Mobile Crowdsensing: Current State and Future Challenges

Raghu K. Ganti, Fan Ye, and Hui Lei
IBM T. J. Watson Research Center, Hawthorne, NY

Abstract—An emerging category of devices at the edge of the Internet are consumer centric mobile sensing and computing devices, such as smartphones, music players, and in-vehicle sensing devices. These devices will fuel the evolution of the Internet of Things as they feed sensor data to the Internet at a societal scale. In this paper, we will examine a category of applications that we term *mobile crowdsensing*, where individuals with sensing and computing devices collectively share information to measure and map phenomena of common interest. We will present a brief overview of existing mobile crowdsensing applications and illustrate various research challenges.

1 Introduction

The integration of sensing and embedded everyday computing devices at the edge of the Internet will result in the evolution of an *embedded Internet* or the *Internet of Things*.

An emerging category of edge devices that are fueling this evolution are consumer centric mobile sensing and computing devices. These include devices such as smartphones (iPhone, Google Nexus), music players (iPods), sensor embedded gaming systems (Wii, XboX Kinect), and in-vehicle sensing devices (GPS, OBD-II). They have become extremely popular recently and are potentially important sources of sensor data. They are typically equipped with various sensing faculty and wireless capabilities and are connected to the Internet. As an example, a sample list of mobile devices and their corresponding sensing capabilities are provided in Table 1.

We observe from Table 1 that these mobile devices can be used to measure various *individual* and *community* phenomena. Individual phenomena are those pertaining to a particular device owner, such as movement patterns (e.g. running, walking, climbing stairs), modes of transportation (e.g. biking, driving, taking a bus, riding the subway), and activities (e.g. using an ATM, visiting a specific store, having a conversation, listening to music, and making coffee). Community phenomena are those pertaining to the aggregate of surroundings and not limited to particular individuals. These include pollution (air/noise) levels in a neighborhood, real-time traffic patterns, pot holes on roads, road closures, and transit timings. Such large scale, community phenomena monitoring is possible when a community of individuals share the sensor data they collect towards a common goal, usually with certain processing involved.

This type of community sensing is popularly called *participatory* sensing [1] or *opportunistic* sensing [2]. Participatory sensing is defined as the kind of sensing where individuals are actively involved in contributing sensor data (e.g. taking a picture, reporting a road closure). On the other hand, opportunistic sensing is where the sensing is more autonomous and user involvement is minimal (e.g. continuous location sampling). We consider that, in terms of the level of user involvement, community sensing spans a wide spectrum, with participatory sensing and opportunistic sensing at the two ends of the spectrum. We therefore coin the term *mobile crowdsensing (MCS)* to refer to a broad range of community sensing paradigms[1].

In the rest of this article, we will survey existing crowdsensing (both participatory and opportunistic) applications (Section 2), identify unique characteristics of MCS applications (Section 3), and discuss the research challenges they face, their solutions and drawbacks. The research challenges

1. The notion that crowdsensing spans a spectrum from participatory sensing to opportunistic was suggested by our colleague Thomas Erickson

Device	Inertial	Compass	GPS	Microphone	Camera	Proximity	Light
iPhone	✓	✓	✓	✓	✓	✓	✓
Nexus S	✓	✓	✓	✓	✓	✓	✓
Nokia 6210	✓	✓	✓	✓	✓		✓
iPod Touch	✓			✓	✓		✓
Garmin ForeRunner 410	✓	✓	✓				

TABLE 1
Sensors on various mobile sensing devices

that we discuss include: (i) *Localized analytics*, (ii) *Resource limitations*, (iii) *Privacy*, (iv) *Aggregate analytics*, and (v) *Architecture*.

2 Mobile CrowdSensing Applications

In this section, we will briefly discuss existing mobile crowdsensing applications, which provide a basis for illustrating various research challenges in the rest of this article. We classify MCS applications into three different categories based on the type of phenomenon being measured or mapped. These include (i) *Environmental*, (ii) *Infrastructure*, and (iii) *Social*.

