An Ontology Based Sensor Selection Engine

Prajit Kumar Das
University of Maryland, Baltimore County
prajit1@umbc.edu

Primal Pappachan
University of Maryland, Baltimore County
primal1@umbc.edu

ABSTRACT

In this project we describe the base ontology that would define sensors as a generic concept while at the same time, being applicable to sensors on a modern mobile smartphone. We used an ontology, in this project, to define the hierarchical concept of sensors on a smartphone. We are using the ontology to introduce the generic properties of sensors that allows us to select the best among the ones available to us. The criterion for the selection being the accuracy of the sensor and energy efficiency rating or energy cost of the sensor. The project is part of our research work, where we are creating a context-aware, energy efficient middleware on an Android smartphone. The primary motivation of this work comes from the domain of context-aware privacy management on mobile devices. Fine-grained, policy based, privacy management can be done in a better manner if the user's context is known at every time instant. However, in doing so we would drain out the battery of the mobile. Thus we propose a technique for managing the energy costs too.

Author Keywords

Semantic Web, Ontology, Sensor Selection, Context-aware, Energy Efficient

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.

However, container based systems ignore user context 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 handle dynamic data flow between the framework and the applications.

In our previous work, we have shown 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]. This second approach to privacy management is finer grained and more robust and carries less overhead than container based systems described above.

The other challenge apart from privacy management comes from the separate but equally vital problem of limited battery capacity, on smartphones. Users' expectations from their smartphones are increasing every day and thus the need for them to do more computation is also increasing. However, 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 necessitating context update at a high frequency. 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

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

UMBC Semantic Web Class Spring '13, Baltimore, MD, USA.

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 project we focus on the creation of the ontology for carrying out these tasks.

In this project we focus on the sub-solution of sensor selection to achieve energy efficiency. We show the architecture diagram of the COntext MANager miDDleware (COMANDD) for the Middelware that is our ultimate goal. However, in this project we only focus on the creation of the Platsens ontology. In order to do this, 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) as seen in Figure 1.

In our research we came up with a three-fold solution to this problem. We are focusing on a single part of this three-fold solution in this project. We extend the concepts of user activity and location notion defined in the Platys project to include the idea of a temporal activity expiration and dirty context information. Our approach to reducing energy costs is 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 a knowledge base supported by an 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, which is based on the acitivity expiration time.

The rest of the project report is organized as follows. Section 2 explains the previous work done. Section 3 briefly mentions the methodology used for the creation of the energy model and relative accuracy values of location sensors. Section 4 dives into the sub-solution of sensor selection and describes the ontology that allows us to select a sensor using an example query and we also describe how we are carrying out sensor selection and a generic energy solution. Section 5 concludes the report with a summary of our conclusions for the project and a description of the future work that we are targeting as part of this research.

PREVIOUS 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 [18]. 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 expiration time, current time. Thus we avoid updating context altogether by using a low energy costing sensor and using a activity based caching mechanism.

ENERGY MODELING WORK

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. We did this energy modeling in our ongoing work [3]. 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 pro-

