CachedSensing: Exploring and Documenting the Environment as a Treasure Hunt

Delphine Christin*, Carsten Büttner*, Nicolas Repp†
*Secure Mobile Networking Lab, Technische Universität Darmstadt,
Mornewegstr. 32, 64293 Darmstadt, Germany
†Research Services Division, Technische Universität Darmstadt,
Karolinenplatz 5, 64289 Darmstadt, Germany
Emails: {delphine.christin, carsten.buettner}@cased.de, repp.ni@pvw.tu-darmstadt.de

***Abstract*—With a user base of billions of mobile phone subscribers worldwide, participatory sensing applications could benefit from environmental data in unprecedented quantity and quality. However, most of existing participatory sensing applications have not yet crossed the chasm. They are still in the early adoption phase and large scale deployments are missing. In order to democratize participatory sensing applications, we propose to introduce sensing tasks in a real-world treasure hunt. Using our approach called CachedSensing, users easily create sensing tasks and write them on Near Field Communication (NFC) tags. The tags are then hidden in the nature by their creators. Other users willing to explore the environment search for the tags and perform the sensing task associated to the tag found. We introduce the CachedSensing architecture and detail our prototype implementation. We finally present our deployment and share the experience gained while hiding NFC tags in the wild.**

***Index Terms*—Mobile sensing, geocaching, NFC tags.**

I. Introduction

Participatory sensing applications leverage mobile phones as a new generation of sensing platforms. Today's mobile phones integrate various sensors, such as accelerometers, cameras, and microphones, and they are resourceful in terms of processing and storage [1]. Carried by users, mobile phones offer an unprecedented coverage. This coverage is particularly of interest to document and monitor environmental phenomena, such as noise pollution. While the outcomes of such applications may benefit the community at large, large scale deployments are still missing.

In order to foster the participation of users, we propose a new approach called CachedSensing, where collecting sensor readings becomes part of a real-world treasure hunt. Inspired from the concept of geocaching, CachedSensing aims to trigger the users' spirit of adventure and exploit their creativity to find and hide caches. In CachedSensing, a cache is composed of a NFC tag. When users find and read a tag with their mobile phone, they obtain the details of the sensing task to complete. In addition to explore the environment, users are rewarded by a logbook entry for each completed task.

The gaming aspect introduced in CachedSensing not only intends to serve as incentives for the participants, but also opens the doors of participatory sensing applications to the geocaching community and its estimated 5 million active members worldwide [2]. By using NFC tags and introducing sensing tasks, we believe that CachedSensing can attract geocachers interested in a novel genre of technology-based caches and hence, foster user contributions to participatory sensing applications.

Our contributions are as follows:

1) We propose the concept of CachedSensing as (a) a novel game-based incentive scheme applicable in various participatory sensing scenarios, (b) a new task distribution scheme, and (c) a new type of geocaches involving sensing tasks and based on NFC tags.
2) We present the CachedSensing architecture and provide details about the implementation of our prototype. Using our proof-of-concept, users can easily create and distribute different types of tasks in order to measure the surrounding sound pressure level and hence, document potential noise pollution. Our proof-of-concept also supports the search for sensing tasks and their execution by other users.
3) We finally detail our deployment in Schöllkrippen, Germany, and share the issues encountered during the deployment as well as the lessons learnt.

The paper is organized as follows. In Section II, we present background information about geocaching and NFC. We discuss related work in Section III and detail the features of CachedSensing in Section IV. We present the CachedSensing architecture in Section V, its implementation in Section VI, and its deployment in Section VII. In Section VIII, we discuss future work, before making concluding remarks in Section IX.

II. Foundations

In this section, we provide background information about geocaching and NFC, both leveraged in this work.

A. Geocaching

Geocaching is a outdoor treasure hunting game, in which users hide objects called *caches* and publish their coordinates along with a brief description online. Based on the published coordinates, other users search for these caches. Once a cache is found, the name of the user(s) who found it is listed in a logbook. There exist multiple types of caches. For example, a *traditional* cache consists of a container in which a pen and a logbook are hidden. Users use the provided pen to write their name in the logbook when they found it. In a *multi-cache*, each cache reveals hints about the location of the next cache, whereas users need first to solve a logical puzzle or an enigma in order to gain access to the coordinates of the cache in *mystery* or *puzzle* caches. Additional caches, such as *virtual* caches do not even require a physical cache. In virtual caches, users has to discover a location and complete a task (e.g., answer a question or take a picture) in order to prove that they actually visited this location. Different platforms, such as Geocaching.com [2] or Opencaching.com [3], support the publication of these different caches and gather the geocaching community. For example, Geocaching.com counts more than 1,820,000 caches and 5 million geocachers worldwide [2].

B. Near Field Communication

Near Field Communication (NFC) is a short-range wireless communication standard specified in ISO/IEC 21481 [4] and operates similar to Radio Frequency Identification (RFID). It, however, presents a shorter range: 10 cm in theory and around 4 cm in practice. The NFC standard offer three different operating modes [5]: (1) *card emulation* mode, (2) *peer-to-peer* mode, and (3) *read/write* mode. In the card emulation mode, a NFC-enabled device acts as a smart card and can be used for, e.g., payments. In comparison, the peer-to-peer mode allows two NFC-enabled devices to exchange data in a bidirectional fashion. In the read/write mode, a NFC-enabled device can read and write the content of a NFC tag or a passive device. There exist four different types of tags, which differ in terms of storage capacity, data rate, read/write states, as well as used standards. In this work, we use NFC tags of type 2 based on the ISO/IEC 14443A standard [6]. The tags present a storage capacity of up to 2 kB and a data rate of 106 kbps. The tags are rewritable, but can also be configured in a read-only state. The data exchanged between NFC devices and NFC tags are formatted according to the NFC Data Exchange Format (NDEF) [7] that ensures the interoperability between NFC devices and NFC tags produced by different manufactures and showing different properties as well as storage schemes.