In environmental MCS applications, the phenomena being measured are those of the natural environment. Examples of such applications include measuring pollution levels in a city, water levels in creeks, and monitoring wildlife habitats. Such applications enable the mapping of various large scale environmental phenomena by involving the common man. An example prototype deployment for pollution monitoring is Common Sense [3]. Common Sense uses specialized handheld air quality sensing devices that communicate with mobile phones (using Bluetooth) to measure various air pollutants (e.g. CO_2, NO_x). These devices when deployed in a large population, collectively measure the air quality of a community or a large area. Similarly, one can utilize microphones on mobile phones to monitor noise levels in communities. Another example is *CreekWatch* developed by IBM Almaden Research Center. It monitors water levels and quality in creeks by aggregating reports from individuals, such as pictures taken at various locations along the creek, or text messages about the amount of trash. Such information can be used by the water control boards to track pollution to water resources.

Another important category are *infrastructure* applications that involve the measurement of large scale phenomena related to public infrastructure. Examples include measuring traffic congestion, road conditions, parking availability, outages of public works (e.g. malfunctioning fire hydrants, broken traffic lights), and real-time transit tracking. Early MCS deployments measured traffic congestion levels in cities, examples of which include MIT's CarTel [4] and Microsoft Research's Nericell [5]. CarTel utilizes specialized devices installed in cars to measure the locations and speeds of the cars and transmit the measured values using public WiFi hotspots to a central server. This central server can then be queried to provide information such as least delay routes or traffic hotspots. On the other hand, Nericell utilizes individuals' mobile phones to not only determine average speed or traffic delays, but also detect honking levels (especially in countries like India where honking is common) and potholes on roads. Another MCS example is ParkNet [6], an application that detects available parking spots in cities using ultrasonic sensing devices installed on cars combined with smart phones.

Finally, another category are *social* applications. As an example, individuals can share their exercise data (e.g. how much time one exercises in a day) and compare their exercise levels with the rest of the community. They can use this comparison to help improve their daily exercise routines. Example deployments include BikeNet [7] and DietSense [8]. In BikeNet, individuals measure location and bike route quality (e.g. CO_2 content on route, bumpiness of ride) and aggregate the data to obtain "most" bikeable routes. In DietSense, individuals take pictures of what they eat and share it within a community to compare their eating habits. A typical use case for this is for a community of diabetics to watch what other diabetics eat and control their diet or provide suggestions to others.

To summarize, the functioning of typical MCS applications is illustrated in Figure 1, which depicts the above research challenges as functional components.

3 MCS: Unique Characteristics

We will first illustrate the unique characteristics of MCS applications that differentiate them from traditional mote-class sensor networks. This will provide the reader with an idea about the research challenges faced by MCS applications.

Compared to traditional mote-class sensor networks, mobile crowdsensing has a number of unique characteristics that bring both new opportunities and problems. First, today's mobile devices have much more computing, communication and storage resources than mote-class sensors, and they are usually equipped with multi-modality sensing capabilities. These will enable a whole lot more applications that require resources and sensing modalities far beyond what mote-class sensors possess. Second, millions of mobile devices are already "deployed in the field": people carry these devices wherever they go and whatever they do. By leveraging these devices we could potentially build large scale sensing application in much shorter time and at much lower cost. For example, instead of installing road-side cameras and loop detectors, we can collect traffic data and detect congestions using smartphones carried by drivers along highways. They allow us to avoid the long time and high investment needed for specialized sensing infrastructure.

The dynamic conditions in the collection of mobile devices and the need of data reuse in different applications in MCS are also quite different from those of traditional sensor networks. In MCS, the population of mobile devices, the kinds of data each can produce, and the quality in terms of accuracy, latency, confidence can change all the time due to device mobility, variations in their energy levels and communication channels, and device owners' preferences. Identifying the right set of devices that can produce the desired data and instructing them to sense with proper parameters to ensure desired quality is a complex problem. In traditional sensor networks, the population and the data they can produce are mostly known a prior; thus controlling the data quality is much easier. The same sensor data have been used for different purposes in many existing MCS applications. For example, the accelerometer readings have found usage in transportation mode identification, pothole detection, human activity pattern extraction. To efficiently support multiple concurrent applications, it is critical to identify common data needs and support the reuse of sensor data across applications. In contrast, a conventional sensor network is typically intended for a single application and reuse for vastly different purposes is rarely needed.

