The SITA principle for Location Privacy – Conceptual Model and Architecture

Mads Schaarup Andersen
Department of Computer Science
Aarhus University, Denmark
Email: masa@cs.au.dk

Mikkel Baun Kjærgaard
Department of Computer Science
Aarhus University, Denmark
Email: mikkelbk@cs.au.dk

Kaj Grønbæk
Department of Computer Science
Aarhus University, Denmark
Email: kgronbak@cs.au.dk

***Abstract*—Most existing location privacy solutions suffer from being binary privacy or constrained to either identity, temporal, or spatial data. Furthermore, solutions which try to embrace location privacy more generally suffer from being overly complex. This limits the expressiveness and general applicability of such solutions, the consequence being that different location privacy implementations are restricted in the sense of which attacks they protect against. Furthermore, this makes it difficult to compare different solutions making it hard for developers to choose and add sufficient location privacy. In this paper we present the SITA conceptual model to solve the aforementioned problems. This novel location privacy model advocates simplicity as principle for location privacy, which is divided into the four fundamental dimensions of spatial, identity, temporal, and activity data. Each of these dimensions are divided into five levels of privacy to be easily comprehensible and complete at the same time. To demonstrate the applicability and feasibility of the conceptual model, we propose a general architecture and provide the AndSITA Android implementation. Furthermore, we demonstrate the applicability by developing an example location based service. We observe through these steps how the properties of the SITA conceptual model provides a more comprehensible and expressive way of providing location privacy, that will help bridge the gap between privacy on a conceptual level and practical use. The contribution of this paper is twofold: (1) we provide a complete, yet simple language to discuss and compare existing solutions and (2) we provide a simple architecture which aids developers in adding SITA privacy.**

Keywords*—*Location Privacy, Framework, Ubiquitous Computing.

I. Introduction

During the last 10 years the broad adaptation of location enabled devices, has enabled the development of location based services (LBSs). The services range from simple querying services, e.g., find nearest hotel, to complex ones which e.g. calculate CO_2 footprint based on location trajectories [1]. In addition, social network services (SNSs) which were originally not location based, are becoming location enabled. The most successful example being Facebook [2], where users can now check-in to places, and where status updates and picture are location tagged by default. Other examples of SNSs which use location include Google Latitude [3] and FourSquare [4].

Ever since LBSs were first developed, location privacy has been raised as an issue. With the continuing increase in smartphones being sold, privacy becomes even more important, as the devices have hit the mass market. This makes the technology and LBSs accessible to everyone, and not only expert users. This leads one to think that privacy should be a main concern for users of such devices. That users should have this concern is supported by Enck et al. who show how much data is actually leaked by popular smartphone apps [5]. However, it has turned out that users in general do not care about privacy [6]. In the current state of commercial applications, it seems that users have so far accepted that in commercial LBSs, privacy is usually handled using access control lists (ACLs) just as when securing access to files. Privacy then seems to be a question of security as the user has to trust that the service provider, who receives the full location data, will handle data according to the ACLs. Examples of commercial LBSs using ACLs include Facebook [2], Google+ [7], and Endomondo [8]. The absence of methods for better protection of location privacy in commercial LBSs is not due to a lack of privacy methods, as the research community have produced a continuum of methods providing better location privacy protection [9]. Such methods are, however, usually quite specific to a certain type of LBS, e.g., Point-of-Interest, Navigation, or Location-based Social Networking. Hence, there is a lack of general applicability of existing methods.

A recent survey of location privacy methods [9], among other divided existing privacy protection methods into the categories of anonymity, classical security, spatial, temporal, and protocol. However, the authors identified that several methods fitted into more than one category. This is interesting as it implies that it might make sense to think of location privacy as not being either, e.g., temporal or spatial, protocol or anonymity, but rather combination of several categories. This characteristic is also partly recognized by Zafeiropoulou et al. as they state that context is very important when sharing location [10]. We call the problem of sticking to one category *the one dimensional restriction*. Hence, we seem to need some generic way of talking about location privacy across categories.

Another problem with existing location privacy methods is that they usually offer privacy controls which adhere to a very specific understanding of privacy [9]. Firstly, by limiting the user to one specific interpretation of, e.g., spatial obfuscation which might be very restrictive. Secondly, by not providing cross category controls, e.g., a spatial privacy method that add noise to the reported positions but which does not consider aspects of other privacy method categories such as anonymity or temporal. Because of such limitations users will be biased towards choosing to disable privacy. We call this problem *the on/off restriction*. It seems that a more dynamic approach might be useful both for developers and users. Here

we argue that methods should take explicit care of all the fundamental data dimensions of spatial, temporal, identity and activity of location-based services and context-aware systems in general [11].

As a solution to these concerns, we present SITA, a conceptual model for providing location privacy. The overall goal of SITA, is to provide location privacy which suffers from neither the *on-off* nor the *one dimensional* restriction. Furthermore, the conceptual model should be simple and, hence, it is based on Maeda's *ten laws of simplicity* [12]. This is due to the observation that the more comprehensive location privacy architectures such as Confab [13] do not seem to be adopted in commercial solutions or broader in research. That simplicity should be a main design focus in IT systems is also recognized by Obendorf [14].