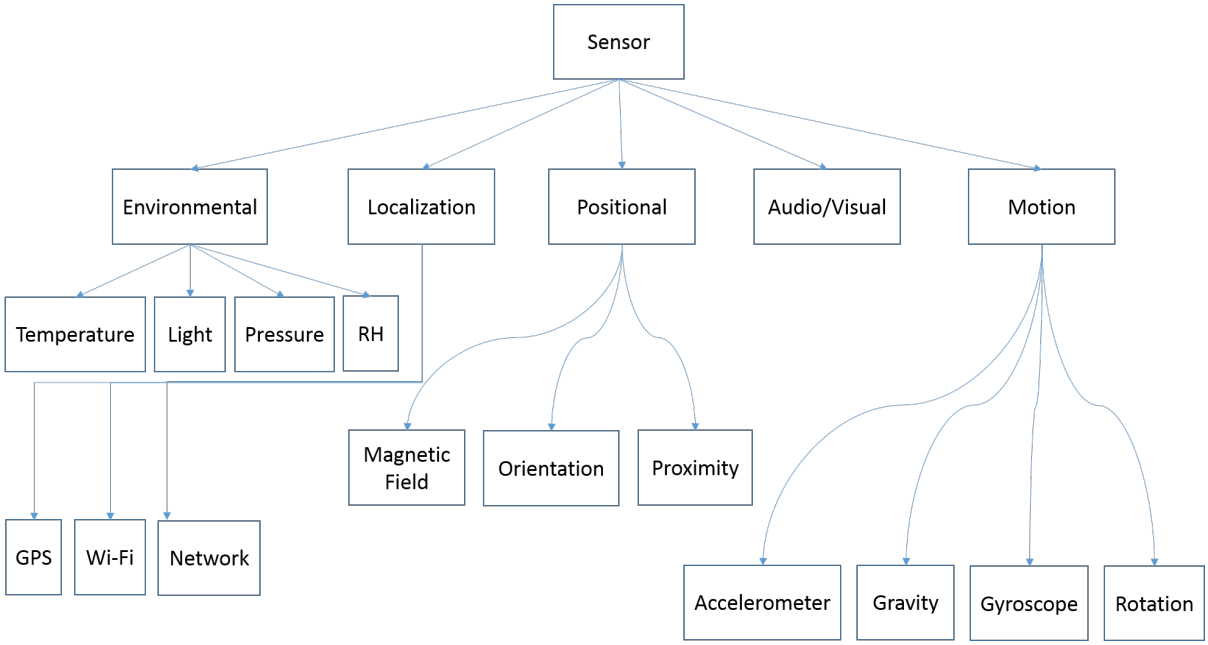

Figure 1. Shows the sensor class hierarchy that we are defining using the ontology

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

vided 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.

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

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.

PLATSENSE ONTOLOGY

In general our solution approach is that on the mobile, apps make 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 **??**: 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 Modules. This rich context data is classified into five sensing groups at present. Current Android documentation broadly defines three categories of sensors i.e. Motion, Environmental and Position. We have included the fourth and fifth categories of sensed context data named localizaion provided by the location sensors i.e. GPS, Wi-Fi and Cellular Network and Audio/Visual sensors.

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

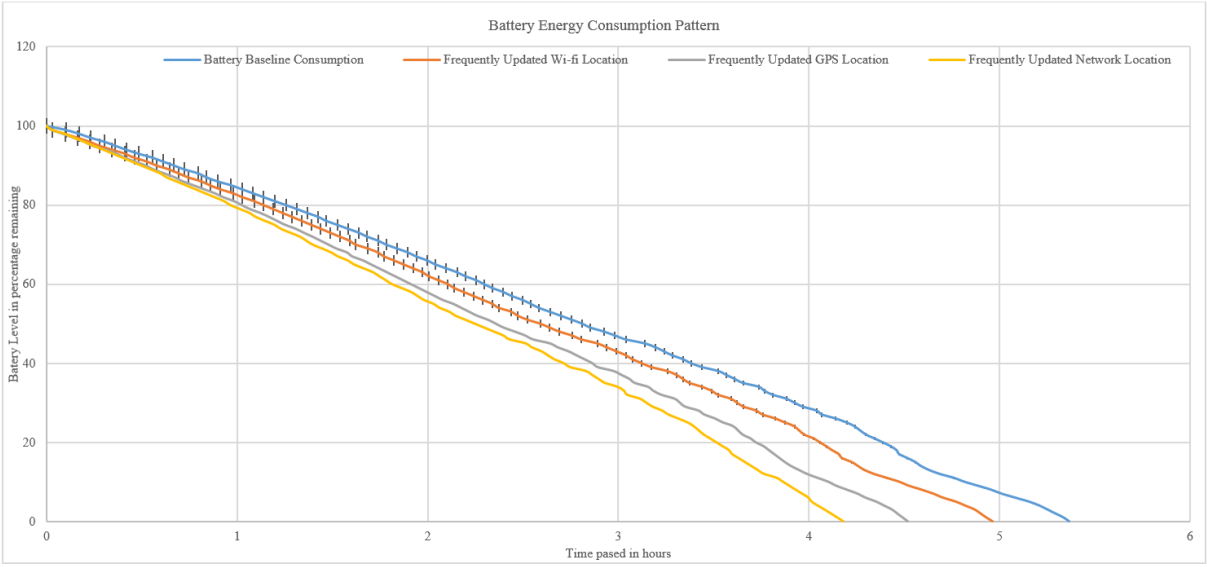

Figure 2. Change in battery level decrease plotted over time

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*. A sensor always belongs to a single specific sensor group denoted by its sensing capability. 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.

In the platsense ontology we have defined the classes of sensors as per their sensing capability. Right now we have a limitation in this research as because there are not enough sensors available as an alternative in each individual group of sensors. Naturally we focused on the Location sensor group, with three alternatives for sensing a user's location this seems to be a logical option. Now let us assume a use case for explaining how this works.

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.