Because devices are owned and carried by individual users, humans are usually involved in the loop. This is a double-sided sword. On the one hand, the intelligence and mobility of humans can be leveraged to help applications collect higher quality or semantically complex data that may otherwise require sophisticated hardware and software. For example, humans can easily identify available street parking spots and report with pictures or text messages, whereas an ultrasound based scanning system not only require special hardware but also sophisticated processing algorithms to ensure the reliability of data. On the other hand, humans naturally have privacy concerns and personal preferences that are not necessarily in the best interests of MCS applications but applications have to live within these constraints. The user may not want to share sensing data that may contain or reveal private and sensitive information, such as their locations; the user may set preferences that can override desired sensing activities, such as avoiding costly GPS sampling when the battery capacity is less than a threshold.

Another important implication for human involvements is incentive mechanism. It may incur energy, monetary costs or even explicit efforts on the owner of the device for sensing, processing and communicating of desired data. Unless there are strong enough incentives, the owners may not be willing to contribute their resources. For MCS applications to succeed, there have to be appropriate incentive mechanisms to recruit, engage and retain human participants. Elaboration on incentive mechanisms and other people-oriented tools are beyond the scope of this paper, since our focus is on system challenges.

4 Localized Analytics

Mobile devices are equipped with various kinds of sensors such as GPS, accelerometer, microphone and camera. The OS allows applications to access

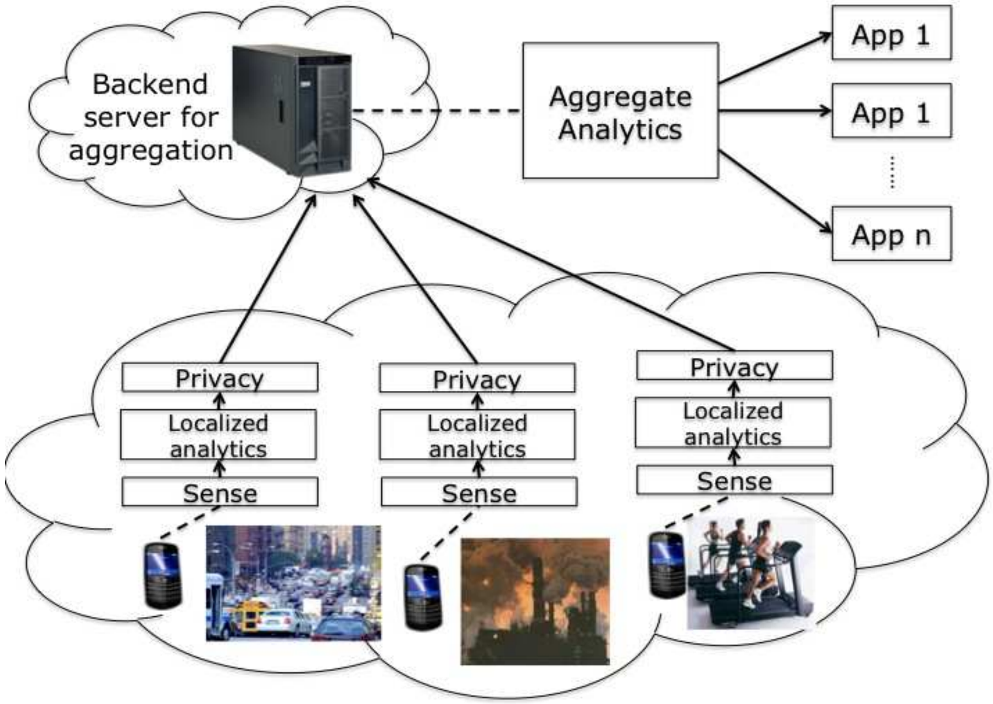

Fig. 1. Typical functioning of MCS applications. Raw sensing data is collected on devices and local analytics process it to produce consumable data for applications. After privacy preservation, the data is sent to to the backend and aggregate analytics will further process it for different applications.

the sensors and extract raw sensing data from them. However, depending on the nature of the raw data and the needs of applications, the physical readings from sensors may not be suitable for the direct consumption of applications. Many times, some *local analytics* perform certain primitive processing on the raw data on the device. They produce some intermediate results which are sent to the backend for further processing and consumption. For example, in a pothole detection [5] application, the local analytic takes the time series of the 3-axis acceleration data and identify concurrent spikes in Z and X axes to identify a potential pothole.

