Energy efficient sensing for managing context and privacy on smartphones

Prajit Kumar Das, Anupam Joshi and Tim Finin

University of Maryland, Baltimore County
Baltimore, MD 21250 USA
{prajit1,finin,joshi}@.umbc.edu

Abstract. Mobile devices can better manage user privacy if they continuously model a user's context, but doing so can result in high energy consumption. The energy costs can be reduced by reasoning about what context information is known, what additional information is needed, how accurate it must be and how to efficiently acquire it. We model the sensors and their data properties, accuracy levels and energy costs in an RDF knowledge base supported by an OWL ontology. We describe a method to manage privacy on smartphones in an energy efficient manner by selecting the best choice sensor for maintaining the user's context information. Sensor selection is done by COntext MANager miDDleware (COMANDD), which maintains a context model and answers queries about it. Context requests are served by capability matching, accuracy level matching and selection of lowest energy cost sensor for reporting context data. A context change detection function is used to decide when the context should be updated.

1 Introduction

Contemporary enterprise work environments are witnessing a significant rise in accommodation or adoption of the Bring-Your-Own-Device (BYOD) model. Concerns over corporate data protection has led to security firms actively researching the challenges and opportunities of using such a model. Ionic Security, an Atlanta based startup recently raised $9.4 million to develop its technology for enabling employees to access data on their own devices [11]. The focus of privacy or security firms has remained on securing the data in case of loss of the device [7] or IT implementation and economic challenges [9]. One approach for providing privacy to corporate data is the container based approach. Samsung and Blackberry, through their SAFE (SAmsung For Enterprises) [12] and Balance [10] systems provided separate containers for corporate data.

Container based approaches provide security to corporate data but with larger overheads of security enhanced operating systems and separate application groups. Moreover, container based systems ignore user context and user data flow and modern smartphones are capable of much more than just storing corporate data i.e. they can gather tremendous amounts of information about the user and her context. User data and context leakage thus, becomes an enormous issue with potentially disastrous results. Naturally, we recognized the move to BYOD model as a major challenge, making privacy management on smartphones an important goal. Advances made in context modeling, location tracking and collaborative localization has resulted in emergence of a class of smartphone apps that can access and share embedded sensor and context data. Current security and privacy mechanisms on Android and other mobile operating systems are not well equipped to effectively control dynamic data flow between the framework and the applications.

In our previous work, we showed application and user context-dependent information sharing policies that dynamically control data flow among applications at a fine-grained level[1]. We use semantically rich policies to dynamically monitor and control the data flow between the sensors and the apps on a smartphone [5]. Our approach to privacy management is more fine-grained and robust and carries less overhead than container based systems described above.

The other challenge apart from privacy management comes from a different but equally vital problem of limited battery capacity, on smartphones. One might claim that users' computation expectations from their smartphones are increasing every day at the same time frustrations due to phones running out of battery remains a significant issue. It can also be claimed that, the ability of batteries to power these devices are not increasing as fast as the processing capabilities [1]. Therefore, there is a necessity to preserve the battery as much as possible, thus extending the battery life.

While policy based privacy management has its advantages, it does require the latest context to be available at all times. Therefore requiring context to be updated frequently. This unfortunately creates a serious hindrance due to the limited battery capacity on smartphones. We created an app that would update the context (location using GPS) with a high frequency and found out that the battery can drain out as fast as five hours, given certain conditions are true. Energy efficient privacy management therefore becomes, an important goal. In our ongoing work we have presented a three-fold solution towards achieving this ultimate goal [3].

In this paper we focus on the solution of sensor selection to achieve energy efficiency. We present the design of the COntext MANager miDDleware (COMANDD) for achieving the goal. We extend a mobile ontology i.e. PlatMob [6, 5] from our previous work to include the concept of a sensor on a device and represent a sensor's capability (the type of data it senses), accuracy level (precision of the sensor) and energy cost (the energy cost of the sensor).