Use case: User at known activity with known expiration time
We are defining the concept of temporal expiration for activities. We say that an activity has a list of associated sensor data. The activity has an expiration time. In this case the user's activity is known and we also know the expiration time for the said activity. Then we are able to respond to a query for a best sensor given the input is a request for location data with low accuracy.

```
SELECT Min(?sensor)
WHERE
?place hasActivity ?activity
?activity hasExpirationTime System:time
?activity hasSensor ?sensor
?sensor hasAccuracy Platsense:low
?sensor Platsense:type Platsense:Localization
?sensor hasEnergyCost ?energyCost
ORDER BY ?energyCost
```

As we can see that we are taking into account the energy cost for a sensor and selecting the minimum value from a list of sensors ordered by the cost. The basic assumption, as always, is that the required sensor accuracy is low when the request comes in. Thus, by modifying the process of continuous context update to a low frequency context update system we achieve energy efficient sensing. As an alternative to using a SPARQL query we defined SWRL rules which do the same selection of sensors. The rules are as follows:-

Rule 1: User has location "UMBC" and activity is "InMeeting" and Accuracy Level Required "Low" $\rightarrow$
$UseWiFilocationupdates$

has_location(?place, ?location),
has_activity(?place, ?activity),
Accuracy(?accuracy) $\rightarrow$
$hasSensor(?activity, ?sensor)$,
$hasAccuracy(?sensor, ?accuracy)$,
$hasEnergyCost(?sensor, ?energy)$

Rule 2: User has location "Home" and activity "Sleeping" and Accuracy Level Required "Low" $\rightarrow$
$Reusethecurrentcontext$

has_location(?place, ?location),
has_activity(?place, ?activity),
integer[≥ 0]$(?time)$,
$hasEndTime(?activity, ?time)$,
$Accuracy(?accuracy) \rightarrow$
$hasContext(?user, ?context)$

Rule 3: User has location "unknown" and activity "IsDriving" and Accuracy Level Required "High" $\rightarrow$
$UseGPSlocationupdates$

has_activity(?place, ?activity),
Accuracy(?accuracy) $\rightarrow$
$hasSensor(?activity, ?sensor)$,
$hasAccuracy(?sensor, ?accuracy)$,
$hasEnergyCost(?sensor, ?energy)$

The query flow can be seen in Figure 5. This figure explains how the context data is searched and returned in case it is present, or else it is updated and stored in the knowledge base and then returned to the requester. This flow occurs when the request is for context information instead of sensor selection request.

CONCLUSION AND FUTURE WORK

In this project we have provided a design of an energy efficient privacy management solution. We are in process of implementing the middleware for which we have provided the design here. Evaluation of the system poses to be a challenge. 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.

REFERENCES

1. A. Boxall. When will your phone battery last as long as your kindle?, December 2012.
2. M. Conti, V. T. N. Nguyen, and B. Crispo. Crepe: Context-related policy enforcement for android. In M. Burmester, G. Tsudik, S. Magliveras, and I. Ilic, editors, *Information Security*, volume 6531 of *Lecture Notes in Computer Science*, pages 331–345. Springer Berlin Heidelberg, 2011.
3. P. K. Das, D. Ghosh, A. Joshi, and T. Finin. Energy efficient semantic context model for managing privacy on smartphones. In *14th International Workshop on Mobile Computing Systems and Applications*. ACM, 2012.
4. W. Enck, P. Gilbert, B.-G. Chun, L. P., Cox, J. Jung, P. McDaniel, and A. N. Sheth. 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*, pages 1–6, 2010.
5. D. Ghosh. Context based privacy and security in smartphones. Master's thesis, University of Maryland, Baltimore County, 2012.
6. D. Ghosh, A. Joshi, T. Finin, and P. Jagtap. Privacy control in smart phones using semantically rich reasoning and context modeling. In *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*, pages 82–85, 2012.
7. T. Goldberg. Securing your enterprise data in a byod world, April 2013.
8. Google. Sensor manager class, April 2013.
9. B. Guptill. Byod managed asap, April 2013.
10. D. Halliwell. Better balance equals better byod living, March 2013.

11. D. Harris. Ionic security raises $9.4m to make byod safe, April 2013.

12. Z. Honig. Samsung announces safe with knox, details plans to secure the enterprise galaxy (hands-on), February 2013.

13. I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, et al. Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21:79, 2004.