The SITA model divides location privacy into four independent properties, or dimensions: *Spatial*, *Identity*, *Temporal*, and *Activity*. These dimensions are orthogonal in the sense that they are independent, so changing one category has no side effects on others. Each of the dimensions are divided into five discrete levels each of these having their own interpretation of privacy in form of a privacy method referred to as a *strategy*. Such strategies can be combined across the dimensions thereby supporting cross dimensional location privacy methods. Additionally, we argue that SITA is complete in the sense that the concept can cover all existing location privacy methods, and that it can be used to classify privacy concepts of existing LBSs. Furthermore, we argue that the principles of SITA are easily comprehensible for users as well as developers. To show the applicability and feasibility of the model we present a general architecture and AndSITA, an Android implementation of this. Furthermore, utilizing this implementation we build an example LBS. We observe through these steps how the properties of the SITA conceptual model, provides a more comprehensible and dynamic way of thinking location privacy, that will help bridge the gap between privacy on a conceptual level and users practices.

II. Related Work

Within the area of location privacy, several attempts have been made to provide high level privacy guidelines for developing LBSs. This has been done by Bellotti and Sellen [15] and Langheinrich [16]. While these guidelines and important, they to not provide architectures or frameworks for enforcing the guidelines.

There has also been work done in providing programming support for developers of context-aware computing, of which LBSs are a subset. This includes middleware and frameworks such as MobiSoC [17], GAIA [18], iRoom [19], one.world [20], and Topiary [21]. These do, however, not include location privacy as a primary concern. Some work has also been done on developing architectures to address privacy explicitly such as Anonysense [22] and SmokeScreen [23] which provide architectures for supporting privacy. However, they are limited by the fact that location privacy can only be achieved through anonymization of users. The most interesting middleware proposed is Confab by Hong et al. [13], which provides users with a comprehensive software framework for building LBSs.

In comparison to existing work, SITA aims to provide a conceptual model which covers all aspects of location privacy rather than only anonymity. This model should also facilitate discussion and comparison of existing methods. In relation to Confab, the SITA software architecture should be more fundamental leaving more choices to the developer, which forcing him to consider all four fundamental dimensions of location privacy.

III. SITA – a Conceptual Model for Location Privacy

To address the problems of the *one dimensional* and *on/off* restrictions and to provide a simplistic model which is complete and comprehensible solution, we present SITA. The spatial dimension is the position of the user in space, e.g., a longitude, latitude coordinate in geographic space. The identity dimension handles the identity of the user who is using the LBS, e.g., as full personal information with name, address, social security number etc. The temporal dimension contains information on when locations are reported to an LBS, e.g., a timestamp. Finally, the activity dimension contains information about what the user is doing when reporting the location. This dimension is quite broadly interpreted as it can contain task performed, movement speed, transportation mode, situation, properties of the surrounding environment, etc.

That the conceptual model should be simplistic, is based on the observation that the more complex models such as Confab [13] do not seem to reach the commercial market or other research projects. From Maeda's [12] ten laws of simplicity, we focus on law 1: *reduce*, law 4: *learn*, and law 8: *trust*. We have reduced the model to only provide information related to location privacy; we have focused on making the model easy to learn for developers; and we have made the model simple as to make it easier for developers to trust that the model will provide what they need and not restrict them more than necessary.

The four dimensions of SITA are inspired by Andersen and Kjærgaard's five category classification of location privacy methods [9] and Dey [11] who states that the fundamental data types of context aware computing (which includes most LBS) are: *space*, *identity*, *time*, and *activity*. The choice to use the dimensions proposed by Dey [11] in relation to location privacy are motivated by the observation that there is a large overlap with the five categories identified by Andersen and Kjærgaard [9]. The differences are that instead of using the name anonymity we use *Identity*, as this reflects the type of data and can also interpreted in a broader context by, among other things, including methods which aggregate users based on an age-range or job. Classical security is omitted as this is thought of as being a security issue rather than privacy. We argue this, because the methods in this category can be applied to all the other categories at some point. For example a server side ACL which determines who can access the extracted data, or store it so that only two users sharing a secret key can decrypt the data. As the protocol category does not denote a type of data it is not needed as a dimension.

While existing work in location privacy focus on the spatial, temporal, and identity aspects of location privacy, we also add the dimension of activity. This is necessary as

newer LBSs often also share location-stamped activity data. An example is usage-based insurance where insurance premiums or deductibles are calculated on the behavior of the user. In this case, the behavior can be unusual acceleration patterns and over-speeding at the time of an accident [24]. Another example is a service to track the tasks of workers. Here the task might be something that the workers have to register manually using an interface. Common for the two examples is that the activity, in this case behavior and tasks, is a central part of the LBS, and hence is very important in relation to privacy. Therefore, the last dimension is *Activity*.

To cope with the *on/off restriction* we divide each of the four dimensions into five discrete levels to reflect different degrees of location privacy. The choice of having five levels is based on how existing methods can be grouped. As most of the existing privacy methods target the spatial dimension this division have mainly been selected based on the grouping of this dimension. The levels are named according to the terminology used in existing privacy methods which are represented at the different levels. We use the following names and numerals from 0 to 4 with 0 being share no information, and 4 being share all information: *0: No Information*, *1:Aggregation*, *2: Obfuscation*, *3: Regulation*, and *4: Full Information*. The same five levels are also defined for the other dimensions, to keep the conceptual model to be easy comprehensible adhering to Maeda's 4th law of simplicity of *learn* [12]. In this way, SITA can also be used to discuss and compare location privacy methods of different LBSs by placing the LBS in levels in the four dimensions. An overview of how these are applied to the four dimensions can be found in Table 1–4. This includes recommendations for what should constitute behavioral feature data on the different levels.