The motivation of such local analytics are twofold. First, the kind of processing they perform usually lead to much less amount and appropriately summarized data. It takes much less energy and bandwidth to transmit such data than the raw sensor readings. This is a well-known tradeoff in conventional mote-class sensor networks: using computation to save energy/bandwidth. Second, it reduces the amount of processing that the backend has to perform. Were all devices to upload their raw sensor readings, the backend would need significantly more processing resources. Shifting part of the burden to mobile devices, which possess significant resources nowadays, enables higher scalability. Finally, some applications may be delay sensitive. Transmitting lots of data to the backend over intermittently connected channels may take much more time than processing the data locally and sending much shortened summarized data.

The main challenge in local analytics is finding heuristics and designing algorithms to achieve the desired function. One category of functions is *data mediation*, such as filtering of outliers, elimination of noise, or makeup for data gaps. For example, due to lack of direct line of sight, sometimes the GPS does not give accurate location coordinates. If it is known that the user is traveling on a highway and the GPS coordinates should be most likely on a straight line, outliers that do not conform to the line can be discarded. Alternatively, based on the traveling speed and past coordinates, some extrapolation can be done to estimate the rough location to make up for such missing gaps of data.

Another common category of functions of local analytics is *context inference*. There are many different types of context, such as the transportation mode (whether the user in on a car, bus, train, or on foot), the kinetic modes of humans (walking, standing, jogging, running), the social settings (e.g., in a meeting, having a phone call, watching TV, etc.), or the occurrence of certain events in the surrounding environments (e.g., potholes on road, stop-and-go traffic, loud noise levels). Exactly what kinds of heuristics and algorithms are needed, usually depends on the requirements of applications. We give one example to illustrate the challenge.

One of the most commonly used type of sensor data is 3-axis accelerometer. By analyzing the amplitude and correlation of the acceleration along different axes, a local algorithm can infer various contexts. For example, in the PotPatrol project, the spikes along the vertical axis are most likely an indication of potholes. Thus by detecting when the Z-axis acceleration exceeds certain threshold, one can tell when the car hits potential potholes. However, other road conditions (such as rail crossing) can also cause sudden spikes on the vertical axis. The authors found that it is most likely that only one side of the tires hit potholes and that causes shaking and acceleration changes in X-axis as well. But for rail crossings, both sides of the tires hit them at about the same time, thus the X-axis acceleration is much less. Eventually further correlation with the acceleration on the side axis is added to achieve reliable detection of potholes.

In general, local analytics can be quite application specific. The exact algorithm they use to make the context inference, greatly depend on the nature of the application and the characteristics of the context. The current practice is to develop analytics solely for one application. This could leads to an "explosion" of analytics when many crowdsensing applications co-exist. Each analytic is working individually and there is possibility they may be accessing the same sensor data, or involve similar computation in their inference.

5 Resource Limitations: Energy, Bandwidth, and Computation

Even though they possess much more computing, bandwidth, and energy resources than mote-class sensors, mobile devices nevertheless face resource limitations. Resource constraints in traditional sensor networks have been well studied. However, MCS applications introduce many new aspects for this challenge.

First, the set of devices that are collecting sensor data are highly dynamic in availability and capabilities. Due to this highly dynamic nature, modeling and predicting the energy, bandwidth requirements to accomplish a particular task becomes much more difficult than in traditional sensor networks. Second, when there are a large number of available devices with diverse sensing capabilities, identifying and scheduling sensing and communication jobs among them under resource constraints becomes more difficult as well.