14. P. Kodeswaran, V. Nandakumar, S. Kapoor, P. Kamaraju, A. Joshi, and S. Mukherjea. Securing enterprise data on smartphones using run time information flow control. In *Mobile Data Management (MDM), 2012 IEEE 13th International Conference on*, pages 300–305, 2012.

15. J. Liu, B. Priyantha, T. Hart, H. S. Ramos, A. A. F. Loureiro, and Q. Wang. Energy efficient gps sensing with cloud offloading. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 85–98, New York, NY, USA, 2012. ACM.

16. A. Pathak, Y. C. Hu, M. Zhang, M. P. Bahl, and Y.-M. Wang. Fine-grained power modeling for smartphones using system call tracing. In *Proceedings of the sixth conference on Computer systems*, EuroSys '11, pages 153–168, New York, NY, USA, 2011. ACM.

17. N. M. Sadeh, T.-C. Chan, L. Van, O. B. Kwon, and K. Takizawa. A semantic web environment for context-aware m-commerce. In *Proceedings of the 4th ACM conference on Electronic commerce*, EC '03, pages 268–269, New York, NY, USA, 2003. ACM.

18. L. Zavala, R. Dharurkar, P. Jagtap, T. Finin, and A. Joshi. Mobile, collaborative, context-aware systems. In *Proc. AAAI Workshop on Activity Context Representation: Techniques and Languages, AAAI. AAAI Press*, 2011.

