, *Geolocation Messaging* demands a geographically based addressing schema. Due to the characteristics of *Geolocation Messaging* an event-based messaging approach is required which enables the delivery of messages to a set of previously unknown number of receivers. This event-based messaging system requires an efficient approach for matching between desired message delivery area of the publisher and the location of the recipients. Figure 2 depicts the different geographic areas where geolocation messages are sent to. Furthermore, it shows the flexibility of defining a variety of shapes in order to meet a diversity of requirements from 3^{rd} parties.

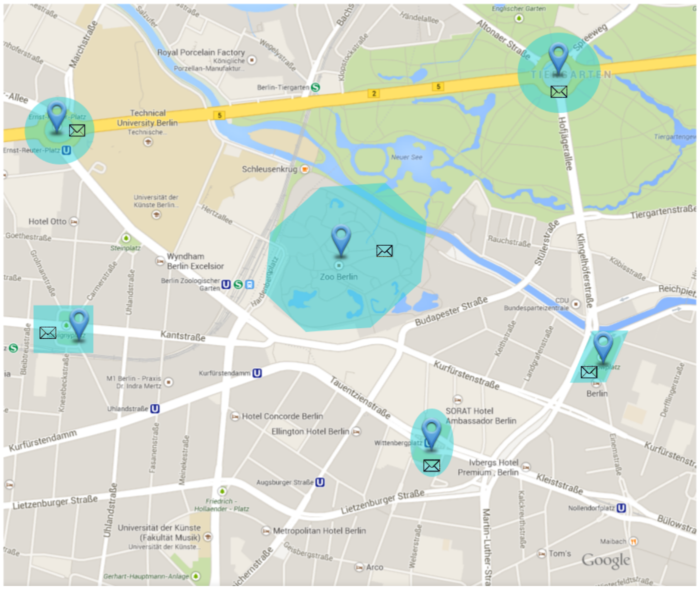


Fig. 2. Depiction of Geolocation Messaging Areas in Berlin - Charlottenburg

1) Geometric Addressing: The *Geolocation Messaging* approach uses geometric addressing for defining the geographic area for message delivery. It allows more flexibility in comparison to symbolic addressing, e.g "country/city/municipality/square". In order to provide a great variety of shapes for geolocation message areas a number of different data structures can be supported which are shown in Table II. Circular areas are represented by a point with WGS84 coordinates and a radius surrounding this point. Lines can be defined with the polyline type which can be used for describing geographic location such as streets or sight lines. In addition, polygons cover an area within a set of coordinates.

TABLE II. ADDRESSING SCHEMES
LAT: LATITUDE, LON: LONGITUDE, R: RADIUS

Geolocation Address Type	**Structure**
Circular	(lat$_{x1}$, lon$_{y1}$, r)
Polyline	(lat$_{x1}$, lon$_{y1}$) , (lat$_{x2}$, lon$_{y2}$) , (lat$_{x..}$, lon$_{y..}$)
Polygon	(lat$_{x1}$, lon$_{y1}$) , (lat$_{x2}$, lon$_{y2}$) , (lat$_{x3}$, lon$_{y3}$) , (lat$_{x..}$, lon$_{y..}$), (lat$_{x1}$, lon$_{y1}$)

2) Geolocation Publish / Subscribe: Messages to geographic locations are characterized by a one-to-many relationship. This means by definition that one publisher sends a message which can have multiple receivers. In *Geolocation Messaging* the users receive a message when they enter or leave the message area. This is an asynchronous process and message sending and receiving is decoupled. In order to fulfill these characteristics, the *Geolocation Messaging* approach uses a publish / subscribe mechanism with location-based filtering for subscriptions. Each location update by the user is considered as a subscription to the current location. Messages are published to a geographic area by using the above mentioned geometric addressing schema. Notifications are sent to the callback address which is contained in the subscription model. The publisher is able to define an entry or exit event upon which the user should receive the message.

3) Message Matching and Delivery: The filtering and matching of messages in *Geolocation Messaging* makes use of spatial predicates and distance computation. The actual matching method depends on the geometric addressing scheme the publisher uses. Polylines and polygons describe whole geographic areas. These can be matched by using spatial predicates such as *equals*, *within*, *overlaps* or *disjoint*. Once a user's location update and thus subscription intersects with a message location, the user is notified. On the other side, the circular areas get matched by a distance-based approach. The distance between the location of the user and the point of the circular message area is determined. If the radius of the circular area is larger than the computed distance the message matches and the user gets the notification. The notification is sent immediately after the message is matched to a user's location. However, it depends highly on the application scenario when the message is displayed. In order to avoid the sending of unnecessary message payload and to increase the flexibility of the system, only a callback id is sent in the notification. This enables the 3^{rd} party to build the messaging service upon its own adjustable requirements.