Another interesting aspect is the interdependencies between various types of sensory data due to multi-modality sensing capability. Different types of data can be used for the same purpose, but with different quality and resource consumption trade-offs. How to leverage their differences to improve the quality while minimizing resource consumption is a new problem. For example, location data can be provided using GPS, WiFi, and GSM, with decreasing levels of accuracy. Compared to WiFi and GSM, continuous GPS location sampling drains the battery much more quickly. One approach to this problem uses low duty cycling to reduce energy consumption of high quality sensors (i.e., GPS), and alternates between high and low quality sensors depending on the energy levels of the device (e.g. sample WiFi often when battery level is less than 70%). This approach trades off data quality and accuracy for energy.

The existence of multiple concurrent applications that require data of different types also complicates resource allocation. A mobile device can be sampling various sensors (e.g. GPS, accelerometer, air quality) on behalf of different applications. The approach proposed by CarTel prioritizes data collection tasks. Depending on the priority of the application that requires sensor data, the sampling rate of other sensors can be reduced (or the sensor completely switched off). For example, during peak travel times, a community may be more interested in obtaining traffic congestion levels as opposed to air or noise pollution levels. As a result, the air/noise sensors sample much less frequently or can be shut down.

A drawback of existing solutions such as low duty cycling is that they are designed for their particular application context and do not scale when many different applications co-exist. An important challenge for large scale deployment of MCS applications is that the resource constraints need to be addressed in a unified manner. How do multiple applications on the same device utilize energy, bandwidth, and computation resources without significantly affecting the data quality of each other? How does scheduling of sensing tasks occur across multiple devices with diverse sensing capabilities and availabilities (which can change dynamically)? We believe that these questions need to be answered before MCS applications can be deployed on a large scale.

6 PRIVACY

An important aspect of MCS applications is that they potentially collect sensitive sensor data pertaining to individuals. For example, GPS sensor readings can be utilized to infer private information about the individual, such as the routes they take in daily commutes, their home and work locations, and so on. On the other hand, these GPS sensor measurements (from daily commutes) shared within a larger community can be used to obtain traffic congestion levels in a given city. Thus, it is important to preserve the privacy of an individual, but at the same time enable MCS applications.

There are several approaches to privacy in existing literature, especially privacy preserving data mining literature. A popular approach is that of anonymization which removes any identifying information from the sensor data before sharing it with a third party. Such approaches are not useful in typical MCS applications. For example, anonymized GPS (or location) sensor measurements can still be used to infer the frequently visited locations of an individual and derive their personal details. Another approach is to use secure multiparty computation, where cryptographic techniques are used to transform the data to preserve the privacy of an individual. Such cryptographic techniques are compute intensive and are not scalable because they require the generation and maintenance of multiple keys. Here we propose a data perturbation based approach that adds noise to sensor data before sharing it with the community to preserve privacy of an individual. The data perturbation approach relies on adding noise in such a manner that the privacy of an individual is preserved [2], but at the same time it is possible to compute the statistics of interest with high accuracy.

Perturbation based approaches for achieving privacy in mobile crowdsensing applications have been applied in [9], [10]. These approaches perturb time series sensor data such that (i) the privacy of application-specific data streams is preserved against common estimation algorithms, (ii) the perturbation allows for computation of community aggregates within proven accuracy bounds, and (iii) the perturbation can be applied by non-expert users without having to trust the application itself. These time series perturbation techniques were applied to a traffic congestion level estimation application and a weight watchers application that computes community weight statistics over time without revealing the true weight values of individuals.

We conclude this section with a few observations in regard to privacy preservation for enabling MCS applications. First, we observe that privacy is very user specific, that is each individual has a different perception of privacy. For example, one person may be willing to share his or her location information continuously whereas another may not. Developing privacy techniques that address variation in individual preferences needs to be addressed. We also observe that the current perturbation based approaches address a class of mobile crowdsensing applications that use time series data and are restricted by the type of noise being added. A generic perturbation technique or a framework of perturbation techniques need to be developed such that privacy can be achieved in a generic setting independent of the nature of the data being shared.