The four dimensions are considered to be orthogonal, i.e. changing the level of, e.g., the spatial dimension has no influence on the other dimensions.

On the conceptual level of SITA, data is received from either sensors, user interfaces, or trusted 3rd party sources. This data is then processed in a number of providers. The data from these providers is considered to be full data without any privacy. This is then processed in strategies to provide the output data according to the privacy of the levels in the current SITA privacy configuration. The output of this is behavioral feature data which is sent to the service utilizing SITA. This is illustrated in Figure 1.

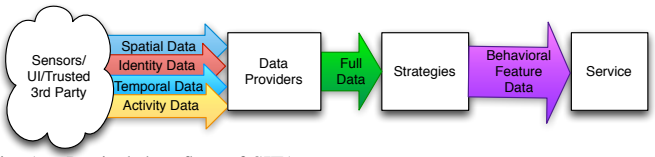

Fig. 1. Logical data flow of SITA.

A fundamental motivation behind the SITA conceptual model is the idea of *privacy-by-substitution* introduced by Andersen et al. [27]. The idea here is that different levels of privacy can be achieved by substituting detailed data with less detailed data. This might even be of another fundamental data type. For example, activity data extracted from spatial data. The detailed data might be fine-grained location traces and the less detailed data might be overall acceleration patterns, kilometers driven or area driven. These different levels of detail are called *extracted behavioral features*. Acceleration pattern and kilometers driven are examples of extracted activity behavioral data, also called activity features, and area driven is a spatial feature. In SITA, this is applied in both data providers and strategies. In data providers this is usually done to provide activity data which is extracted from spatial data. Acceleration patterns and kilometers driven are examples of this. In strategies, data from privacy level 1 to 3 is also found by using behavioral extraction. Area driven is an example of this. Another example is to extract sex and age from full identity data to provide level 1 identity data. As behavioral feature extraction can occur in providers as well as strategies, one can imagine a scenario where activity data is inferred using the full spatial data. That is, if level 4 spatial data is provided the activity data might be inferred even though it is is set to 0 as level 4 spatial data could be the only data needed to calculate activity in the activity provider. This should not be solved by the SITA model, and should, hence, be considered by developers implementing the model.

The conceptual model is also inspired by the UNIX access control model for files and directories. In this control model a user sets access by specifying a sequence of digits from $0-7$ for user, group, and all using the `chmod` command. The number specifies which access the user, group, and all have on the specific file. For example `chmod 644 testfile` gives read and write access to the user and read access to group, and all to testfile. In SITA, the syntax is the same (with an additional digit) but the semantics are different. Here the four digits represent the level of privacy of each dimension (in the order: spatial, identity, temporal, activity) and this is set on an user group or application, depending on how the conceptual model is applied in practice, rather than a file. E.g. `set-sita 4321 user-group` sets level 4 spatial, level 3 identity, level 2 temporal, and level 1 activity privacy for *user-group*.

To illustrate the principle of the SITA conceptual model, an example is given in Figure 2 with individual examples of Spatial, Identity, Temporal, and Activity strategies. The figure focus on exemplifying visually the completely different strategies of achieving the goal of less information within one dimension. Spatial strategies are illustrated on a map, identity on a portrait photo, temporal on time, and activity for mode of transportation. Privacy level is set to `4231`. Vertically, we have the four dimensions and horizontally the five privacy levels from 0 to 4. In the bottom the resulting strategies are shown.

IV. From Conceptual Model to Architecture

The following section will propose a software architecture for implementing the SITA conceptual model. The idea of the software architecture is to be the basis for a fundamental and simplistic middleware and, hence, not provide an all including software package. The middleware should be an implementation of the SITA conceptual model and aid developers in adding multi-dimensional location privacy while helping them to consider all aspects of SITA. To develop the architecture, the method of architectural prototyping was utilized [28] where several prototypes were build through out the design process.

TABLE I. SPATIAL PRIVACY LEVELS.

0	**No information**: No spatial information is shared.
1	**Aggregation**: Spatial position is aggregated among other positions. This might be implemented by discretizing the space to squares and returning such a square as done in The New Casper [25].
2	**Obfuscation**: Spatial position is replaced by a fake position. This can be a perturbed position as proposed by Gidofalvi et al. [26].
3	**Regulation**: Rules are applied for a specific part of the space. Position is returned as normal except within these zones where rules from aggregation, obfuscation, or no information apply.
4	**Full information**: Full spatial information is returned.

TABLE II. TEMPORAL PRIVACY LEVELS.

0	**No information**: No temporal information is returned.
1	**Aggregation**: Time is aggregated to a period as ,e.g., 10 minutes intervals, or less precise as, e.g., only the date.
2	**Obfuscation**: Time is obfuscated by changing the timestamp. A method on this level can be to randomly delay data, e.g., between 0 and 10 seconds.
3	**Regulation**: Time data is changed according to rules within a specified time interval. For example not allowing any timestamp on the 24th of January.
4	**Full information**: Full temporal information is returned.

