System Support for Privacy-preserving and Energy-efficient Data Gathering in the Global Smartphone Network
Opportunities and Challenges

Jochen Streicher, Orwa Nassour and Olaf Spinczyk
Technische Universität Dortmund, Dortmund, Germany

Keywords: Privacy, Energy Management, Execution Platform, Collective Apps, Internet Queries.

Abstract: Smartphones are becoming the predominant mobile computing device of the decade. By end of 2012 more than 150 million devices have already been sold and the yearly market growth is about 40%. The ubiquity of the smartphone creates tremendous opportunities for collecting and mining data from end users. Smartphones are becoming a global sensor network that could answer questions about user (customer) behavior and their interests, device status, security threats, and a vast amount of derived information such as CO_2 footprints, traffic conditions, etc. However, so far only very few market-leading companies, such as Apple and Google, are able to exploit this data source. Even though technologically possible, end user concerns such as privacy protection, energy consumption, and the general lack of incentives, make it difficult for smaller companies and private app developers to make use of the smartphone network. This paper will present the vision of an open system support platform for running flexible "Internet Queries" and "Collective Apps" in the global smartphone network. We analyze the problems of the current state of the art, derive platform requirements, and sketch the envisioned platform's architecture. The discussion will culminate in a list of important research directions to be followed.

1 VISION

The proliferation of smartphones and similar devices in home and office environments is a big step towards Mark Weiser's vision of Ubiquitous Computing (Weiser, 1991). They provide sensors, actuators, and decent computational power in everyday situations and share the Internet as a global communication infrastructure. It would be a waste of resources if we used these devices only for individual end users instead of exploiting the world's biggest "sensor network" to answer questions of global scale, which we have never been able to answer before. Figure 1 illustrates the gain of information by mining sensor data and locally derived data from the vast number of smartphones in the Internet. As applications of this data are so manifold, the figure contains only a few examples and does intentionally not strive for completeness.

Our discussion of the state of the art in Section 2 will show that a (small) number of market leading companies are already using this tremendous data source. Several research projects are also interested, but have only very limited access. Smaller companies

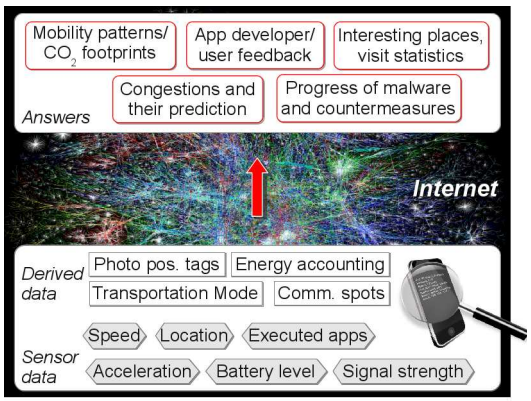

Figure 1: Examples for sensor data taken on smartphones, locally derived data, and information that would be available if the data was collected from the whole smartphone network (or large fractions).

and private app developers have virtually no chance to access the smartphone network. It is interesting to note here that the problem is–at first glance–*not* technical. Apps downloaded from a market can technically access a smartphone's sensors and transmit the data via the Internet to a central server. It is more a

Streicher J., Nassour O. and Spinczyk O..
System Support for Privacy-preserving and Energy-efficient Data Gathering in the Global Smartphone Network - Opportunities and Challenges.
DOI: 10.5220/0004339200800085
In *Proceedings of the 3rd International Conference on Pervasive Embedded Computing and Communication Systems* (PECCS-2013), pages 80-85
ISBN: 978-989-8565-43-3

matter of *trust*, *transparency*, *simplicity*, and *incentives*. We will revisit these points throughout this paper.

The key enabler for our vision is a powerful system software that solves the aforementioned problems. Section 3 will present a sketch of its architecture, while Section 4 will discuss related work and essential future directions of research that is needed for the solution and its optimization. If we assume that the necessary system support was available, two new classes of smartphone applications could be developed, which potentially have huge market impact:

Internet Queries. Companies, developers, and communities could formulate and run queries as if the smartphone network was their own huge sensor network. Of course, end-user privacy protection does not allow to access personal information, such as individual location traces, but in most cases that is not relevant for large-scale data mining anyway. As an example consider a query that determines the global market share of smartphone hardware vendors. This information is relevant for various stakeholders who would clearly accept to pay for it and thus compensate the resources consumed by the query on the data providers' phones.