7 AGGREGATE ANALYTICS

The local analytics running on mobile devices give data about the local area to applications. For many applications, they may also need to run *aggregate analytics* at the backend. These analytics detect patterns in the sensor data from large number of mobile devices. These patterns occur in certain spatial scope and within some temporal duration; they signify the features and characteristics of the physical or social environment that are interested by the user. For example, the transportation authority of a city may be interested in the spatial distribution of traffic hot spots around the road network, and how the distribution evolves over various time scales. Such insight can help them better coordinate the traffic lights to ease traffic depending on the time of the day, and better plan future road expansion

2. We do not discuss how to quantify the preservation of privacy of an individual in this article as it is beyond the scope. It will suffice to say that privacy preservation is equivalent to making it algorithmically hard for estimating the original sensor data from the perturbed data.

in the long term to reduce congestion. Another example is in public works maintenance. Citizens can report problems in public facilities, such as broken water pipes, dysfunctional traffic lights. By looking at the counts of complaints, the maintenance personnel can infer to a certain degree the scope and severity of the incident, and use that information to help prioritize and schedule their repair resources.

The patterns may also help users build models about the physical or social phenomena they are observing. These models can be used to make predications. One example is the monitoring of pollutants such as car exhaust. An important aspect of environment protection is to build models to understand the dissemination of pollutants in the air, soil and water. By collecting large amount of data samples about air pollutants such as car exhaust, one can not only monitor the concentration of pollution, but also detect patterns to model how the concentration evolves spatially and temporally as temperature, humidity and wind change. These models can help the environment authority make forecasts and give alerts and directions to the public to better shield themselves from pollutions in their daily activities.

The challenge in identifying patterns from large amount of data is usually application specific and it involves certain data mining algorithms. Depending on the amount of incoming data, and the delay sensitivity of applications, there are two possible approaches for data mining. One is a traditional approach where data is stored in a database first, then one can apply various mining algorithms against the database to detect patterns. However, if the amount of continuous data input is too much for storage, or the application requires fast detection of patterns, stream data mining algorithm may be required. They take continuous input data streams and identify patterns, without the need to first store the data. Data mining algorithms are very domain-specific. They are usually developed together with domain experts. Exactly how the algorithms work is closely related to the application and out of the scope of this paper, so we will not further elaborate.

8 ARCHITECTURE

In this section, we illustrate the current state of the architecture of existing MCS applications and point out its drawbacks. Currently, a typical MCS application has two application specific components, one on the device (for sensor data collection and propagation) and the second in the backend (or cloud) for the analysis of the sensor data to drive the MCS application. This architecture is depicted in Figure 2. We refer to this as *application silos* because each application is built ground-up and independent from each other. There is no common component even though each application faces a number of common challenges in data collection, resource allocation and energy conservation.

Such an architecture hinders the development and deployment of MCS applications in several ways. First, it is hard to *program* an application. To write a new application, the developer has to address challenges in energy, privacy, and data quality in an ad hoc manner, reinventing the wheel all the time. Further, he may need to develop different variants of local analytics if he wants to run the application on heterogeneous devices using different OSes. Second, this approach is *inefficient*. Applications performing sensing and processing activities independently without understanding the consequences on each other will result in low efficiency on an already resource constrained platform. There is a high likelihood of duplicating sensing and processing across multiple applications. For example, traffic sensing, air and noise pollution all require location information, but these applications would each do its own sampling without reusing the same data samples. Further, there is no collaboration or coordination across devices. Devices may not all be needed (e.g. traffic sensing in a given location) especially when the device population is dense. Finally, the current architecture is not *scalable*. Only a small number of applications can be accommodated on each device (e.g. limitations imposed by the device operating system, human capacity to keep track of a large number of applications). Also, the data gathered from societal-scale sensing may overwhelm network and backend server capacities, thus making the current architecture non-scalable.

