Enabling Crowd Sensing for Non-Experts

Ajay Palleri Kesavan*, Neil Prakasam†, Ambika Hegde‡ and Brent Lagesse§
Computing and Software Systems
University of Washington Bothell
Bothell, WA USA
Email: *ajayre@uw.edu, †npraka@uw.edu, ‡ambikagh@uw.edu, §lagesse@uw.edu

***Abstract*—Crowd sensing utilizes a large group of users with devices capable of sensing to collect data for analysis. The motive of crowd sensing is to generate inferences from collected data. Examples of these applications include measuring traffic congestion, bike route quality, parking spot availability, and pollution levels. Since these applications are not necessarily the job of software developers, one of the challenges for crowd sensing is the effort required for task initiators to create and deploy a task, especially when they lack technical expertise. We are developing a framework to ease the development of crowd sensing tasks for non-experts by enabling them to create crowd sensing tasks and analysis using automated GUI-based tools for many common cases. These tasks are then pushed to a marketplace to make it easier for users to discover tasks that they want to participate in (or that have incentives that are appealing to them). We will demonstrate this system and allow participants of the demo session to create their own tasks and Percom attendees can participate in collecting data for those tasks throughout the week of Percom.**

I. Introduction

Commonly available smartphones have a wide variety of sensors available on them. This has inspired innovative use of the mobile phones to collect and aggregate data from the on-board sensors in order to perform analysis and decision making for a variety of applications. These crowd sensing applications include domains such as smart cities, air quality monitoring, and human sciences [1][2] [3].

To take advantage of these opportunities, developers need experience in a variety of fields ranging from mobile app development, networking, security, data analytics, and visualization. The focus of our work is to lower the barriers for entry to non-experts to deploy crowd sensing tasks. To this end, we are developing intuitive GUI-based system to generate crowd sensing tasks and to provide common data analysis techniques.

Other crowd sensing frameworks like Medusa[4] APISENSE [5] need programming experience to deploy a task. Medusa[4] needs the deployer to learn MedScript Programming Language to implement a new task. APISENSE[5] is a Software-as-a-Service solution for crowd sensing.

We will demonstrate a system that is targeted towards people with no experience in mobile application development. We provide an interface to deploy a crowd sensing task to a marketplace, an Android application to act as a sandbox for the tasks to execute, and an interface for data to be analyzed and displayed.

II. Design

Figure 1 illustrates the architecture and exchange of data on the devices. In this we have two primary users of our system. The deployer of crowd sensing task (who provides the task) and the worker (who opts-in to the task). For a deployer to deploy an application they need to log into the admin portal with OAuth, choose from the total list of sensors, provide incentives (if desired), provide the frequency of data collection and submit a task to the market place. Deployers can also define how the data will be analyzed. In this case they select from a library of analysis algorithms and compatible visualization mechanisms. Figure 3 demonstrates an example GPS-based visualization that a task deployer can create from a GUI. Additionally complexity can be added in the description file, but these common tasks are all automated on behalf of non-expert users.

For a worker to take part in the system they download mobile application. To maintain unique identity the worker has to log in with Oauth into the application to start or stop contributing data. To opt-in to a crowd sensing task, the market place tab renders all the available tasks for the users mobile phone and the task is switched on in the market place as shown in figure 2. Start job button is used to start contributing data to the system. A foreground service is launched notifying the start of the service. To stop the service the user has to click the mobile application and touch the stop service button.

III. Demo Setup

We will demonstrate the full process from creation to visualization of an application built with tasks from our marketplace. We will also allow users to sign up to act as workers during the course of the week of Percom.

To contribute as a worker for this application, the user downloads the application (we will provide a QR code to make app discovery and installation easier) and logs into our Android application through OAuth, goes to the Available tasks section to find the tasks that can be performed by the phone's sensors. The user then gets a list of tasks on the marketplace tab with the description, permissions and the required data. The permissions necessary Android permissions are requested at tasks switch on time.