TABLE III. IDENTITY PRIVACY LEVELS.

0	**No information**: No identity information is returned. This is often called anonymity.
1	**Aggregation**: Identity is aggregated to be indistinguishable among other identities. This can be a group, or part of the full identity which places then in a group.
2	**Obfuscation**: Identity is obfuscated by providing false identity information. This might be as simple as changing the age or as complex as providing a fake social security number.
3	**Regulation**: Rules are applied so that different positions can be tied to the same identity, but not necessarily any details about identity data. An example is pseudonyms.
4	**Full information**: Full identity information is returned.

TABLE IV. ACTIVITY PRIVACY LEVELS.

0	**No information**: No activity information is returned.
1	**Aggregation**: Information aggregated so that the activity is indistinguishable from other activities. An example method for this level is to place activities within groups and only return the group.
2	**Obfuscation**: Activity is obfuscated. This can be to modify the full data to appear as valid full data, but with partly or entirely false information.
3	**Regulation**: Rules are applied so that full activity information is returned unless activity is within the rules.
4	**Full information**: Full activity information is returned.

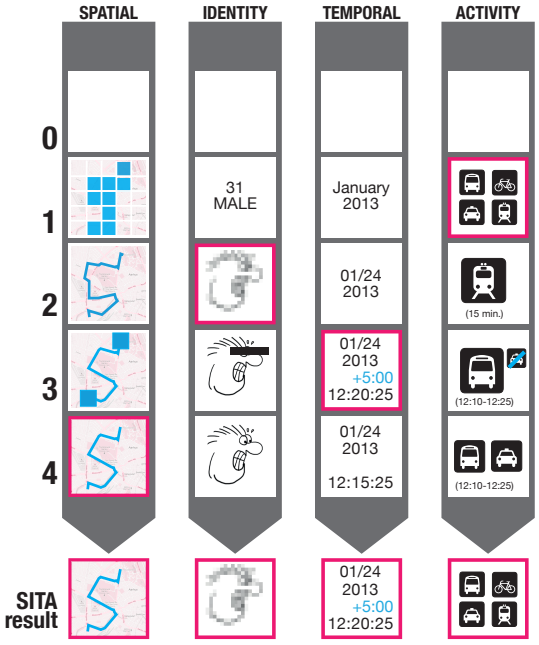

Fig. 2. SITA implementation with `set-site 4231`. Each arrow represents five strategies of each dimension. The box in the bottom contains the result.

A. Data Persistence Trade-Offs

When it comes to privacy, it becomes a major concern when data is somehow shared with others (users or services) and potentially stored. If data is kept on the device where it is collected, privacy can only be compromised if an adversary gains physical or logical access, e.g., as part of an application [5] and here security measures such as pin codes or cryptology can be utilized. Therefore, we will focus on the case where data is shared. Here, it becomes a question of how much trust is put into the receiving party, from now on referred to as the server. If complete trust is put in the server, full data can without issues be sent to the server, and then the server can handle privacy if it further distributes the data. In the case the server is not trusted filtering need to be done at the client side, and only filtered data should be shared. Hence, we have two data models for the two scenarios: *server restricted data*, in which we trust the server to restrict the view on the data, and *client restricted data*, where we only share the data that others are allowed to see.

The implications of sharing full data is that the data can be used by the receiving party to do computations later which were not thought of at the time when data was collected. This can, e.g., be the case when collecting traffic information, initially to calculate CO_2 footprint, but later it is found that the same data might be used to calculated road bottle necks. If data had been restricted at the client, and the only data sent was a sequence of CO_2 footprints, there will be no identity, no spatial, and no temporal data, and hence it would be impossible to calculate potential bottle necks in traffic. Collecting the data in the first place might be a very time consuming and expensive task, so saving the data and only restricting the view might save a lot of time and money. However, sharing only limited data has the benefit of guaranteeing that data cannot be used to infer things beyond the original purpose. For example, if an employer (with ownership of the server) decides that the data collected to calculate CO_2 consumption as part of green accounting suddenly decides to fire an employee for wasting too much time on transportation, this can only be done with server restricted data. This is because the data needed to compute time waste was never shared in the client restricted data model. Therefore, a trade-off analysis is needed before choosing one or the other.

B. Common and Service-specific Components

A main consideration for an software architecture is to determine which parts should be common for all implementations, and which should be service specific. The following division of responsibilities was chosen given the objective of building a reliable and flexible architecture with a high degree of decoupling and a single point of entry for the developer. The proposed design was chosen after experimenting with different options during the architectural prototyping phase.

The software architecture for SITA proposed to achieve these goals, consists of seven major components each of which has responsibilities as listed below.

