

Privacy Policies for Location-Aware Social Network Services

Ingrid Hjulstad

Master of Science in Communication Technology
Submission date: June 2011
Supervisor: Svein Johan Knapskog, ITEM
Co-supervisor: Serge Gladyshev, ITEM

Problem Description

Name of student:

Ingrid Hjulstad

Problemdescription:

Privacy Policies for Location-Aware Social Network Services

Social Network Services have already become one of the most noticeable and widely-used Internet-based technology of the 21st century. Location-Aware Social Networks have recently appeared (2009/2010), made fast progress and gained huge popularity, e.g.: Foursquare, Gowalla. Consequently, the largest social network platforms have also introduced new location-aware features in 2010, e.g.: Facebook Places and Google Buzz. While such new technologies are opening exciting opportunities by giving plenty of benefits to the humanity, at the same time they can possess serious threats to privacy. Indeed, privacy in social networks has become a very hot topic attracting huge amount of attention and raising lots of research and technical challenges. Moreover, privacy in location-aware social networks is even more important, if not to say dramatically critical for the nearest future.

The task for this Master thesis is thus to investigate the new privacy implications of location-aware social networks (e.g. Facebook Places feature, Google Buzz and others). End-user scenarios (use-cases of location-aware social networks) will be used as a starting point for such a privacy analysis. The planned goal of this work is to design a privacy policy framework for location-aware social networks.

Assignment given: 15.01.2011

Supervisor: Serge Gladyshev

Name of responsible professor: Svein Knapskog

Abstract

The combination of location-awareness and social networks has introduced systems containing an increased amount of protection-worthy personal information, creating the need for improved privacy control from a user point of view.

End-user privacy requirements were derived from identified end-user privacy preferences. These requirements were used to evaluate current Location-Aware Social Network Services' (LASNSs') end-user privacy control as well as help develop relevant enhancements.

These requirements allows users to be able to control (if they wish) which of the objects related to them are accessed by whom, in what way and under which conditions. Two enhancement ideas which together helps fulfill this requirement have been presented. The few LASNSs offering the user access control rule specification only provides a small list of pre-defined subjects (e.g. "Friends", "Everyone"). This list is too limited for specification of many fine-grained privacy preferences. With a more extensive implementation of Role Based Access Control (RBAC) in LASNSs, with the user as the system administrator of roles, users will be able to create roles (e.g. "colleague", "close friend", "family"), assign them to their connections, and specify these roles as subjects in access control rules. The user will also be allowed to specify conditions, under which subject(s)/role(s) can access an object. These conditions can be based on system attributes of the object owner (e.g. location), the subject requesting access (e.g. age) or external attributes (e.g. time). A suitable user-friendly access control user interface has been proposed, showing how this can be presented in an effective and understandable way to the user. A few example user privacy preferences, each one representing one of the identified end-user privacy control requirements have been translated from data sent to the system through the proposed interface, into formal languages like Datalog and XACML.

Current end-user privacy control can be improved, by making more fine-grained access control rule specification possible, through the proposed enhancements, suitable both from an end-user perspective and from a developer's point of view.

Keywords: Location-Aware Social Network Services, Privacy preferences, Enhanced end-user privacy, Access control rules, Role Based Access Control, Use of XACML, Use of Datalog.

Acknowledgements

I would like to express gratitude to my supervisor, Serge Gladysh, for help and guidance throughout the work with this thesis. From beginning to end, he has been there to help with interesting scientific directions, insights and advice, as well as support.

I would also like to thank professor Svein Knapskog for helpful advice and challenging discussions. The discussions gave me a deeper understanding of both the relevant research fields, and the challenges and possible solutions in this master thesis.

Guri Tronstad and my fellow students, especially Nicolai Berthelsen, also deserve a thank you, as they have provided me with constant motivation and served as valuable discussion partners and support system.

Contents

1	Introduction	17
1.1	Motivation	17
1.2	Objectives	18
1.3	Scope	19
1.4	Methodology	19
1.5	Outline	20
2	Background	21
2.1	Access control	21
2.1.1	Basic concepts	22
2.1.2	Discretionary Access Control	23
2.1.3	Mandatory Access Control	24
2.1.4	Role Based Access Control	27
2.1.5	Extensions of RBAC	33
2.1.6	Logic in Access Control	36
2.2	Privacy policies	36
2.2.1	Policies and privacy regulation	37
2.2.2	P3P	37
2.2.3	EPAL	38
2.2.4	XACML	39
2.2.5	Privacy policy enforcement architectures	41
2.2.6	Privacy policy referencing	42
2.3	Emerging privacy issues in the new IT-technologies	43
2.3.1	Online Social Networks access and privacy control re- quirements	43
2.4	Conclusion	45
3	Scenarios	47
3.1	Scenario 1, SNS	47
3.2	Scenario 2, LASNS	49
4	Privacy in Social Network Services and Platforms	51
4.1	General overview of SNSs	51
4.1.1	Introduction to SNSs	52
4.1.2	Data models for SNSs	52
4.2	Facebook as an example SNS	53
4.2.1	End-user Interface	54
4.2.2	The Facebook Platform	57

4.2.3	Facebook user privacy requirements	59
4.2.4	Facebook user privacy control	63
4.2.5	Facebook privacy control analysis	71
4.3	Privacy requirements for SNSs in general	74
4.4	Conclusion	74
5	Privacy in Location-Aware Social Networks	77
5.1	Location-aware mobile services	78
5.1.1	Underlying technology overview	78
5.1.2	Privacy implications	81
5.2	Location-awareness with SNSs	83
5.2.1	Integration: how it works	83
5.2.2	Privacy implications	85
5.3	Facebook Places	86
5.3.1	Overview	86
5.3.2	Privacy control	87
5.4	Other LASNS examples	89
5.4.1	Foursquare	89
5.4.2	Gowalla	92
5.4.3	Loopt	94
5.4.4	Brightkite	96
5.5	LASNS privacy control analysis	97
5.6	Conclusion	99
6	Enhanced Privacy Control Framework for LASNSs	101
6.1	Privacy-enhancing access control for LASNSs	102
6.1.1	More fine-grained subject separation	102
6.1.2	Rule conditions	105
6.2	Implementation	108
6.2.1	User interface	108
6.2.2	From user scenario case to system-interpretable access control rule	115
6.3	Comparison with existent solutions	123
7	Future work	125
8	Conclusions	127
A	XACML examples	129
	Bibliography	137

List of Acronyms

AC Access Control	21
ACP Access Control Point	22
ANSI American National Standards Institute	28
API Application Programming Interface	57
DAC Discretionary Access Control	21
DBMS Database Management System	53
DoD U.S Department of Defense	21
DSoD Dynamic separation of Duties	28
EPAL Enterprise Privacy Authorization Language	38
FB Facebook	86
FBML Facebook Markup Language	57
FBP Facebook Places	86

GPS Global Positioning System	77
GUI Graphical User Interface	19
IETF Internet Engineering Task Force	41
iOS iPhone OS	79
ISO International Organization for Standardization	41
IT Information Technology	38
LA Location-aware	78
LASNS Location-Aware Social Network Service	17
MAC Mandatory Access Control	21
OASIS Organization for the Advancement of Structured Information Standards	39
OGC Open GeoSpatial Consortium	34
OS Operating Systems	57
PDP Policy Decision Point	41
PEP Policy Enforcement Point	41
P-RBAC Conditional Privacy-aware RBAC	35

RBAC Role Based Access Control	
SDK Software Development Kit	57
SNS Social Network Service	17
SoD Separation of Duties	31
SSoD Static Separation of Duties	28
TRBAC Temporal RBAC	34
W3C World Wide Web Consortium	37
Wi-Fi Wireless Fidelity	79
WWW World Wide Web	52
XACML eXtensible Access Control Markup Language	39
XML Extensible Markup Language	37

Glossary

This glossary provides an explanation of how to interpret the technical terms used in this master thesis. A star (*) is specified for explanations created or modified (from standard or generally accepted definitions) by the author.

Access control rule* - A rule which may be used to grant or deny a user or set of users access rights to an object or an operation (and possibly specify how and when).

App - A piece of software. It can run on the Internet, on a computer, or on a phone or other electronic device.

Owner - The subject of personally-identifiable information.

Privacy Control* - Users' control over their own privacy, in this case through access control.

Privacy Control Panel* - The interface where users try to match the privacy settings for their account with their own privacy preferences.

Privacy Policy - A set of privacy rules.

Privacy Preference* - A user-preferred level of privacy in a certain setting.
Example: "I would like only my friends to see my profile images".

Subject - An entity requesting access to an object.

Services

This master thesis will mention and discuss many different Social Network Services (SNSs), Location-Aware SNSs (LASNSs) and other types of services. The following is a list of all the mentioned services, uniquely identifying them through name and relevant sites where they can currently be found. There will be times when the text will refer to a service or a part of service, e.g the privacy control panel for accounts, which is not accessible unless one is logged into such a service with a valid user. This list will thus serve as references for these services, such that one can access (possibly through creating an account) the discussed material. Where it is necessary screenshots of the material will be included.

Android - A software stack for mobile devices, includes an OS, middleware and applications. Home page: <http://source.android.com/>.

Android Market - Android's app marketplace. Home page: <https://market.android.com/>. For more information about Android see the Android service.

Booyah - A social web and entertainment company. Home page: <http://www.booyah.com/>.

Brightkite - A LASNS for exploring locations and connecting with people. Home page: <http://brightkite.com/>. (Access to the privacy control settings discussed in this master thesis requires log in with a valid account)

Buddy - A mobile service for locating friends. Home page: <http://www.mbuddy.no/>.

Classmates.com - A SNS for re-connecting with (past) classmates. Home page: <http://www.classmates.com/>

Digg - A SNS for discovering and sharing web content. Home page: <http://digg.com/>. Further information about the service can be found at <http://about.digg.com/>.

Facebook - A SNS for connecting with people and sharing content. Home page: <http://www.facebook.com/> (Access to the privacy control settings discussed in this master thesis requires log in with a valid account).

Facebook Developers - Site for help using the Facebook API. Home page: <http://developers.facebook.com/>.

Facebook Places - A LASNS addition to Facebook. Home page: <http://www.facebook.com/places/> (Access to the privacy control settings discussed in this master thesis requires log in with a valid account).

Foursquare - A LASNS for exploring locations and connecting with people. Home page: <https://foursquare.com/> (Access to the privacy control settings discussed in this master thesis requires log in with a valid account).

Gowalla - A LASNS for sharing and exploring locations plus connecting with friends. Home page: <http://gowalla.com/> (Access to the privacy control settings discussed in this master thesis requires log in with a valid account).

LinkedIn - A SNS for professional profiles and connections. Home page: <http://www.linkedin.com/>

Loopt - A LASNS for exploring locations and connecting with people. Home page: <https://www.loopt.com/> (Access to the privacy control settings discussed in this master thesis requires log in with a valid account).

Socialcast - A SNS for enterprise collaboration. Home page: <http://www.socialcast.com/>

Yelp - A social networking site offering help through user reviews and a local search web site. Home page: <http://www.yelp.com/>

List of Figures

2.1	Access Control in general	23
2.2	Discretionary Access Control	24
2.3	Mandatory Access Control	25
2.4	Role Based Access Control	27
2.5	Core RBAC	29
2.6	Hierarchical RBAC	30
2.7	Static Separation of Duty relations	32
2.8	Dynamic Separation of Duty relations	32
2.9	RBAC with constraints	33
2.10	EPAL and XACML policy enforcement model	42
4.1	Facebook features and resources	55
4.2	Facebook privacy panel	64
4.3	Facebook privacy control list	65
4.4	Facebook privacy control; Customize	66
4.5	Facebook privacy control; Connecting on Facebook	67
4.6	Facebook privacy control; Apps and Websites	69
4.7	Facebook privacy control; App privacy settings	70
4.8	Facebook privacy control; Block Lists	72
5.1	Android phone and iPhone GPS app	80
5.2	iPhone Tracker result map	82
5.3	Facebook Places	87
5.4	Facebook Places privacy control; friends	88
5.5	Facebook Places privacy control; sharing	88
5.6	Facebook Places privacy control; sharing drop down menu	89
5.7	Foursquare	90
5.8	Foursquare privacy control	91
5.9	Gowalla	92
5.10	Gowalla privacy control	93
5.11	Loopt	94
5.12	Loopt privacy control	95

5.13	Brightkite	96
5.14	Brightkite privacy control	97
6.1	Role assignment example	104
6.2	Role access assignment example	105
6.3	Rule table example	106
6.4	Generalized condition format	107
6.5	Enhanced privacy panel	110
6.6	Textual AC rule format	114
6.7	Access control rule data	117

Chapter 1

Introduction

Online social networks have the last two decades become more and more popular, and have been developed in different directions providing different service-types. The recent development, facilitated by smartphones and other new phones capable of calculating the phone's current position, is social network services paired with location-aware features. We call each such service a Location-Aware Social Network Service (LASNS). New phones often contain, or are capable of downloading and installing, software connecting them with these services. People usually carry their location-aware phones with them at all times, and are now able to share their current location with their social network connections through the click of a few buttons. Social network services and now LASNSs, are often used for social interaction and fun, and development is driven by a search for new and fun ways of using existing data with new technology. People share more and more of their personal information through these services, yet focus on user privacy control has not followed the same pace. For many of these new services the user has little control of how the content they produce is handled and shared with other users. It appears to us like user privacy is not the priority of the LASNS developers as the user control panels usually are limited to coarse-grained control settings. While new fun and interactive features are released, privacy control often remain, in our opinion, inadequate.

1.1 Motivation

As a user of a Social Network Service (SNS), we value the ability to control how the information we create, or that in some way is related to us, is shared and treated. A large portion of this kind of data is personal and not the type of data everyone would like to share with all people in all situations.

Some SNS privacy control panels partially fulfill these requirements, yet we have not come across one SNS privacy panel where there is no room for improvement. It is important to realize that different users have different privacy preferences. Some users might find current control panels completely adequate for reflecting their preferences, and others might not even care about privacy. Still, the goal should be to be able to reflect every user's privacy preferences, including the ones who find that privacy control panels for these services currently provides too coarse-grained control for their taste.

The addition of locational information in SNS systems pose new requirements for user privacy control. A person's location might not only tell you where that person is at the time, it can also indicate other personal information when paired with additional data like what time it is, possibly who else is there at that time and/or other surrounding factors. In this thesis we hope to illustrate why it is important to control locational information as well as other data in these LASNSs, shed light on the shortcomings of current user privacy control for SNS and LASNS available today, and at last suggest enhancements to improve such user privacy control. Our goal is to present suggestions which will enhance the privacy control panels such that they are able to reflect nearly all users' possible privacy preferences, providing the users with the tools to protect their locations and other personal data from undesired disclosure.