IV. User Privacy Protection in Geolocation Messaging

Geolocation Messaging faces main challenges regarding the user's privacy with respect to the location-based background tracking. The mechanisms, which are introduced in the previous chapter, raise major concerns as it allows to detect the moving behavior of the user. For example, by knowing the full location track, a user's home and work environment or frequently visited places (shopping mall, restaurant, bar) could be easily detected. Even more personal information could be inferred when considering that locations are linked to the users' activity and health condition (fitness center, tennis court or hospital). The position of a user could not only help the interlinking between the user to a certain area but also to other users. Social relationships between users could be spotted by analyzing the group location context of multiple users. Regular meetings between users are detected by comparing two location tracks over time. In addition, the first-mentioned linking to areas can be combined with the group location context. For example, the combination of multiple user location tracks can lead to the assumption that a set of users belongs to the same family and lives in the same house. For this reason, it is crucial for the users' privacy that a set of mechanisms are considered when building such a service. This includes the obfuscation of the user's location so that the position is only as precise as required by the 3rd party. Additionally, several approaches are discussed which limit the background tracking and protect the user's information in the implementation of the platform. This protection also ensures the privacy of the users against breaching attackers of the platform.

A. Obfuscation of Location Information

The obfuscation of geographical coordinates increases the anonymity of users. As mentioned, the device delivers the most current position fix with the accuracy which depends heavily on the positioning method used. However, for certain application scenarios only a very coarse grained level of accuracy is required. In these cases the position can be modified by simply truncating the position fix information or by adding a randomized offset to it.

1) Truncation: Depending on the required accuracy level of the LBS provider the location information is truncated by rounding the values of latitude and longitude in the position fix. Considering the equatorial circumference of the planet earth (approx. 40075 km) and the characteristics of the WGS84 geographic coordinates the loss in accuracy by truncating decimal places of the coordinates can be determined. For example, by truncating the fourth decimal place of the longitude at the equator would lead to an accuracy loss of roughly $\pm 50m$. The truncation method can be adapted based on the LBS provider accuracy requirement and the accuracy level of the positioning method.

2) Randomized Offsets: Similar to the truncation process the randomized offsets increase the user's anonymity by modifying the position fix. The user's location is further obfuscated by adding or subtracting random offsets to the decimal places of WGS84 coordinates. In comparison to truncation, this methods allows a more fine grained obfuscation of user locations. The range of possible offsets and the decimal place to be edited can be varied according to the required accuracy of the 3rd party application.

B. Avoidance of Unnecessary Location Updates

The *Positioning Enabler* estimates the required location update frequency by taking into consideration multiple context parameters of the user's device, the zones of interest and the requirements of the 3rd party service, as shown in the previous chapter. Therefore, the reduction of communication between mobile clients and the *Positioning Enabler* platform enables an optimized workflow and retrieves the user's location only when necessary. This limitation leads to a fragmented track of the user, which enables a better privacy protection and ensures the feasibility of the 3rd party service.

C. Protection of the User Identity

The protection of the user identity is an important tasks for all services that keep track of the user's behavior. As mentioned above, the user's identity can be inferred by using its location and thus this information is worth protecting from external attackers. This can be achieved by the combination of a number of practices that enable a privacy-aware architecture.

1) Separation of User Profile and Location Context: Besides the location context, the *Positioning Enabler* also maintains a user profile with sensitive data for unique identification. This data should be stored in a specific database which is only responsible for the user management mechanisms. The location and other context information of the user could be stored separately in an optimized context database.