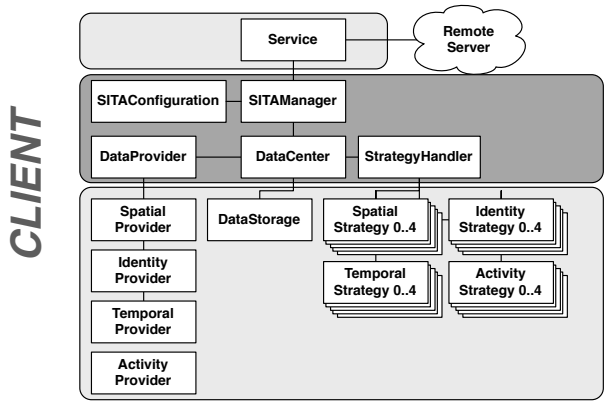

Fig. 3. SITA Architecture with client restricted data. The dark gray areas contains common fully implemented components used across SITA implementing applications, and the light gray contains service specific components which are the interfaces to be implemented by the specific service.

a) SITAManager: The SITAManager, is the single point of entry for a service that wants to obtain data controlled by SITA privacy controls. The responsibility of the SITAManager is to notify subscribers of data updates. The manager can also be queried for larger amounts of data, e.g., location data of last week or simply last known location.

b) SITAConfiguration: The SITAConfiguration holds the privacy configuration of the client, and it is queried for current settings every time the manager needs to process and deliver data.

c) Service: This is the client service which needs location privacy. Depending on the service, data can either be received by subscribing to updates from the SITAManager, or querying for the needed data.

d) DataCenter: The DataCenter is the main entry point for full location data handling, i.e., before any behavioral feature extraction is done. This being both in storing the data through the DataStorage, getting raw input data from the DataProvider, and calling the appropriate behavioral feature extractor using the StrategyHandler.

e) DataStorage: The DataStorage has the responsibility to store the data in some persistent way.

f) DataProvider: The DataProvider provides the full input data necessary for the system. This is done by four separate data providers for activity data, spatial data, temporal data, and identity data.

g) StrategyHandler: The StrategyHandler makes sure that the appropriate strategies are returned to the DataCenter in agreement with the SITAConfiguration.

In the client restricted data model, these components are all located on the client. This is illustrated in Figure 3. In this figure the distinction between common and service specific components is made by enclosement in a dark gray box for common and light gray for interfaces which have to be implemented by the specific service.

In the server restricted data model, the common components exist on both client and server. Essentially, the architecture from the client restricted data model, is mirrored on the server, but StrategyHandler and DataStorage are only found on the server side since full data is transmitted to the server, and the behavioral feature extraction is only done as other users or services request access to the location data on the server. Furthermore, the DataProvider and its providers do not exist on the server as location data is only collected at the client. The SITA architecture with server restricted data can be found in Figure 4. Again, dark gray boxes indicate common components and light gray service specific.

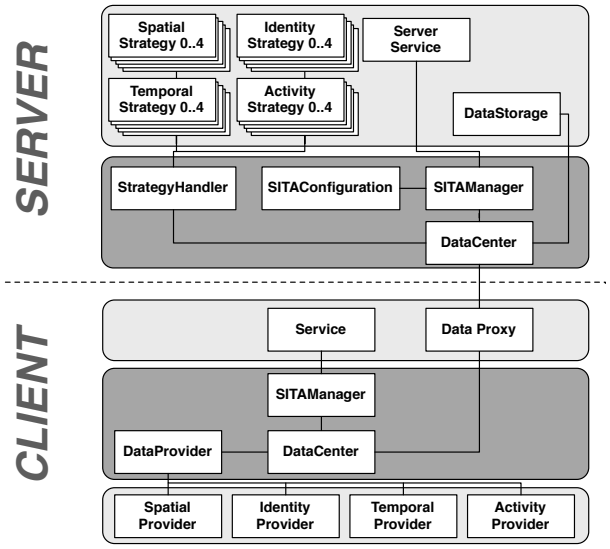

Fig. 4. SITA Architecture with server restricted data. The dark gray areas contains common fully implemented components used across SITA implementing applications, and the light gray contains service specific components which are the interfaces to be implemented by the specific service.

The five components: SITAManager, SITAConfiguration, DataCenter, StrategyHandler, and DataProvider constitute the common part of the architecture which is similar for all implementations of SITA location privacy. DataStorage should be defined so that it can store the specific data types, i.e., full data that are application specific. The two remaining components: Service and DataStorage, the developer will have to implement. Furthermore, the developer has to define a spatial, identity, and activity provider and three extraction strategies for each of the four dimensions. The reason that only three (and not five) strategies have to be provided is that level 4 is always full data, and level 0 is no data. This is handled using a full data pass-through and a null strategy, respectively.

C. Two Alternative Data Flows

The data flow can be initialized in two different ways. The most common is to subscribe to updates through the SITAManager using the observer pattern. When this approach is used, the DataProvider receives raw input data from its four providers. As spatial and activity data always changes more rapidly than identity, a new full data entry for the data center is constructed every time a new location appears from the spatial provider. This data is then passed to the DataCenter. Here the data flow changes depending on whether client restricted data or server restricted data is chosen as persistence model.

h) Client Restricted Data: The full data is stored using the DataStorage. Then it is passed on to the StrategyHandler,

which call a strategy for each of the four dimensions in accordance with the SITAConfiguration. The output data of these four strategies is then collected in a set of behavioral features which are passed on to the SITAManager, which then in turn notifies the service that new data is available.

i) Server Restricted Data: The full data is passed directly to the DataCenter, which notifies the server-side DataCenter through the Data Proxy that new data is available. The server-side DataCenter, stores it in the DataStorage. Upon data requests, the full data is passed to the StrategyHandler, and the resulting behavioral feature data is extracted based on the SITAConfiguration, and returned to the requester.