1.2 Objectives

Based on our observations and motivational factors, research questions arise. The objectives we wish to reach and the research questions we wish to answer in this thesis are:

1. What kind of access control features exist in current LASNS to control end-user's privacy?
2. What kind of privacy preferences may end-users have in LASNSs?
3. Are existent access control features in LASNSs able to satisfy end-user's privacy requirements/preferences?
4. Which privacy-enhancing access control features in LASNSs should be added (or improved in which way) to satisfy end-user's requirements/preferences? (Illustrated with examples.)
5. How can the privacy-enhancing access control features be represented to end-users?

6. How can the privacy-enhancing access control features be represented in terms of logic rules and in machine readable format (e.g. XACML)?

1.3 Scope

The main focus of this thesis and its proposed results will be limited to LASNSs. LASNSs will be our focus when we discuss current user privacy as well as suggesting relevant enhancements. Yet, this thesis contains one chapter discussing privacy in SNSs, namely Chapter 4. This is because SNSs are a part of LASNSs, and it is thus useful to discuss and understand the social network aspect of LASNSs. Even though the proposed access control enhancements are meant for LASNS, it does not necessarily mean they can not be applied to SNS privacy control.

The result chapter, called "Enhanced Privacy Control Framework for LASNSs", will present the suggestions of how to improve user privacy control through enhanced access control rule specification. This master thesis will only suggest general ideas and recommendations based on discussion and analysis, for how to enhance LASNSs. These will not be applied in practice to any LASNS, due to time-limitations.

1.4 Methodology

To analyze existing user privacy control, and propose possible enhancements, we have to define some sort of goal to measure against. This goal is for users to be able to reflect their LASNS privacy preferences through their accounts' privacy control settings. This requires us to reflect upon and understand what kind of privacy preferences users can have. As explained by the project UbiCompForAll (Ubiquitous service composition for all users) in their collection of scenarios ([Ubi10]), the use of scenarios is a good method for coming up with new system ideas, and to understand the system's users. We have therefore selected a scenario-driven approach. End-user privacy requirements will be derived from identified end-user privacy preferences. These requirements will help us analyze current LASNSs' end-user privacy control as well as help develop and evaluate relevant enhancements. We will use fine-grained access control as a mean towards developing these enhancements. In order to show how the enhancements can be implemented, we use translation of data from end-user privacy preferences through a proposed Graphical User Interface (GUI) into a logic programming language Datalog and a declarative functional language XACML representation.

1.5 Outline

This thesis is outlined in the following way. After this introductory chapter, a background chapter will be presented. The background chapter will provide the reader with sufficient knowledge of the techniques and technologies discussed in this thesis. To be able to analyze and acquire an understanding of the user's privacy perspective, Chapter 3, called "Scenarios" follows the background chapter. This chapter consists of a list of many different user privacy preferences for different objects, subjects and access types. They take place in either the context of 1) a regular Facebook user (Scenario 1), an example of a SNS setting, or 2) a Facebook Places user (Scenario 2), an example of a LASNS settings. Then follows a chapter discussing privacy in SNS systems, Chapter 4. This chapter contains a general presentation of SNSs, a concrete example of such services (Facebook), and based on the previous discussion of general and specific SNSs, our proposed privacy requirements for SNSs in general. Following is a chapter called "Privacy in Location-Aware Social Networks" (Chapter 5), circling in on the main focus of this thesis, location-awareness in SNSs. This chapter discusses underlying technologies, integration with SNSs, a few example LASNSs and for each part, a discussion of privacy in the context of the relevant part. This chapter also contains a privacy analysis of current user privacy control in LASNSs. Chapter 6, "Enhanced Privacy Control Framework for LASNSs" is the chapter where we present our results; the proposed privacy enhancements, and discuss the implementation of these enhancements. A discussion of future work in Chapter 7 will then be presented, before Chapter 8 provides a conclusion of the work with this thesis.

Chapter 2

Background

This master thesis work will be focused mainly on privacy control through Access Control (AC) and privacy policies, and the interaction of services offered within these two fields. It is therefore useful to spend some time discussing these subjects in general. This chapter will present a necessary overview of each subject in turn, not only discussing the basic concepts, but also some newer models and developments.

2.1 Access control

The field of AC is one which has been created alongside the development of many kinds of computer systems. These systems have had different security and access control needs, which in turn has created the demand for different access control schemes.

[oD85] was one of the first papers to officially formalize AC, and was issued by the U.S Department of Defense (DoD) in 1983. This document, named the Trusted Computer System Evaluation Criteria, mainly introduced the two AC schemes Mandatory Access Control (MAC) and Discretionary Access Control (DAC). These two schemes were developed to fulfill the DoDs requirements for access control in military computer systems. The MAC scheme is the one that has the closest relations to the military domain, and one example of this relation is the basis for MAC decisions (as stated in [oD85]):

”...subjects and objects shall be assigned sensitivity labels that are a combination of hierarchical classification levels and non-hierarchical categories, and the labels shall be used as the basis for mandatory access control decisions”

The hierarchical classification levels are directly translatable to the DoD clearance hierarchy, "unclassified", "confidential", "secret" and "top secret", while the non-hierarchical categories can be translated into the different military departments and work fields. This is one of the reasons why these AC schemes work well within the military domain, but might not be as efficient in other domains, like SNS systems, and why other AC schemes have been developed. Such schemes will be presented below, after a brief introduction to the basic concepts of AC and a more thorough presentation of MAC and DAC.

2.1.1 Basic concepts

Access control is the task of limiting and controlling what resources an authenticated user is allowed to access, and in what way. AC will obviously not be effective unless secure authentication of a user was performed in advance. After the authentication process is complete, a user might start requesting access to different resources. When AC is present in the system, the user will make the request through an Access Control Point (ACP), which will decide whether or not to grant the request. The ACP will base its decisions on previously defined rules, which can include current conditions and other factors decided by the system developers and administrators.

Most AC systems makes their control decisions based on rules of some sort. These rules can have different types of authors, and can even be induced by more general rules or the owner of the resource in question. For basic AC systems, a rule will consist of three entities. The subject (S), the object (O) and the type of access to be granted (T). An example of such a rule can be: S has the permission to perform T on O. The subject S can be a user or an application running on behalf of a user, and is the entity requesting permission. The object O is the resource in which the subject requests access to, and could take the shape of a system resource, a file or anything else one might wish to restrict access to. T is the type of access the subject wishes to perform on the object. When O is a file, T might be operations such as read, write, execute, append, etc.

One important extension to the basic AC access rule format worth mentioning is the concept of conditions. Each access rule will then not only consist of S, O and T, but also conditions C. With this extension the rules will have the following shape: S has the permission to perform T on O, given C. Many types of conditions exist, but the two which are the most interesting for this master thesis are the temporal and the geographical. A temporal condition describes at which points in time S can access O. A geographical condition will rely on the location of the subject or object, and describe

Access Control: General Model

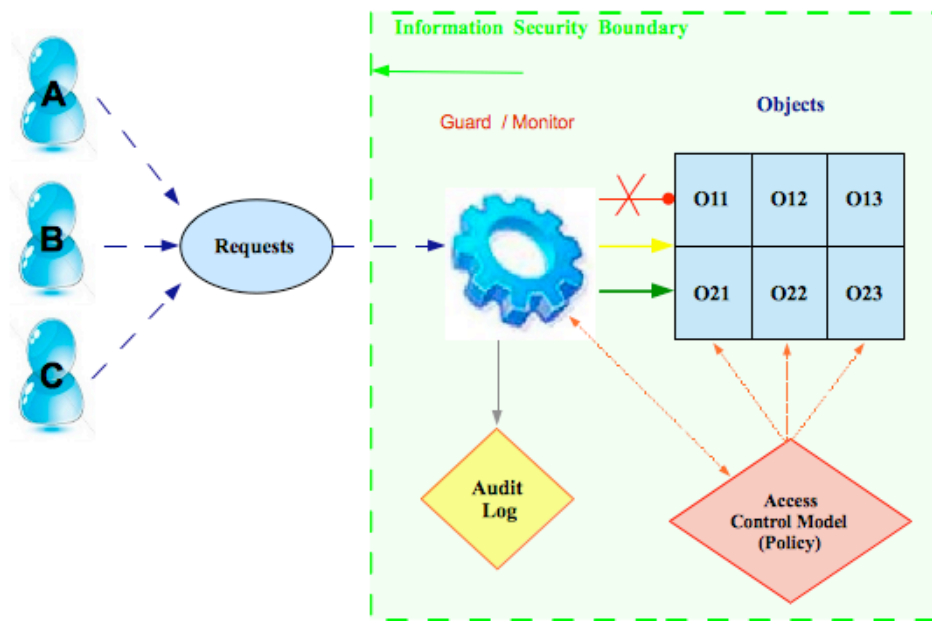


Figure 2.1: This figure shows how access control is performed in general. Entities will request access to certain objects, and the guard/monitor will make decisions on whether to grant access or not, based on policies (and in some cases log the request).

at which locations S is granted access to O . Both these, plus Conditional Privacy-aware RBAC will be further explored below as RBAC extensions.

2.1.2 Discretionary Access Control

DAC was officially presented in [oD85], and in it defined as:

”A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject (unless restrained by mandatory access control).”

DAC was developed as a way to control different kinds of access to objects by subjects. In a purely discretionary access control system there are no

Discretionary Access Control (DAC)

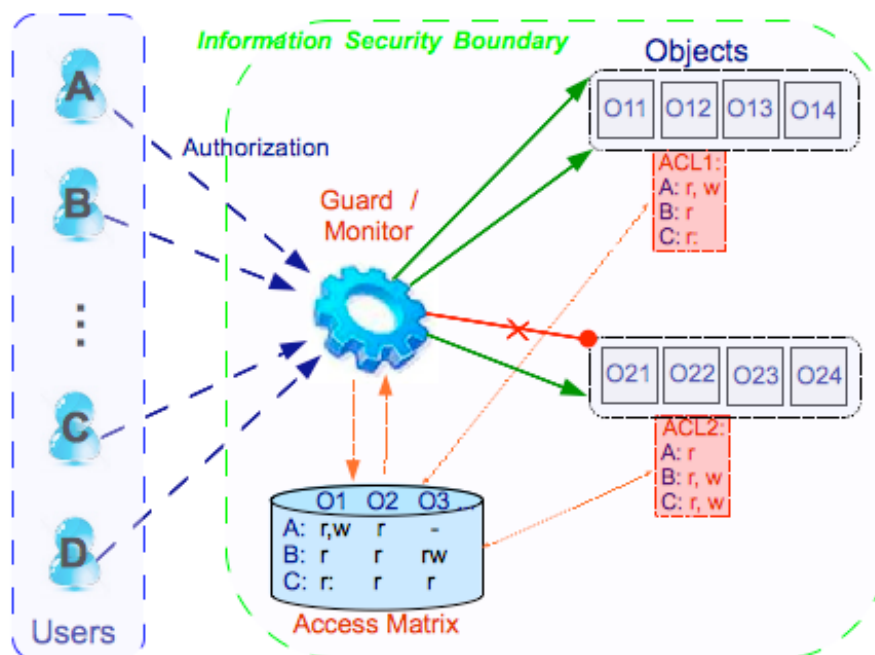


Figure 2.2: This figure shows how Discretionary Access Control is performed. It is similar to the general AC model in Figure 2.1, only here the guard/-monitor will make the access control decisions based on an access matrix.

mandatory access control policies, meaning that access control policies are defined by subjects. A subject can even hold certain access rights giving it permission to grant other subjects the same access rights.

2.1.3 Mandatory Access Control

As mentioned previously, MAC was developed by the DoD, and is custom-made to manage classified information flow within the military. The goal is to protect confidentiality by keeping classified information from leaking down the classification hierarchy. [oD85] defines how MAC achieves this goal through this precise description:

”A means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the

objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity”.

MAC is therefore based on policies and the control by a security policy administrator and will not, as opposed to DAC, allow users to override these policies to control access to different objects. This means that an owner of e.g. a file, will not be able to grant access to that file to a user which based on the policies would otherwise be denied access.

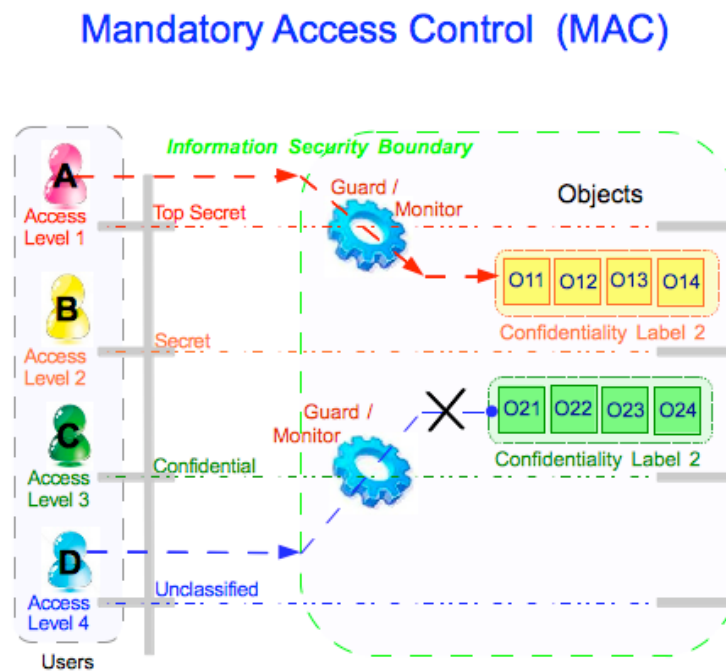


Figure 2.3: This figure shows how Mandatory Access Control can be modeled. Users and objects are divided into different classification levels. The guards/monitors will make access control decision based on the levels of the subject and object, plus global rules.

The Bell and LaPadula model

One of the first MAC models was the Bell and LaPadula model ([BP76]), and it is therefore presented as an example of a MAC model. It was developed to control the confidentiality of objects with a certain sensitivity/classification level for the operating system Multix ([BP76]), often used within the military. In other words: Preventing information-leakage to subjects or objects with lower classification than what was defined for that information.

According to this model every subject and object are assigned an access class, which consists of a security level plus a set of categories. To exemplify the model we will use security levels and categories found in the military domain. The assigned security level has to be an element of a totally ordered set, and will for this example consist of the following elements with relationships accordingly; Top Secret (TS) > Secret (S) > Confidential (C) > Unclassified (U). The categories Nuclear, Navy and Army will serve as the relevant categories for this explanation. A subject or object can be assigned multiple categories, for instance both Nuclear and Navy.