The rest of the paper is organized as follows. Section 2 explains the three-fold energy efficiency solution design. Section 3 dives into the sub-solution of sensor selection and describes the ontology that allows us to select a sensor using an example rule. Section 4 describes the methodology used for the creation of the energy model and relative accuracy values of location sensors. Section

[1] Application and App are both used to refer to the same concept, of an Android App in this paper

5 summarizes the related work from the literature. Section 6 summarizes the ongoing work and concludes the paper.

2 Energy efficiency solution

Our three-fold solution for energy efficient privacy management is based upon a study and creation of an energy model for an Android smartphone's component-wise battery consumption pattern. The base assumption in the formation of the model is that there are more than one sensor in a smartphone that can provide the same kind of data. For example location can be obtained using a phone's GPS or Internet connection. The steps in our three-fold approach are as follows.

First, we assume a time period in which a sensing request would be coming in. During each time period, we would only enable the sensors that are required to satisfy the antecedents of relevant policy rules. And, if a set of the policy rules being enforced require the same sensor data (e.g., location) then we take one reading and use it for all of the rules.

Second, if certain information can be gathered from multiple sensors, we use the sensor with the lowest energy footprint or one that is already being used unless the requester asks for a higher precision on the data provided. We trade-off accuracy for energy efficiency depending on the granularity requested by the policy. That means, if we have implemented a policy at the current time which states that provide only country level location data then we ignore the fact that the app is requesting a fine location as provided by the GPS. If the GPS is currently not switched on, we do not consider switching it on even though the app requested GPS location since the policy clearly stated that the data to flow has to be at country level.

Third, if there are multiple conditions in the rule's antecedents and the rule is a conjunction of the conditions then negating a low cost condition will negate the whole rule. We therefore try to reorder the rule in order of the cost of the conditions in the rule's antecedents.

In this paper we have focused on the second part of this three-fold solution. The time period based context sensing step and the reordering of the conditions in the rule's antecedents are beyond the scope of this paper.

3 Energy Aware Sensor Selection

In an Android phone, apps can make requests for various sensor data access. Our "Privacy Management Module" developed in previous work [5, 6] requires context information to be updated frequently, creating energy cost issues. In our current work, we present a design of the COntext MANager miDDleware (COMANDD) that consists of four parts, as shown in Figure 1: a context provider service, query engine, knowledge base with inference engine and context provisioning modules.

In this solution we gather and store the latest context data sensed from various sensors on the phone, in a knowledge-base using the Context Provisioning

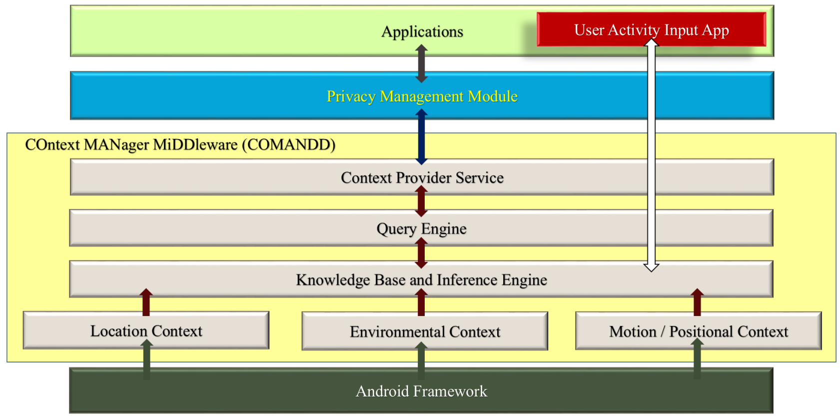

Fig. 1. Context Manager Middleware for maintaining and providing context data at a low energy cost

Modules. This rich context data is classified into four sensing groups at present. Current Android documentation broadly defines three categories of sensors i.e. Motion, Environmental and Position. We have included a fourth category of sensed context data named location provided by the location sensors i.e. GPS, Wi-Fi and Cellular Network.

A sensor can be characterized by properties like what is the sensing capability of the sensor or how accurate a sensor is or how much power or energy does the sensor consume for sensing purposes. Such characteristics may be represented in the form of an ontology. We extend the ontology names PlatMob defined in our previous work [5, 6] to represent the sensor characteristics. Our ontology includes classes for *Sensor* and *SensorGroup*. A sensor always belongs to a single specific sensor group. A sensor has a predefined accuracy level. A sensor has an energy cost. The previous ontology had defined classes for *Device* with a sub class *Mobile*. We define the *Mobile* class as having a listOfSensors property that enlists all the sensors that are available on it.