2) Pseudonymization of Users: In order to decouple these two databases the user's identifier could be stored in a pseudonymized manner at the context database. This can be accomplished by hashing the user's identifier with a secure hash function as shown in (1). This mechanism is also often used for storing passwords.

$$hash\_function(user\_id) = pseudo\_id \qquad (1)$$

The following example shows the pseudonymization using the SHA-256[2] hash function and an UUID[3] for the user identifier:

$$SHA256(3b56551c-1a6e-456b-b875-838e77bf907b)$$
$$= c465cc931b2c828648473b89be84e50b703f193476e6c54...$$

This protection mechanism has several implications for external attacks. Given that the external attacker is able to gain access on the secured user profile database, he will be able to identify the users connected to the database. However, the context information is not accessible. The same accounts for gaining access solely on the context database. The attacker will be able to know the context information of pseudonymized users, but cannot link to any profile information. However, given the case that the attacker gains access to both protected databases he would be able to infer between the profile and the pseudonymized context information by knowing the hashing functionality. This can be avoided by extending (1) with a generated secured random *salt* and adding it to the user's identifier in the hash function (2).

$$hash\_function(user\_id + salt) = pseudo\_id \qquad (2)$$

The attacker would need additional access to the *salt* parameter. The level of anonymity can be further increased by introducing a random *salt* per user and changing these salts in a chronological manner. These mechanisms ensure that an external attacker could gain access to both databases and would not be able to connect the user profile to its context information.

D. Consent of User

As stated the *Positioning Enabler* deals with sensitive information regarding the user profile and context information. The previous sections show that there are several mechanisms for building a privacy-aware platform and protecting this sensitive information. However, the 3rd party services and the platform cannot be provided without the user's consent (opt-in). Therefore the user should have an insight on the functionality of the background tracking. The application can only be privacy-aware if the user has full knowledge on and agrees to the use of her/his data. Additionally, current opt-in solutions only allow the user to decide whether to use the LBS application or not. More advanced approaches would allow the user to specifically determine the way of how the LBS application can use the user's information, e.g. the accuracy granularity or storage time of her/his data.

V. FLASHPOLL: CONTEXT-AWARE POLLING FOR CITIZENS

In order to show the applicability of the *Positioning Enabler* middleware as a 3rd party *Geolocation Messaging* platform we implemented the *FlashPoll Tool*, a context-aware citizen polling application which helps public municipalities and communities in the decision-making process. Currently, the main objectives of the tool are the enhancement of citizen involvement and the insight of the public opinion on urban development challenges. In urban development, most projects are related to physical places such as public squares, streets, parks or whole neighborhoods. This means that most questions on the public opinion are of concern to a limited set of citizens in the respective geographical area. For this reason, the *FlashPoll Tool* introduces *Context-aware Polls* which are delivered to contextual matching citizens in the vicinity of an area under development. Besides the content (description, questions and answers), the polls contain also enriched context information which defines the zone of interest of the poll (polling zone) and the time period of the polling phase. These additional context parameters help filtering the polls according to the geographic location and time. The *FlashPoll* application basically consists of the *FlashPoll Server* on the server side and of the *FlashPoll App* which is developed as a prototype for Google's Android platform on the mobile client side.