Access classes, containing both security level and categories, are a partially ordered set with the following formal dominance relationship \geq :

Definition : Dominance relationship One access class $AC_1 = (Level_1, Categories_1)$ dominates another access class $AC_2 = (Level_2, Categories_2)$, $AC_1 \geq AC_2$, if both of the following conditions hold: i) $Level_1 \geq Level_2$ (the security level of AC_1 is greater or equal than the security level of AC_2) and ii) $Categories_1 \supseteq Categories_2$ (the category set of AC_1 includes the category set of AC_2).

If $Level_1 > Level_2$ and $Categories_1 \supset Categories_2$ then $AC_1 > AC_2$, AC_1 strictly dominates AC_2 . AC_1 and AC_2 are incomparable if neither $AC_1 \geq AC_2$ nor $AC_2 \geq AC_1$ (neither dominates the other).

This model is based on the concept of a state machine. An initial "secure state" is defined, and each transition in accordance with the rules will result in another secure state. The state of the system is denoted (A, L). A is the set of current accesses under execution in the form of (s, o, p), subject s is exercising privilege p on o. L is the level function for each subject and object: $L : O \cup S \rightarrow AC$. O and S represents all objects and subjects, respectively, while AC is the set of access classes in the system.

For this model there are two MAC state rules and one DAC state rule (based on an access matrix) with three security properties. When making a decision of whether to grant an access request for access (s, o, p), the resulting new state is examined based on the following three properties. Only if the new state satisfies all the properties, the transition is made, and the system is still secure by definition.

The simple security property - This property is designed to prevent subjects from reading objects with access classes dominating or that are incomparable to their own, also known as the "no-read-up property". A new state (A, L) satisfies the simple security property if, for each element (s, o, p) \in A, where p=read or p=write, the following holds:

$L_{(s)} \geq L_{(o)}$ (where $L_{(s)}$ and $L_{(o)}$ are the level functions of the subject and object in question).

The * property - This property was designed to prevent write operations on lower or incomparable objects, preventing unauthorized information-leakage downwards. It is also known as the "no-write-down property". A state satisfies this property only if for each element $(s, o, p) \in A$, where p =append or p =write, the following holds: $L_{(s)} \leq L_{(o)}$.

The discretionary security property - This property ensures some level of DAC, stating that a state is secure if the access requested (s, o, p) is allowed by the system's access matrix.

2.1.4 Role Based Access Control

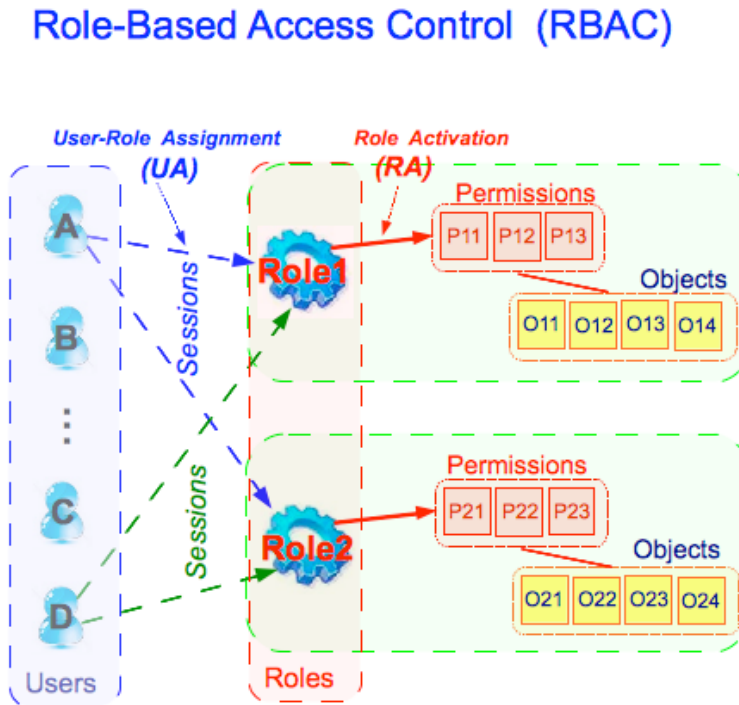


Figure 2.4: This figure illustrates how Role Based Access Control can be modeled. Compared to Figure 2.2 showing DAC, RBAC is somewhat similar. The main difference is how all users are assigned to roles. Roles are assigned to permissions (the same way as DAC does for users), where a permission defines which operations are accessible to which role on each relevant object.

Today's computer systems and programs pose different and new challenges regarding access control. Operating systems and applications for smartphones, laptop computers, servers, mainframes, and different types of applications such as embedded systems, desktop applications, web applications, web services and console applications will require different AC models. The "old" AC models using only DAC, MAC or a combination of the two, no longer fulfill all types of access control requirements. Many of these systems are fairly complex, with numerous subjects and objects, and when using DAC or MAC, there will be a vast amount of overhead when it comes to managing and administering access control. Access control in systems with a high churn rate of subjects or objects are especially difficult and time-consuming to manage. RBAC is a model developed to ease the job of assigning to and revoking from subjects access authorizations to protected objects.

The basic concept is this, as seen in Figure 2.4: Do not assign access rights to subjects, but assign it to roles, and in turn, assign subjects to these roles. This extra step in the link between subjects and access rights, will ease administration of access permission assignments as there are usually less roles than subjects. In a scenario where all managers in a company are to be granted access to certain reports in a system without RBAC, there has to be a change in permissions for each subject representing a manager. If the system implements RBAC, and the managers are assigned to the role "Manager", only one permission has to be changed, that of the role Manager's access to the reports.

The American National Standards Institute (ANSI) has published a standard for RBAC in the document American National Standard 359-2004 ([fITS04]). The standard defines different functional components of an RBAC implementation. Included are the following components: Core RBAC, Hierarchical RBAC and Constrained RBAC. The latter includes the two concepts Static Separation of Duties (SSoD) and Dynamic separation of Duties (DSoD). The standard acknowledges that different systems have different requirements, and that not all components are appropriate in every system. That is the reason why Core RBAC is required, while the other components are optional.

Core RBAC

Core RBAC is the basic component of the RBAC standard, and consists of a defined set of elements and with defined relationships between these. There are six elements that make up the building blocks for Core RBAC. These elements and the most important relationships between them are depicted in Figure 2.5 and explained below:

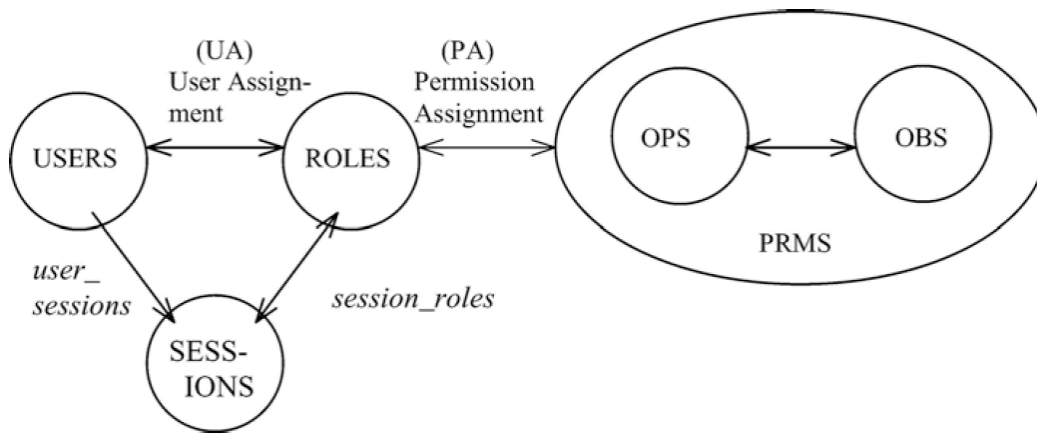


Figure 2.5: This figure illustrates how the different entities defined in the Core RBAC standard relates to each other. The figure is taken from [fITS04]

USERS - The set of users accessing the system.

ROLES - The set of roles that USERS can be assigned to.

OBS - The set of objects that can be accessed by USERS.

OPS - The set of operations that can be performed on OBS.

PRMS - The set of permissions allowing OPS to be performed on OBS, more precisely a set of pairs: $(\text{object} \in \text{OBS}, \text{operation} \in \text{OPS})$.

SESSIONS The set of PRMS available to a USER during a SESSION (Sessions have been introduced to model sessions where a user logs in to a system, and thereby activating a subset of roles assigned to that user).

The standard also defines the relationship between these. The important ones are (as described in Figure 5.1 in [Fer10]):

UA $\subset \text{USERS} \times \text{ROLES}$ - Specifies the ROLES that USERS are allowed to play.

PA $\subset \text{PRMS} \times \text{ROLES}$ - Assigns ROLES the permissions necessary to perform the tasks of that ROLE.

Assigned_prms: $\text{ROLES} \rightarrow 2^{\text{PRMS}}$ - Maps ROLES into a set of PRMS.

Assigned_users: $\text{ROLES} \rightarrow 2^{\text{USERS}}$ - Maps a ROLE into a set of USERS.

Session_users: **SESSIONS** \rightarrow **USERS** - Maps SESSIONS into the corresponding USERS.

Session_roles: **SESSIONS** $\rightarrow 2^{ROLES}$ - Maps a SESSION into a set of ROLES.

Avl_sess_prms: **SESSIONS** $\rightarrow 2^{PMRS}$ - Describes the PRMS available to a user during a SESSION.

Hierarchical RBAC

Hierarchical RBAC is an optional component and describes a hierarchy of roles, introducing a partial order relation on ROLES. This relation is depicted as the Role Hierarchy (RH) in Figure 2.6. This role-hierarchy often aims to reflect the lines of authority and responsibility within the organization using this system. Every role inherits the permissions of all its descendants, reflecting the relationship between superiors and subordinates within an organization. The Role Hierarchy ($RH \subseteq ROLES \times ROLES$) is defined as \geq , and identifies pairs of roles (r_i, r_j) where r_i inherits the permissions of r_j . Given two roles $r_i, r_j \in ROLES$, $r_i \geq r_j$ implies the following: i) $Assigned_prms(r_i) \subseteq Assigned_prms(r_j)$ and ii) $Assigned_users(r_i) \subseteq Assigned_users(r_j)$.

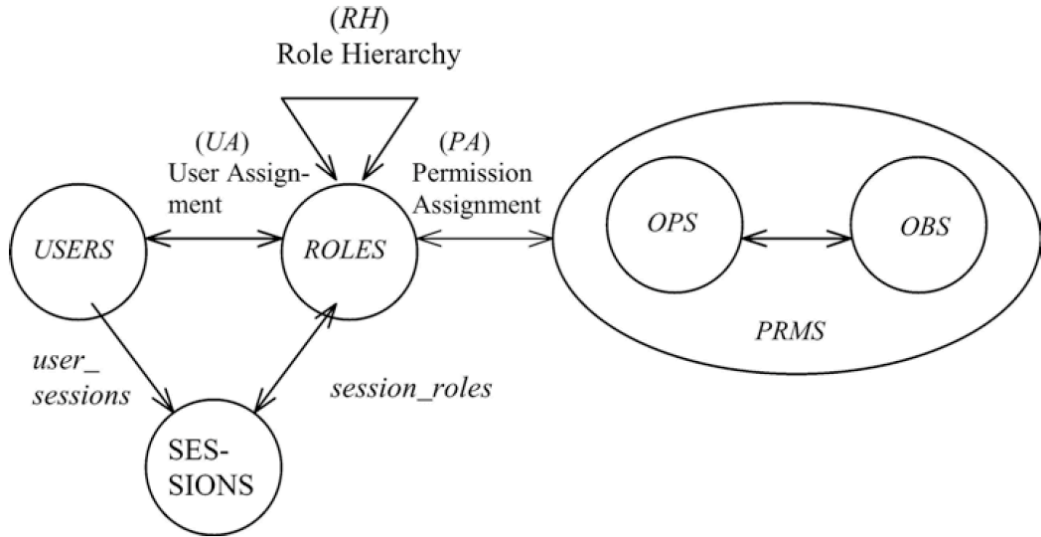


Figure 2.6: This figure shows how Hierarchical RBAC adds to Core RBAC, showing the added relation called "Role Hierarchy", which illustrates that roles are related to each other, forming a hierarchy. The figure is taken from [fITS04]

Constrained RBAC

As discussed in the presentation of basic AC in Section 2.1.1, constraints have been added to the basic AC model, defining conditions under which the permissions are to be granted. Constraints have also been added to the standard as an RBAC component, but it supports only one type of constraint. The standard only supports a Separation of Duties (SoD) constraint, which in effect is constraining the assignment of ROLES to USERS both directly and indirectly, through the assignment of SESSIONS to ROLES (while USERS are assigned to SESSIONS). Rules restricting which roles can be simultaneously assigned to a user can sometimes be necessary.

One example include the two roles of "internal financial auditor" and "financial bookkeeper" within the same company. There should be a company policy stating that an employee is not allowed to audit his or her own work, meaning that no employee can be assigned one of the roles while assigned to the other. If not, an auditor can audit his/her own bookkeeping-work, increasing the chance of undiscovered fraud. This policy should be enforced by the AC system, and the enforcement is ensured by implementing SoD constraints with RBAC. Another example includes the two roles "Doctor" and "Patient", where a person can be both a Doctor and a Patient during a certain time period, just not during the same session (doctors should not examine themselves or write their own prescriptions). This constraint should therefore be tied to SESSIONS. The two examples have created different requirements, therefore two kinds of SoD constraint classes are supported by the standard, Static SoD and Dynamic SoD respectively.

Static Separation of Duties SSoD constraints are statically enforced whenever a user is assigned to a role, depicted in Figure 2.7, and define a mutual exclusion among different roles a user can play. An SSoD constraint is formally a pair (RS, n) , where $RS \subseteq \text{ROLES}$ and n a natural number where $n > 1$. (RS, n) means that a user can play no more than $n - 1$ roles among those in the set RS . If SSoD is implemented together with Hierarchical RBAC the SSoD constraints will propagate along the hierarchy. This means that when evaluating whether a user can safely be assigned a role according to the constraints, both directly assigned roles and inherited roles are considered.