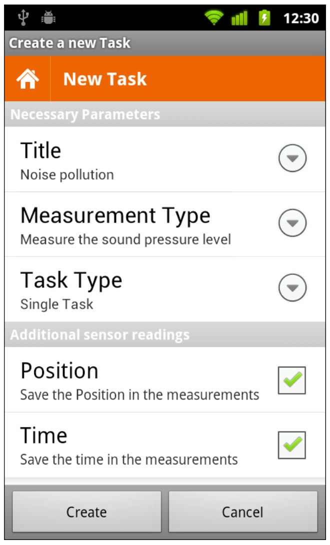
III. Related Work

Different incentive mechanisms specially tailored to participatory sensing applications have been proposed in recent years. For example, the incentive model presented in [8] is based on the assumption that users of such applications are both providers and consumers. Users who contribute sensor readings to the applications obtain access to further services provided by other users. This scheme, however, requires mutual relationships between providers and consumers that may not be applicable in all application scenarios. Additionally, monetary incentives based on reverse auctions were introduced in [9] and [10]. In these schemes, users place a bid for the sensor readings they collected. Service providers can accept this bid and pay for the sensor readings. Micro-payments were also investigated in [11]. The results show that not only monetary incentives motivate users to contribute to participatory sensing applications, but also that altruism and competitiveness are important factors. In contrast to these approaches, CachedSensing does not rely on momentary incentives, but on the spirit of competition between users. For example, users may compete to be the first to find a newly hidden NFC tag, or to complete the greater number of tasks. As CachedSensing, *flora caching* [12] is based on the principles of geocaching. Instead of searching NFC tags, users search for specific plant species. When users find one of the species of interest, they take a picture of the plant(s) and report their observations to the application. The contributions of the users are then displayed on a map and provide insights about the flora to specialists or other users. While CachedSensing shares the most similarities with flora caching, our approach aims to provide means for users to create sensing tasks based on a wider range of sensing modalities and hence, to expand the range of potential environmental subjects monitored.

In addition to foster user contributions, CachedSensing proposes a new kind of caches for the geocaching community. Whereas the majority of caches are still based on a paper logbook in which users write their name, the number of existing challenge-based and technology-oriented geocaches continuously increases. For example, using *Geocaching Challenges* [13], users can create challenges for other users to accomplish. The challenges can require the users to make a particular action, such as climbing, singing, or hiking, or to take a picture of themselves while making such action. In comparison to CachedSensing, the challenges focus on the physical actions accomplished by the users rather than on their environment. Moreover, they do not involve NFC tags and are limited to the capture of pictures. Instead of using NFC tags, existing technology-oriented caches leverage wireless beacons or QR codes. Wireless beacons called *Chirp* [14] broadcast within a radius of 10 m information, such as the coordinates of the next cache, the date of the last visit to the beacon, or the total number of visits, to nearby users equipped with a compatible device. These beacon-based caches, however, do not require users to perform sensing tasks. Moreover, each beacon has a limited lifetime as it is battery-powered, and it costs around 23 USD. As reference, the price of a NFC tag is around 1 USD. Other technology-oriented geocaches make use of QR codes called *munzees* [15]. As in CachedSensing, users search for munzees based on their coordinates published online and scan the QR codes for gaining points. Users, however, do not perform sensing tasks. In summary, CachedSensing differs from existing geocaches since our approach leverages NFC tags as information support and requires users to perform a sensing task.

IV. Key Features and Application Scenario

We highlight the key features of CachedSensing and illustrate its principles by describing a potential application scenario. Note that CachedSensing is not restricted to this particular scenario, but can be applied in a wide range of other settings.



12:30

Create a new Task

New Task

Necessary Parameters

Title
Noise pollution

Measurement Type
Measure the sound pressure level

Task Type
Single Task

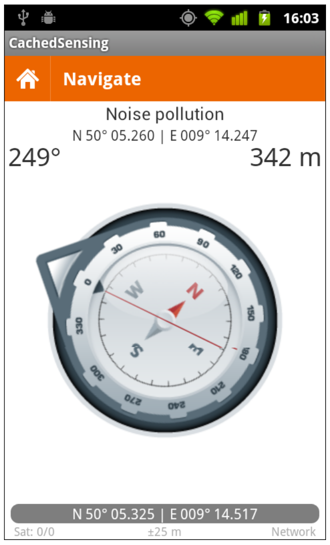
Additional sensor readings

Position ✓
Save the Position in the measurements

Time ✓
Save the time in the measurements

Create | Cancel

Fig. 1. Task settings



16:03

CachedSensing

Navigate

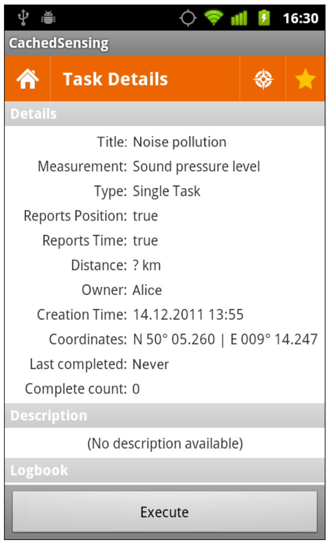
Noise pollution
N 50° 05.260 | E 009° 14.247

