FlashPoll: A Context-aware Polling Ecosystem for Mobile Participation

Bersant Deva, Sandro Rodriguez Garzon, Axel Küpper
Service-centric Networking
Telekom Innovation Laboratories, TU Berlin
Berlin, Germany
{bersant.deva | sandro.rodriguezgarzon | axel.kuepper}@tu-berlin.de

***Abstract*—The rise of smart mobile devices created new opportunities for electronic and mobile participation. These devices provide means to sense the current environment citizens are situated in, which can be then used to carry out targeted opinion polls. The context-aware polling system *FlashPoll* takes advantage of this notion and enables the conduct of short and fast context-specific polls which are matched against the citizens' individual situation. Thus, the target groups are more focused, which increases the relevance of polls and results to the citizens and the executing authorities. This work presents a complete overview of the FlashPoll ecosystem, which comprises poll management tools for public authorities and mobile applications for participants. It has been tested in several field trials to prove the applicability of the approach.**

I. Introduction

Today, smart mobile applications have a major impact on how we deal with our daily environment and how we fulfill our needs. Mobile services have transformed the way we do ordinary things such as social communication, reading news, shopping or even requesting services such as banking, taxis or restaurant reservations. An increasing number of these services is taking the users' context information into consideration in order to provide an experience that is better suited to their environments [1]. Especially *Location-based Services* (LBS), as a subset of context-aware services, have emerged in recent years and became ubiquitously available on almost everyone's mobile device. The application scenarios are manifold from mapping and navigation to social networking and personal assistance. Besides the advantages for personal use, LBS are increasingly used to generate benefit for the general public, such as traffic congestion estimation, crime mapping or the tracking of spreading diseases. In this respect, there has been an increasing effort of municipalities and governments to take advantage of the societal digitalization and improve the participation of citizens in the form of electronic participation (e-Participation) solutions. These tools are expected to advance the serviceability and administration of public entities as well as boost decision and policy making processes in increasingly complex urban environments. The use of mobile devices advances the e-Participation efforts towards mobile participation (m-Participation) which take advantage of the capabilities of modern smart mobile devices and allows for a more customized experience and better targeted results.

One of the means to improve the participation process is by directly asking for the citizens' opinion (polling) and, thus, obtaining a qualitative analysis of the item in question. The outcome of this interrogation can then be used to proceed with the further decision making process, whether the citizens' opinion is legally binding or not. While polls can be relatively generic in terms of content, objective and context, there is a set of scenarios where the poll's content is highly dependent on the contextual environment it refers to. Polls targeted to a specific set of people (*target group*) are able to provide higher qualitative feedback. These polls provide questions that are specific to the situation and knowledge background of a potential responder. For example, polls concerning a specific social event, such as a sports match or concert, should only be answered by visitors of that event. On the other side, urban developers would like to understand the citizens' needs in specific urban environments such as buildings, neighborhoods or even whole cities. In addition to urban development and social event polling, there are manifold opportunities for a context-aware polling system. Schools and universities are trying to improve the teaching and learning experience by proactively asking students for feedback. The use of context-specific polls enables valuable results that are specifically answered by a respective target group of students, such as math class students or library visitors. In addition, context-specific polls can be used by companies to inquire feedback for their various branch stores which could increase the service quality across all branches.

The challenges of context-specific polls are manifold and the general question is raised on how to provide these polls, according to their context parameters, to the best potential target group of participants. It is assumed that contextually selected potential participants are most interested in the process of participation in a customized context-specific poll. Today's smart mobile devices are ideally equipped for the task of context-specific polling as they are able to provide the required context information (e.g. location or current activity) and present a direct link to the citizens through the device interface and wireless connection. This paper describes the main concept and implementation of *FlashPoll*[1], a tool for m-Participation in urban development. The presented work

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

provides a thorough technical view of the general polling ecosystem and its key roles and components. The FlashPoll ecosystem contains a complete and distributed set of tools and systems that are required for the full mobile-assisted polling process, ranging from the management to the delivery of polls and the visualization of poll results. Furthermore, the paper focuses on the challenges of context-monitoring for context-specific polls. Several real-world field trials have been conducted to evaluate the applicability of FlashPoll for m-Participation.