The Context Provider Service runs in the background and it receives the requests as provided by the Privacy Management Module. The requests are for context data. The input specifies the category of context data required and the accuracy level requested. The Query Engine then makes a query to the knowledge-base for a matching context data with the accuracy level required.

At this point the inference engine can take two separate solution paths. First, if the required accuracy for the context requires the highest possible level, it has to search for the most accurate context data in the knowledge-base or access the framework and provide the required data. Second, if the accuracy is lower than the accuracy level of multiple sensors from the sensor capability group then it again searches for the context data in the knowledge-base and if not found, queries the low cost sensor to provide the requested context data.

Two examples: Let's consider that the user's calendar states that user is in meetings from 9:00am to 5:00pm on a weekday at a corporate location. We also have the GPS coordinates of the user at the corporate location at 9:00am in the morning. We may now switch off the GPS and not update the location information till 5:00pm in the evening.

In another example, we have the user at his home at 6:00am in the morning on a weekend. The user is naturally connected to her home Wi-Fi. The user activity gathered from a learning system we developed in earlier work [18] we know that the user is Watching TV. In this case we do not know when the user would stop the current activity and may be go out. Therefore the context would need to be updated at an unknown time. However, using the accelerometer available on the phone we detect that user is in motion now [18]. Since the accelerometer is a low cost sensor we use it to determine that the user is driving or is stationary. At which point we update the context data using the costlier sensors and store it back in our knowledge-base.

As seen from the two examples above, we can ascertain that context information can be associated with activities. Therefore, we define a context change detection function. The inputs to this function are an activity end time, a low cost sensor's input denoting change in user activity state. The function determines if, the current context data is stale or not. At which point we update the context and store it back to the knowledge-base for future use.

Thus, by modifying the process of continuous context update to a low frequency context update system we achieve energy efficient sensing.

4 Energy Modeling

In order to make a system on a mobile, energy efficient and making selections of sensors to be based upon energy cost of individual sensors, it is first and foremost necessary to explore the energy consumption of individual components on a smartphone. Towards that goal we created an Android app capable of collecting data of current battery levels. The app records any change in battery levels along with the timestamps and stores the information on an external storage on the phone. We created a baseline for the bare bone Android system in airplane mode. This ensured that there was no network communication. We also ensured that no other apps were running on the system. Using Android programming constructs we ensured that the system was running only the operating system on its own and nothing else. We drained the battery out in this condition. Our technique thus provided us with battery consumption pattern which formed the baseline for our energy model.

Creation of the energy model that was carried out included the task of selecting a component that we wanted to model. Selection was done for all possible sensors in each individual sensor group (mentioned in previous section). Once a component was selected we yet again used an Android app to run that component, refreshing the data obtained over small time intervals and storing the battery level changes, timestamps and the sensor data obtained. At present we have created models for location detection sensors. Figure 2 shows the comparative battery drainage time for Wi-Fi, Network, GPS and the baseline system.

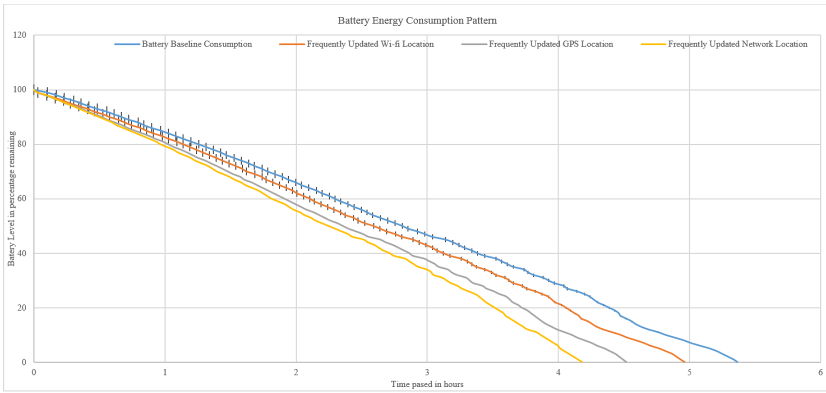

Fig. 2. Change in battery level decrease plotted over time

From the model we were able to calculate the average values of energy consumption of the sensors under test as follows:-