249° 342 m

N 50° 05.325 | E 009° 14.517

Sat: 0/0 ±25 m Network

Fig. 2. Navigation compass



16:30

CachedSensing

Task Details

Details

Title:	Noise pollution
Measurement:	Sound pressure level
Type:	Single Task
Reports Position:	true
Reports Time:	true
Distance:	? km
Owner:	Alice
Creation Time:	14.12.2011 13:55
Coordinates:	N 50° 05.260 | E 009° 14.247
Last completed:	Never
Complete count:	0

Description

(No description available)

Logbook

Execute

Fig. 3. Task details

A. *Key Features*

CachedSensing is inspired by the concept of geocaching and serves as an interface between people interested in documenting the environment and those interested in exploring it. People interested in collecting sensor readings about the environment can easily create a sensing task using a NFC-enabled mobile phone, write it on a NFC tag, hide the tag, and publish its coordinates. Compared to existing geocaches, CachedSensing offers thus a new kind of caches since it leverages the NFC technology and is associated with sensing tasks. Instead of uniquely searching for an object, users also contribute to the application and these contributions benefit to the creators of the tasks. The introduction of sensing tasks as part of the geocaches opens the doors to an infinity of novel scenarios, in which the multiple sensors embedded in mobile phones come into play and allow to draw different portraits of the environment. Moreover, CachedSensing allows the task creators to digitally and automatically keep track of users having found and completed the tasks, instead of relying on physical logbooks in which users have manually written their name. CachedSensing hence prevents potential abuses and ensures that users having found and completed the task are listed chronologically. In case of multi-stations caches, CachedSensing also ensures that users have performed the sensing tasks in the order imposed by the task creator.

B. *Exemplary Application Scenario*

We assume that Alice and Bob are registered in the CachedSensing application. Alice is interested in buying an apartment and she would like to know whether the apartment she visited is exposed to noise pollution. Since no noise map of her city is available, Alice decides to create a new task in CachedSensing in order to obtain information about the sound loudness in proximity of this apartment. For the creation of the task, Alice uses the CachedSensing application on her mobile phone and determines the title of the task, the type of measurement she is interested in, the type of the task, and if the measurement should be annotated with time and location information as illustrated in Fig. 1. As she is interested in noise pollution at a precise location, Alice selects “sound pressure level” as measurement type and “single task” as task type. Additionally, she indicates that the measurements should be annotated with spatiotemporal information. Once Alice has entered the corresponding parameters, she places a NFC tag close to her mobile phone, which writes the created task on the tag. Next, Alice goes to the street in which the apartment is located and searches a place to hide the tag. When the tag is hidden under a windowsill, she updates the coordinates of the location where the tag is hidden in the CachedSensing application and activates the task, which position becomes visible to other users of CachedSensing.

Bob just moved in the same city as Alice and he aims to discover the city while searching for hidden tasks. During the weekend, he browses the CachedSensing map on his mobile phone in order to select a new area to visit. As several tags are hidden in the same area as the apartment visited by Alice, Bob decides to explore this area and chooses three interesting tasks based on their descriptions, one being Alice’s task. When Bob arrives in the area of interest, he uses the compass provided by the CachedSensing application (see Fig. 2) and navigates to the published coordinates of Alice’s task. Bob reaches his destination and searches for the tag hidden by Alice. After a few minutes, Bob finds the tag and holds his mobile phone near to it in order to read the task and display the related information illustrated in Fig. 3. Bob chooses to execute the task that triggers the automatic collection of a sound sample and the computation of the corresponding sound pressure on his phone. The measurement is annotated with the time and location of its capture and later, transmitted to the CachedSensing server. Once the task is completed, the logbook of Bob is updated and a new entry for Alice’s task is created.

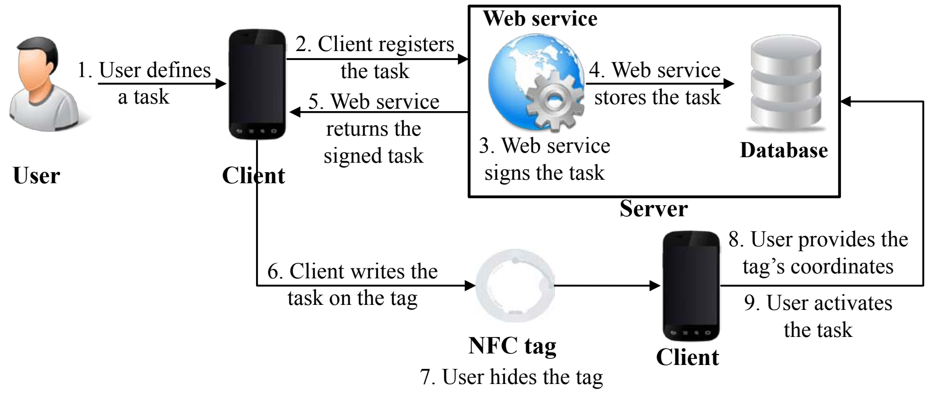

User
1. User defines a task
Client
2. Client registers the task
5. Web service returns the signed task
Web service
4. Web service stores the task
Database
3. Web service signs the task
Server
6. Client writes the task on the tag
NFC tag
7. User hides the tag
Client
8. User provides the tag's coordinates
9. User activates the task