II. Related Work

The decrease of citizen participation during the last decades initiated a large set of e-Participation activities by various governments [2], [3]. As part of the e-Government and e-Democracy process, e-Participation promotes the citizen participation in the policy and decision making processes with the use of new information and communication technologies as described by the UK government [4]. As Macintosh [5] considers, the identification of the target audience's (*group*) type and size, e.g. a geographical community, and the technologies used are key components for the e-Participation process. However, the use of mobile technologies for participation activities has only been introduced recently and has not been considered in traditional e-Participation models. In addition, m-Participation models take advantage of capabilities provided by mobile services by taking into consideration location and time-specific participation models [6]. Furthermore, the concept of opinion polls through mobile networks has been defined to allow for direct feedback through the usage of SMS [7]. A similar concept has been executed in the field by deploying SMS surveys for election monitoring [8]. Additionally, a personalized survey tool that aims to inform and consult citizens has been piloted with the Swindon Borough Council to improve consultation's efficiency and effectiveness and the personal citizen's engagement [9]. In addition, there are many commercial polling and survey tools and applications available that support research in the field with web and mobile polls, such as *SurveyMonkey*, *polltogo* or *AudienceOpinion*. As argued by Kamal [2], the importance of location-based services in policy making processes is limited as practical systems enabling citizens involvement are lacking. Salim and Haque [10] reviewed a set of tools and technologies which provide an overview on the role of participation and engagement in urban computing. They identified *mobile crowdsourcing and crowdsensing* that use location services as one of the most common interaction methods for realizing urban computing and participation where the user knowledge and opinion of the crowd is being shared. Common mobile applications following the mobile crowdsourcing and crowdsensing approach that also take context information into account are among others *Foursquare*, *Waze* and *Yelp*. The work by de Reuver et al. [11] is among the first to combine context-sensitivity with means of mobile crowdsourcing and crowdsensing. This approach uses position, time and the requested service to extend public services by routing citizens' requests to authorities according to the context parameters. Taylor et al. [12] presented situated voting by deploying a interactive public display for situation-specific polling. However, neither of the above mentioned approaches provide any means to deploy large-scale context-specific polls in a mobile environment. This field has been targeted by the FlashPoll tool. It uses polls to improve the mobile participation in urban development [13], [14] and schools [15]. The tool supports mobile and context-specific polling by providing polls proactively to citizens that are in the vicinity of respective urban development projects.

III. Concept

The main purpose of FlashPoll is to carry out context-specific short and fast opinion polls that increase the mobile participation in urban development projects. FlashPoll supports the notion of micro democracy where effective decisions are made in an increasingly local manner with a distinct set of selected participants, which is crucial in nowadays urban development projects. Accordingly, the selection process that defines which poll is received by whom is based on the matching between the context specified in the poll description and the citizens' contextual environment. FlashPoll takes advantage of the capabilities of current mobile devices by using available context information on smartphones to retrieve polls. In order to provide polls to citizens, an infrastructure is required that manages the context-specific polls from the initiation and creation, to the execution and delivery to participants. In addition, the collection of poll results is required to provide access to the visualization and analysis of the participants' feedback in an appropriate and extensive manner.

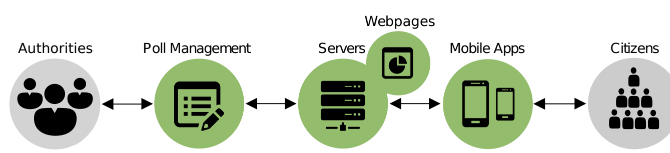

Fig. 1. The end-to-end FlashPoll concept for polls in urban development.

As shown in Figure 1, the FlashPoll concept reflects the real-world environment of poll participation in urban development. The polls are created in the different stages of policy making, e.g. *agenda setting*, *policy creation* or *policy monitoring*, as defined by Macintosh [5]. The role responsible for the initiation, creation and management of polls is the *poll manager*. The poll manager typically is a representative of local authorities, such as governments, municipalities or other entities, responsible for the mobile participation process. Once the polls are created they are ready for publishing. The published polls are distributed through a server infrastructure to the mobile devices of potential poll participants, e.g. regular citizens or other specific groups of poll responders. The participants are ready to consume the poll's content and provide their subjective opinion on the topic in question. The opinions of all participants are gathered immediately together and the aggregated results are shared with citizens, authorities or the general public including non-participants. The results visualization can be either handled by the poll management, the mobile applications or separate ready-to-publish implantable web pages. The citizen's context is crucial for conducting polls in urban development. FlashPoll uses the location and time context to distribute polls to citizens which are, assumingly, more interested in polls in their vicinity than others. For example, the participation in a restructuring project of a public space is more interesting to neighboring local residents than to residents on the other side of the city. This classifies FlashPoll as an LBS application that uses additional contextual parameters to present and adapt information based on the citizens' location.

To understand the full mobile participation process with FlashPoll, it is important to understand the basic concept behind reactive and proactive LBS [16]. Most of currently available LBS are explicitly invoked with an initial request by the user that the service handles in a reactive manner, e.g. the most commonly known point-of-interest search. On the other hand, proactive LBS, for example *Geofencing*, are event-based and provide information proactively when a user enters or leaves a dedicated geographical zone, called *geofence* [17], even when the device itself is in stand-by mode. Proactive LBS have been handled typically in a device-based fashion, where solely the mobile device is responsible for tracking the whereabouts of a user. However, current mobile operating systems only support a limited set of simplified circular geofences per device, which is unsuitable for a polling system with potentially thousands of opinion polls at the same time around the globe. Major challenges for proactive LBS are scalability and the battery consumption on mobile devices due to continuous background tracking. For these reasons, FlashPoll follows a server infrastructure-assisted approach [18] that uses mobile devices as thin clients which only provide the devices' position to a server infrastructure that handles the task of complex spatial context processing. FlashPoll can be scaled linearly by adding more distributed server instances that deal with the synchronization of the mobile devices' state with the server infrastructure. However, this is realized through *stateless* connections to all mobile clients which increase the exchanged messages between mobile clients and server infrastructure, and also the battery consumption on mobile devices. These two metrics are set to be optimized and should be kept at a minimum level.