In the case where the SITAManager is called instead of being notified of changes, a range query is issued with the time for which to return behavioral feature data.

D. Data Providers

In a SITA implementation there are four data providers: activity, spatial, temporal, and identity.

Each data provider can be implemented in one of three ways. Using sensors, a user interface, or a trusted third party. The reason that it cannot just be the server, is that it would compromise the idea of the server only receiving restricted data. Data providers are application specific, so these have to be implemented or chosen in a library of existing providers. For example, one can imagine a spatial provider based on a smartphones GPS receiver, and that the data it provides is (longitude,latitude,altitude,accuracy,...) tuples, so that the data type can be reused. One could image the same for an activity provider which returns transportation mode based on GPS readings as proposed by Zheng et al. [29].

E. Plugin Strategies

Each dimension has five different levels, but as mentioned only three need to be provided by the developer per dimension. As data providers, the strategies are service specific, and share the idea that over time a library of different strategies can be built, and developers can choose from existing strategies.

Some strategies are quite simple and will in a few steps compute the outputted extracted behavioral features from the full data. Strategies can, however, also be more complex, e.g., involve heavy processing or communicate with a trusted 3rd party. An example with a trusted 3rd party is the implementation of k-anonymity as an aggregated identity strategy, as the strategy will have to know the location of the k-1 other users [30]. Moreover, temporal strategies are always a bit more complex as they will also have to control when notification of new location data is done. To motivate this, take the example that the level 1 temporal strategy is to aggregated time to a day rather than a precise timestamp. If the timestamp is changed from *2013-01-24T11:14:13* to *January 24th, 2013*, it does not make sense to notify of a new location immediately, as the data receiver (for example the server) will be able to timestamp the location data itself. Therefore, temporal strategies will, in addition to an extraction strategy, have to specify whether notification should be done at all and when it should be. In the example, this could be when the date changes, and then all data from the 24th could be reported, as there would be no logical link indicating the order of the location data from this date. Another example is when random obfuscation of the timestamp is used. If listeners are notified instantly it will be easy for an adversary to determine that this timestamp is false. The notification should be delayed until the timestamp will seem to be correct, and to make it even harder, this obfuscation should be random.

F. AndSITA – Android Implementation

As the architecture was designed using architectural prototyping [28], To test the feasibility of the SITA Architecture we gradually implemented AndSITA, an Android implementation of the software architecture described above.

The entry point for services utilizing AndSITA is the `SITAManager`. This is implemented using the singleton design pattern. This is chosen to simulate the manager structure which is extensively used by the Android API, so that AndSITA follow the design guidelines of the Android framework. The SITAManager should then be used by the service instead of the existing Android sensor managers as AndSITA will guarantee location privacy on top of them. On creation, the developer has to set the persistence model to either client or server restricted data. This has an influence on what service specific components the client-SITAManager will expect. All the service specific components are dependency injected through the `SITAManager`.

V. Case: Piccollo Management System

To test that the software architecture is complete and comprehensible, a case study was implemented. Based on using both the client and server persistency models to show that both can be handled in a real case. The case study covers a system for piccollos to better coordinate work tasks by knowing activities and positions of each other. The system is being prototyped independently of the present project. As the case is addressed without prior knowledge of the system the case study gives a real indication of the completeness of SITA. In addition, it will evaluate how comprehensible AndSITA is to developers. The piccollo management system has the following requirements: (1) Activities of the piccollos must the registered, and these activities must be divided into activity groups. (2) WiFi positioning has to be used for positioning. (3) Identities of piccollos are known in detail down to social security number. The system will among others expose the position, identity, and activity of the piccollos on a web-page and therefore there is a fundamental need to demonstrate how privacy can be added to the project.

A. Implementation

To implement the system using SITA we need to consider the following: (1) What constitutes the full location data? (2) How should the four providers be implemented? (3) Which twelve strategies should be chosen (3x4) and what data should they output?

In the system SITA has to handle the following types of data. Spatial data is an (x,y,z) coordinate originating where x and y determines the placement in the plane, and z is the altitude which can be used to determine floor. Identity data is first and last name, job, address, and social security number.

In Denmark, the latter uniquely identifies an individual and is prone to many types of attack including identity theft. Therefore, privacy and security controls are very important when such information is part of the data. Temporal data, is a timestamp attached when the spatial provider sends new spatial data. Activity data is the name of the activity, an activity group name of which all activities are members, and a timestamp indicating when the activity begun. Activities are listed in the following:

Activity Group	Activity Name
Complex Task	Sorting Mail Repairing Projector
Simple Task	Transporting Coffee Cleaning Room
Idle	Drinking Coffee Facebooking
Break	On Break

As providers for the system we implemented the following providers. For wifi positioning we use a system developed at the department as described in [31]. On the client the system is available as an Android activity which provides an (x,y,z) coordinate set. We integrate this system as a provider by modifying the Positioning (Android) activity to broadcast an action whenever a position is ready. This action is then capture by `WifiSpatialProvider` which extracts the data, bundles it in a `SpatialFeature`, and notifies the `DataProvider` that new spatial data is ready. For the identity provider, we hard-code for the evaluation a person with relevant data. This provider notifies the `DataProvider` on creation. The activity provider is also hard-coded for the test so that it changes activity based on time since the application was started. The background thread which does this, is started on provider creation. The `DataProvider` is notified on activity changes. This is to be replaced in future versions with user updated information or automatically sensed data.