Dynamic Separation of Duties DSoD constraints differ from SSoD in that they are dynamically enforced when a user activates different roles through a session, as depicted in Figure 2.8. DSoD will at runtime prevent a user from activating conflicting roles within the same session. DSoD is also represented by pairs (RS, n) , where $RS \subseteq \text{ROLES}$ and $n > 1$, but

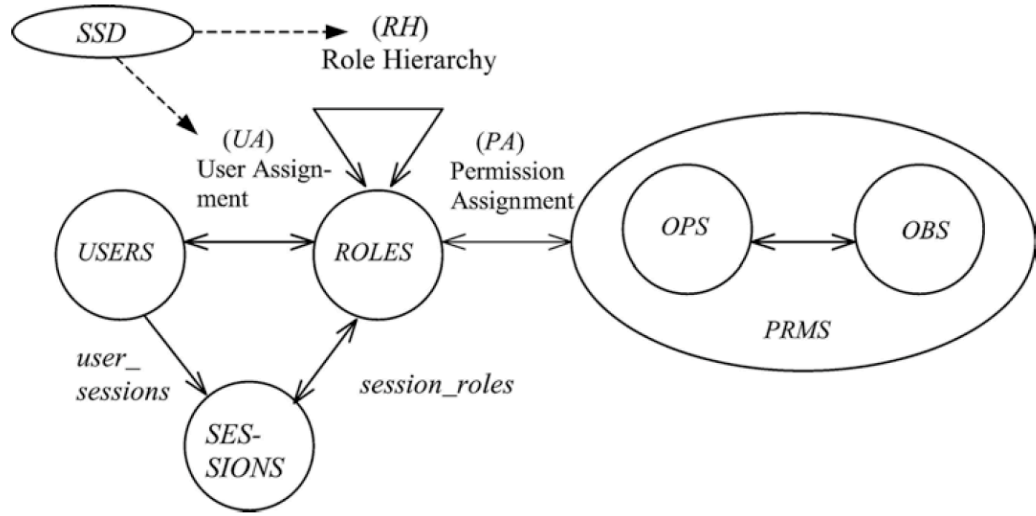


Figure 2.7: This figure shows the addition of Static Separation of Duty (SSoD) constraints on an RBAC implementation with both Core RBAC and Hierarchical RBAC. The SSoD component will control which User Assignments (UA) and Role Hierarchy (RH) assignments are made. The figure is taken from [fITS04]

(RS, n) will in this case mean that a user may not activate n or more roles from RS within the same session.

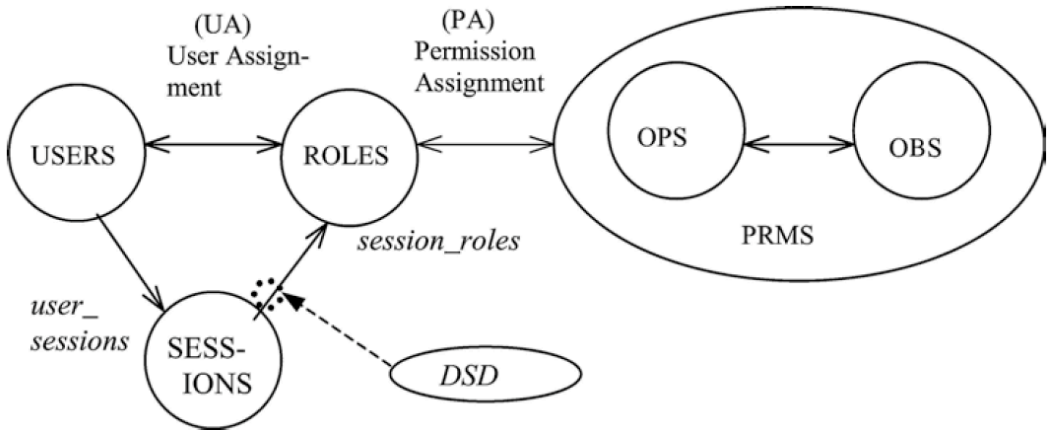


Figure 2.8: This figure shows a Core RBAC implementation with the addition of Dynamic Separation of Duty (DSoD) constraint. The DSoD component will constrain which combinations of roles are allowed to be activated during a session. The figure is taken from [fITS04]

2.1.5 Extensions of RBAC

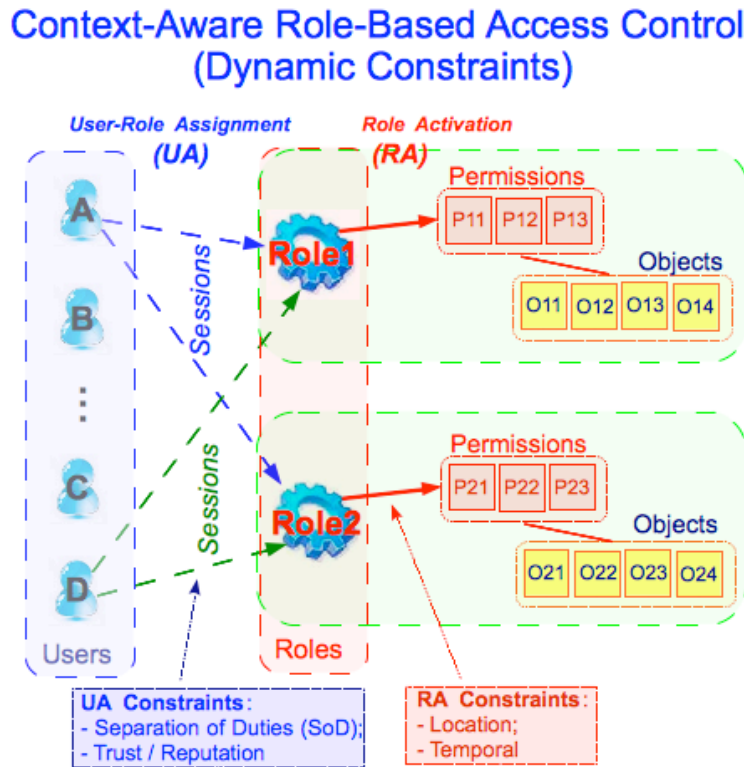


Figure 2.9: This figure is similar to Figure 2.4, with the exception of two types of constraints. User assignment (UA) constraints will control the assignment of users to roles. Role Activation (RA) constraints will control which combination of roles will be activated during a session.

To be able to keep up with new access control requirements some new extensions to the RBAC standard have been proposed. The direction mainly taken by new extensions is that of using context information and different kinds of constraints to make AC decisions. Figure 2.9 shows how context-aware RBAC with constraints can work. For user-role assignment (UA) the standard RBAC component SoD, plus trust/reputation constraints, can be applied. However, it is through the role activation (RA) constraints the most interesting new constraint extensions have been proposed. Temporal, Locational and Conditional constraints are the most interesting new concepts, proposed through Temporal RBAC, Geo-RBAC and Conditional Privacy-aware RBAC, and discussed below.

Temporal RBAC

Temporal RBAC (TRBAC) described in [BBF01], provides the option to enable and disable different roles based on temporal conditions. These conditions can be based on different times a day, weeks, months or any time-based constraint. TRBAC basically supports periodic enabling and disabling of roles, plus takes into consideration temporal dependencies between the enabling and disabling of these roles. As stated in [BBF01]:

”Such dependencies expressed by means of role triggers (active rules that are automatically executed when the specified actions occur) can also be used to constrain the set of roles that a particular user can activate at a given time instant.”

A trigger can be described as a list of preconditions for the activation of the trigger. When a trigger is fired a role might be enabled or disabled at a specific time, now or after a while. It is possible to assign priorities to such enabling and disabling actions, and when conflict arises, the one with the highest priority will be executed.

Geo-RBAC

Spatial-aware RBAC is described in [FV03] and [DBCP05]. Geo-RBAC is a spatially aware RBAC, extending the RBAC standard with spatial roles. It has been developed to enhance access control in location-aware services and mobile applications, and is therefore highly relevant to this master thesis. With Geo-RBAC access control decisions can be made based on rules using spatial conditions, like object locations and user positions. The spatial model adopted by Geo-RBAC is compliant with Open GeoSpatial Consortium (OGC), which is a

”...non-profit, international, voluntary consensus standards organization that is leading the development of standards for geospatial and location based services.”

as stated on their web site [Inc].

Geo-RBAC adds to RBAC the notion of a spatial role. A spatial role is defined (in [DBCP05]) as a

”..geographically bounded organizational function”.

The boundaries for a spatial role is represented by a so-called ”feature”. A feature is an instance of a feature type, which are types of geographical places. Examples of feature types could be a city, a road address or a country.

Examples of features could thus be "London", "1600 Pennsylvania Avenue" or "Norway", correspondingly. The physical location of a user will be directly or indirectly fetched from a location-aware device, like a mobile phone, GPS, etc. In addition to a physical location, a user will also be mapped to a logical location, which will represent the feature where the user is located within. The logical location (feature) will often be computed by mapping from the physical location, and will be the boundary for the area the user has to be within to play the relevant spatial role.

Conditional Privacy-aware RBAC

In [NB], the authors presents Conditional Privacy-aware RBAC (P-RBAC), a model which incorporates an efficient way of expressing privacy policies with RBAC. The Core P-RBAC includes seven entities: Users(U), Roles(R), Data(D), Actions(A), Purposes(P), Obligations(O) and Conditions(C). These are expressed through a customized language which they call LC_0 . A user is a human being, a role is the same as for regular RBAC, data means any information relating to an identifiable user and an action is some function executed for the user. Purposes, conditions and obligations in Core P-RBAC originates from the OECD Guidelines on the Protection of Privacy and Trans-border Flows of Personal Data ([NB]).

Conditional P-RBAC, an addition to Core P-RBAC, allows more powerful and complex condition specification, while still being able to evaluate whether a condition in a permission assignment can be satisfied (a problem essentially the same as the classic NP-complete satisfiability problem, SAT). According to [NB] Conditional P-RBAC is characterized by the addition and definition of:

- "...a more expressive condition language LC_1 and introduce the concept of simple permission assignment set, for which SAT is tractable."
- "...a fully expressive condition language LC_2 and introduce the concept of advanced permission assignment set, for which SAT is theoretically intractable but remains tractable in practice given a reasonable assumption."

In addition to Core P-RBAC and Conditional P-RBAC, they also present the extension Universal P-RBAC, which is Conditional P-RBAC together with the concept of hierarchies.

Through these additions to RBAC the authors are able to present a solution for expressing privacy policies together with RBAC, where more complex conditions can be specified, containing algorithms for detecting conflicts, redundancies, and indeterminism for a set of permission assignments.

2.1.6 Logic in Access Control

Martín Abadi in his paper called "Logic in Access Control" ([Aba03]), sums up the work done with logic in AC. It particularly discusses logically founded languages for representing AC when programming or stating policies. The work described in this paper is much more general than what we are going to work with in this thesis, yet relevant as a way of combining system elements into access control rules describing, in our case, user privacy policies. The user may have many different privacy preferences and may be able to input these to the system through a well-designed GUI, but in the end, the system has to interpret them correctly, consult them when a subject requests a certain permission to a certain object, and ultimately make the right access control decision and enforce it. This thesis is limited to LASNSs, and our main focus is user privacy. Because of the complexity of data in LASNSs, and users' myriad of possible preferences, we need a more complex representation of access control rules than what a regular access control matrix, for instance, can provide, only listing which accesses each subject has on each object. Since we operate only in the context of LASNSs, this simplifies our requirements for a complex logical language. Yet, this paper by Martín Abadi provides us with clues as to how we can represent users' privacy preferences as access control rules in a general format, in the context of a LASNS system. Logic in access control is also discussed in Elisa Bertino, Barbara Catania, Elena Ferrari and Paolo Perlasca's article; "A Logical Framework for Reasoning about Access Control Models" ([BCFP03]). This article, however, proposes a formal framework as a tool for reasoning about and discussing access control models.

2.2 Privacy policies

Privacy policies can be found in different forms and contexts. A company will typically have a privacy policy statement, stating how data of different sensitivity levels and types are to be handled. A web site, like the University of California, Santa Barbara web site: <http://www.ucsb.edu/>, states in their Privacy Notification Statement, [UoC], how the system will handle the information gathered while visitors navigate their web sites. Privacy policies can be enforced by human protocols and/or routines, but also by computer systems and programs. The latter is the most relevant, and there are different ways of representing and enforcing privacy policies in a machine-readable way.

2.2.1 Policies and privacy regulation

To be able to regulate and ensure any degree of privacy it is crucial to establish what privacy entails for the subject in question. The subject can be a natural person or an organization. Some people wish to share more than others, for instance the statement "I do not wish to share pictures with strangers" does not necessarily suit everyone in every situation. A photographer trying to create awareness of her work might not agree, while a private person uploading images intended for view by only her friends will hopefully demand this of her account settings. Organizations own and use information with different sensitivity levels. The policy "This information should not be disclosed to anyone else besides company employees" will suit sensitive internal strategies, while it might be counter-productive for PR-material intended for the public. That is why privacy needs to be defined before it is enforced, and this can be done by creating privacy policies. A policy will usually be relevant for a specific environment, a computer system, an organization or similar. It will mostly consist of different rules governing how information should be handled. Each rule is often concerned with protecting a resource, which can be specified in general terms, like types of documents, everything regarding personal information, all images, etc., or describe a specific entity like a document containing the 2011 company budget. The rules will describe how different subjects are allowed to access the resource. An example company policy might be; "No one, except the people working in the Administration and Human Relations departments, should have access to other employees' salaries." An example of a person's privacy preferences for a SNS, stated as a policy or more specifically a privacy access control rule, might be "I only want to share my interests with my friends".

Privacy policies clearly exist outside computer system environments, and certain challenges arise when they need to be translated into computer commands in order to be enforced. Natural languages, like English, are often not precise enough for a computer to read unambiguously. That is why languages, some based on the Extensible Markup Language (XML) syntax, have been introduced, as a way for system administrators to translate policies stated in human languages, into something a computer can understand. Examples of such languages are P3P, EPAL and XACML which are presented in the following subsections.

2.2.2 P3P

P3P, is the Platform for Privacy Preferences, a standard developed by the World Wide Web Consortium (W3C), described in [Conb]. The work with

P3P was initiated after the first W3C Privacy Workshop ([Conc]) that took place in Dulles/Virginia in 2002, as a way for web sites to communicate their privacy practices and to increase user privacy awareness. The standard describes a way for websites to express their privacy practices in a standard format that is automatically retrievable by user agents. After retrieval, the web site's privacy practices will be easily interpreted by the user agent, and users can be informed of its content in both human- and machine-readable formats. The standard thus allows software to interpret web sites' privacy practices and make automated decisions based on these practices. The user are then able to configure such software to make decisions based on matching the user's privacy preferences against the site's privacy practices where the user is only bothered when conflicts arise.

This standard will hopefully create incentives for web sites to make their privacy practices more open to the public, and therefore push them to behave in a more privacy friendly way. Browsers get the chance to implement smart interfaces to make the users aware of and understand privacy implications of Internet browsing, and thus creating the incentive for web sites to be open and well behaved when it comes to privacy handling.

2.2.3 EPAL

Enterprise Privacy Authorization Language (EPAL) is a formal language for stating enterprise privacy policies, where an Information Technology (IT) system governing data handling can make decisions based on such policies. The data handling is governed according to fine-grained positive and negative authorization rights. It was submitted to W3C by IBM on November 10. in 2003, and is described in detail in [AHK⁺]. In [AHK⁺] you can find their mission statement, and it is as follows:

"The EPAL Working Group exists to develop a interoperability language for the representation of data handling policies and practices within and between privacy-enabled enterprise tools, which serve to

- Enable organizations to be demonstrably compliant with their stated policies.
- Reduce overhead and the cost of configuring and enforcing data handling policies.
- Leverage existing standards and technologies."