FlashPoll requires the installation of an application containing the LBS service on citizens' mobile devices in order to be able to provide the current location of the device to the server infrastructure. The selection of poll participants is primarily based on citizens' whereabouts and movement behavior. Polls have a time duration and a pre-defined region in which they are active. The notifications are delivered proactively to the mobile device when a participant enters the region during the time of the poll's activity. This should not require any prior action or knowledge of surrounding polls by the mobile participant. Therefore, potential participants do not necessarily know when and where polls are taking place. FlashPoll uses unintrusive notifications to not bother citizens extensively with surrounding polls. The polls' regions, or geofences, can be of arbitrary shape to represent real-world venues, such as apartment blocks, schools, airports, streets, neighborhoods or city blocks. In addition, polls are received only once at the first time a region is entered. Polls can either be answered immediately or later in time but only once to avoid severe manipulation of the results. In general, by using context parameters to match polls and citizens faking the result data is less likely as deceivers are required to match the user's context with the one required to retrieve the poll. Additionally, there is a set of options that define in which stage the participants can view the accumulated poll results which strongly influences the execution and outcome of a poll. For example, in a competitive challenge the results could be shown to increase the number of answers by potential participants. On the other side, to avoid the impact of current results on the poll, the options can be set to show the overall results only after the answers are submitted by a participant. This restriction can be further strengthened by allowing results to be shown only after the poll has ended or, if required, leave the participants without any results at all.

IV. Ecosystem

The concept described in Section III served as the foundation for the prototypical implementations of all the technical components required to create and to conduct context-specific polls and to automatically process and to share the results. The development yields to the FlashPoll ecosystem which comprises the infrastructure, the administrative user interfaces as well as multiple mobile applications. A comprehensive architectural overview of the ecosystem is given in Figure 2. In the following subsections all components and their functional interactions are introduced.

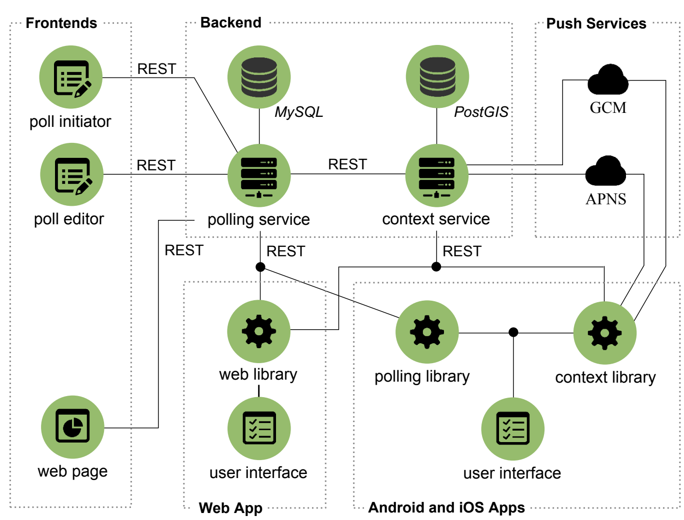

Fig. 2. The FlashPoll ecosystem. GCM: Google Cloud Messaging, APNS: Apple Push Notification Service, REST: Representational State Transfer

A. Components

First of all, the administrative frontends used by the poll managers to create, modify and delete polls and to view the results are presented. This is followed by a brief introduction of the core component of the ecosystem: the FlashPoll backend. The mobile applications used by the participants as well as the poll-specific web pages for the visualization of results are described thereafter.

1) Administrative Frontends: The central administration of the ecosystem is carried out by the poll managers through the use of the *poll initiator*. It enables a poll manager to create, edit and delete context-specific polls as well as to get insights into the corresponding results. Its access can be restricted to particular poll managers or groups of poll managers in order to cope with the particular structural fragmentation of public institutions or companies. The poll initiator is organized into different sections: poll/location management and poll results. Context-specific polls are created and managed in the former sections while the results can be queried in the latter section. The creation of a context-specific poll is mainly divided into two steps. First, the poll's context needs to be specified. This includes the definition of the start and end dates/times of a poll - forming the *activation period* - and the geofence itself. The activation period is the time in which the poll is being distributed to the mobile clients and in which the citizens are able to participate in it. The geofence encompasses the dedicated region in which potential participants will be proactively notified about the poll. A date/time picker is provided as an easy means to select the activation period while the geofence is created by drawing a polygonal shaped region onto a two-dimensional map of the world. A geofence is defined independently of a particular poll within the location management section and can be reused for other polls as well. The second step deals with the definition of arbitrary questions and selectable answers within the poll management section. Thereby, a poll manager can choose from the following question types:

- **Single and multiple choice:** Participants can either select one or multiple answers from a set of predefined answers. A predefined answer might also give participants the option to formulate the answer in text form.
- **Open text:** The participant can freely phrase the answer in text form.
- **Ranking question:** The participant must rank the predefined answers among each other.