Fig. 4. Overview of the steps necessary to create a task

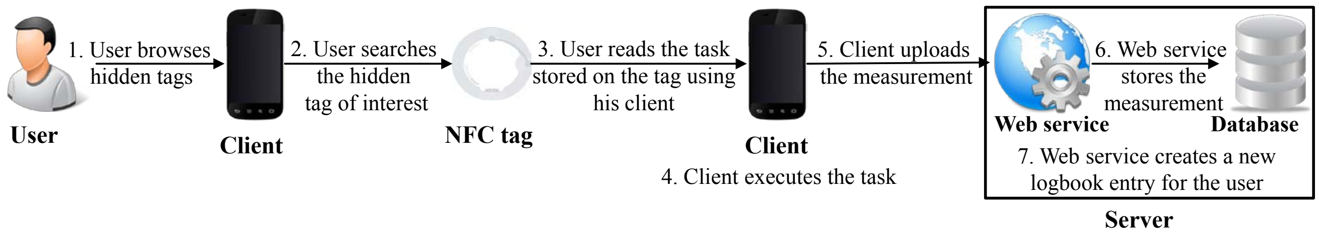

User
1. User browses hidden tags
Client
2. User searches the hidden tag of interest
NFC tag
3. User reads the task stored on the tag using his client
Client
4. Client executes the task
5. Client uploads the measurement
Web service
6. Web service stores the measurement
Database
7. Web service creates a new logbook entry for the user
Server

Fig. 5. Overview of the steps necessary to complete a task

Moreover, Alice can consult the measurement uploaded by Bob, which document that the sound pressure level in the street of interest is low on a Saturday afternoon. Assuming that other users of CachedSensing complete Alice's task on different days and at different time of day, Alice obtains a fine-granular picture of the noise pollution in this street and hence, can decide whether to buy this apartment or not.

V. The CachedSensing Architecture

The CachedSensing architecture comprises clients (i.e., the mobile phones of CachedSensing users), a server, and NFC tags. Using the client, a user can (1) register to CachedSensing, (2) create a new task, (3) perform a task, and (4) consult information about available tasks, created tasks and/or performed tasks. On the other side, the server manages the registration of the users and the created tasks. It also stores the registered tasks and the sensor readings captured by the clients. In the following, we address the mechanisms involved in the creation and completion of tasks, respectively. We provide details about their implementation in Section VI.

A. Task Creation Mechanisms

As illustrated in Fig. 4, a user starts the creation process by defining the characteristics of the task using the CachedSensing application on his client. He first gives a title to his task and indicates the type of measurement he wants to collect. For example, he may be interested in collecting pictures, videos, sound samples, sound pressure levels, etc. Then, he defines the type of task. For example, he can choose between (1) a single task, (2) a periodic task, and (3) a multi-station task. A single task collects a unique sensor reading, whereas a periodic task collects multiple sensor readings, which number and period are both defined by the task creator. In a multi-station task, subtasks are distributed on different tags and one subtask can only be completed if the previous subtask has been completed. A multi-station task is thus completed if all subtasks have been completed in the correct order. Moreover, he can define a start and end date for the task and provides additional information to find it. Once the user has defined all characteristics of his task and validated it, the task is transmitted by the client to the server. The server registers the task by signing it, and attributing a unique identifier to the task. The signature guarantees that the task is conform to the CachedSensing requirements. The server stores the task, its identifier, and its signature, before transmitting them to the client. Next, the user having created the task writes task and its signature on a NFC tag using his client. The tag contains all information necessary to fulfill the task even if no connection between the client and the server can be established. The user then hides the tag near to the location of interest, and provides the coordinates of the hidden tag to the server, which updates the task details accordingly. Finally, the task creator activates his task, which becomes visible on a map available to other CachedSensing users.

B. Task Completion Mechanisms

Fig. 5 shows the main steps necessary for the completion of a task. First, a user browses tasks hidden by other users in a specific area of interest (e.g., his neighborhood, a new area

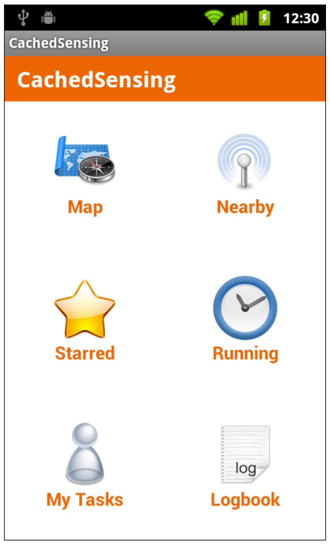


Fig. 6. Screenshot of the main interface

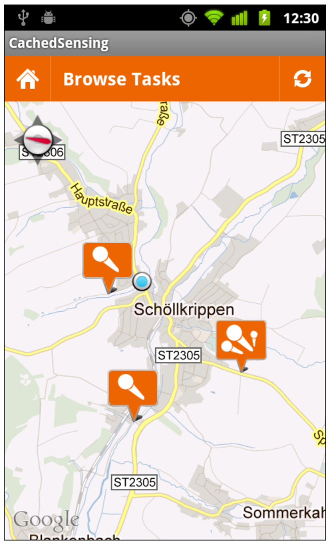
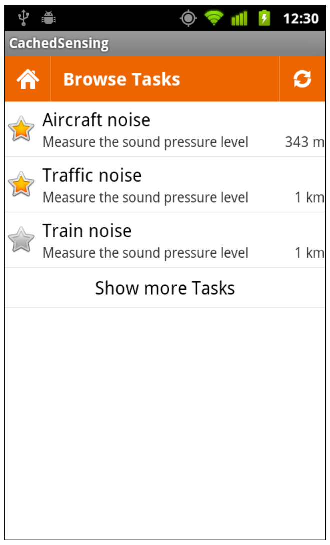