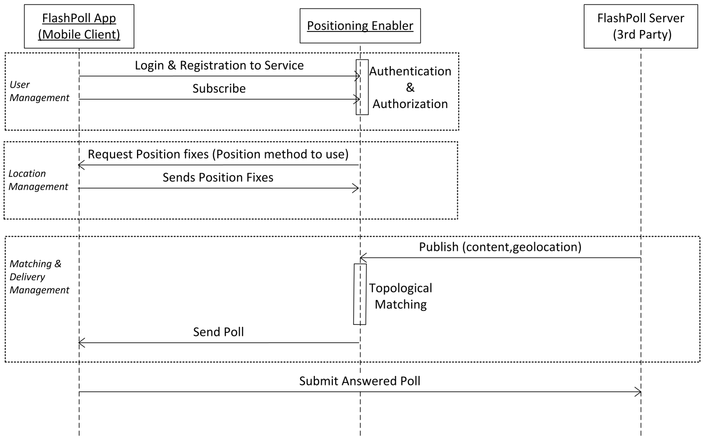
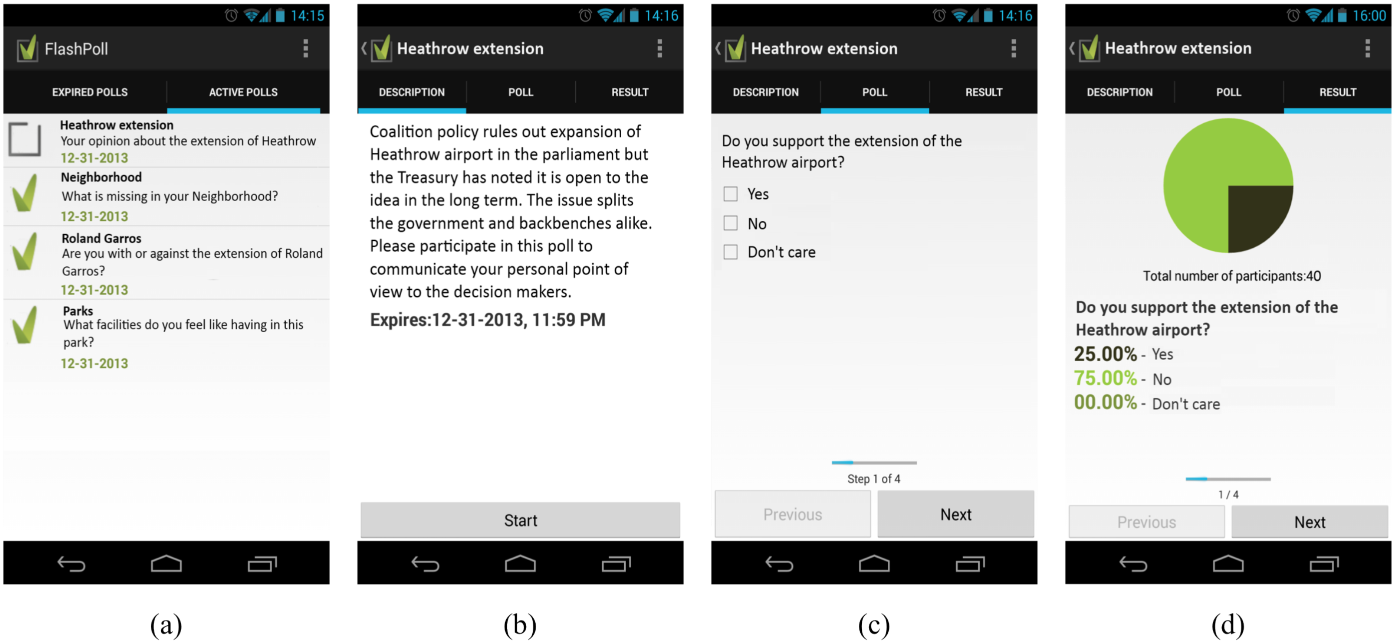


Fig. 3. Workflow of the *FlashPoll* App

Figure 3 presents the general workflow of the *FlashPoll* application. Using a mobile client application, a user first registers to a 3^{rd} party service, on the *Positioning Enabler*. The registration process is provided by using single-sign-on authentication & authorization login mechanisms by trusted 3^{rd} party providers such as *Google* or *Yahoo*. The platform currently supports *OpenID* and *OAuth 2.0* login functionalities. After the login process, the user subscribes to the 3^{rd} party LBS service, *FlashPoll* in this case, by sending an HTTP request through the *Mobile Client API*. The *Positioning Enabler* starts the background location tracking for a user upon his subscription. It first requests the mobile device to send its current position fix, and is then followed by informing it of the most appropriate positioning method to use as well as the positioning update strategy. These steps are repeated during the tracking process according to the location tracking approach described earlier in Section III-B. When the poll message is published by the *FlashPoll Server* to a geographic location (polling zone), the

[2] RFC4634

[3] RFC4122



FlashPoll

EXPIRED POLLS | ACTIVE POLLS

□ Heathrow extension
Your opinion about the extension of Heathrow
12-31-2013

✓ Neighborhood
What is missing in your Neighborhood?
12-31-2013

✓ Roland Garros
Are you with or against the extension of Roland Garros?
12-31-2013

✓ Parks
What facilities do you feel like having in this park?
12-31-2013

(a)

Heathrow extension

DESCRIPTION | POLL | RESULT

Coalition policy rules out expansion of Heathrow airport in the parliament but the Treasury has noted it is open to the idea in the long term. The issue splits the government and backbenches alike. Please participate in this poll to communicate your personal point of view to the decision makers.

Expires:12-31-2013, 11:59 PM

Start

(b)

Heathrow extension

DESCRIPTION | POLL | RESULT

Do you support the extension of the Heathrow airport?

□ Yes
□ No
□ Don't care

Step 1 of 4

Previous | Next

(c)

Heathrow extension

DESCRIPTION | POLL | RESULT

Total number of participants:40

Do you support the extension of the Heathrow airport?