Additionally, the poll manager is also able to give a poll a title, a subtitle, a description and a conclude message. The description should briefly introduce the purpose of the poll to a potential participant while the conclude message will be shown to the participant after the completion of all questions. Finally, it can be selected whether a participant is able to view the latest results of the poll within the mobile application before answering the poll, after answering the poll, after the end of the activation period of a poll or not at all. The results are aggregated among all participants of the poll, e.g. the amount of times a selectable answer was ranked with the highest priority. To give a poll manager all-time access to the aggregated results, they are shown in form of diagrams within the poll initiator, regardless of the visibility option selected for the participants. A complete and anonymized list of all detailed answers of a poll can be exported in order to enable a deeper statistical analysis by using a different tool set. Another administrative frontend, called *poll editor*, is provided as an alternative to the easy-to-use and intuitive poll initiator. It is targeted towards expert users which make use of build-in experimental features like poll tags. Therein, each poll can be assigned to an arbitrary amount of tags like "sports" or "public transport". Given a mobile application that is aware of the tags a citizen is interested in, these tags can be used by the ecosystem to narrow down the set of location-wise relevant polls to an even more relevant subset.

2) Backend: As the focal point of the ecosystem, the backend is responsible for persisting polls, monitoring the context of all mobile clients, delivering relevant polls to mobile clients, collecting and processing the answers, providing the aggregated and detailed results. It also provides means to access the aggregated results within the mobile applications or by the interested public via publicly available web pages. From a general point of view, it can be distinguished between backend tasks which are exclusively related to the domain of polling and backend tasks which are dealing with the challenge of proactively notifying mobile clients about relevant content depending on their context. Owing to this differentiation, the backend is divided into two subcomponents: a domain-specific *polling service* (PS) and a *context service* (CS). With the support of mobile applications, the CS monitors the context of all mobile clients and proactively delivers context-specific notifications to mobile clients in case their context matches a given context of interest. The CS is used in conjunction with a domain-specific 3^{rd} party service which uses the CS as a broker for the delivery of its domain-specific notifications to the mobile clients. For example, the CS could be used to deliver location-based coupons to a promotion application of a grocery store on behalf of a couponing service. In our setup, the PS acts as the 3^{rd} party service of the CS. As a domain-specific subcomponent, the PS is responsible to store polls, provide polls, store poll answers, process poll answers and provide poll results. In case a new poll is created within the PS, it instructs the CS to monitor the context of all mobile clients with respect to the given poll context and to deliver the unique poll identifier to those mobile clients which are moving into the given poll context. With a unique poll identifier, a mobile client is then able to retrieve the poll of interest from the PS. The CS is built in a generic way so that context-specific notifications are delivered transparently to the mobile clients regardless of its content. Moreover, the CS is even generic in the way context is handled. In the given use case, the context comprises a geofence and the activation period. But one can easily imagine a geofence-based poll without an activation period that will be active forever. These properties increase the reusability of the CS for different kinds of ecosystems in which proactive context-specific notifications need to be delivered to mobile clients.

3) Mobile Applications: The ecosystem comprises in total three mobile applications. Two applications were developed for the Android and iOS mobile operating systems. The third application was implemented as a mobile web application in order to support all mobile operating systems that provide state-of-the-art mobile browsers. Although supporting a similar set of features, the user interfaces of all mobile applications differ slightly from each other. The noticeable differences are caused by the use of platform-specific user interface elements and by taking the platform-specific interaction concepts into consideration. Both the Android and the iOS application are able to receive relevant polls even if the applications are not shown in the foreground or if the mobile devices are screen-locked or in stand-by. This ability fulfills the requirement defined in Section III to notify the citizen about a poll proactively, regardless of whether the citizen is currently using the mobile device actively or not. This requirement can not be fulfilled by the web application because the required location tracking is stopped as soon the citizen closes the execution environment, namely the web browser. A first public working draft of the W3C to support proactive location-based notifications in web applications even when the browser is closed is in discussion at the time of writing, but it is neither specified yet nor supported by state-of-the-art mobile web browsers[2]. One might argue that from the citizen's perspective this specific feature is not strongly missed because citizens are not considered to know of any relevant context-specific polls. In other words, if a citizen is moving through a region which is associated with a poll but the citizen is not expecting it, the citizen will not miss the proactive notification. But this contradicts the main goal of the proactive context-aware polling system to inform the potential participants about polls that are not known in advance but which are of high relevance to them. Sample screenshots of all mobile application are shown in Figure 3.