Collective Apps. A new class of apps might appear in the markets that do not only process data locally, but analyze the data from all users on central servers. As an incentive to provide their data, users will benefit from the results of the analysis. For example, a hotel finding app could collect hotel visit statistics from its users. Based on this data, hotels visited more than once by the same user could be marked, because this might indicate customer satisfaction.

The two examples mentioned above show what *incentives* we have envisioned for data providers and data consumers. They also show that privacy protection is crucial for the success of these application classes. We assume that a data provider will *trust* his hardware and the system software, but not the various data consumers. However, as long as an Internet Query or Collective App does not consume too many resources–especially battery power–and as long as it does not *transmit* critical data, end users are not harmed by executing it on their phone. However, *transparency* with respect to energy consumption and to the criticality of transmitted data is an important platform requirement. For the sake of *simplicity*, data providers shall be able to configure their privacy requirements, i.e. the amount of data they are willing to provide, as simple rules that can be used by the platform in order to accept or reject queries automatically on behalf of the user. This reliefs the data provider from the burden to understand and accept various different "terms and conditions" with individual privacy regulations.

2 STATE OF THE ART

Today most apps provide only local services (e.g. games) or solely use the Internet to provide users with a convenient interface to access centrally stored data. Collective Apps and Internet Queries use real-time or historical data from a community of mobile users in order to offer improved services or for data mining purposes. In this section we present two examples that can be regarded as prototypes of Collective Apps and Internet Query applications. Based on these examples we revisit the concerns about privacy, transparency, simplicity, and the lack of incentives that we have mentioned in the previous section from the data provider's and consumer's points of view.

Google Maps[1]**.** is a web-based map service application and technology, which gives the mobile user the ability to look up his current position on a map of the environment and to use navigation services. The location information is determined using one of the positioning technologies installed on the device such as GPS, cellular identification, or even its IP address. By transmitting their location data, users can get the possible routes to their desired destination. Google Maps collects user's data in order estimate how much time the user needs to reach his destination. Data from other smartphones is used to evaluate the current and future traffic situation. Therefore, Google Maps can be considered as an early Collective App.

Cambridge Device Analyzer[2]**.** is an example for the first prototypes for Internet Queries, whose purpose is research. It collects usage statistics from geographically distributed devices for statistical purposes. Data from all participants are aggregated at a central server. This data will then be filtered and examined in order to extract useful information from it. Location information is collected by sending the ID of the GSM cell to which the phone is connected.

2.1 Trust

Currently, smartphone users either have to accept that an app will have access to certain sensors and system

[1] http://en.wikipedia.org/wiki/Google_Maps
[2] http://deviceanalyzer.cl.cam.ac.uk/

state or they will not be allowed to use it. Therefore, many people just accept, even though a bad gut feeling remains, because they do not really trust the app developers. Users especially have no influence on the data that will be transmitted to servers in the Internet. Other users just do not accept. This gives big (more "trustworthy") companies an advantage over small companies or research projects: Based on the latest statistical expectations by market analysts, Google and Apple, combined, will capture 98% of the worldwide mobile market by the end of 2012. Consequently, this is no problem for Google Maps, but an important issue for, e.g., the Cambridge Device Analyzer.

The Collective Apps that we are aware of typically assume that a trusted centralized server or coordinator exists, which collects data from all devices in one database and anonymizes it before use. However, central aggregation and anonymization of data is problematic, because it increases the number of parties a user has to trust: For example, the server might be located in an untrusted cloud or the communication link could be not properly secured.

Many proposals have been suggested to avoid this problem, such as introducing a trusted proxy as a middle layer between the data sources and the data consumer, which anonymize all data transferred. Yet, eventually all this boils down to the simple fact that, as soon as the data has left the smartphone, the user no longer has control over his data and the analyses performed on it.

2.2 Transparency

Another cause of privacy concerns, which is related to trust, is the lack of adequate control over the disclosure of real-time personal information. Most of the existing location-sharing apps do not detail their policies for the collection and use of personal information. Likewise, Google's privacy policy ensures that user information is shared across Google's network of sites, which means that the participants' data is being collected and processed. It is not yet clear which data is actually being transmitted, since the whole process is obscured. In the case of Apple, two developers have recently discovered that the iPhone has been regularly recording the device's location since the introduction of iOS 4. There is not enough transparency of data processing in today's smartphone applications.

A second important transparency issue is the resource consumption of the device while collecting data, especially with respect to battery power. Most Collective Apps in use today do not take any consideration for conserving the data providers' battery power and do not provide an estimate of the overall resource consumption the user has to expect. For example, frequent location updates from and to the mobile device consume a significant amount of its battery power. Users will not accept Collective Apps if they have to live with unexpected battery drain.