Fig. 7. Map showing available tasks



Aircraft noise Measure the sound pressure level	343 m
Traffic noise Measure the sound pressure level	1 km
Train noise Measure the sound pressure level	1 km

Show more Tasks

Fig. 8. List of tasks in the vicinity of the user

to explore, etc.). For each area of the map, the client requests the corresponding tasks from the server and updates the map according to the responses of the server. The user selects a task to complete. Note that users can browse tasks either in their vicinity or independently of their current location. Next, the user uses the published coordinates of the cache and follows the indications provided by the task creator in order to find the location where the tag is hidden. When he finds the tag, he holds his client near to the tag. The client first checks the validity of the task stored on the tag by verifying its signature. This prevents clients from executing tasks written by malicious users and not respecting the CachedSensing requirements. The details of the task are then displayed on the client and the user can choose to execute the task or not. Depending on the type of task, the execution of the task is either manually managed by the user or automatically managed by the client. The collected sensor readings are then transmitted to the server. The server stores the sensor readings and creates a new logbook entry for the user having successfully completed the task. The server also provides details about the completed task and the collected sensor readings to the creator of the task and other CachedSensing users.

VI. The CacheSensing Prototype

In this section, we present our prototype implementation. We first provide details about the implementation of the clients, before addressing the implementation of the server.

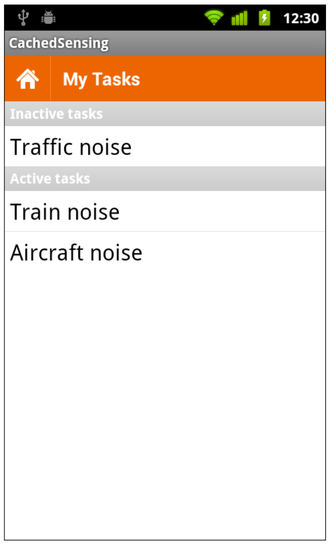
A. Clients

The clients are implemented on Nexus S mobile phones running the Android operating system and supporting the NFC technology. We present the different functions supported by the clients and we discuss their implementation according to the user interfaces with which they are associated. Additionally, we detail the implementation of the tasks on NFC tags.

1) User Interfaces and Associated Functions: Users first register to the CachedSensing application by either creating a new user account or log in with their existing account. Next, they access the main interface of CachedSensing application illustrated in Figure 6. Each icon allows users to access one of the following function/interface.

a) "Map" interface: By selecting the *Map* icon, users navigate to the map interface shown in Fig. 7. Since each task is represented by an icon, users can browse for tasks and, e.g., choose the next task they want to complete. Each symbol provides a hint about the nature of the task. For example, a single microphone indicates that the task is based on a sound sample and requires the collection of a unique sensor reading, while three microphones indicate a periodic task. By selecting an icon, users access to the main characteristics of the task, such as its title, the measurement type, the task type, and the distance to the task. If they are interested in additional information, they can access the *Task Details* interface illustrated in Fig. 3. This interface provides, e.g., the name of the task creator, the exact task coordinates, the number of times the task was completed, and who completed the task. Moreover, a star indicates whether the users have already marked this task as of interest. The indicated distance to the task is determined based on positioning information provided by both the built-in GPS receiver and the network (i.e., current cell IDs and Wi-Fi signals). For coarse-grained positioning, we privilege the network-based positioning system, while we activate the GPS positioning system for fine-grained positioning. We apply the same strategy for further functions based on the location of the users.

b) "Nearby" interface: By selecting the *Nearby* icon, users access a list of tasks located within a radius of 3 km centered on their current location (see Fig. 8). If no task is found or users are not satisfied with the results, they can manually extend the radius. Additionally, users can toggle the star of each task in order to mark it as of interest.


12:30

CachedSensing

My Tasks

Inactive tasks

Traffic noise

Active tasks

Train noise

Aircraft noise

Fig. 9. Summary of created tasks

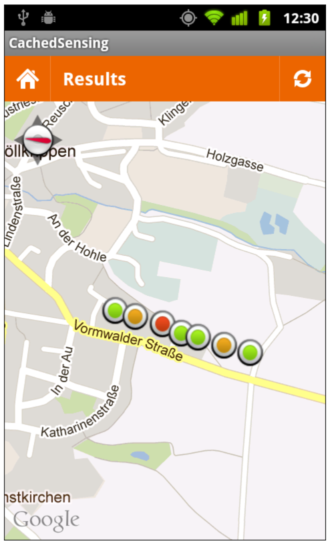
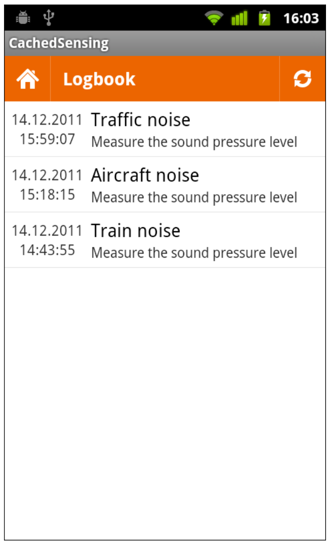