A company in need for privacy enforcement will typically add a privacy enforcement system in their software package. Privacy rules are then set

up by one or more privacy administrators, requiring no programming skills. The privacy enforcement system will then enforce these EPAL privacy rules and obligations. EPAL is thus only a language where the privacy rules and obligations are described, and require the implementation of an enforcement system to enforce privacy.

The EPAL language is written in well-formed XML, conforming to XML 1.0, and must be validated by the already defined XML Schema for EPAL, [AHK⁺]. A schema defining the vocabulary for sector-specific privacy policies is also required. Based on this vocabulary, the privacy rules and obligations for the specific sector are defined in XML, where what is allowed and what is denied is formalized. Each policy contains the following elements;

Policy information - Information describing the policy.

EPAL vocabulary reference - Reference to the vocabulary.

Conditions - Zero or more condition elements (conditions can be evaluated to true or false, and a rule can be applied only if all the conditions are evaluated to true).

Rules - Zero or more rule elements which will define the actual privacy rules of the policy which together define the authorizations of the policy.

2.2.4 XACML

eXtensible Access Control Markup Language (XACML) is a standard for stating, interpreting and enforcing privacy policies. It has been developed by the Organization for the Advancement of Structured Information Standards (OASIS), which is a not-for-profit consortium working on making new and better IT-standards ([ftAoSISO]). XACML 2.0 is currently the last published version, and XACML 3.0 is under construction. The original version, XACML 1.0 is described in [ftAoSISO03].

XACML consists of two parts, the declarative access control policy language, and an associated processing model, interpreting such XACML policies. The processing model will be discussed in Section 2.2.5 below ("Privacy policy enforcement architectures"), as XACML share such a processing model with other privacy policy language standards.

XACML privacy policy language

The XACML policy language is written as XML. It is structured into 3 levels of elements:

Policy Set - Can contain other Policy Sets and Policy elements, plus a policy-combining algorithm and optionally a set of obligations. Obligations are descriptions of what must be carried out before or after access is granted (e.g. logging the request).

Policy - Will typically contain one or more Rule elements, a rule-combining algorithm identifier and optionally a set of obligations.

Rule - Consists of a Target, an Effect and Conditions.

Target - Consists of Subjects, Resources and Actions.

A privacy policy is defined with a collection of Rules contained in a Policy element. The XACML rules refers to subjects, resources, etc. that previously have been formally defined in XML for the relevant domain. The following is an example of a Rule defined in XACML taken from [ftAoSISO03]:

```
<Rule RuleId= "urn:oasis:names:tc:xacml:1.0
:example:SimpleRule1" Effect="Permit">
  <Description>
    Any subject with an e-mail name in the medico.com
    domain can perform any action on any resource.
  </Description>
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0
:function:rfc822Name-match">
          <SubjectAttributeDesignator AttributeId="
urn:oasis:names:tc:xacml:1.0:subject:subject-id
" DataType="urn:oasis:names:tc:xacml:1.0
:data-type:rfc822Name" />
          <AttributeValue DataType="
urn:oasis:names:tc:xacml:1.0:data-
type:rfc822Name">medico.com</AttributeValue>
        </SubjectMatch>
      </Subject>
    </Subjects>
    <Resources>
      <AnyResource/>
    </Resources>
```

```
<Actions>
  <AnyAction/>
</Actions>
</Target>
</Rule>
```

The XACML Privacy Policy Profile, as presented in [ftAoSISO10], also has the framework for stating purposes. The holder of information can state the purpose for why that piece of information was gathered, and the entity requesting access must accordingly state the purpose for why the information is requested. This creates the foundation for matching these two purposes such that privacy policies can be enforced under the correct conditions. Access should only be granted if the requester intends to use the information for the same purpose as it was gathered.

GeoXACML is an extension to the OASIS XACML specification, which has been adopted by the OGC and is currently under control of the specification Revision Working Group, before releasing the final GeoXACML specification ([Cona]). This addition will add geo-specific elements to the XACML language, and allow decisions to be based on geographical data.

2.2.5 Privacy policy enforcement architectures

There exists multiple languages for defining privacy policies, where P3P, EPAL and the XACML language have been described above. Only defining such policies by means of different kinds of rules is not enough to enforce such policies, there must exist a system who will interpret and enforce them in the correct manner. [And05] explains how Internet Engineering Task Force (IETF) and International Organization for Standardization (ISO) have defined an abstract model for EPAL and XACML policy enforcement. Figure 2.10 shows how EPAL or XACML policies are interpreted and enforced in a Policy Decision Point (PDP) and Policy Enforcement Point (PEP) respectively. It also illustrates how these two components relates to applications, resources and attributes in a system. As the figure shows, an application will request access to a resource from the PEP. The PEP will, in abstract terms, lie between the entity requesting a resource and the resource itself, controlling access. The request will be forwarded to where the decision will be made, the PDP. The PDP retrieves relevant policies and attributes, and makes a decision based on these. It will then notify the PEP of the decision, and the PEP will grant or not grant access to the resource in question accordingly. In addition to notifying the PEP of whether or not to grant access to a resource, the PDP can forward relevant Obligations found in policies used.

Obligations were presented in the section above, as part of the XACML language, and states what the PEP is obligated to do when a certain policy is consulted. One example is making the PEP log the request.

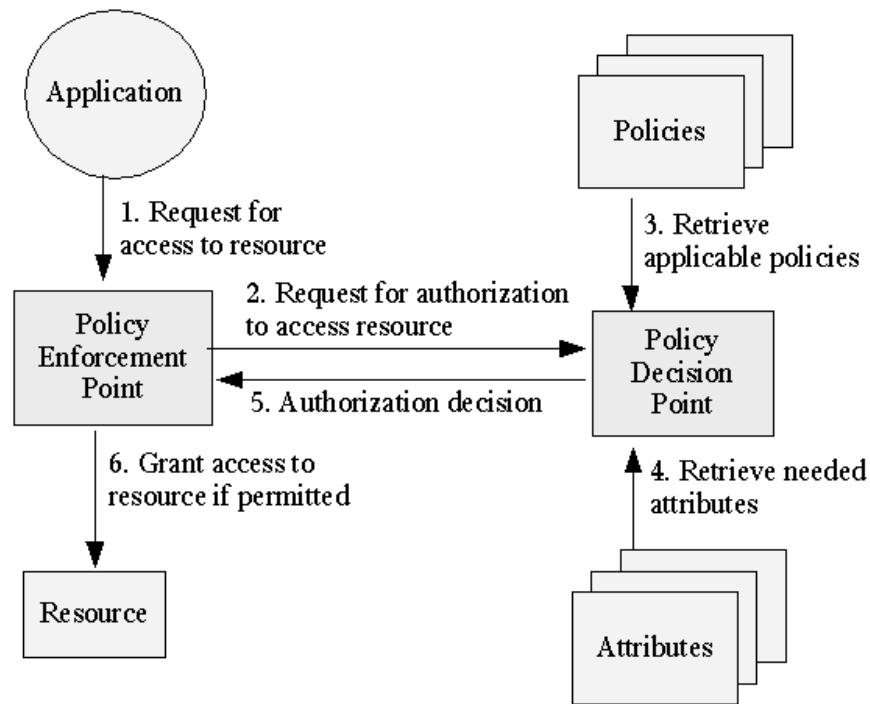


Figure 2.10: This figure illustrates the abstract privacy enforcement model for EPAL and XACML described above. It shows how access control is enforced through the Policy Enforcement Point(PEP) and the Policy Decision Point(PDP), and how they relate to the environment of applications, resources, policies and attributes. The figure is taken from [And05]

2.2.6 Privacy policy referencing

Privacy policies stated in a legal context by companies or through a GUI by a user should not only be enforced in a system, but understood by users and verifiable by outsiders so that they are able to audit how systems handle personal data. This sort of audit and verification can reveal whether these services' technical practices for data handling matches what they claim in their stated privacy policies. A paper by Audun Jøsang, Lothar Fritsch and Tobias Mahler called "Privacy Policies Referencing", [JFM10], propose an

infrastructure for privacy policy referencing. The infrastructure is composed of a technical, a policy, a management and a legal framework for how service providers can specify privacy policies for personal data in their systems, and how technical compliance with the policies can be verified by users or third-party auditors. The paper proposes adding metadata to personal information, with a discussion of the format standardization, the creation, the management and legal aspects of such metadata. This kind of metadata will state how the relevant data should be treated in the system. When data travels the system or collaborating systems, this metadata will ensure it is treated in compliance with the metadata description, and thus the stated privacy policies.

2.3 Emerging privacy issues in the new IT-technologies

This section is loosely based on Chapter 6, Section 4, in the book by Elena Ferrari [Fer10], called "Further research directions in access control". This subchapter discusses four such new research directions, "Trust compilation", "Access control and privacy for mobile users and location-based services", "Going beyond traditional access control and privacy-preserving mechanisms" and "Information accountability". Following is a compilation of the most relevant points for this master thesis.

2.3.1 Online Social Networks access and privacy control requirements

There is no denying that social networking has become a big part of today's information society. SNSs are everywhere, examples include the largest ones; Facebook, LinkedIn, MySpace, countless online dating sites; FriendFinder, eHarmony, etc., the discussion and collaboration SNSs; Orkut, SlideShare, Socialcast, etc. and last but not least the relatively new LASNSs; Foursquare, Gowalla, Brightkite, Google Buzz and Facebook Places.

Driving the development of such services are often the anticipation of new features, cool tools, access to all sorts of data in all kinds of ways. People and enterprises are sharing more and more personal and sensitive data in search for social and organizational benefits or other ways to exploit these services fully. There are benefits to be reaped, but there are also negative consequences when it comes to privacy. Seeing as these services will handle vast amounts of personal and sensitive data they should have correspondingly

strict access control requirements to protect the privacy of its users. When new features drive the development of these services, and there is a race for deploying them as fast as possible, privacy often suffers. The concept of privacy will in some cases only be brought to people's attention when there is a severe problem or breach of trust within one of the services they are currently using. For many users, privacy will not be a deciding or even existent factor in the decision-making process of whether or not to use such services (or which one to choose). Lack of privacy in some shape might even be a requirement for a service or feature to be able to work properly. This leads to the pattern where services are deployed without much concern for privacy, and when such concerns are raised, at a later point in time, privacy will be more or less addressed. This pattern exists because the developers of SNSs do not have adequate incentives for implementing strict access and privacy control, seeing as large portions of the public seems to not understand the risks or do not care about this kind of privacy (or they might not consider it an important enough factor in the choice of whether or not to use different kinds of SNSs). That is why one important further research area is a way to make the public understand privacy and the risks involved when sharing personal or sensitive information through SNSs. In addition to understanding the risks, the public should be able to check the privacy policies of an SNS (for instance by using P3P plus easily understandable GUI interfaces to display a website's privacy policies in a browser). This might create incentives for the developers of SNSs to include more strict access and privacy control in their product.

Mobile and Location Privacy

As mentioned above, LASNSs, like Facebook Places, Foursquare, Gowalla, Brightkite and Google Buzz, have been developed and are now in use. The location-awareness will open up a world of possible features for different scenarios. One example of such a scenario can be a crowded music concert where you could check a service on your phone to see if any of your other friends are nearby, or even at the same concert. Another example is the possibility of a clothing store communicating daily sales to potential customers in the area. Still, mobile and location-aware services pose new challenges when it comes to user privacy and security. You might not want your employer to know that you were at a bar until 4 am last night, might not want certain people to know you were at a hospital, and certainly not your stalker or any untrustworthy person to know that you are home alone at night, or even that your home is empty. These examples certainly raise awareness of the risks involved in using LASNS without any concern for privacy. The nature

of location-awareness causes these SNS systems to face stricter privacy requirements. It should be possible and even easy for any user to tune the account's access control panel to match their preferences for how much, to whom, when, where and in which scenarios they wish to share their location.

2.4 Conclusion

Achieving privacy through access control and privacy policies has been the focus of previous research. Both computer systems in general, and lately SNSs, will in most cases implement this kind of control in some form. Today's systems, like LASNS systems, handle more and more information, mostly personal but also corporate sensitive data, and this has created new requirements for increasingly strict privacy policies with corresponding enforcement architecture, protecting both organizations and citizens. Researches and developers have previously come up with different access control models to control security and privacy, each applicable for their usual environments. The new LASNS systems pose new privacy challenges, which still have not been completely solved, requiring us to use and tailor existing access control models to fit this new environment.

Chapter 3

Scenarios

To be able to analyze privacy and access control in SNSs and LASNSs, we need to come up with privacy requirements for such systems. To be able to come up with perfectly general privacy requirements we need to consider all types of user privacy preferences, in all kinds of roles, in all types of scenarios. Covering all such possible preferences will take an unnecessary amount of time and effort, and might not even be possible. Instead we come up with two scenarios in this chapter, which we will base our privacy control analysis on.

We have to be able to cover many types of users, roles, services and surroundings through these scenarios, to be able to base our privacy requirements on enough actual user preferences. To write these scenarios we will therefore not use the traditional scenario form with a lengthy textual description of a certain user, using a certain service, in certain surroundings, at a certain time, with much unnecessary information. The term "scenario" is in this case defined as a type of user using a specific service or type of service, with different privacy preferences based on different roles, surroundings or other factors. Each scenario will thus contain the user type, the service used and a list of privacy preferences such a user might have.

We will only present two scenarios, confined to two different services. The first one describes a regular person using a SNS and the second a regular person using SNS with location awareness (a LASNS). The preferences are based on the author's experience, imagination and cases where lack of privacy can be thought to have undesirable effects.

3.1 Scenario 1, SNS

User: Regular Facebook user

Service: Facebook (only features and apps developed by Facebook's own developers, except the location-aware Places feature)

Preference list: (based on time, features/resources or connections/relationships):

1. As a regular user, I only want close friends to see my profile information.
2. As a student, I want only the people in the same network as me to be able to find me in searches.
3. As a 50 year old teacher, I want to be invisible to all users except the ones I have myself requested to be friends with.
4. As a regular user, I only want friends and friends of friends to be able to send me messages.
5. As a regular user, I want to show the videos I am tagged in only to friends, except my boss.
6. As a 14 year old girl, I do not want men who are not my friends or friends of friends to contact me.
7. As a parent, I do not want strangers who are not of a similar age as my daughter to be able to contact her.
8. As a teenager, I only want to show my interests to friends with similar interests.
9. As a Norwegian citizen, I wish only to be contacted by strangers who speak one of the same languages as I do.
10. As regular user, I do not want to be contacted by strangers, unless I have been tagged in the same photo as them.
11. As a 25 year old guy, I want to show the photo album from the guy-trip to London only to my close friends, except the close friends who are also in my family.
12. As a business woman, I do not want my colleagues to see the photos I am tagged in.
13. As a business woman, I only want my colleagues and acquaintances to see the places I have checked in to during business hours.
14. As a regular user, I do not want users without a profile picture and who are not my friend to contact me.