Depending on all the tasks opted-in by the users, the application collects data from the user whenever the phone is switched on. The data is stored locally on the phone temporarily to ensure that data is not lost because of intermittent

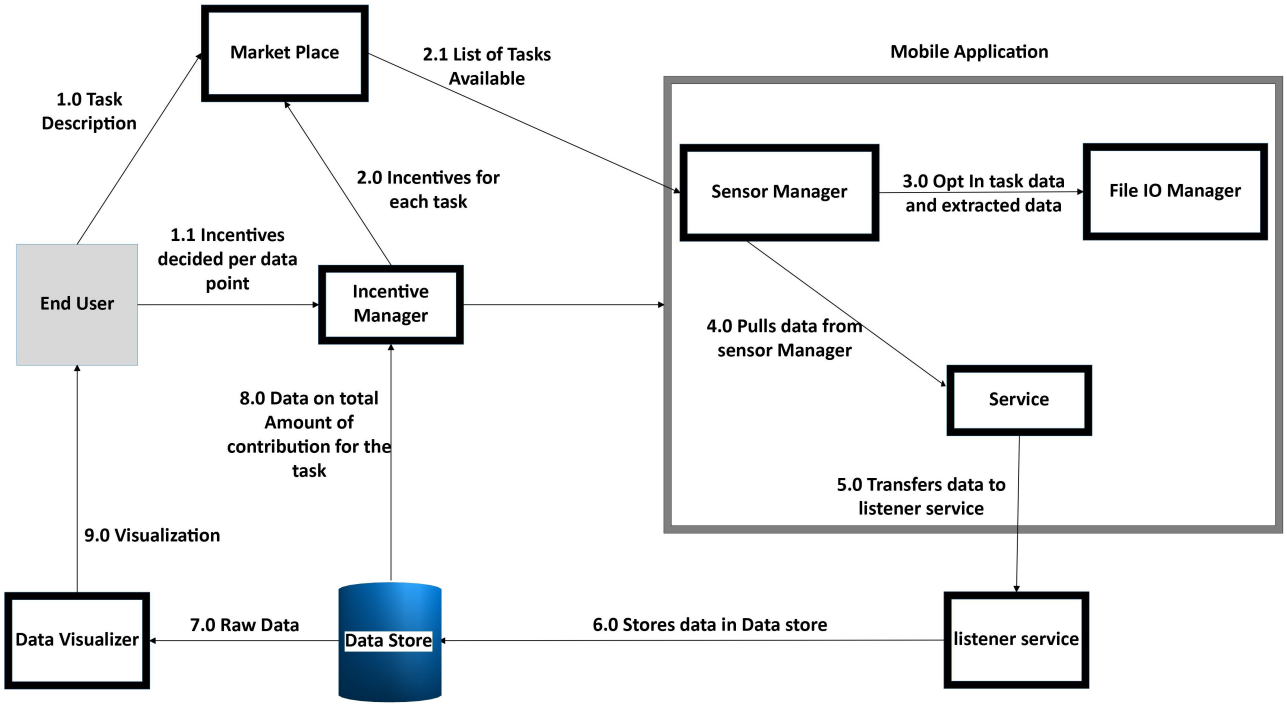

Fig. 1. Architecture of crowd sensing framework

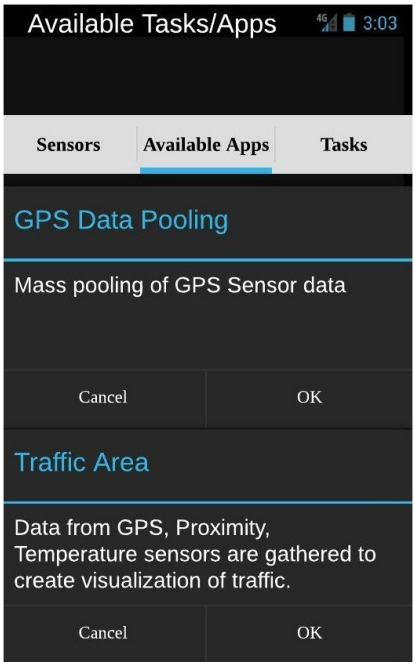

Fig. 2. Screen shot of the marketplace

Internet connections. At regular intervals the stored data is uploaded to the data store. The data is then distributed from the data store to individual task owners.

A. *Scenario*

A user wants to study the movement patterns of people during the weekend. The data needed by the task creator is the time stamp and location data. He/she logs into the admin portal and creates a crowd sensing task and selects the data needed in the list of sensors. The user then uses the portal to define that the data is to be displayed in a mapping service for visualization. Later, workers go to the market place and query the tasks available in the marketplace with details on permissions required, description on the application of sensor data, conditions for the data to be valid and sensors required to complete the task. If the worker would like to participate in the task she/he switches on the task in the application and the data collected at the aggregation point and then distributed to the task creator for analysis. Incentives are given to the users based on the task that they are involved in if that was defined in the task description.