Within the Android and iOS applications, it is distinguished between the domain-specific parts of the applications and the parts that are responsible for the context-awareness. The user interfaces are highly domain-specific because they are dealing with the presentation of polls and results. Other strongly required domain-specific components within the applications are the *polling libraries*. These libraries encapsulate the access to the PS. Their main tasks are to provide the polls from the PS, to submit the answers to the PS and to retrieve the aggregated results from the PS. Encapsulating this functionality into single libraries, one for each mobile operating system, makes it possible to extend a 3^{rd} party mobile application with polling functionality by simply integrating this library. The context-awareness of the mobile applications is enabled by the *context libraries*. They encapsulate all the required logic for a mobile application to receive proactive context-specific notifications from the CS. The synchronization mechanism between the CS and the context libraries is fully explained in Subsection IV-B2. Since the context libraries are not domain-specific, they

[2]http://www.w3.org/TR/geofencing/

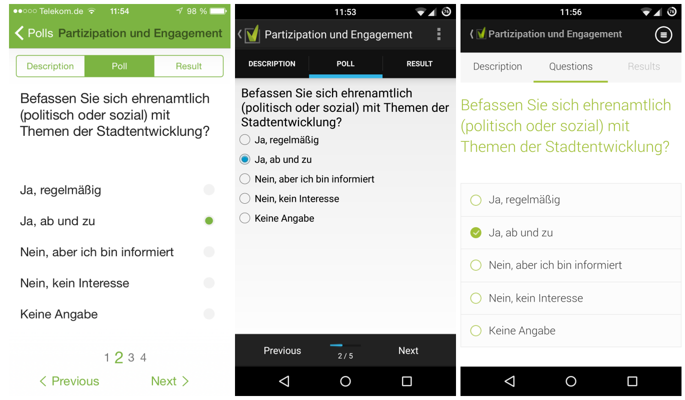

Fig. 3. Screenshots of the FlashPoll mobile applications starting with the iOS App from the left and followed by the Android App and Web App from a real world field trial poll executed in Berlin, Germany.

can also be used by any kind of 3^{rd} party mobile application to adapt its content proactively to the environment, e.g. extending a weather application with location-based storm warnings. In the given use case, they are providing the mobile applications with location-based notifications which are containing the aforementioned unique poll identifiers.

4) Web pages: Poll managers can retrieve the aggregated and detailed results via the poll results section of the poll initiator. On the other side, the participants of a poll can access the aggregated results in form of diagrams via the mobile applications. The ecosystem provides poll-specific web pages to give citizens that could not participate in the poll still the possibility to access the aggregated results of a poll. Each poll-specific web page contains the list of questions of a particular poll together with diagrams visualizing the aggregated results. These web pages are based upon a web page template which is automatically filled by poll-specific content as soon as a poll manager requests a poll-specific web page within the poll initiator. A poll manager can then either distribute these static web pages, e.g. via email, or deploy them on a publicly available web server.

B. Interaction of Components

The core functionalities of the backend are offered via carefully defined RESTful APIs. The *polling service API* (PS API) is used by the poll initiator and poll editor to create polls, to edit polls, to delete polls and to access the aggregated as well as detailed results. The PS API is also used by the polling library within the mobile applications to submit the poll answers and to obtain the aggregated results. The poll-specific web pages are accessing the aggregated results via the PS API as well. As stated above, the CS is used by the PS as a means to notify mobile clients proactively about relevant polls in form of unique poll identifiers. Via the generic *context service API* (CS API), a 3^{rd} party service is able to register or deregister context-specific notifications with the CS. In case of the FlashPoll ecosystem, a context-specific notification comprises the unique poll identifier, the activation period and the associated geofence. In addition, the CS API is used by the context libraries within the mobile applications to register a mobile application to the CS and to subscribe for the reception of context-specific poll notifications.

1) Operational Workflow: A typical workflow of a poll manager starts with creating a poll and the geofence with the poll initiator or poll editor. The new poll can be either created as a genuine poll or as a copy of an older poll. During the whole lifetime, a poll successively traverses the following states:

1) **draft** A poll starts in the draft state. In this state, a poll can be edited, e.g. questions can be added or deleted, arbitrary times until it perfectly suits the requirements.
2) **inactive** A poll has been explicitly declared for distribution to mobile clients but is not active yet.
3) **preactive** A poll becomes preactive 4 hours prior to the activation period. The reason for introducing a preactive state is presented in Subsection IV-B2.
4) **active** A poll gets active at the start of the activation period. Here, it will be distributed to the mobile clients and it can be answered by the participants.
5) **expired** At the end of the activation period, the poll is expired. In this state, it will not be distributed anymore and answers will not be accepted by the PS anymore. The poll is declared to be finished.

After declaring the poll to be ready for distribution within the poll manager user interfaces, the PS announces a new context- and domain-specific notification to the CS. This triggers the CS to start the observation of the poll-specific context for all mobile clients which have been previously registered successfully via the CS API to receive context-specific polls. In case a poll gets active, it will be pushed to any Android or iOS mobile client that enters the corresponding geofence. The mobile client receiving a context-specific notification will extract the unique poll identifier from it and will query the PS via the PS API in order to retrieve the corresponding poll, including among other things the questions and answer options. Due to the reasons explained in Subsection IV-A3, the location context of a WebApp can not be monitored by the CS in case the web browser is closed. As a consequence, the CS will not monitor the context of any WebApp. This implies that the WebApp can deliver polls only in reactive manner, when citizens actively requests polls related to their current position. In any case, after receiving a poll, a citizen is able to answer the poll until it is expired. Immediately after answering a poll, the citizen-specific answers are sent to the PS via the PS API in an anonymized manner. During the activation period, each submitted answer gets immediately processed by the PS in order to provide aggregated results of the poll in real-time to the poll manager user interfaces, the mobile applications and the poll-specific web pages.