2.3 Simplicity

Most Internet Queries and Collective Apps today ask the participant to accept the terms of data processing and anonymization. However, these documents are usually long and complicated. Reading the individual statements is time consuming, tedious, and difficult for non-experts in law and privacy-preserving data mining. Google's new (simplified) privacy policy has about 2300 words, not including product-specific regulations. The description of the Device Analyzer has "only" about 800 words. It should be clear that dealing with each Collective App separately does not scale.

2.4 Incentives

Some location-based apps, like Google Maps, receive position information from its participants while providing them with the desired navigation service in return. Other apps, e.g. the Cambridge Device analyzer, collect data solely for research purposes, in which the data provider might not be interested. Thus, it provides no direct incentive for the data providers to share their information. Therefore, only enthusiasts participate in this effort.

3 PLATFORM PROPOSAL

Having identified the roadblocks for Collective Apps and Internet Queries in the current state of the art, we will now come up with a set of design principles we deem suited to address those problems. With these principles in mind, we will then proceed to sketch how the respective queries may look like and coarsely describe the envisioned platform's architecture.

3.1 Design Principles

Simple Absolute Control. Transparency is not only about knowing what applications are doing with a provider's device and data, but also incorporates control. Obviously, this comprises the possibility of opting out of data collection temporarily or permanently at any time. However, providers should also be able to declare in a more fine-grained manner *how*

much they are willing to give. Regarding energy consumption, this means control over the additional battery drain that is caused by data collection. With respect to privacy, a provider should be able to declare the extent of potentially privacy-critical data he is willing to disclose. Defining a list of non-disclosable attributes or even more sophisticated rules should of course be possible. But regarding the need for *simplicity*, we need a means to *quantify* the potential privacy threat that stems from collected data.

Device-based Privacy. Because we explicitly want to avoid the need for signing a contract with every possible data consumer, we have to assume that data transmitted from the device is accessible for everyone, forever. Although this sounds like a horrible scenario, we believe that it is actually not too far away from the current state. Designing our on-device platform with this assumption in mind, we are completely independent of any external infrastructure with respect to privacy. Thus, the data provider only needs to trust his own device, which *simplifies* the *trust* requirement. This, of course, makes classical anonymization procedures that are based on data from many individuals more difficult if not impossible at all. However, many envisioned queries are still possible, since non-critical data like the battery charge level can also provide important insights if collected on such a large scale. Besides that, research on privacy-preserving data mining techniques like perturbation (Agrawal and Haritsa, 2005) shows that it is indeed possible to get aggregate values of sensitive data without abandoning individual privacy.

Flexible Privacy. It is widely accepted that there is no general method for privacy-preserving data publishing that also preserves the utility of the data. Generally spoken, privacy *and* utility is only possible if you already know the analysis procedure for the data. Thus, anyone wishing to issue an Internet Query or writing a Collective App should be able to specify the privacy-preserving procedures himself in order to also preserve data *utility*. To satisfy the need for *transparency*, the query has to be analyzable with respect to the quantification of the potential privacy threat.

Modified Stream Semantics. Most work on privacy-preserving data mining focuses on static, existing, data. The continuous stream of data from smartphones poses an additional challenge. Existing work on privacy-preserving stream mining considers *vertically separated data streams* that would have to be joined to gain access to the whole range of attributes. In our case, however, we have *horizontally*

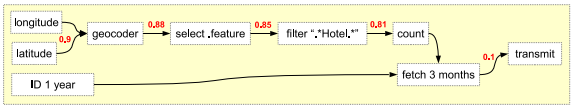

Figure 2: A query example: counting hotel visits.

separated data. In the classic database-oriented view, all data ever produced by a smartphone would form a row in the database, with billions of attributes, smashing any attempt of privacy preservation with the *curse of dimensionality*, impeding anonymization (Aggarwal, 2005) and perturbation(Aggarwal, 2007). Therefore, the data from one device must never be perceived as one complete stream. Instead, only small chunks of the provider's data may be perceived as a continuous data stream.

Lazy Transmission. Various work has shown that it is generally beneficial to transfer mobile data rather infrequently and in larger bursts. It is even better to wait for more energy-efficient connections or situations, where the device is plugged into an AC connector. However, if queries are time-critical, intelligent scheduling is needed for trading off power consumption against latency.