As there were no explicit requirements for privacy support in the project, we analyzed the scenario to identify relevant strategies. The strategies were chosen to demonstrate how different types of location privacy can be used in combination with SITA.

As spatial strategies we consider the following for level 1 to 3 (as 0 and 4 are given): (1) extract a 20x20 meter zone in which the position is located; (2) randomly obfuscate the position between 0 and 10 meters in a random direction; (3) provide full position unless the position is within a restricted zone. If this is the case, the zone is returned. The choice of restricted zones are based on the location of bathrooms and coffee room.

Identity strategies chosen for level 1 to 3: (1) aggregation of users based on job, (2) identity obfuscated by providing full identity data, but with a false name and false social security number, (3) name replaced by pseudonym (calculated from the social security number).

Activity strategies chosen for level 1 to 3: (1) aggregation of task to an activity group, (2) obfuscation of activity by providing full data, but with a false activity within the correct activity group, (3) restriction of activities by return full activity data unless activity group is *Break*, then no information is returned.

Temporal strategies chosen for level 1 to 3: (1) temporal aggregation by changing timestamp to only date, (2) temporal obfuscation by aggregating locations with 10 minutes intervals, (3) time data is changed based on the rule that timestamps between 8:00 am and 9:00 am will never occur. In this case one hour will be added to the timestamp. However, temporal strategies are more complex than the other types as they also specify when notifications should be done. In relation to level 1, notifications should be done every time the date changes. All positions of the previous date should be returned. For level 2, notification should be done after each 10 minute interval and all locations of the previous interval should be returned. For level 3, a background timer should be started and notifications should only be send after this timer runs out.

While using UNIX file-system access control model might be easy comprehensible to computer experts, it will most likely not be to a regular user. Hence, to maintain the property of easy comprehensible, a graphical user interface to control SITA is needed. In relation to the on/off restriction, the user interface plays an important role. As our current implementation of the piccollo management application was developed mainly to test out how easy it is to utilize SITA for developers, how users perceived the system was not a major concern. Still, we expect that providing the user with the ability to control each dimension independently while receiving concurrent feedback will aid users in making informed decisions, nudging them towards choosing a more strict privacy setting. We do this by providing the user with a slider for each dimension, and a text field with immediate feedback for each dimension. In Figure 5 we see an example which is similar to `set-sita 4131`. Here we see four sliders, one for each dimension, which is used to set the current privacy levels. In the text field below these, we see four lines representing the received behavioral feature data, so that users can see how changes in privacy settings are reflected on what they are sharing.

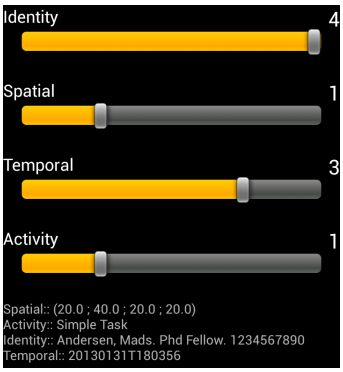

Fig. 5. User interface of the piccollo management system.

B. Experiences

Implementing the case of the piccollo management system project, on a conceptual level showed that the SITA architecture can model a real world case. Furthermore, with little effort we were able to implement the strategies, data providers as listed above and plug these into the AndSITA implementation of the proposed software architecture. For the evaluation we ran the system and observed that when we changed settings in the user interface this was reflected in the used strategies and the data outputted. In our evaluation we also considered the challenges in switching from the client to the server restricts data persistence model. To facilitate this evaluation

we in addition to AndSITA implemented JEntSITA on top of a Java Enterprise Edition server based on a number of Java Servlets connected to a PostgreSQL database. This process was straight forward as most of our code were standard Java source code, except from a few Android specific things. Hence, we demonstrated that the system could run in either mode. With the combination of Android and Java Enterprise Edition, most of the same source code could be used for both platforms (with different compilers), however, other combinations, e.g., iOS and Ruby on Rails, would make this process more tedious. While it was not a major goal in the evaluation of SITA to consider an user interface for aiding users to take informed decisions about location privacy, we envision that SITA has to potential to do so. We hypothesize that if more effort is put into investigating the user interface potential of SITA, it can be demonstrated that the software architecture is easy comprehensible to users as well as developers.

VI. Conclusion

In this paper, we presented the SITA conceptual model for location privacy to address the problems that existing location privacy methods suffer under. SITA is a conceptual model featuring a simplicity principle and describing location privacy in four fundamental and orthogonal dimensions of spatial, identity, temporal, and activity, and dividing each of them in five discrete levels. This enables the model both to be complete and comprehensible. Complete in the sense that all existing privacy methods can be described using SITA and that the model can be applied to all types of LBSs. Furthermore, we proposed a flexible and simplistic software architecture to implement the concept and backed with experiments for our Android-based prototype AndSITA. Therefore we also claim that the conceptual model and it's realization is easy comprehensible in the sense that it is easy to utilize for developers. In relation to earlier work in the area of location privacy, SITA provides a high level model to discuss and compare location privacy, while aiding developers to add location privacy to LBSs by providing the SITA software architecture. Finally, we indicated that SITA is complete by considering the case of a student assistant coordination application implemented using AndSITA. This turned out to be easy and, hence, in addition provide evidence for the statement that SITA is easily comprehensible for developers.