Fig. 10. Map showing the collected sound levels


16:03

CachedSensing

Logbook

14.12.2011 15:59:07	Traffic noise Measure the sound pressure level
14.12.2011 15:18:15	Aircraft noise Measure the sound pressure level
14.12.2011 14:43:55	Train noise Measure the sound pressure level

Fig. 11. Summary of completed tasks

c) "Starred" interface: Users directly access the list of the tasks they marked using the star associated with each task. The list can include, e.g., the next tasks the users want to complete, or their favorite tasks.

d) "Running" interface: Users obtain a list of still active periodic and/or multi-station tasks executed by their client.

e) "My Tasks" interface: This interface allows users to create tasks using the *New Task* interface introduced in Fig. 1. Using the *New Task* interface, users determine the task characteristics as detailed in Section V-A. By selecting the *Create* button, the new task is registered on the server. Users can choose to write the task on a NFC tag either directly after its creation or later on. For a postponed activation, users select the inactive tasks in the list shown in Fig. 9 and follow the same steps as for a direct activation. This implies holding the tag close to the client in order to write the task on the tag, and providing the coordinates of the task to the server using the *Task Details* interface introduced in Fig. 3. Note that the format in which a task is written on a tag is detailed in Section VI-A2. In addition, tasks creators can activate/deactivate the tasks they created, write these tasks on tags, and update the coordinates and the description of the tasks using the *Task Details* interface. Furthermore, users can consult the results of the completed tasks on a map. For example, Fig. 10 shows collected sound levels. The color of the marker indicates the sound pressure level – green for low, orange for medium, and red for high.

f) "Logbook" interface: Users can consult the tasks they completed via the *Logbook* interface (see Fig. 11). Each task is listed according to its date and time of completion and includes its title and the collected sensor readings.

2) Writing and Reading CachedSensing Tasks: In our prototype implementation, we use MIFARE Ultralight C type 2 NFC tags [16]. NFC tags of type 2 are based on the ISO/IEC 14443A standard [6] and have a storage capacity of up to 2 kB with a transmission speed of 106 kbps. We selected these tags because of their low cost (around 1 USD) in order to limit the investment of task creators for each created task as much as possible. Additionally, they can be set into a read-only state after their creation in order to prevent malicious users from abusing tags and deploying their own tasks on tags deployed by other users. Note that the utilization of other tags is, however, possible.

When users are in the task creation process and hold a tag close to their client, the corresponding task is automatically written on the tag by the client according to the NFC Data Exchange Format (NDEF) [7]. Despite the low storage capacity of the tags, we chose not to only write the task IDs on the tags, but also all necessary information for the execution of the task. Consequently, the tasks can still be executed even in the absence of connection to the server due to, e.g., unavailable Internet services in remote locations. As a result, each tag contains the following information organized according to the structure presented in Fig. 12:

- The *Task ID*.
- The field *Type* indicates the type of task: 0 for a single task, 1 for a periodic task, 2 for a multi-station task.
- The field *Measure* indicates the type of measurement: 1 corresponds to sound pressure measurement. Note that additional measurements can be included. We, however, consider this extension as future work (cf. Section VIII).
- The flag L indicates whether the collected sensor reading should be annotated with location information.
- The flag T indicates whether the collected sensor reading should be annotated with time information.
- The flags S and E indicate whether the task has a start date and a end date, respectively. If yes, the start date is written in the *Start date* field and the end date is written in the *End date* field.
- For periodic tasks, the field *Unit* indicates the unit of

```
Bit   0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
0    |                    Task ID                    |
16   |                                               |
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
32   |    Type    |  Measure  |L |T | Res |S |E | Res |
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
48   |                  Start date                   |
64   |                                               |
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
80   |                   End date                    |
96   |                                               |
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
112  | Unit|    Period     |            Count        |
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
128  |                                               |
...  |                Task signature                 |
480  |                                               |
     +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Fig. 12. Data structure on a NFC tag

the period: 0 for minutes, 1 for hours, and 2 for days. The field *Period* indicates the duration of the period in the corresponding unit, and the field *Count* indicates the total of the sensor readings to collect.

- The field *Task signature* contains the signature of the task generated by the server.
- *Res* corresponds to a reserved field.

When users find a hidden tag and hold their client close to the tag, the client automatically parses the task stored on the tag in the background and stores it in a SQLite database. The client displays the information stored on the tag and the users can execute the task by selecting the *Execute* button. The capture of the sound samples in the background and the computation of the corresponding sound pressure levels are realized based on the *HermitAndroid* library [17]. Additionally, the execution of periodic tasks is supported by the built-in *AlarmManager* [18]. Note that we plan to include additional sensor modalities in a future version of our prototype implementation as discussed in Section VIII.

B. Server

In our prototype implementation, the server comprised an Apache Tomcat web server, a Jersey web service, and a MySQL database. The communication between the server and the clients is synchronized using the built-in Android *SyncAdapter* [19]. The authentication of the clients relies on the Basic Access Authentication scheme [20] over SSL. Each authenticated client transmits the tasks to register and the collected sensor readings to the server, which stores them in the database. The web service supports the realization of the CachedSensing functions detailed in Section VI-A. For example, using the web service, clients can request tasks stored in the database based on their ID and their location. They can also request sensor readings and logbook entries based on the corresponding task IDs. The responses of the server are stored in the SQLite database of the clients for offline and faster access.

VII. The CachedSensing Deployment

Fig. 13 shows eight selected caches of our deployment in Schöllkrippen, Germany. We hid the tags (1) on the backside of a sign, (2) on a wooden cart, (3) on a metallic container, (4) in a tree trunk, (5) on a tree, (6) on a bench, (7) on a traffic sign, and (8) on an electrical enclosure. Note that the tags we chose for our deployment are metal-isolated and hence, can also be deployed on metallic surfaces. Each tag is associated with the collection of one or several sound pressure levels. In caches 5 to 7, users only need to collect one measurement in order to complete the task, while they need to collect five measurements (one measurement every minute) in cache 8. Caches 1 to 4 are multi-station caches. This means that users must first perform the task of cache 1, before performing the one of cache 2 and so on.

A. Deployment Issues and Lessons Learnt

By deploying the tags, we have encountered the following issues. Firstly, choosing an appropriate location for a NFC tag can be quite challenging. If a tag is placed in a visible location, users may find it easily. While this may be appropriate for beginners, experienced users may, however, expect more demanding caches. On the other hand, the more difficult the search of the tag is, the fewer users may find it and hence, the fewer sensor readings may be collected. Consequently, task creators must find a tradeoff between simple and complex

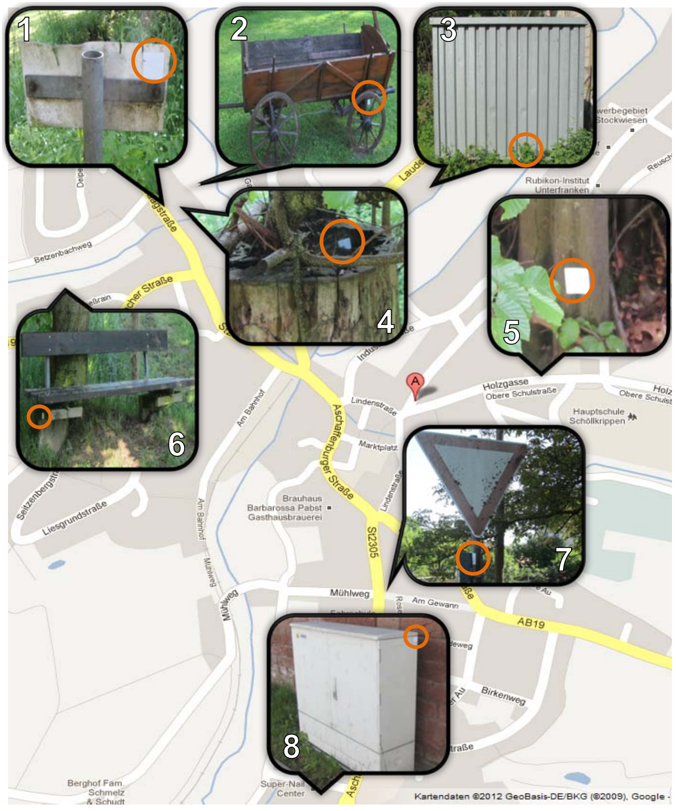


Fig. 13. Map of the CachedSensing deployment

caches in order to encourage the participation of users with different degrees of experience, and foster the contributions of sensor readings to the application. Furthermore, task creators need to carefully select the location of the caches, since hiding and searching tags can attract the attention of passersby. For example, users searching tags in proximity of habitations or schools may appear suspect to passersby who may call the police. Note that during the deployment of the tags, we experienced suspicious persons who stared at us in an inquisitive fashion. Again, the cache locations should be sufficiently isolated in order not to attract unnecessary attention, but be still challenging to search for. On the other hand, visible locations, such as advertisement boards, may offer appropriate locations, since tags may blend into the background and remain unnoticeable to unaware people. Finally, each cache should offer an appropriate surface for the tag to properly adhere and sufficient space to be easily accessible by a mobile phone.

Secondly, off-the-shelf NFC tags often present either a unicolor surface or a printed logo. As shown in Fig. 13, this results in a high contrast between the tags and their surroundings that may not only attract the attention of CachedSensing users, but also of passersby. This poses the risk that passersby degrade or remove deployed tags, preventing CachedSensing users from completing the associated sensing tasks. The financial loss incurred by such degradation is, however, limited due to the low cost of NFC tags (around 1 USD) compared to existing devices, e.g., the Chirp beacons introduced in Section III. However, it prevents users from being able to perform the associated tasks. In order to lower the risk of degradation and make the tags more difficult to find for CachedSensing users, the tags may be camouflaged by customizing their surface according to the material they are fixed on. The customization can be realized using, e.g., waterproof pens or printed motives.