3.2 Query Structure

As stated in the previous section, data consumers should be able to flexibly specify the method used to enhance the privacy of transmitted data. Since we cannot expect providers to inspect every query for potential threats to their privacy, we need to *quantify* the criticality and identifiability of the generated data. Thus a query may only use a defined set of data sources and processing operators with known semantics. Additionally, the data flow from those sources through different operators has to be specified in a way that is automatically analyzable.

While there are many possible representations, we use a data flow graph to explain the query structure. The query for the Collective App example from the introduction, the hotel visit count, is depicted in Figure 2. Some operators are designed to enhance the privacy compared to the raw data. The red figures after each processing step are meant to illustrate how privacy quantification might look like in the future; currently, they are of course just pure fiction.

Our proposed query language consists of basic operators that may be chained together:

Aggregators. These operators compute an aggregate value of a time-dependent discretely sampled input value. There are simple ones, like the computing the average, and advanced ones, like an op-

erator that counts occurrences of discrete values (hotel names, in our example). Temporal aggregators have to be followed by the **fetch** operator, which determines the points in time where aggregate results are actually produced (in the example, every 3 months; the computation might of course be done continuously). Thus, two or more aggregators can be synchronized. The purpose of aggregators is to lower the criticality as well as the volume of the generated data.

Filters. A filter's purpose is to reduce the data set by removing attribute values which fulfill a certain condition.

Perturbators. Perturbators distort attribute values by, e.g., adding random noise. They reduce the criticality by rendering values unusable with respect to a specific device, but still allow to compute aggregate values for many devices.

Interpreters. An interpreter enriches data semantically by using publicly available information, and is always something that could be done also at the consumer's site. However, the resulting data might offer a better trade-off between utility and privacy and should thus be part of the standardized processing stage. In our example, the time-location pair is highly critical and would thus have to be filtered or perturbed, which renders it unusable for hotel rating. However, if we use an interpreter to transform the numeric location into a semantic location (e.g., via a **geocoder** that allows *reverse geocoding*), filter everything besides hotels, and use a counting aggregator to record visit counts for every hotel, we can preserve data utility, while drastically reducing its criticality.

This list is certainly not complete. Besides these operators, we could also think of mathematical (stateless) functions, which might also influence the criticality.

Everything that goes into the **transmit** node in Figure 2, is sent to interested consumers, together with a timestamp. An **ID** for the data stream is needed, when individual patterns have to be monitored. However, in adherence to our modified stream semantics principle, the ID is generated randomly and not a constant value. In fact, the change frequency of an ID (in our example: one year) has to be considered during criticality assessment. We have, of course, to assume that there is a data transport mechanism that does not allow linking two stream chunks with different IDs to the same device. This, however, is our only assumption regarding the external infrastructure's trustworthiness.

3.3 Architecture Sketch

Collective Apps, as well as general consumers issue queries as described in the previous section. As depicted in Figure 3, queries are subject to privacy checks before actually executed. Queries make use of data sources and operators. Although it is not within the scope of this paper, we see energy-efficient sensor management, e.g., for location sensing (Zhuang et al., 2010) as a crucial ingredient here. Some operators or data sources might access publicly available knowledge, like the geocoder from our example. Both the generated data as well as requests for supplementary data are subject to energy-efficient lazy transmission scheduling, like it is done in (Ra et al., 2010).

Keeping the privacy issues inside the device, the off-device infrastructure's main challenge is to ensure fairness between all data consumers. It has to intelligently distribute queries to devices all over the planet, keeping track of them and transporting the generated data back to the consumers. Additionally similar queries should be combined to reduce the mobile data transfer volume and thus the individual energy consumption, allowing every consumer to query a larger set of devices. Due to the large number of devices, consumers and queries, the common infrastructure itself has to be a distributed system, which brings its own challenges.

4 RESEARCH DIRECTIONS

To wrap it up, we present the most crucial research challenges in the pursuit of this vision.

Power models for mobile devices are necessary to assign energy costs to queries (transparency and control). Although much work has been done to model power consumption of hardware components (Zhang et al., 2010), the cross-layer nature of data collection yet poses a challenge.

Aggregation and Perturbation are well known concepts in the data mining community and are amenable for distributed execution. (Agrawal and Haritsa, 2005) However, most research is focused globally on static databases instead of locally on the data generation site.