15. As a regular user, I wish to show my attending status on events only to my friends and other people who are attending the same events.
16. As a regular user, I wish to hide my photos from strangers.

3.2 Scenario 2, LASNS

User: A Facebook Places user

Service: Facebook Places

Preference list: (based on location or connections/relationships):

1. As a Facebook Places user, I would like the places I check into to only be visible to my friends.
2. As a Facebook Places user, I would like only people in the vicinity of where I am to see my check-in.
3. As a Facebook Places user, I would like only my friends who are in the vicinity to see my location.
4. As a Facebook Places user, I would like to strictly forbid all but a few trusted users to see my location when I am home.
5. As a Facebook Places user, I would like for my colleagues to be able to chat with me only when I am at work, at Bank of America.

Chapter 4

Privacy in Social Network Services and Platforms

This master thesis will focus mainly on LASNSs. However, in order to analyze relevant privacy control and requirements for these kinds of services it is useful to take a look at privacy control and requirements for SNSs in general. It is also valuable for the reader to become familiar with the idea and structure of SNSs. The first section of this chapter will thus be spent on a general overview of SNSs.

Section 4.2 will be an analysis of Facebook. We have had more experience with SNSs in general (seeing as they have been around for longer), and are particularly familiar with Facebook as a specific case of SNS. We have therefore chosen to use it as a case to study user privacy control. It is a useful exercise, applicable to more general cases, to use a specific SNS case to learn its stated privacy policies, come up with our own privacy requirements for that service, and eventually analyze the existing privacy control with regards to those expectations.

Section 4.3 will use the experience gained in the Facebook case study plus work with different SNSs and LASNSs to present the privacy requirements we demand from SNSs in general. These requirements will be used as a foundation to create privacy control requirements for LASNSs in Chapter 5.

4.1 General overview of SNSs

This section gives the reader a useful introduction to SNSs and presents the most common data model for such services.

4.1.1 Introduction to SNSs

A Social Network Service (SNS) is an online service or platform, that is often web based. Its main focus is to build and reflect the social networks or relations that exist in real life through common arenas or interests (friends, colleagues, class mates, fans etc.). The core of the service is the representation of a user, the user's relations to other users and communication among them. SNS's are thus mostly individual-centered, as opposed to online communities, which are group centered. The service is often spiced with additional features, like user profiles, with user-provided information, different kinds of relationships, and additional features for sharing different types of content.

From the beginning, during the nineties, the World Wide Web (WWW) was used for sharing content ([War06]). Throughout that decade many services emerged for people to interact and share content. In the beginning these services were often group centered online communities, but in the late nineties user profiles became a central feature of these web based social networks ([Wikf]). One step further was the ability to compile lists of friends, and the next step was to facilitate more advanced features to find and manage relations. Soon SNSs, as we know them today, became very popular, and now they are a part of most peoples' online communication habits.

Most SNSs have been developed to serve a particular purpose, or to reflect a real life social structure. Many online dating services are basically SNSs, with the goal of finding and communicating with potential romantic partners. The core of Classmates.com is to find and re-connect with old classmates. LinkedIn is a SNS specialized in professional profiles and connections. Facebook, was initially created to link students at Harvard, through friendships, sharing of content, interaction and user-contributed profile information. The newest development is the generation of location-aware SNSs (LASNSs). The explosion of smartphones and app development for these devices have opened up possibilities for using a person's (devices') location in SNSs. Facebook has released Facebook Places where users can share their location with other users through their mobile devices. Other new SNSs have location awareness as their core service, like Foursquare and Gowalla. This master thesis will focus mainly on the location-aware SNSs, but some of the work will be applicable for SNSs in general.

4.1.2 Data models for SNSs

SNS systems contain a lot of data of different types. Relationships, user information, shared entities and everything else that is the core of such a system. All this data has to be structured and stored in the back-end sys-

tem. Many different technologies are used to distribute, access, organize and display the data, yet ultimately all the information and relations have to be stored in a storage medium using storage software to manage it. For this purpose a Database Management System (DBMS) is often used. Facebook, Twitter and Digg (a web based social network for sharing stories and links) uses Cassandra ([Cas]). Cassandra is a

”highly scalable second-generation distributed database”

according to their web site, and a system which is especially good at handling large amounts of data spread out across many physical servers. Linkedin and Myspace uses Aster Data systems as their DBMS. According to the Aster Data web site, [Dat], their solution is a

”An Analytic Platform: MPP row and column database with an integrated analytics engine”.

These kinds of DBMSs hold and manage the huge amounts of data that is aggregated through the use of SNSs. On top of the DBMSs there will be layers of different technologies, to pull up and display the data in the way we are used to seeing it on Facebook, Twitter, Linkedin, etc. Examples of such technologies includes the DBMS query languages, caching systems, business layers and web GUI layers.

4.2 Facebook as an example SNS

We have decided to use Facebook as an example SNS. Facebook is a hugely popular SNS, with millions of users, is well documented, and has support for mobile features; e.g. Facebook app-development for iOS and Android, with location-awareness through Facebook Places among others. Facebook was a huge success from the beginning, first available only to Harvard University students, and eventually all people of 13 years or older ([Gol09]). The core of their service is the ability for a user to connect to others (friends), and create a social network for that user. That social network is called ”The social graph”, and a user has direct and indirect connections to friends, friends of friends, etc. Social graphs are not at all unique for Facebook, most social networking services are probably based on a similar structure, but the Facebook system had other traits which differentiated it from its competitors. In addition to the networking capabilities, their developers created popular applications themselves on top of the social graph. Examples include sharing images and videos, wall-posting and sharing personal information (relationship status, current city, interests, etc). This section will examine Facebook’s end-user

interface, developer platform, user privacy requirements, privacy control and an analysis of this privacy control in light of the requirements.

4.2.1 End-user Interface

The Facebook system uses a web-interface, where a user will log in at `http://www.facebook.com`, and enter the view related to that user. Giving a detailed description of all the parts offered through the Facebook web interface would be too time- and page-consuming and unnecessary. This section will focus only on the parts which will be useful in the work with this thesis. The basic features and resources related to a user will be listed. To protect a user's privacy, it is useful to define which objects is in need of such protection.

Features and Resources

To start the Facebook experience, one has to register as a user, and log into the user view. The user view will provide options to view or change its own profile or related elements, or browse the available data in Facebook (open profiles, search lists, groups, etc.). Facebook offers different features, not only features and services provided by the Facebook developers, but also Facebook-related apps and websites by third-party developers through the Facebook Platform. Examples of features created by the Facebook developers are; uploading and sharing images and videos, creating, inviting to and sharing groups and events, writing on friend's walls and sending Facebook messages.

After some time the user will be part of an elaborate social network, with its own wall and profile information plus relations to other users, images, videos, places, notes, status posts, events, groups and similar, through Facebook's numerous types of relationships. A lot of information will be gathered about the user's life, friendships, images, videos, religious views, interests, visited places, etc, and the user will wish to control who has access to all this information. When speaking of objects or resources to protect, they could include traditional objects like an image, a wall-post, a message or similar. Yet, other types of information, which might need protection, are all types of relationships. A friend-list is a list of relationships for that user, and he or she might not want strangers to see that list, or any relationship contained in the list. Sometimes the object to protect can be a relationship plus a traditional object, like an image plus the connection to the person tagged in it. Thus, when using the term "object" or "resource" when discussing privacy, this could mean all entities containing any type of information about users and/or their relationships.

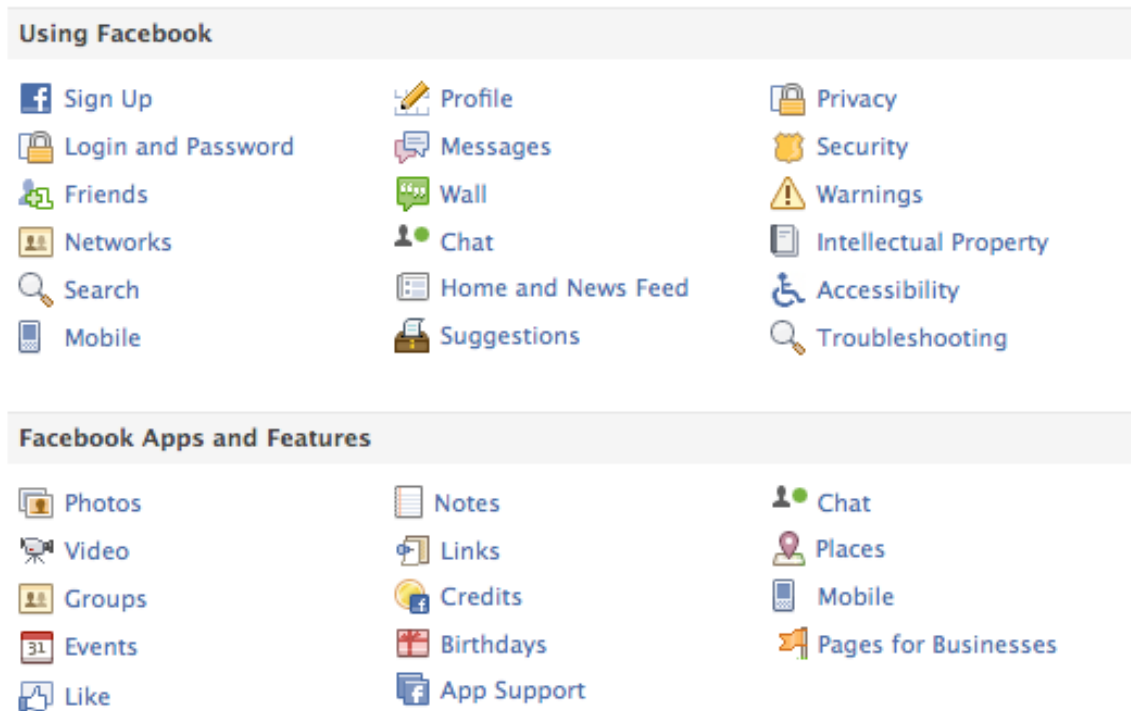


Figure 4.1: This figure shows a screenshot of the main Facebook features and resources, as presented in the Facebook Help Center.

Figure 4.1 gives an overview of the basic usage and features in the Facebook realm. Further inspection will reveal many user resources, either directly, like images in "Photos", or more indirectly, like a friend relationship in "Friends" or any information added to a users "Profile". Following is a break-down of possible objects or resources loosely based on the lists "Using Facebook" and "Facebook Apps and Features" in Figure 4.1.

Friends - A user-user relationship. Either a friendly, romantic or family relationship.

Networks - A user-network relationship.

Search - Visibility in searches.

Profile - Username, Profile picture(s), gender, spoken languages, relationship status, religious and political views, quotes, entertainment preferences (books, movies, television, music and games), sports (sports you play, favorite teams or athletes), activities, interests and all contact information (email, phone number, address, etc.).

Messages - A message from one user to another. The content of the message and the fact that they are messaging each other.

Wall - A wall-post from one user to another.

Home and News Feed - A users status updates, likes and shared links. The users who are tagged in those status updates.

Photos - An image and its relationships to users through tags.

Video - A video and its relationships to users through tags.

Groups - Group membership, group status (regular member or administrator), group activity (wall-posts, image uploads, etc).

Events - Attendance status (attending, maybe attending and not attending), event status (regular or administrator), event activity (wall-post, image uploads, etc.)

Like - The user-like-object relationship.

Notes - The author-note relationship and the content.

Links - The shared link-user relationship.

Birthday - Day and month, plus year.

Chat - The user-user chat relationship, chat log and times chatted.

Places - Places visited and mobile device checked in from (Iphone, Android, etc.).

All these resources are a part of the Facebook features and user experience. For one user, a lot of personal information will be collected when using and being a part of the Facebook network. Different users have different preferences when it comes to privacy and protection of these objects, and can even have different preferences based on different scenarios. That is why Facebook provides a privacy control panel to tune access control for these objects. The question is whether the control options they provide are good enough for this purpose, and will facilitate settings which can match these preferences.

4.2.2 The Facebook Platform

The Facebook platform was released on May 24th in 2007, a good while after thefacebook.com was started by Mark Zuckerberg ([Gol09]). It was not until that point that third party developers were able to utilize all that Facebook information and infrastructure to build creative applications. The Facebook Platform gives developers access to information contained in the Facebook systems through well-defined interfaces, making possible the deployment of numerous application on top of the platform, probably many more than what the Facebook developers would be able to think of and create themselves.

Facebook Platform overview

High-level information regarding the Facebook Platform can be found at <http://developers.facebook.com>, while the details of the platform, the Application Programming Interface (API) and language syntax can be found on their wiki site <http://wiki.developers.facebook.com/>. There seems to be three high-level usages for the platform, each introduced through browsing the "Getting started" section.

Websites - Incorporating Facebook features into your own web site, like including a "Like Button" connected to a user's Facebook account.

Apps on Facebook.com - Creating a Facebook application integrated with the Facebook core.

Mobile apps - Developing mobile apps, for iOS, Android or other mobile Operating Systems (OS) using the Facebook Platform.

To achieve this, the Facebook Platform offers a Software Development Kit (SDK), helpful tools, and most importantly provides the following five core concepts for development on the platform.

Social Plugins The concept of Social Plugins will allow web site developers to add social Facebook components directly into their web page. These components are related to showing likes, comments or shared material from friends of a Facebook authenticated user. Examples of components listed on [Devb] includes; The Like Button, Activity Feed (related to your site), Recommendations, Like Box, Login Button, Registration, Facepile (pictures of friends who have liked your site), Comment and Live Stream.

[Devb] presents how to include these components into your web site only using iframes or Facebook's own markup language Facebook Markup

Language (FBML) together with JavaScript SDK calling Facebook methods.

Graph API The core of the Facebook Platform is the Graph API, an interface for external developers to read from and write to Facebook [Devb]. Through this API one can access the social graph (explained above in subsection 4.2.2), meaning a unique representation of people and everything they are connected to (friends, photos, events, pages, etc) through different types of relationships (friendships, relationships, photo tags, shared content, comments, etc).

Social Channels The concept of Social Channels offers access to different Facebook-powered social channels which can be incorporated into your Website, mobile or Apps on Facebook.com. There are three types of channels; News Feed, Requests and Automatic Channels. At this moment several Automatic Channels are provided; Bookmarks, Notifications, Dashboards, Usage Stories and App Profiles & Searches. A further explanation of the concept of social channels can be found at [Devb].