Thirdly, most of our deployed tags are exposed to natural elements, such as rain, pollution, sun, etc. We therefore chose outdoor tags for our deployment [21] and tested their robustness. In July 2011, we deployed a tag in a natural and exposed environment. Almost eleven months later, the tag can still be read. Moreover, we investigated the resistance of the tags to water, dirt, and erosion, in artificial settings. We first immersed a tag under water for one week and periodically controlled its readability. We observed that the tag was still readable at the end of the immersion period. Next, we tested whether dirt perturbs the reading of a tag. To this end, we covered the tag with soil and dirt layers of different thicknesses. We observed that the tag was readable until the thickness of the layer reached 2 cm, which roughly corresponds to the range of the NFC communication. Consequently, dirt and soil do not prevent tags from being read, except if the thickness exceeds the NFC range. Finally, we scraped a tag on a rough surface in order to artificially simulate erosion and observed that the tag was still functioning despite the presence of scratches on its surface. As a result, the tags have been shown to be resistant to natural as well as artificially amplified environmental conditions.

VIII. Discussions and Future Work

Using our current prototype implementation, users can only create sensing tasks involving sound pressure level measurements. While these measurement are helpful to gain insights about noise pollution, the range of offered sensing tasks need to be extended in the future. This may not only increase the interest of task creators, but also the interest of users in contributing to the application. To this end, the CachedSensing architecture has been designed to allow an easy integration of additional sensor modalities. For example, we plan to offer new sensing tasks based on the cameras and accelerometers embedded in the mobile phones of the users. We can also imagine sensing tasks leveraging the wireless interfaces of the mobile phones.

Moreover, our prototype implementation allows users to define the constraints associated to the collection of the sensor readings only on a temporal basis by e.g., defining the sampling frequency of periodic tasks. We however intend to offer additional options to the users, with which they could define spatial and temporal conditions with finer degrees of granularity. For example, users could parameterize a task such that the sensor readings should only be captured between 2.00 and 4.00 pm and in a range of 10 m around the cache location. Furthermore, additional supports to store the tasks could be introduced, such as QR codes. Again, this would increase the diversity of the sensing tasks and potentially foster the contributions of both task creators and CachedSensing users.

In the near future, we plan to release the CachedSensing application at large scale. This includes making the application available in the online market Google Play and advertising our approach on geocaching websites. A large scale deployment completed by questionnaire-based studies would allow us to investigate and answer important questions, such as analyzing the real impact of the gaming aspect on the motivation of users to contribute to the application. Finally, we could gather information about, e.g., the usability of our application. This would help us to further develop and improve our concept and prototype.

IX. Conclusions

In this paper, we have proposed a novel concept called CachedSensing. Our approach aims to foster user contributions to participatory sensing applications, and also provides tools for users to easily create and distribute sensing tasks. We have presented the CachedSensing architecture as well as our prototype implementation. We have leveraged our prototype to deploy sensing tasks in the wild and discussed the lessons we learnt during the deployment process. In the long term, we believe that adding a gaming component to participatory sensing applications may broaden the user community by attracting new profiles of users (such as geocachers) and hence, help these applications to cross the chasm.

Acknowledgments

This work was supported by CASED (www.cased.de).

References

[1] D. Christin and M. Hollick, "We Must Move - We Will Move: On Mobile Phones as Sensing Platforms," in *Proceedings of the 10th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN)*, 2011.

[2] "Geocaching," Online: http://www.geocaching.com (accessed in 05.2012).

[3] "Open Caching," Online: http://www.opencaching.com (accessed in 05.2012).

[4] "ISO 21481. Information Technology – Telecommunications and Information Exchange between Systems – Near Field Communication Interface and Protocol (NFCIP-2)," Online: http://www.iso.org (accessed in 05.2012).

[5] "NFC Forum," Online: http://www.nfc-forum.org (accessed in 05.2012).

[6] "ISO 14443. Identification Cards – Contactless Integrated Circuit Cards," Online: http://www.iso.org (accessed in 05.2012).

[7] "NFC Data Exchange Format (NDEF) Technical Specification," Online: http://www.nfc-forum.org (accessed in 05.2012).

[8] T. Luo and C. K. Tham, "Fairness and Social Welfare in Incentivizing Participatory Sensing," in *Proceedings of the 9th Annual IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2012.

[9] J.-S. Lee and B. Hoh, "Dynamic Pricing Incentive for Participatory Sensing," *Pervasive and Moblie Computing*, vol. 6, no. 6, pp. 693–708, 2010.

[10] L. G. Jaimes, I. Vergara-Laurens, and M. A. Labrador, "A Location-based Incentive Mechanism for Participatory Sensing Systems with Budget Constraints," in *Proceedings of the 10th IEEE Conference on Pervasive Computing and Communications (PerCom)*, 2012.

[11] S. Reddy, D. Estrin, M. Hansen, and M. Srivastava, "Examining Micropayments for Participatory Sensing Data Collections," in *Proceedings of the 12th ACM International Conference on Ubiquitous Computing (UbiComp)*, 2010.

[12] K. Han, E. Graham, D. Vassallo, and D. Estrin, "Enhancing Motivation in a Mobile Participatory Sensing Project through Gaming," in *Proceedings of the 4th IEEE International Conference on Social Computing (SocialCom)*, 2011.

[13] "Geocaching Challenges," Online: http://www.geocaching.com/challenges (accessed in 05.2012).

[14] "Chirp," Online: http://www.garmin.com/chirp (accessed in 05.2012).

[15] "Munzee," Online: http://www.munzee.com (accessed in 05.2012).

[16] "MF0ICU2 - MIFARE Ultralight C," Online: http://www.nxp.com (accessed in 05.2012).

[17] "HermitAndroid Library," Online: http://moonblink.googlecode.com.

[18] "AlarmManager," Online: http://developer.android.com (accessed in 05.2012).

[19] "AbstractThreadedSyncAdapter," Online: http://developer.android.com (accessed in 05.2012).

[20] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication. RFC 2617 (Draft Standard)," Online: http://www.ietf.org/rfc/rfc2617.txt (accessed in 05.2012), 2009.

[21] "Outdoor Type 2 NFC Sticker (PVC) - Square (35mm x 35mm)," Online: http://www.tagstand.com (accessed in 05.2012).