Table 1. Average energy consumption data for location sensors in an Android Google Galaxy Nexus phone

Sensor	Average (mJ/s)
Baseline Power Consumption	401.12
Wi-Fi Power Consumption	414.15
Network Power Consumption	453.67
GPS Power Consumption	458.67

The energy model data was incorporated into our extension of the place ontology explained in the previous section.

The other aspect of location sensors that we studied in our work was of location accuracy with respect to a baseline. We observed that although Wi-Fi consumed relatively low energy. It showed fairly accurate location results, given the assumption that we were able to connect to a Wi-Fi location and even if we are connected the Wi-Fi access point has been updated to the Wi-Fi hotspot databases. Network localization on other hand were highly inaccurate and unstable. The high energy cost associated with the network can be attributed to 3G data transmission energy cost. Figure 3 shows the distance predicted, from the absolute location provided by GPS, for Wi-Fi and Network.

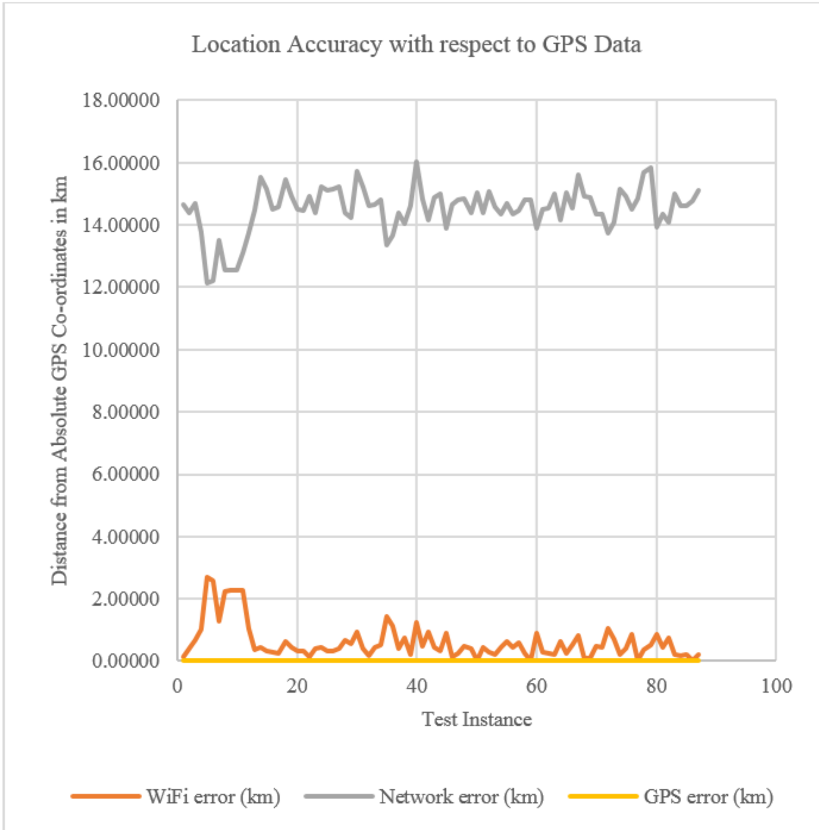

Fig. 3. Location accuracy for 100 test instances

5 Related Work

Privacy management: Our approach for privacy management differs from those in the literature [2, 17, 4] on context based privacy and security. Using semantically rich policies and the user and user application context we undertake a reasoning method to decide the choice of either releasing or obfuscating the sensor/context data being shared with the application [5, 6, 14]. We obtain rich context mapping between a location and its surroundings, the presence of people and devices, inferred activities and the roles people fill in them. All the facts are inferred by a model created by using a machine learning system trained on user data [19]. The context realized as a dynamic knowledge-base of RDF triples is grounded in an ontology expressed in the semantic web language OWL. All policies are encoded in form of SWRL [13] rules and use conjunctions of facts in the context knowledge-base in their conditions. The rules control the sensor data flow from the Android framework to the requesting app. When policies necessitate protection of certain data flow form sensors to an app the data is obfuscated. Our ontology represents the concept of application provenance that is used in policies. The resulting system provides fine-grained, context-dependent control to sensitive user data [5].