Authentication The Facebook platform offers a single sign-on mechanism for both Web, mobile, and desktop applications. For an application to be able to access the Graph API in the context of a user, both the user and the app needs to be authenticated and authorized to access the data. When this has been successfully performed, the app is issued a user access token which grants access to a user's information, and to perform actions on behalf of that user.

Open Graph Protocol The Open Graph Protocol is used to integrate a Web page into a social graph, and is compatible with and useful for Web pages representing a real-world entity, a famous person, a food dish, a music album or similar. If a user browses your page, and clicks your Facebook Like Button, the entity will be listed in that person's Likes and Interests section, and a connection is made between your entity and that user. This kind of connection enables you to publish updates to all connected users.

Facebook Platform privacy issues

The Facebook Platform enables third-party developers to gain access to Facebook-related information, through Facebook apps, their web page or a mobile app. The Facebook Privacy Policy [Fac10] begins by informing the users that

"This privacy policy covers all of Facebook. It does not, however, apply to entities that Facebook does not own or control, such as applications and websites using Platform".

Further, section 4 called "Information You Share With Third Parties: Facebook Platform" states the following:

"As mentioned above, we do not own or operate the applications or websites that use Facebook Platform. That means that when you use those applications and websites you are making your Facebook information available to someone other than Facebook. Prior to allowing them to access any information about you, we require them to agree to terms that limit their use of your information (which you can read about in Section 9 of our Statement of Rights and Responsibilities) and we use technical measures to ensure that they only obtain authorized information. To learn more about Platform, visit our About Platform page."

It seems that while Facebook has taken some measures to protect users of Facebook Platform services, like making third-party developers formally agree to limiting terms and ensuring that they only obtain information which somehow have ended up being labeled "authorized", they have no actual control over what happens with a user's information after it has been provided to the third-party developer systems. It is up to the developer to state which information categories an app requires access to, even if such access might not really be necessary. The developer of a mobile Facebook app might ask the user for permission to use the current location of the mobile device, and the user might press "Agree" without much thought. The app server might store information, without telling the user, and even if the user presses "Agree" to a long textual list of terms to get access to the app, it is a common fact that not all users tend to read the agreements carefully. These scenarios can pose some harmless or more serious privacy issues. Opening up the Facebook system to third party developers, gives away some of the back-end privacy control, and might result in vulnerable users taking an app into use without agreeing with or even knowledge of the privacy implications.

4.2.3 Facebook user privacy requirements

To be able to analyze existing privacy, and further to propose improved privacy control for LASNSs, we need to take a look at existing privacy policies plus privacy and access control requirements to compare against the actual implementation. We will first examine Facebook's own privacy policies. The

analysis of whether the relevant policies and other privacy requirements have been fulfilled will be aimed towards Facebook's user privacy and access control. As explained in Chapter 3 "Scenarios", it is impractical to consider all scenarios, for all user types and roles, in all surroundings. We will therefore base these requirements, in which we will evaluate Facebook privacy control against, on Scenario 1. Scenario 1 is based on a regular Facebook user, and contains a variety of roles, surroundings, and possible preferences.

After the presentation of Facebook privacy policies, we will establish our own privacy and access control requirements in which to compare the actual implementation against, in Section 4.2.5. There are two general high-level requirements when it comes to user privacy control. The first one is privacy awareness. The user should be aware of what data is stored, which personal data needs to be protected and how to protect it. The second is actual privacy control. The system should offer a reasonably extensive control menu or control panel to limit access to data according to their preferences. User privacy preferences will vary from user to user and scenario to scenario, and there are no universal conclusions as to what one should expect from the Facebook system in this regard. The privacy control requirements we set forth will thus be based on possible user privacy preferences, as well as the author's own opinion of the level of privacy a user can reasonably expect.

The Facebook privacy policy

One can read the current Facebook privacy policy at [Fac10]. It consists of a set of privacy policies regarding different types of information with regards to different receiving subjects. It consist of the following sections:

1. Introduction
2. Information We Receive
3. Sharing information on Facebook
4. Information You Share With Third Parties
5. How We Use Your Information
6. How We Share Information
7. How You Can Change or Remove Information
8. How We Protect Information
9. Other Terms

Each section describes in quite detail what kind of information it is regarding, and often what kind of control a user has of such sharing. They do not state any broad privacy policies, but addresses each (group of) information-subject relations separately. The first sections (except the introduction) describes what kind of information a user will directly and indirectly share with the Facebook system, everyone in general and third parties, respectively. The two following sections describe how Facebook uses and shares your information. "How You Can Change or Remove Information" explains exactly that, while "How We protect Information" explains which steps Facebook has taken to protect personal information. That section also contains a small subsection describing "Risks inherent in sharing information", which seems to in fact be a section which releases Facebook from any responsibilities for security or privacy breaches in their systems, by informing that

"...no security measures are perfect or impenetrable."

The privacy policy has however changed multiple times since the beginning of Facebook. This blog post from 2010, "Facebook's Eroding Privacy Policy: A Timeline" [Dee10], describes how the company's stated privacy policy has changed from 2005 to 2010, in the direction of eroding privacy and less user privacy control. When searching the web for cites of the Facebook privacy policy one can find different citations, which can no longer be found in the current policies. The article at Web Identity [Ide08] writes in 2008 that

"Facebook's privacy policy begins with a statement of Facebook's 'core principles'".

These two core principles are further listed:

1. "You should have control over your personal information."
2. "You should have access to the information others want to share."

This allegedly stated core privacy policy can not be found in the current privacy policy, which can lead us to believe that it was removed as the privacy policy has evolved and might no longer suit the first principle in all cases.

Privacy awareness requirements

Users can not make enlightened privacy control decisions if they are not aware of which objects they need to protect. As Facebook users spend time on Facebook, a lot of personal data will be gathered. Initially, when new

to Facebook, the user should be informed by the Facebook system what information they should protect, and how to do it. A profile should have the recommended privacy settings for a regular user as a default, and the user should be informed of the potential dangers of exposing too much personal information and how to set up adequate access control for this information.

In addition, there should be clear guidelines and rules for which data third-party developers will get access to, and how they handle it (how long it is stored, who it is disclosed to, etc.). In addition to the rules, the user needs to be informed of apps' data gathering and handling before taking such services into use.

As diving into the issue of privacy awareness would greatly increase the scope of this thesis, and demand much more time than what is set for such a thesis, this issue will not be discussed further.

Privacy control requirements

The 16 privacy preferences stated in Scenario 1, can be transformed into more general privacy and access control requirements for Facebook. As a user I wish to tune the access rights for certain subjects to all my personal information objects based on:

1. **Type of object** - Set different access control rules for different types of objects, like images, tags and videos, friend-list, profile information, etc.
2. **Subjects** - Allow or disallow certain users access to an object, with the ability to except others. Capabilities to block certain users from all contact.
3. **Relationships** - Allow or disallow access to an object based on the subjects relationship with the object owner. Examples could include romantic relationships, friendships, Facebook history (chat, wall posting, etc.) and family relationship.
4. **Roles** - Allow or disallow access to an object based on the role of the subject in relation to the owner. Example roles could include colleagues, parents or system administrator.
5. **My attributes** - Allow or disallow access to an object based on my profile attributes. For instance: "I wish to show my interests only to people with similar interests".

6. **Subject's attributes** - Allow or disallow access to an object based on the subject's attributes. For instance, allowing access only to subjects of a certain age, speaking a particular language, of a certain gender, part of a specific network, with the existence of a profile picture.
7. **External attributes** - Allow or disallow access to an object based on external factors, like time of day, month or year.

4.2.4 Facebook user privacy control

Facebook accounts have a control panel to tune account privacy settings to match the user's privacy preferences. The purpose of the Facebook privacy control panel is to control which people have which types of access to all these objects. The shape of the GUI for privacy control is thus often a list of textual descriptions of the object and type of access in question, plus the corresponding user choice for which subjects are allowed to perform such access on the object. Each row in the panel can be seen as an access control rule with the following format: "Subject S is allowed to perform access A on object O" , where object and access type are pre-defined by Facebook, while only the subject is user defined. Figure 4.2 is a screenshot of the main entrance view of the privacy control panels:

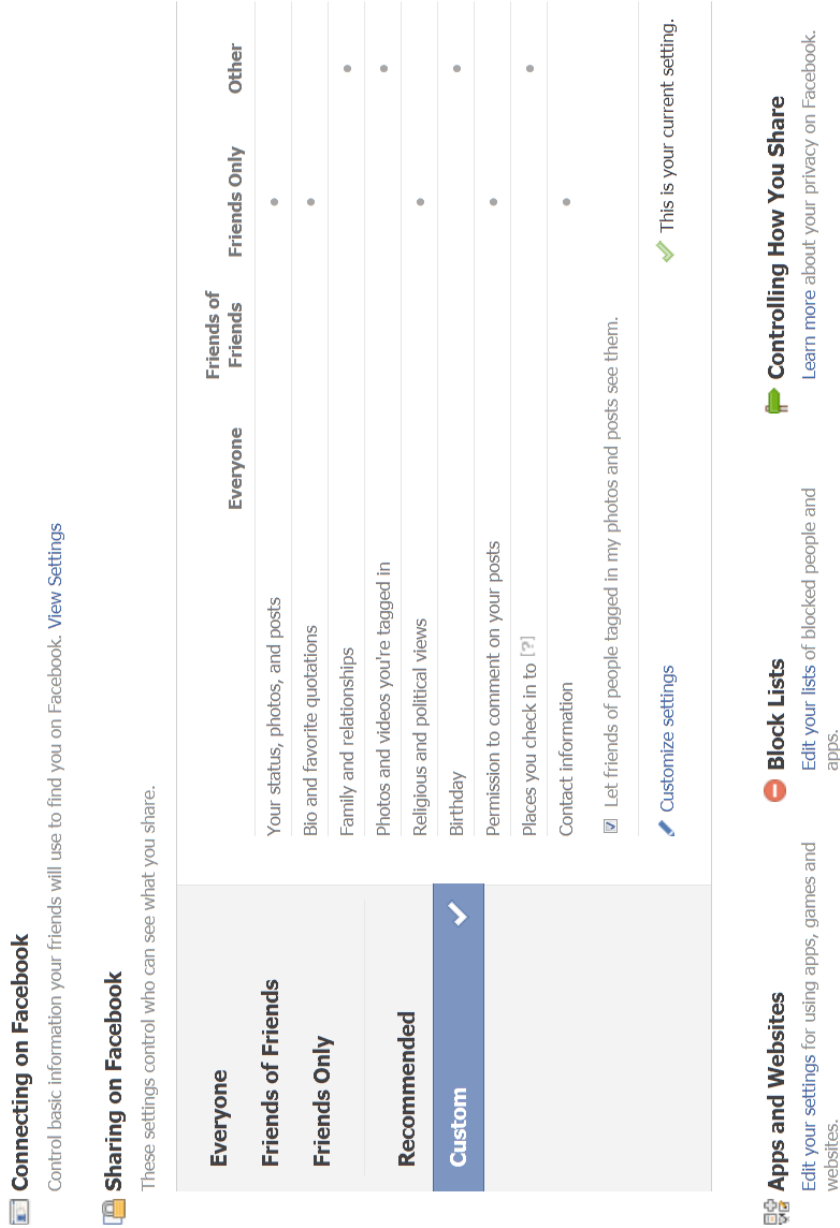


Figure 4.2: This figure is a screenshot of the main Facebook privacy panel, where current privacy settings are displayed, plus links to the sites where the privacy can be changed.

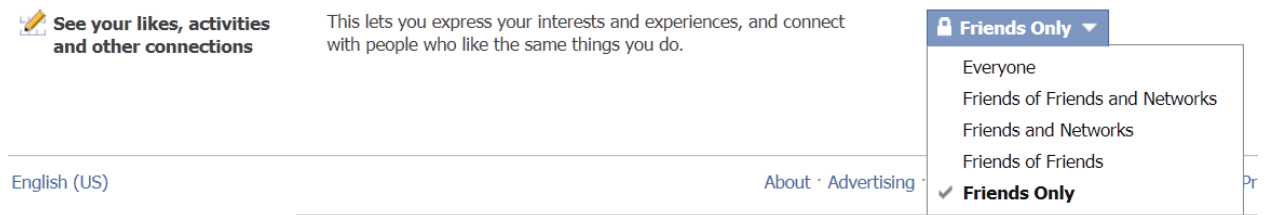


Figure 4.3: This figure shows a screenshot of an example Facebook privacy control rule, with a pre-defined object with a certain type of access, and the user choice for which subjects should be allowed access to that specific access type-object combination.

Figure 4.3 shows an example access control rule from Facebook's privacy control panel. As can be seen in the figure, the subject being granted access to the object can be chosen from a drop-down list. This list will differ slightly with different combinations of object and access type, but the following list includes all the possible drop-down subject choices Facebook provides.

- Everyone
- Friends and Networks
- Friends of Friends
- Friends Only
- Customize

Through the Customize option the user is redirected to a special panel with two specification options; "Make this visible to..." and "Hide this from...". As can be seen in Figure 4.4, the user is able to make the object visible to "Friends of Friends", "Friends Only", "Only me" or "Specific People..." which is specified as a list of Facebook usernames or self-defined groups. The user can specify who the object will be hidden from as well, and this is also stated as a list of Facebook usernames or self-defined groups. A self-defined group can be created in the "Edit Friends" menu, and is a list of friends, with a self-defined group name.

The granularity of subject specification is relatively coarse. Only Friends, Networks, Individuals or self-defined groups can be granted or denied access. The "Customize" choice creates more fine-grained privacy control possible, through the use of self-defined groups, but this option is not available for all access rules in the category "Connecting on Facebook". One example of a rule without this choice is the ability to "Send you message", which can only

Custom Privacy

✓ **Make this visible to** _____

These people: Friends Only

And these networks: Friends of Friends of Science & Technology

Friends Only

Specific People...

Only Me

Only friends can see this.

✗ **Hide this from** _____

These people: _____

Save Setting **Cancel**

Figure 4.4: This figure is a screenshot of the interface for customization of which subjects can access an object in certain ways.

be specified for "Everyone", "Friends of Friends" and "Friends Only", which means that you can not block a group of users from sending you messages, unless you block all of them through the "Block List" interface, blocking all Facebook interaction.

As stated before, Facebook has defined which objects and which types of access a user can control. The control menu allows a user to tune the privacy for four basic information object types; "Connecting on Facebook", "Sharing on Facebook", "Apps and Websites" and "Block Lists". Each of these types contain relevant objects, for instance "Sharing on Facebook" contains a list of your uploaded images, relationships and your birthday among others, while "Connecting on Facebook" will let you tune which people are allowed send you messages or see your friend-list.