25.00% - Yes
75.00% - No
00.00% - Don't care

1 / 4

Previous | Next

(d)

Fig. 4. Screenshots: (a) Overview of active & expired polls, (b) Poll description, (c) Poll question 1 / 4, (d) Overview of the accumulated result for this question

Matching & Delivery Component on the middleware spatially matches the poll against subscribed citizens. In this process, the polling zones are spatially defined by geographic coordinates using the widely used WKT[4] format. The performance of the topological matching process is highly dependent on the underlying spatial functionalities of the database. Spatial indexes are required for low query execution times on queries with even larger data sets. Additionally, the support for spatial predicates and geometry types are necessary for efficient query matching between polls and citizens. For this reason, the polling zones and the citizens location are separately stored in an optimized *PostgreSQL* database with *PostGIS* support. The poll matching is mainly based on the approach of *Geolocation Messaging* as described in chapter III-C. To cope with the requirements of municipalities for defining diverse shapes of polling zones, there are two types supported which are handled by the matching entity differently: Circular geofences and polygon geofences. The matching of circular geofences is realized by comparing the distance between the citizen's location and the point coordinate. If the distance is smaller than the radius of the poll zone, the citizen is considered to be in the vicinity of the poll and it is sent to the citizen. With polygon geofences the poll matching makes use of the spatial predicate *intersects* for matching citizens and polls. The main advantage of the polygon geofence is the flexibility for creating polling zones. With polygons real world environments such as squares, building complexes or neighborhoods can be defined and matched to users. However, the technical requirement for spatial predicates increases the computational complexity compared to circular geofences. Once a poll is matched to a citizen it is delivered to the *FlashPoll App* using the push notification service *Google Cloud Messaging* (GCM). Due to message load limitations of GCM notifications only a callback identifier of the poll is sent to the mobile application with which the poll content can be retrieved from the FlashPoll service. Furthermore, the *FlashPoll App* uses the activity context parameter providing a smart approach for displaying the polls. When the device owner and thus citizen is moving fast (ON_BICYCLE or IN_VEHICLE) the *FlashPoll App* will not disturb in a situation where the citizen is unlikely to answer. In addition, the mobile application is able to detect passersby by requiring a minimum amount of time for being in the vicinity of a matching poll. These measures try to avoid bothering notifications and only display polls to citizens that are capable of answering.

Figure 4 shows a first prototype of the *FlashPoll App*. In 4(a) a number of polls can be seen on an overview screen. Checked polls have already been answered, whereas unchecked ones are still open for answering. This screen differentiates between active and expired polls which are shown by tapping at the respective tabs. Once a user taps on a poll in the list the 4(b) shows up. This screen contains a title, a description and the expiration date and time to which the poll can be answered. By pressing the start button, swiping to the right or tapping the ”Poll” tab, the next screen is revealed which represents the actual poll answering process. Here the citizen is able to select, depending on the question type, one or multiple answers. Once all questions are answered by the citizen the results can be submitted to the *FlashPoll* service. Shortly afterwards, the citizen is able to immediately see the accumulated results of all participating citizens in a visualized form through a pie chart and the proportional percentage of each answer.

The *Positioning Enabler* and the *FlashPoll* tool introduce several mechanisms for protecting the citizen's privacy. The application demands the citizen's consent before it can be used through the login procedure. As described, by using trusted 3rd parties for authentication & authorization neither the *Positioning Enabler* nor the *FlashPoll* service keep login credential information of citizens which could be a poten-

[4] Well-known Text

tial risk towards attackers. After the login the application is fully available and the continuous background tracking is activated. However, the citizen is able to switch off the *FlashPoll* service by canceling the subscription when there is no current desire to use the service. This action will not only disable the background tracking process on the mobile application side, but also delete any contextual information of the citizen at the *Positioning Enabler*. The citizen's location context can only be retrieved by manually re-enabling the service. This mechanism provides a privacy-aware approach where the citizen has full control over the location tracking service. As described above, the *Positioning Enabler* holds the geographic location of polling zones and the location of citizens in a spatially optimized PostgreSQL database. In order to increase data protection this data is stored separately from the citizen's profile information in a pseudonymized manner. In addition, this simplifies the matching between polling zones and citizens as both matching parameters are accessible in the same database. The pseudonymization process is realized as described in the section IV by hashing the unique user identifier with an additional unique salt per citizen. The *FlashPoll* application is currently using the secure hash function SHA-256. Finally, it should be noted that the presented approach only stores the most recent location update of the citizen and does not record any privacy data-sensitive citizen specific long term location tracks.