Although there is work on privacy quantification (Venkatasubramanian, 2008; Agrawal and Aggarwal, 2001) for selected privacy-preserving data mining techniques, a comprehensive approach that allows to quantify chains of operators does not yet exist. Furthermore, the resulting measures often depend on the concrete data. Another approach would be to come up with a model for data- and operator-specific privacy

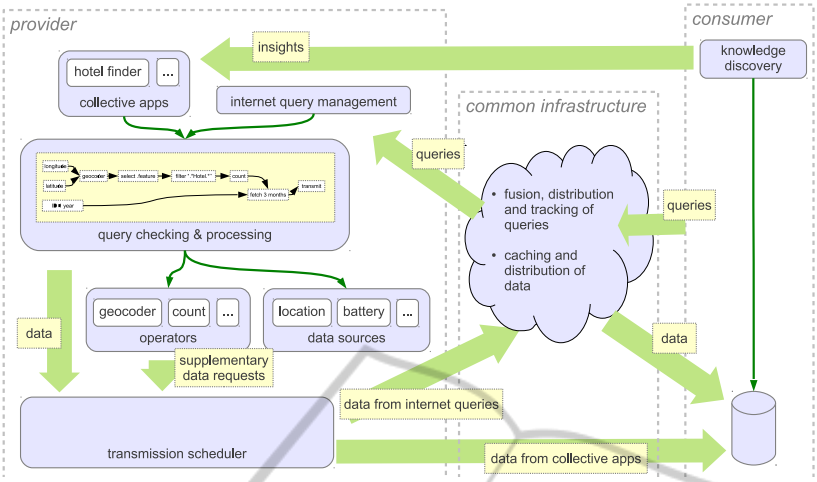

Figure 3: The proposed architecture.

policies. These policies would impose rules on the selection of data sources, operators and their parameters. Then, instead of deriving a privacy level from a query, one would map privacy levels to policies.

5 CONCLUSIONS AND OUTLOOK

In this paper we have presented the vision of a privacy-preserving and energy-efficient platform for smartphone applications from an infrastructure point of view. The platform would exploit theoretical research results from the privacy-preserving data mining community for a very practical goal: An open global smartphone sensor network. The platform would enable two novel application classes, namely Internet Queries and Collective Apps. This might have a huge impact on smartphone use, for both, the data providers and data consumers.

The motivation for this work was practical experience with data collection tasks on smartphones and the concerns of users we have faced. Our goal is to perform a step-wise transformation of our own data gathering infrastructure into the sketched platform. In parallel we aim at refining the described design. In order to write a self-contained short paper, we have focused on the smartphone-side of the architecture here. However, the development of the distributed "common infrastructure", which globally processes, optimizes, and dispatches queries, is not less challenging.

ACKNOWLEDGEMENTS

Part of the work on this paper has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876, project A1.

REFERENCES

Aggarwal, C. (2005). On k-anonymity and the curse of dimensionality. In *31st int. conf. on Very large data bases*, pages 901–909. VLDB Endowment.

Aggarwal, C. (2007). On randomization, public information and the curse of dimensionality. In *IEEE 23rd Int. Conf. on Data Engineeering*, pages 136–145. IEEE.

Agrawal, D. and Aggarwal, C. (2001). On the design and quantification of privacy preserving data mining algorithms. In *20th ACM SIGMOD-SIGACT-SIGART symp. on Principles of database systems*, pages 247–255. ACM.

Agrawal, S. and Haritsa, J. (2005). A framework for high-accuracy privacy-preserving mining. In *21st Int. Conf. on Data Engineering*, pages 193–204. IEEE.

Ra, M., Paek, J., Sharma, A., Govindan, R., Krieger, M., and Neely, M. (2010). Energy-delay tradeoffs in smartphone applications. In *8th int. conf. on Mobile systems, applications, and services*, pages 255–270. ACM.

Venkatasubramanian, S. (2008). Measures of anonymity. In Aggarwal, C. C., Yu, P. S., and Elmagarmid, A. K., editors, *Privacy-Preserving Data Mining*, volume 34 of *Advances in Database Systems*, pages 81–103. Springer US. 10.1007/978-0-387-70992-5_4.

Weiser, M. (1991). The computer for the 21st centrury. *Scientific American*, 265(3):94–104.

Zhang, L., Tiwana, B., Qian, Z., Wang, Z., Dick, R., Mao, Z., and Yang, L. (2010). Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *8th IEEE/ACM/IFIP Int. Conf. on Hardware/software codesign and system synthesis*, pages 105–114. ACM.

Zhuang, Z., Kim, K., and Singh, J. (2010). Improving energy efficiency of location sensing on smartphones. In *8th int. conf. on Mobile systems, applications, and services*, pages 315–330. ACM.