In our demonstration we will allow users to:

1) Create and deploy a new crowd sensing task from our GUI.
2) Create a new analysis and display site for your crowd sensing data from our GUI.
3) How to opt-in to a crowd sensing task and start providing data as a worker.

IV. Conclusion

The goal of this demo is to enable people who do not have extensive experience with mobile application programming to be able to create crowd sensing tasks. Through the crowd

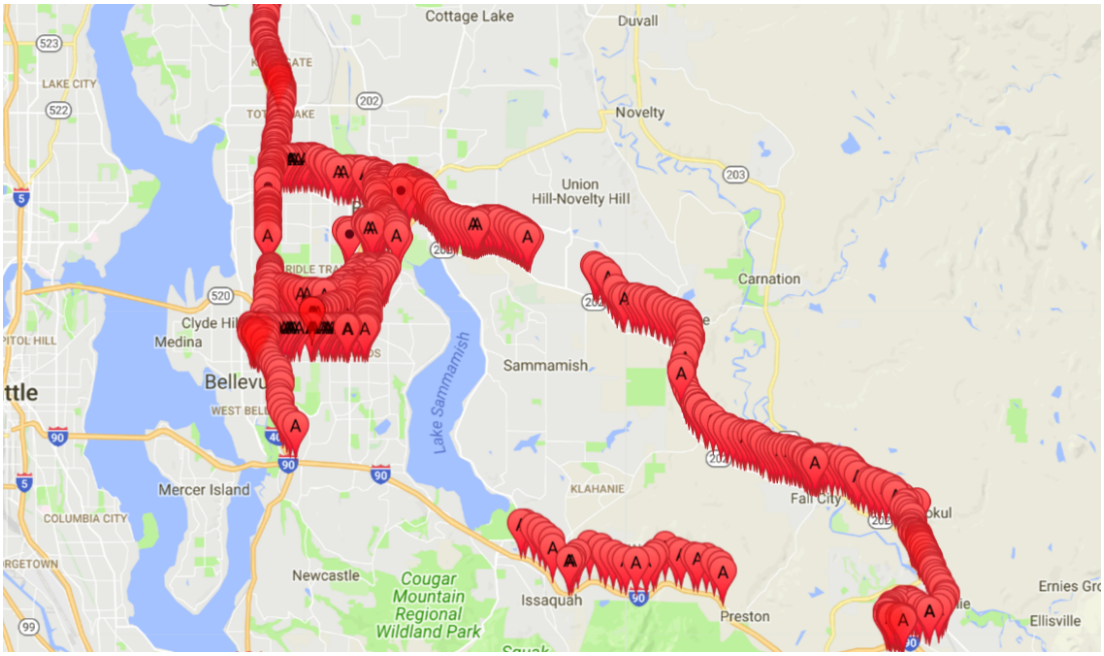

Fig. 3. Example GPS-based visualization

sensing framework we have created, users will be able to deploy these tasks as well as collect data from their users with minimal effort. When crowd sensing is exposed as a service, deployers will not need expensive hardware or knowledge of application development to spread their crowd sensing task, but rather depend on our crowd sensing framework to contribute data.

REFERENCES

[1] C. Leonardi, A. Cappellotto, M. Caraviello, B. Lepri, and F. Antonelli, "SecondNose: An Air Quality Mobile Crowdsensing System," in *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, ser. NordiCHI '14. New York, NY, USA: ACM, 2014, pp. 1051–1054. [Online]. Available: http://doi.acm.org/10.1145/2639189.2670273

[2] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li, "Jigsaw: Indoor Floor Plan Reconstruction via Mobile Crowdsensing," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '14. New York, NY, USA: ACM, 2014, pp. 249–260. [Online]. Available: http://doi.acm.org/10.1145/2639108.2639134

[3] J. Weppner and P. Lukowicz, "Bluetooth based collaborative crowd density estimation with mobile phones," in *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Mar. 2013, pp. 193–200.

[4] M.-R. Ra, B. Liu, T. F. La Porta, and R. Govindan, "Medusa: A Programming Framework for Crowd-sensing Applications," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '12. New York, NY, USA: ACM, 2012, pp. 337–350. [Online]. Available: http://doi.acm.org/10.1145/2307636.2307668

[5] R. Rouvoy, "APISENSE (http://apisense.io)," in *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, vol. 2, Jun. 2016, pp. 61–61.