Energy efficiency: Current work in the literature on the energy consumption study focuses on exact battery utilization of specific applications and also refers to tail energy issues [16], but has not dealt with creating an energy efficient context inference system that can be used for security. In ongoing work we have carried out studies on Android smartphones to find out energy consumption pattern of individual sensors and their accuracy values. [3]. We group sensors according to the type of data they sense. Android's developer documentation has a basic classification of the sensor categories [8], i.e. Motion, Position and Environment. The context data we are trying to infer or gather or pass includes a fourth category of sensors called Location. We utilize the localization capable components as sensors in this case.

Acquisitional Context Engine (ACE) is a work done in energy efficient context inference. This work includes the notion of *Inference Caching* and *Context Correlation Mining*. In our current work we have adopted a unique approach of defining a function that would specify that the context is outdated and needs to be updated. Given an initial location and activity context, we use a function with inputs of activity length, current time and a low cost motion sensor like accelerometer to determine whether there was any change in context or not. Thus we avoid updating context altogether by using a low energy costing sensor and avoid the need to maintain a caching mechanism.

As far as we know a cross-device context discovery framework has not been designed or implemented by anyone. We believe this new technique would allow us to achieve extended battery life beyond any current mechanisms.

6 Conclusions and Ongoing Work

In this paper we have described the design of an energy efficient privacy management system for smartphones running the Android operating system. We are currently working on the implementation of the middleware to support the system and designing evaluation experiments to test it. We plan to implement two separate systems with and without the context manager middleware and compare and contrast the results to evaluate our system for various privacy manager use cases.

The research work done here is based on the data that can be successfully obtained using Android framework provided data. The best precision of data available through the framework is a one percent change in the battery level. Although the expected battery drain is linear. It may not be exactly linear. We are working on other phones and carrying out research by varying the parameters that affect accuracy and efficiency. We eventually intend to collect enough data to run a learning algorithm and generate a model file. When this model would be put on a phone it will be able to predict the energy consumption patterns of the phones components and adjust the model if necessary.

In course of our research, we observed that frequent update to location context had a significant impact on the battery. GPS position fix although takes substantial time initially [15] but once obtained, provides high accuracy in location information. Wi-Fi, on the other hand, had relatively lower precision but got initial position fix faster at known Wi-Fi access point locations. We do take advantage of this trade-off of location precision versus energy measurements to optimize our energy efficiency algorithm. But we have not considered the time to obtain this fix with respect to energy consumption. We would like to study this aspect in the future. We have created the policy-based security mechanism in the Android framework [5, 6] and have designed and partially implemented the energy efficient privacy framework. Evaluating this system, however remains to be done.

When we talk about evaluating the work, we can look at this from three different aspects. We have to evaluate the policies, in order to prove that the policies are capable of providing the privacy the user needs. We have to evaluate the accuracy of the user context that we gathered, in order to prove that the technique of not actually using the sensors or using inaccurate sensors to gather user context does not reduce context accuracy below a certain a threshold. Finally, we have to evaluate the energy efficiency claim for our system. At present, we are working on the evaluation of this work. We have designed two methods of carrying out the evaluation.

The first is the ideal evaluation scenario where we plan to put our system on a number of phones. We then ask the user to use their phones with a specific selection of privacy policies applied. We record the user's battery usage for efficiency evaluation and user's responses on a questionnaire for evaluating the privacy aspect. We do the same task without our system on the phone for a few user's and compare the results for evaluation purposes.

The second technique we have designed is to use a simulated environment for some phones and using a markov model of real human users simulate activities and location changes. Thereafter we can compare the results for an energy efficiency perspective by logging the battery usage on two phones, one with the system and one without. Such evaluations would still have to be carried out. At the moment we are in the process of creating the experimental setups and we are working on the evaluation process.