VI. Conclusion & Future Work

In this paper we presented the *Positioning Enabler* service middleware platform which enables cross-referencing LBS applications. We demonstrate the functionality of the platform by applying a 3^{rd} party *Geolocation Messaging* approach. This approach uses an innovative and efficient background tracking mechanism which takes into consideration the available positioning methods, state-of-the-art location update strategies as well as additional context information of users and requirements of 3^{rd} party services. Furthermore, the paper tackled user privacy aspects in LBS applications, especially regarding the application scenario of *Geolocation Messaging*. Finally, we have implemented the *Context-aware Polling* application, *FlashPoll*, which leverages citizens' participation in decision-making processes for urban development in smart cities. In the future we plan to extend the context model of users beyond location and provide more customized matching, e.g. age, personal interests or other topics. The integration of open data such as public event information or POIs can further enrich context model of the published messages. Ultimately, the *Positioning Enabler* will be extended to a distributed system for providing its services in a scalable manner.

Acknowledgment

This work is carried out within the scope of the FlashPoll and FI-WARE Projects, which are funded by the EIT-ICT and the FP7 European Community for research respectively. The authors would like to thank Maija Poikela and the students Gabriel Pilz and Stefan Medack for their support.

References

[1] M. Salem, U. Bareth, and A. Küpper, "Positioning Enabler for realizing Location-based Community Services," in *Proceedings of the 2nd International Conference on Mobile Services (MS 2013)*. Silicon Valley, California, USA: IEEE, Jun 2013.

[2] A. Küpper, G. Treu, and C. Linnhoff-Popien, "TraX: A Device-Centric Middleware Framework for Location-Based Services," *IEEE Communications Magazine*, vol. 44, no. 9, pp. 114–120, Sept. 2006.

[3] U. Bareth, A. Küpper, and P. Ruppel, "geoXmart - a Marketplace for Geofence-Based Mobile Services," in *Proceedings of the IEEE 34th Annual Computer Software and Applications Conference (COMPSAC 2010)*. Seoul, South Korea: IEEE, Jul 2010, pp. 101–106.

[4] A. Zimmermann, "Vendor Survey Analysis: Four Issues Facing Location-Based Service Providers," *Gartner Market Analysis and Statistics*, 16 November 2010.

[5] M. Salem, P. Ruppel, U. Bareth, and A. Küpper, "X-centric Positioning: A Combination of Device-centric and Multi-RAT Network-centric Positioning Approaches," in *Proceedings of the 4th Intl. IEEE Workshop on Open NGN and IMS Testbeds (ONIT 2012)*. Anaheim, California, USA: IEEE, Dec 2012, pp. 1866–1871.

[6] M. B. Kjærgaard, "Minimizing the power consumption of location-based services on mobile phones," *IEEE Pervasive Computing*, vol. 11, no. 1, pp. 67–73, 2012.

[7] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," in *IEEE Computer*, vol. 34, 2001, pp. 57–66.

[8] M. Chen, T. Sohn, D. Chmelev, D. Haehnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky, "Practical Metropolitan-scale Positioning for GSM Phones," in *Ubiquitous Computing*, 2006, pp. 225–242.

[9] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A Framework of Energy Efficient Mobile Sensing for Automatic User State Recognition," in *Proceedings of the 7th international conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM, 2009, pp. 179–192.

[10] A. Grlach, A. Heinemann, and W. Terpstra, "Survey on location privacy in pervasive computing," in *Privacy, Security and Trust within the Context of Pervasive Computing*, ser. The International Series in Engineering and Computer Science, P. Robinson, H. Vogt, and W. Wagealla, Eds. Springer US, 2005, vol. 780, pp. 23–34.

[11] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*, ser. MobiSys '03. New York, NY, USA: ACM, 2003, pp. 31–42.

[12] A. Beresford and F. Stajano, "Location privacy in pervasive computing," *Pervasive Computing, IEEE*, vol. 2, no. 1, pp. 46–55, 2003.

[13] M. Gruteser, J. Bredin, and D. Grunwald, "Path privacy in location-aware computing," June 2004.

[14] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, H.-W. Gellersen, R. Want, and A. Schmidt, Eds. Springer Berlin Heidelberg, 2005, vol. 3468, pp. 152–170.

[15] X. Chen, Y. Chen, and F. Rao, "An Efficient Spatial Publish/Subscribe System for Intelligent Location-Based Services," in *Proceedings of the 2nd International Workshop on Distributed Event-based Systems, ACM Press*, San Diego, California, 2003, pp. 1–6.

[16] M. J. Egenhofer and J. Herring, "Categorizing binary topological relations between regions, lines, and points in geographic databases," National Center for Geographic Information and Analysis, Tech. Rep., 1994.