2) Context Monitoring and Reactive Poll Delivery: The technically challenging task of the CS is to monitor the contexts of all Android and iOS mobile clients in an energy and data-traffic efficient manner. Within the ecosystem, the mobile clients and the CS are assumed to be time synchronized with respect to a predefined time zone. Observing the time context of the mobile clients with respect to the activation period of a poll can therefore be simplified to the task of keeping track of the activation period at the CS, using the internal clock. In contrast to the time, the locations of the mobile clients are differing from each other. Since the CS needs to monitor locations of mobile clients with respect to multiple geofences (one for each poll), a location synchronization mechanism between the CS and the mobile clients is required. Otherwise, the CS would not be able to detect whether a mobile client entered a certain geofence or not. But as stated in Section III, synchronizing a location context between a mobile device and the infrastructure increases the wireless data traffic and impacts the battery life of the mobile devices significantly. In the following, a new location synchronization mechanism is introduced (based upon ideas presented by Küpper [19] and Bareth [20]), which reduces the amount of location measurements and the amount of messages exchanged wirelessly to a minimum, while still guaranteeing the CS to detect mobile devices entering a geofence at a reliable level.

The location synchronization mechanism and the location update strategy respectively between the mobile clients and the CS are based upon the basic idea of an omniscient CS which informs each mobile client individually about the next contextual condition under which the mobile client has to synchronize its position with the CS. Based on this strategy, a mobile client will only measure and synchronize its position in case it is strongly needed. If there are no polls active or preactive, the CS can recommend the mobile clients to stop measuring and synchronizing positions for a longer period of time. On the other hand, if the last known position of a mobile client is close to a geofence of an active or preactive poll, the CS can recommend the mobile client to measure and synchronize its position after a shorter time period because it might cross the geofence limits in the near future. To be more precise, each time a mobile client has sent its position to the CS in form of a location update, the CS calculates the distance to the closest active or preactive geofence and provides it as a response to the mobile client. Preactive geofences are introduced in order to provide a more reliable poll delivery with newly added polls. The CS considers preactive polls for the synchronization with mobile clients once they are created and set for delivery by the authorities and become active within the next four hours. This way, mobile clients are already aware of upcoming nearby polls and adapt their client-specific location update strategy efficiently to the current situation.

Figure 4 visualizes the location synchronization mechanism by means of a sample trace. Depending on the distance, the Android and iOS mobile clients switch to one of the following modes:

1) **Idle mode.** Condition: The are no (pre)active geofences known by the CS (negative distance to closest (pre)active geofence). A mobile client measures and synchronizes its position every four hours, as it is done by regular mobile phones within today's mobile operator networks. During this time, it will not measure its position nor communicate with the CS.

2) **Far mode.** Condition: A mobile client is located far away from the edge of the closest (pre)activated geofence, thus the distance between client and geofence is above a certain threshold. The given distance spans a circular region around the mobile client, called *safety zone*, in which the client can move freely without the need to send its location to the CS. Within this situation, the Android and iOS Apps make use of energy-efficient mechanisms given by the mobile operating systems to monitor the position with respect to the safety zone. The mobile device initiates a location update as soon as the citizen leaves the safety zone.
3) **Near mode.** Condition: The distance to the edge of the closest (pre)active geofence falls below a certain threshold, thus the citizen is located near the edge of the closest (pre)active geofence. Depending on the operating system, the mobile client switches to a time periodic update strategy (Android App) or to a distance update strategy (iOS App). Within the periodic update strategy, the mobile client measures and synchronizes its position periodically with an activity-specific time interval, which is determined through evaluation in [18]. The activity recognition engine is provided by the operating system. Within the distance update strategy, the client's distance traveled is recorded and initiates a location update as soon as a certain distance threshold is exceeded.

In case the CS detects a mobile client to be located within an active geofence the response to the location update will be enriched with the corresponding poll identifier. The mobile client is able to grab the poll from the PS immediately after receiving the response. Although the citizens experience these notifications as being proactive, they are technically delivered in a reactive way in form of a response to a location update. As mentioned in Section III, every poll should only be delivered once. Hence, for the mobile client, the geofence is not considered to be active anymore as soon as the poll is delivered successfully. The response enriched with the poll identifier will not contain the distance to the corresponding geofence but the distance to the next closest (pre)active geofence.