In our ongoing work we address the following: First, we explore SITA in relation to being comprehensible for users by using the concept to aid users in controlling location privacy in several cases. We expect, that SITA can nudge users towards taking more informed decisions about location privacy and therefore set more suitable privacy settings. Finally, we propose to add SITA privacy to a wider array of LBSs, and in particular existing LBSs to test more extensively for completeness.

References

[1] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir, the personal environmental impact report, as a platform for participatory sensing systems research," in *MobiSys*. ACM, 2009.

[2] "Facebook," http://www.facebook.com/.

[3] "Google latitude," http://latitude.google.com/.

[4] "Foursquare," https://foursquare.com/.

[5] 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 *OSDI*. USENIX, 2010, pp. 1–6.

[6] G. Danezis, S. Lewis, and R. Anderson, "How much is location privacy worth?" in *Online Proc. of the Workshop on the Economics of Information Security Series*, 2005.

[7] "Google plus," http://plus.google.com/.

[8] "Endomondo – community based on GPS tracking of sports," http://www.endomondo.com/.

[9] M. Andersen and M. Kjærgaard, "Towards a new classificaion of location privacy methods in pervasive computing," in *Mobiquitous*. Springer, 2011.

[10] A.-M. Zafeiropoulou, D. E. Millard, C. Webber, and K. O'Hara, "Privacy implications of location and contextual data on the social web," in *Websci*. ACM, 2011.

[11] J. Krumm, *Ubiquitous Computing Fundamentals*, 1st ed. Chapman & Hall/CRC, 2009.

[12] J. Maeda, "Ten laws of simplicity," http://lawsofsimplicity.com/.

[13] J. I. Hong and J. A. Landay, "An architecture for privacy-sensitive ubiquitous computing," in *MobiSys*. ACM, 2004, pp. 177–189.

[14] H. Obendorf, *Minimalism - Designing Simplicity*, ser. Human-Computer Interaction Series. Springer, 2009.

[15] V. Bellotti and A. Sellen, "Design for privacy in ubiquitous computing environments," in *ECSCW*, 1993.

[16] M. Langheinrich, "Privacy by design - principles of privacy-aware ubiquitous systems," in *UbiComp*. Springer-Verlag, 2001.

[17] A. Gupta, A. Kalra, D. Boston, and C. Borcea, "Mobisoc: a middleware for mobile social computing applications," *Mob. Netw. Appl.*, vol. 14, no. 1, pp. 35–52, Feb. 2009.

[18] M. Román, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt, "A middleware infrastructure for active spaces," *IEEE Pervasive Computing*, vol. 1, no. 4, pp. 74–83, Oct. 2002.

[19] B. Johanson, A. Fox, and T. Winograd, "The interactive workspaces project: Experiences with ubiquitous computing rooms," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 67–74, Apr. 2002.

[20] R. Grimm, J. Davis, E. Lemar, A. Macbeth, S. Swanson, S. Gribble, T. Anderson, B. Bershad, G. Borriello, and D. Wetherall, "Programming for pervasive computing environments," Tech. Rep., 2001.

[21] Y. Li, J. I. Hong, and J. A. Landay, "Topiary: a tool for prototyping location-enhanced applications," in *UIST*. ACM, 2004, pp. 217–226.

[22] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, "Anonysense: privacy-aware people-centric sensing," in *MobiSys*. ACM, 2008, pp. 211–224.

[23] L. P. Cox, A. Dalton, and V. Marupadi, "Smokescreen: flexible privacy controls for presence-sharing," in *MobiSys*. ACM, 2007, pp. 233–245.

[24] "Alka boksen (translated: The alka box)," http://www.alka.dk/Privat/Forsikringer/Bilforsikring/AlkaBoksen, usage based insurance from the Danish insurance company Alka. Source in Danish.

[25] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: query processing for location services without compromising privacy," in *VLDB '06*, 2006, pp. 763–774.

[26] G. Gidofalvi, X. Huang, and T. Pedersen, "Privacy-preserving data mining on moving object trajectories," in *MDM*, May 2007.

[27] M. S. Andersen, M. B. Kjærgaard, and K. Grønbæk, *Using Extracted Behavioral Features to Improve Privacy for Shared Route Tracks*. Springer, 2012, pp. 107–118.

[28] J. Bardram, H. Christensen, and K. Hansen, "Architectural prototyping: an approach for grounding architectural design and learning," in *WICSA*, 2004.

[29] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma, "Understanding transportation modes based on gps data for web applications," *ACM Trans. Web*, vol. 4, no. 1, Jan. 2010.

[30] B. Gedik and L. Liu, "A customizable k-anonymity model for protecting location privacy," in *ICDCS*, 2004, pp. 620–629.

[31] M. B. Kjærgaard, "Indoor positioning with radio location fingerprinting," *CoRR*, vol. abs/1004.4759, 2010.