We envision that a unifying architecture could address the current limitations of how MCS applications are developed and deployed. It will satisfy the common needs for multiple different applications. First, it should allow application developers to specify their data needs in a high level language. It should identify common data needs across applications to avoid duplicate sensing and processing

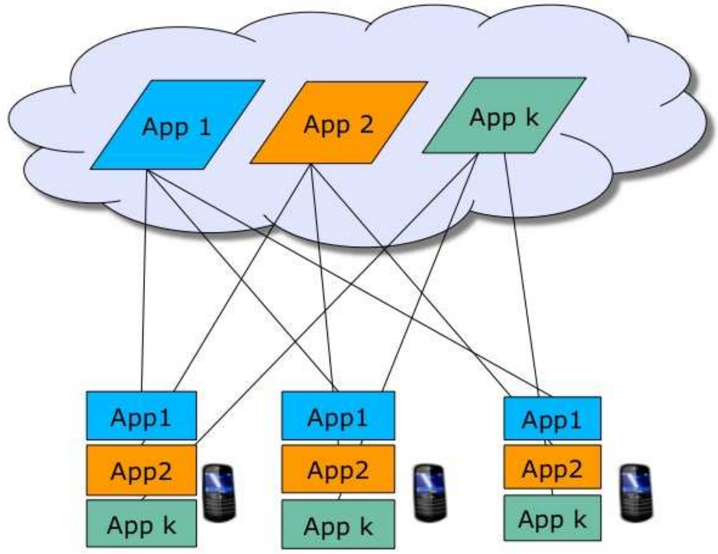

Fig. 2. Existing MCS applications take an "application silo" approach where each application is built from scratch without any common component even though they face many common challenges. Such an architecture hinders the development of new MCS applications and we envision a unifying architecture should address its limitations.

activities on devices. Second, it should automatically identify the set of devices that can provide the desired data, and produce instructions to configure the sensing activities on devices properly. When dynamic changes happen, it should adapt the set of chosen devices and sensing instructions to ensure the desired data quality. Finally, to avoid writing different versions of local analytics on heterogeneous devices, a layer that can shield the differences in physical sensor access APIs and provides the same API upwards should be added on top of the OS. This makes it possible to reuse the same local analytics across different device platforms, assuming these platform all support a common programming language such as Java. We plan to investigate the design of such a unifying architecture in the future.

ACKNOWLEDGEMENTS

We would like to thank our team members, Joel Branch, Han Chen, Sastry Duri, Minkyong Kim, Johnathan Reason, and Zhe Zhang for the several stimulating discussions that provided the basis for this paper. Further, we would also like to thank Maria Ebling, Thomas Erickson, and Wendy Kellog for their valuable inputs to this paper.

REFERENCES

[1] J. Burke *et al.*, "Participatory sensing," Workshop on World-Sensor-Web, co-located with ACM SenSys, 2006. [Online]. Available: http://www.sensorplanet.org/wsw2006/

[2] N. Lane *et al.*, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[3] P. Dutta *et al.*, "Demo abstract: Common sense: Participatory urban sensing using a network of handheld air quality monitors," in *Proc. of ACM SenSys*, 2009, pp. 349–350.

[4] B. Hull *et al.*, "Cartel: a distributed mobile sensor computing system," in *Proc. of SenSys*, 2006, pp. 125–138.

[5] P. Mohan, V. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *Proc. of ACM SenSys*, 2008, pp. 323–336.

[6] S. Mathur *et al.*, "Parknet: Drive-by sensing of road-side parking statistics," in *Proc. of ACM MobiSys*, 2010, pp. 123–136.

[7] S. B. Eisenman *et al.*, "The bikenet mobile sensing system for cyclist experience mapping," in *Proc. of SenSys*, November 2007.

[8] S. Reddy *et al.*, "Image browsing, processing, and clustering for participatory sensing: Lessons from a dietsense prototype," in *Proc of EmNets*, 2007, pp. 13–17.

[9] R. K. Ganti, N. Pham, Y.-E. Tsai, and T. F. Abdelzaher, "Poolview: Stream privacy for grassroots participatory sensing," in *Proc. of SenSys '08*, 2008, pp. 281–294.

[10] N. Pham, R. Ganti, Y. Sarwar, S. Nath, and T. Abdelzaher, "Privacy-preserving reconstruction of multidimensional data maps in vehicular participatory sensing," in *LNCS Proc. of EWSN*, 2010, pp. 114–130.