3) Proactive Poll Delivery by Push Notifications: So far, a poll identifier could be delivered reactively as part of the response to a location update. But there are still two situations in which this is not sufficient. Consider a citizen entering a preactive geofence, like shown in Figure 4. In case the citizen stays close to the edge of the geofence and stops moving (detected by the activity recognition engine), the mobile client will pause any scheduled location updates because measuring and synchronizing a not-changing location will obviously waste energy and wireless resources. If the citizen will not move for a longer period, the poll will not be delivered at the time the geofence gets active because of the missing location updates. A similar situation happens if the citizen continues to move deeper into the preactive geofence, which causes the mobile client to switch to the far mode. If the citizen will not leave the safety zone for a longer period, there will be no location update initiated by the mobile client. The only

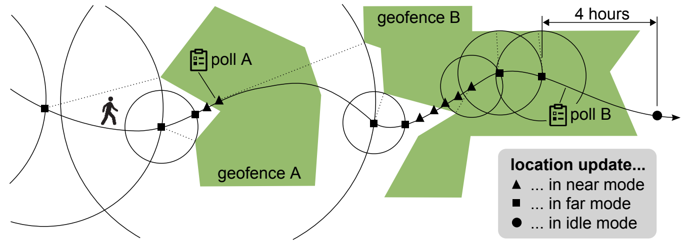

Fig. 4. A sample trace of a citizen passing an active geofence A and thereafter entering a preactive geofence B. The latter is getting active while the citizen is passing through it. Safety zones are spanned as circular regions around the last location update using the distance to the nearest active or preactive geofence as the radius.

way to notify mobile clients in both situations is to send push notifications, initiated by the CS, as shown by poll B in the example of Figure 4. Since the CS knows the last location of all citizens, it can compile a list of mobile clients which should be notified proactively via push notifications. Unfortunately, beside mobile-terminated calls and SMS/MMS messages there is no build-in mechanism within a cellular network to deliver packet-switched push notications. Therefore, all major mobile operating systems offer mechanisms at the application layer to deliver data packets proactively to mobile clients via separate and energy efficient connections between mobile clients and the server infrastructure of the mobile operating system provider. The ecosystem makes use of the Google Cloud Messaging (GCM) service and the Apple Push Notification Service (APNS) to deliver packet-switched push notifications to the Android and iOS mobile clients.

V. Field Trials

The FlashPoll ecosystem with all its components has been tested in several field trials. Varying polls were published at different occasions and events targeting diverse user groups. The largest trial has been established at the university library of the Technische Universität Berlin. Students and faculty stuff were asked to give their opinion on the organizational restructuring of sparsely available workspaces in the library, resulting in 660 participants. After the first install, the mobile application will receive an initial country-specific poll. This poll asks citizens on general questions on mobile participation and engagement and has received roughly about 150 responses. FlashPoll also enabled new democratic processes at schools in Sweden [15]. The ABC public elementary school in Stockholm participated in pilot polls that targeted the involvement and engagement of pupils and the use of technology in schools. The authorities used FlashPoll to target different groups of students and teachers with over 100 participants. A key aspect for the trial was to analyze the pupils' understanding of e-Democracy. In addition, other trials were conducted to improve the management of neighborhoods and sport grounds around the city of Berlin. FlashPoll has also been tested on several public events[3], such as *Lange Nacht der Wissenschaften* (Long Night of Science) in Berlin or the *Future en Seine* (Future at Seine-River) in Paris. FlashPoll was also part of

[3]http://www.flashpoll.eu/en/event

the *Solutions COP21 Paris 2015*. During all trials FlashPoll has been received as intuitive to use and easy to understand, which is an important aspect for the acceptance of mobile participation. The trials and pilots were overall successful and gave beneficial insights for improving policy and decision making processes. However, authorities as well as participants raised the issue of location privacy within FlashPoll as the shared location information can be used to track and identify the user's behavior beyond the sole purpose of delivering polls. In a sensitive environment such as m-Participation with decreasing trust in public authorities, privacy concerns like these are crucial for the usage of the overall ecosystem. Therefore, a context-sensitive location privacy extension has been introduced, which targets context-sensitive proactive LBS [21]. This extension allows users to influence the context information shared by the mobile clients.

VI. Conclusions

In this paper we presented a complete view on the concept and architecture of the FlashPoll ecosystem. FlashPoll uses context-specific mobile polls to increase the mobile participation of citizens. Its main case of application is the improvement of urban development processes from the initiation of projects to the satisfaction monitoring afterwards. FlashPoll provides a comprehensive end-to-end tool set that comprises the intuitive management of polls through administrative frontends as well as the backend that uses a sophisticated service for contextual matching of polling notifications. Furthermore, the ecosystem includes mobile applications for both leading mobile operating systems, Android and iOS, and an additional context-sensitive web application for all other devices. The submitted responses are immediately aggregated to viewable results and are accessible through the management tools, the mobile applications as well as through publicly available web pages. FlashPoll uses generic components which allow the immediate reuse in parts or as a whole. It can be easily integrated into other ecosystems and applications and can serve as a white-labeling solution for different purposes beyond the use case for mobile participation in urban development. Finally, further additions to proactive LBS have been made that allow the configuration of so-called *geofence scenarios* which enable temporal logic between a set of geofences [22]. With this approach it is possible to conduct multiple polls which are spatially and temporally dependent, e.g. a set of polls that evaluate the consecutive experience during trips with public transportation services.

Acknowledgments

This work has been carried out within the EIT Digital FlashPoll activity. We thank all partners, namely Alfstore (Salem Tirane, Franck Villedieu, Richard Ewelle), CTS - TU Berlin (Carolin Schröder), KTH Stockholm (Konrad Tollmar), Missions Publiques (Antoine Vergne), Nexus Institute (Christoph Henseler, Angela Jain), Q&U Lab - TU Berlin (Maija Poikela), Siemens (Michael Thalhofer, Reinhold Ufrecht) and Stockholm University (Patrik Hernwall).