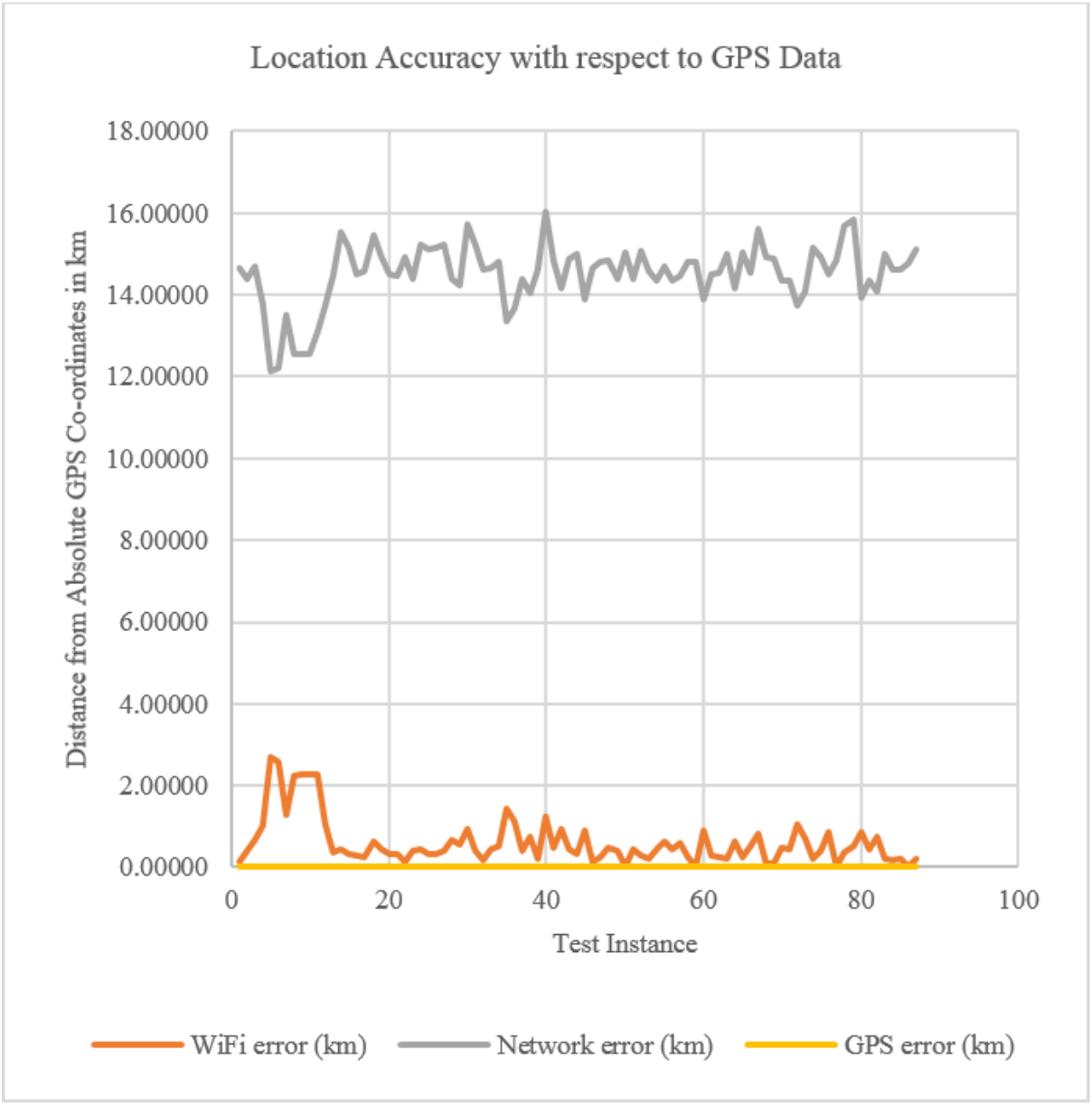

Figure 3. Location accuracy for 100 test instances

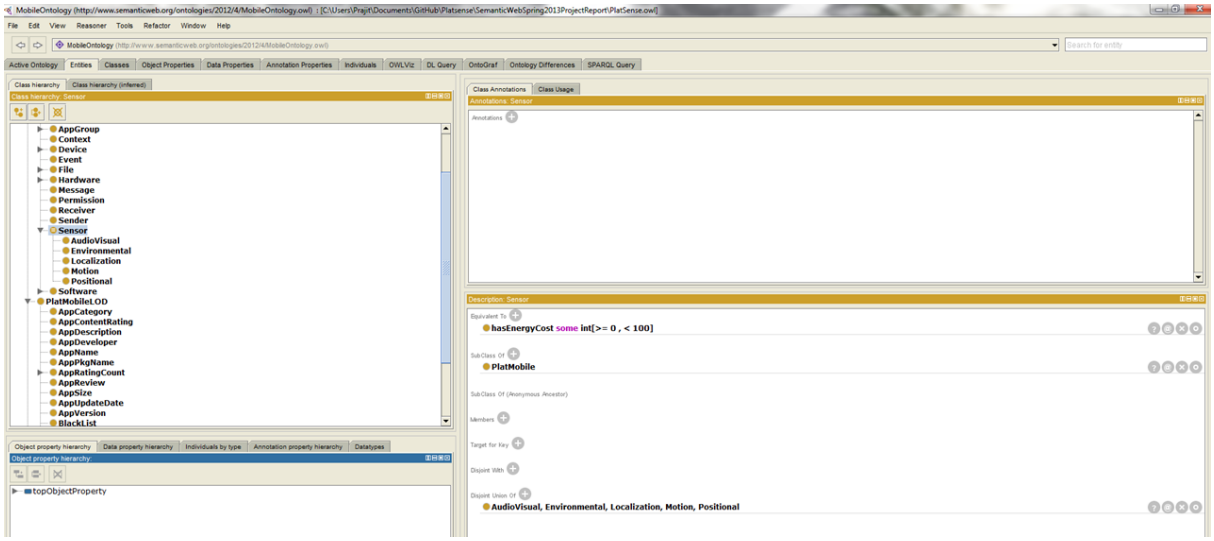

Figure 4. Location accuracy for 100 test instances

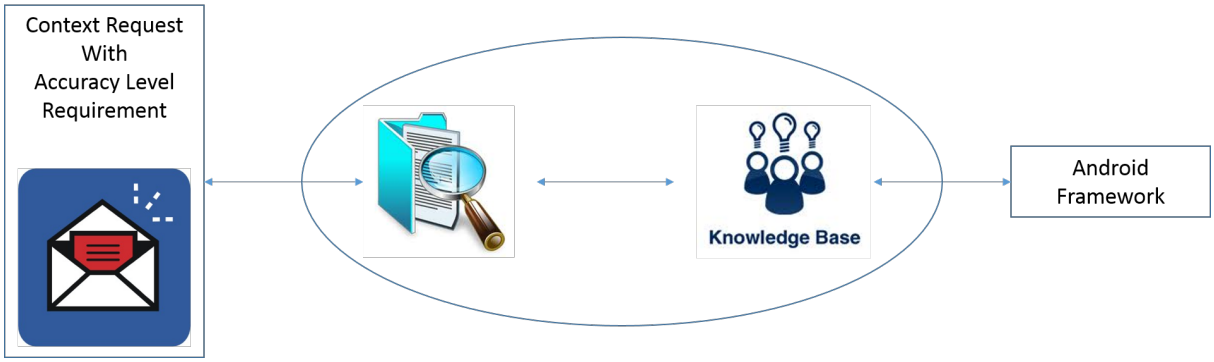

Figure 5. Query flow from Privacy management to Context management