7 Acknowledgments

Support for this work was provided by NSF grant 0910838, MURI award FA9550-08-1-0265 from the Air Force Office of Scientific Research. Varish Mulwad provided valuable inputs throughout the design process of the COntext MANager miDDleware (COMANDD).

References

1. Boxall, A.: When will your phone battery last as long as your kindle? (December 2012), http://goo.gl/T2T0R

2. Conti, M., Nguyen, V.T.N., Crispo, B.: Crepe: Context-related policy enforcement for android. In: Burmester, M., Tsudik, G., Magliveras, S., Ilic, I. (eds.) Information Security, Lecture Notes in Computer Science, vol. 6531, pp. 331–345. Springer Berlin Heidelberg (2011)
3. Das, P.K., Ghosh, D., Joshi, A., Finin, T.: Energy efficient semantic context model for managing privacy on smartphones. In: 14th International Workshop on Mobile Computing Systems and Applications. ACM (2012)
4. Enck, W., Gilbert, P., Chun, B.G., P., L., Cox, Jung, J., McDaniel, P., Sheth, A.N.: Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. In: Proceedings of the 9th USENIX conference on Operating systems design and implementation. pp. 1–6 (2010)
5. Ghosh, D.: Context based privacy and security in smartphones. Master's thesis, University of Maryland, Baltimore County (2012)
6. Ghosh, D., Joshi, A., Finin, T., Jagtap, P.: Privacy control in smart phones using semantically rich reasoning and context modeling. In: Security and Privacy Workshops (SPW), 2012 IEEE Symposium on. pp. 82–85 (2012)
7. Goldberg, T.: Securing your enterprise data in a byod world (April 2013), http://goo.gl/na8So
8. Google: Sensor manager class (April 2013), http://goo.gl/1nMJ7
9. Guptill, B.: Byod managed asap (April 2013), http://goo.gl/OoW99
10. Halliwell, D.: Better balance equals better byod living (March 2013), http://goo.gl/OQpfO
11. Harris, D.: Ionic security raises $9.4m to make byod safe (April 2013), http://goo.gl/j835y
12. Honig, Z.: Samsung announces safe with knox, details plans to secure the enterprise galaxy (hands-on) (February 2013), http://goo.gl/97DCc
13. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M., et al.: Swrl: A semantic web rule language combining owl and ruleml. W3C Member submission 21, 79 (2004)
14. Kodeswaran, P., Nandakumar, V., Kapoor, S., Kamaraju, P., Joshi, A., Mukherjea, S.: Securing enterprise data on smartphones using run time information flow control. In: Mobile Data Management (MDM), 2012 IEEE 13th International Conference on. pp. 300–305 (2012)
15. Liu, J., Priyantha, B., Hart, T., Ramos, H.S., Loureiro, A.A.F., Wang, Q.: Energy efficient gps sensing with cloud offloading. In: Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems. pp. 85–98. SenSys '12, ACM, New York, NY, USA (2012)
16. Pathak, A., Hu, Y.C., Zhang, M., Bahl, M.P., Wang, Y.M.: Fine-grained power modeling for smartphones using system call tracing. In: Proceedings of the sixth conference on Computer systems. pp. 153–168. EuroSys '11, ACM, New York, NY, USA (2011)
17. Sadeh, N.M., Chan, T.C., Van, L., Kwon, O.B., Takizawa, K.: A semantic web environment for context-aware m-commerce. In: Proceedings of the 4th ACM conference on Electronic commerce. pp. 268–269. EC '03, ACM, New York, NY, USA (2003)
18. Sane, A.: Predicting the Activities of mobile phone users with Hidden Markov models. Master's thesis, University of Maryland, Baltimore County (2012)
19. Zavala, L., Dharurkar, R., Jagtap, P., Finin, T., Joshi, A.: Mobile, collaborative, context-aware systems. In: Proc. AAAI Workshop on Activity Context Representation: Techniques and Languages, AAAI. AAAI Press (2011)