References

[1] A. K. Dey, G. D. Abowd, and D. Salber, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-aware Applications," *Human Computer Interaction*, vol. 16, pp. 97–166, 2001.

[2] M. M. Kamal, "An analysis of e-participation research: moving from theoretical to pragmatic viewpoint," *Transforming Government: People, Process and Policy*, vol. 3, pp. 340–354, 2009.

[3] L. Komito, "e-Participation and Governance: Widening the net," *The Electronic Journal of e-government*, vol. 3, pp. 39–48, 2005.

[4] R. Cook, "In the Service of Democracy - a consultation paper on a policy for electronic democracy," *HM Government and UKOnline*, 2002.

[5] A. Macintosh, "Characterizing E-Participation in Policy-Making," in *Proc. of the 37th Annual Hawaii Int. Conf. on System Sciences*. IEEE Computer Society, 2004.

[6] N. Shankaranarayanan and B. Kim, "Location and Time Specific Mobile Participation Platform," 2012, US Patent 8,335,526 B2.

[7] P. Heinonen, H. Okkonen, and I. Tuomi, "Opinion Poll Utilizing a Wireless Data Transmission Connection," 2002, US Patent 6,418,308.

[8] J. Hellström and A. Karefelt, "Participation Through Mobile Phones: A Study of SMS Use During the Ugandan General Elections 2011," in *Proc. of the Fifth Int. Conf. on Information and Communication Technologies and Development*. ACM, 2012, pp. 249–258.

[9] A. Macintosh and A. Whyte, "Towards an evaluation framework for eParticipation," *Transforming Government: People, Process and Policy*, vol. 2, pp. 16–30, 2008.

[10] F. Salim and U. Haque, "Urban computing in the wild: A survey on large scale participation and citizen engagement with ubiquitous computing, cyber physical systems, and Internet of Things," *Int. Journal of Human-Computer Studies*, vol. 81, pp. 31 – 48, 2015.

[11] M. de Reuver, S. Stein, and J. F. Hampe, "From eParticipation to mobile participation: Designing a service platform and business model for mobile participation," *Information Polity*, vol. 18, pp. 57–73, 2013.

[12] N. Taylor, J. Marshall, A. Blum-Ross, J. Mills, J. Rogers, P. Egglestone, D. M. Frohlich, P. Wright, and P. Olivier, "Viewpoint: empowering communities with situated voting devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 1361–1370.

[13] M. Salem and B. Deva, "3rd Party Geolocation Messaging: A Positioning Enabler Middleware for Realizing Context-aware Polling," in *Proc. of the 6th Int. Conf. on Mobile Wireless Middleware, Operating Systems, and Applications*. IEEE, 2013, pp. 100–109.

[14] C. Schröder, "A Mobile App for Citizen Participation," in *Proc. of the 2014 Conf. on Electronic Governance and Open Society: Challenges in Eurasia*. ACM, 2014, pp. 75–78.

[15] L. Elkenberg, P. Hernwall, and K. Tollmar, "Enabling new democratic processes in Schools," in *Int. Reports on Socio-Informatics (IRSI)*, vol. 11, 2014, pp. 21–33.

[16] P. Bellavista, A. Küpper, and S. Helal, "Location-Based Services: Back to the Future," *Pervasive Computing, IEEE*, vol. 7, pp. 85–89, 2008.

[17] A. Küpper, U. Bareth, and B. Freese, "Geofencing and Background Tracking - The Next Features in LBS," in *Proc. of the 41th Annual Conf. of the Gesellschaft für Informatik e.V.*, vol. 192. Köllen Druck + Verlag GmbH, 2011.

[18] S. Rodriguez Garzon, B. Deva, G. Pilz, and S. Medack, "Infrastructure-assisted Geofencing: Proactive Location-Based Services with Thin Mobile Clients and Smart Servers," in *2015 3rd IEEE Int. Conf. on Mobile Cloud Computing, Services, and Engineering*, 2015, pp. 61–70.

[19] A. Küpper and G. Treu, "Efficient Proximity and Separation Detection Among Mobile Targets for Supporting Location-based Community Services," *SIGMOBILE Mobile Computing and Communications Review*, vol. 10, pp. 1–12, 2006.

[20] U. Bareth and A. Küpper, "Energy-Efficient Position Tracking in Proactive Location-Based Services for Smartphone Environments," in *IEEE 35th Ann. Comput. Software and Applications Conf.*, 2011, pp. 516–521.

[21] B. Deva, S. Rodriguez Garzon, and S. Schünemann, "A Context-sensitive Privacy-aware Framework for Proactive Location-based Services," in *9th Int. Conf. on Next Generation Mobile Applications, Services and Technologies*, 2015.

[22] S. Rodriguez Garzon and B. Deva, "Geofencing 2.0: Taking Location-based Notifications to the Next Level," in *Proc. of the Int. Joint Conf. on Pervasive and Ubiquitous Computing*. ACM, 2014, pp. 919–930.