Connecting on Facebook This panel, shown in Figure 4.5, will control the information people use to find you on Facebook. Your name, profile picture, gender and networks will be visible to everyone by default, and this can not be changed. You can however control who can find you in searches, send you friend requests and messages, see information like your















 Search for you on Facebook	This lets friends and family find you in Facebook search results. Set this to Everyone or you could miss friend requests.	 Everyone ▼
 Send you friend requests	This lets you receive friend requests. Set this to Everyone to avoid missing out on chances to connect with people you know.	 Everyone ▼
 Send you messages	This helps you make sure you know people before adding them as friends.	 Everyone ▼
 See your friend list	This lets you connect with people based on friends you have in common. Your friend list is always available to applications and your connections to friends may be visible elsewhere.	 Friends Only ▼
 See your education and work	This helps you connect with classmates and colleagues, and discover new professional opportunities.	 Friends Only ▼
 See your current city and hometown	This helps you get in touch with neighbors and old friends. Note: you can separately control how you share places you check in to on the main privacy page.	 Friends Only ▼
 See your likes, activities and other connections	This lets you express your interests and experiences, and connect with people who like the same things you do.	 Friends Only ▼

Figure 4.5: This figure is a screenshot of the Facebook interface for privacy settings controlling the information people use to find a you as a user on Facebook.

friend-list, education, work, current city, hometown, likes, activities and other connections.

Sharing on Facebook The panel used for customizing the privacy of information shared about you, contains three categories of objects to be protected. "Things I share", "Things others share" (which are related to you in some way) and "Contact information". Following are listed all the objects contained within these three categories. They are listed with the object name followed by the type of protection Facebook offers for that object. The type "Subjects" indicates that the user can choose from a drop-down menu one of the subjects from the subject-list presented above. The type "Enable/Disable" indicates that access to the object can be enabled or disabled by the user. "Things I share" includes these objects:

- Posts by me - Subjects
- Family - Subjects
- Relationships - Subjects

- Interested in - Subjects
- Bio and favorite quotations - Subjects
- Website - Subjects
- Religious and political views - Subjects
- Birthday - Subjects
- Places I check in to - Subjects
- Include me in "People Here Now" after I check in - Enable/Disable

"Things others share" includes these objects:

- Photos and videos I'm tagged in - Subjects
- Can comment on posts (Includes status updates, friends' Wall posts, and photos) - Subjects
- Suggest photos of me to friends - Enable/Disable
- Friends can post on my Wall - Enable/Disable
- Can see Wall posts by friends - Subjects
- Friends can check me in to Places - Enable/Disable

"Contact information" includes these objects:

- Address - Subjects
- IM screen name - Subjects
- <email addresses> - Subjects

Apps and Websites When a user connects to an app or website, the app or website will automatically have access to username, profile picture, gender, networks, the user's friend-list and any information the user has chosen to share with everyone. For everything else, the user has to specify privacy preferences related to app and website use and interactions, and for every app, define which objects the user wishes to share with that app. The menu for general privacy preferences for app and website use is depicted in Figure 4.6, while the panel for controlling the privacy settings for one app in particular is shown in Figure 4.7:



Figure 4.6: This figure is a screenshot the Facebook interface for general privacy towards apps and websites.

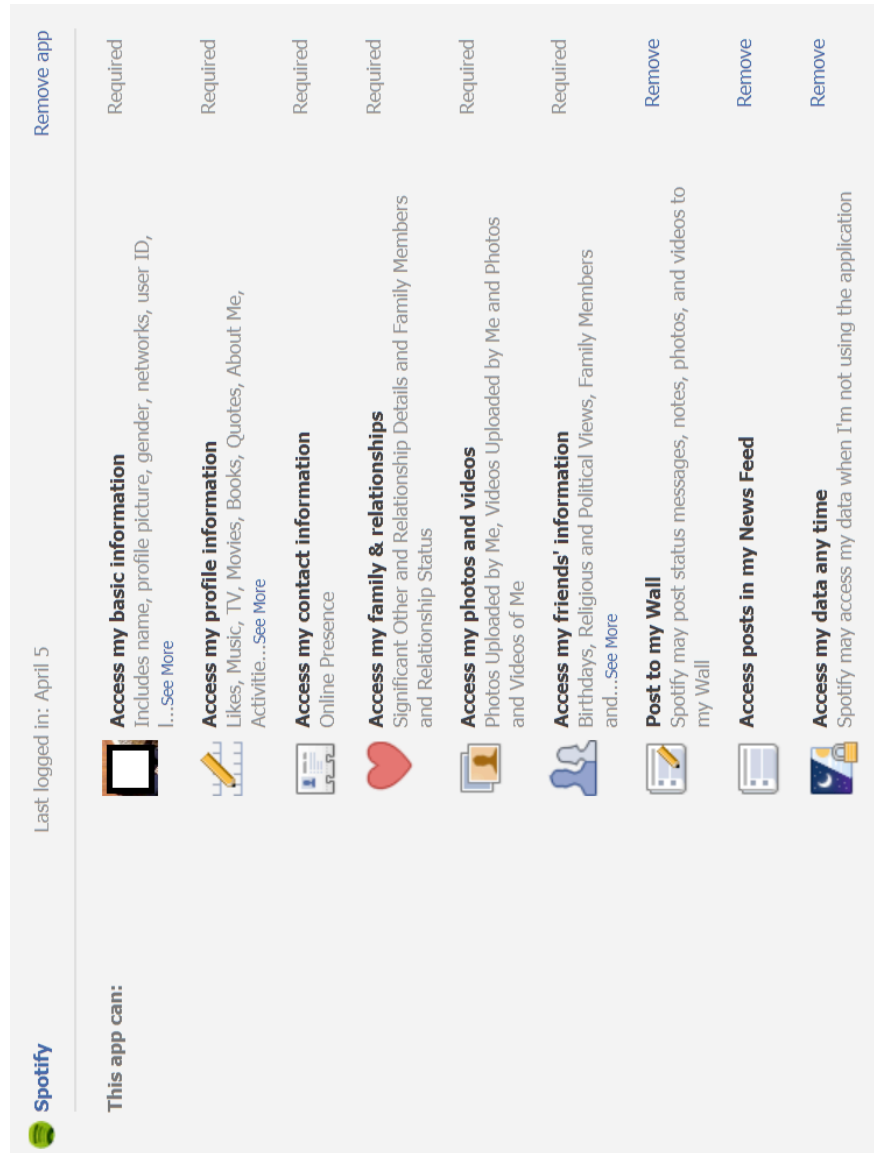


Figure 4.7: This figure is a screenshot of an example Facebook interface where users can tune their privacy towards specific apps, in this case the Spotify Facebook app.

Facebook has grouped different information objects into categories (for instance "Access my basic information"), and defined the access type and content in the textual description. As can be seen in Figure 4.7, the app developer will mark access to different information categories as required or not. The user can only remove access to the categories which are not required by the app. One example where a category is strictly required for the app to work properly could be an app which uses family and relationship information to create a family tree for the user in question, and thus requires access to the family and relationship information category. One example where an information category is required by the app, for no apparent reason, is the required access to a user's photos and videos by the Spotify app, seeing as Spotify is a music streaming program (as seen in Figure 4.7).

Block Lists This panel, shown in Figure 4.8, allows Facebook users to administrate blocked people, apps and the interaction from these. From this menu a user is able to block interaction from specific strangers, block app and event invites from specific friends and block certain apps from contacting you or using your information. All blocked strangers, friends or apps have to be uniquely specified by email, username, appname or similar, thus Facebook does not provide the option to block entities based on certain criteria (e.g. a group membership).

4.2.5 Facebook privacy control analysis

We will analyze the Facebook privacy control presented in the section above with regards to the current Facebook privacy control in Section 4.2.3 and our Facebook privacy requirements outlined in Section 4.2.3, in turn.

Facebook privacy policy It seems that Facebook has, instead of keeping a general privacy policy saying for instance that a user should be able to control what is shared about that user, tailored their privacy policy statement to address information elements separately to suit all the cases where the user can not control what is shared. It is also a place to release all their responsibilities for security and privacy, and place it on the user. The following citation from their privacy policy section about how they protect a user's information embodies this point:

"Although we allow you to set privacy options that limit access to your information, please be aware that no security measures are perfect or impenetrable. We cannot control the actions of other

Block users

Once you block someone, that person can no longer be your friend on Facebook or interact with you (except within applications and games you both use).

Name:

Block This User

Email:

Block This User

You haven't added anyone to your block list.

Block app invites

Once you block app invites from someone, you'll automatically ignore future app requests from that friend. To block invites from a specific friend, click the "Ignore All Invites From This Friend" link under your latest request.

Block invites from:

You haven't blocked invites from anyone.

Block event invites

Once you block event invites from someone, you'll automatically ignore future event requests from that friend.

Block invites from:

You haven't blocked event invites from anyone.

Blocked apps

Once you block an app, it can no longer contact you or use your information. To block an app, go to the app's Facebook Page and click the "Block App" link on the left.

Friends For Sale

Unblock

Slide FunSpace

Unblock

Texas HoldEm Poker

Unblock

Zombies

Unblock

Mob Wars

Unblock

Figure 4.8: This figure is a screenshot of the Facebook interface for block list administration.

users with whom you share your information. We cannot guarantee that only authorized persons will view your information. We cannot ensure that information you share on Facebook will not become publicly available. We are not responsible for third party circumvention of any privacy settings or security measures on Facebook. You can reduce these risks by using common sense security practices such as choosing a strong password, using different passwords for different services, and using up to date antivirus software.”

This means that Facebook promises little in their privacy policy statement, opening for less privacy in their systems, and less user control. Because of the lenient privacy policy, one can say that the implemented privacy control, described in Section 4.2.4 is very compliant with their privacy policy statement.

Our privacy requirements Analyzing the Facebook user privacy control with regards to our privacy requirements in Section 4.2.3 will reveal weaker privacy control. In that section we presented seven requirements for how a user should be able to control personal information. Out of the seven requirements, Facebook's current privacy control only fully meet the first two requirements. Requirement 1, setting different access control rules for different types of objects, has been met through the function where users can, for Facebook's previously defined object-access type couples, set which people can perform that type of access on the objects of that type. Requirement 2, regarding the granularity for which subjects can be allowed access to an object, stating that one should be able to allow or disallow certain users access to an object, is also met. This is not met through the drop-down menu choices "Everyone", "Friends and Networks", "Friends of Friends", "Friends Only", but through the "Customize" option where one can specify certain individuals or group of individuals one wishes to allow or hide the object from.

The rest of the requirements, 3 through 7, are not fully met with current privacy control. Requirement 3, allowing or disallowing access to an object based on the relationship between the relevant user and the subject requesting the access, is only partially met. The previously mentioned access control rule subject drop-down menu choices: "Friends and Networks", "Friends of Friends" and "Friends Only" will grant subjects access to the object based on the relationship between the user and subject. Still, the choices are, in our opinion, too limited. The relationships "Friendship" and "in the same Network" applies to such a coarse-grained grouping of people. Experience using Facebook tells us that most users have hundreds of friends, and these often include close friend, childhood friends, co-workers, business associates, classmates, family, acquaintances and other people one has encountered in all sorts of settings. Calling all these relationships "friendships" can seem very coarse-grained and is not always adequate for making rules that reflect a person's privacy preferences. Requirement 4 is closely tied to the demand for more fine-grained relationships, and is not met with current privacy control. Being able to state access control rules based on which role a subject has towards the relevant user, is not possible, and like requirement 3 calls for a more fine-grained grouping of a user's "Friendships". The three last requirements, being able to set access rules based on the user's, the subject's or external attributes, are by no means met with the existing control panel. The control panel fully lacks the option to set conditions under which access to a specified object should be granted to the specified subject(s). To be able to fulfill requirement 5, 6 and 7, such options to specify conditions must be implemented.

It becomes clear that seen in the light of our privacy control requirements, Facebook's current privacy control panel does not provide sufficiently fine-grained user privacy control through access control of personal information. The panel lacks sufficiently fine grained subject specification and grouping of "Friends", plus does not provide the option to set conditions under which the access should be given.

4.3 Privacy requirements for SNSs in general

Through the case study of the privacy control panel in Facebook, we have thus come up with six general requirements for end-user privacy control:

1. **Type of object** - Set different access control rules for different types of objects, like Images, tags and videos, Friend-list, profile information, etc.
2. **Subjects** - Allow or disallow certain users access to an object, with the ability to except others. Capabilities to block certain users from all contact.
3. **Relationships/Roles** - Allow or disallow access to an object based on the subjects relationship with or role towards the owner.
4. **Owner attributes** - Allow or disallow access to an object based on owner profile attributes. For instance: "I wish to show my interests only to people with similar interests".
5. **Subject's attributes** - Allow or disallow access to an object based on the subjects attributes. For instance, allowing access only to subjects of a certain age, speaking a particular language, of a certain gender, part of a specific network, with the existence of a profile picture.
6. **External attributes** - Allow or disallow access to an object based on external factors, like time of day, month or year.

4.4 Conclusion

It is commonly known that SNSs face privacy issues as they often deal with huge amounts of personal data. We can thus demand more strict user privacy control from such services. We have used Facebook as an example, analyzing current privacy control against their own policy and our requirements based on Scenario 1. This has shown that not even Facebook, one of the biggest

and most commonly used SNSs, which is based on people sharing their personal lives with others, is able to meet our demands for user privacy control. This analysis has provided us with an expected level of granularity for subject specification and the realization of the need for users to be able to set conditions under which access should be granted, through a set of privacy requirements for SNSs in general. This information will be used to improve privacy control for LASNSs.

Chapter 5

Privacy in Location-Aware Social Networks

As the field of IT progresses and new devices are introduced to the market, developers come up with creative ways to use all the new hardware and software to make innovative features and new areas of use, some which we could not have imagined we would need or even want.

Some older phones have been trackable by triangulation (e.g. Buddy) or simple Global Positioning System (GPS) receivers used for tracking of emergency calls ([Shi10]). Yet, we will narrow the discussion of LASNSs to primarily include **smartphones** as possible devices for such services. There are two reasons for this; 1) smartphones often have the required positioning hardware and software making location awareness accessible to additional software (apps and programs), and 2) as this master thesis focus on location awareness in SNSs, we need to limit the device type to include only those regularly used for social networking, thus including most smartphones.

The volume of smartphones being developed, produced and sold is growing, and more and more people carry a smartphone around wherever they go. PCmag.com defines a smartphone in the following way:

Definition A cellular telephone with built-in applications and Internet access. Smartphones provide digital voice service as well as text messaging, e-mail, Web browsing, still and video cameras, MP3 player, video viewing and often video calling. In addition to their built-in functions, smartphones can run myriad applications, turning the once single-minded cellphone in to a mobile computer.

Most of these new smartphones contain embedded GPS receivers and software taking advantage of the received data ([Shi10]). Locational data from a smartphone can be used in a variety of ways to provide the user with

new functionality and features. Tracking a user on jogging trips, to measure progress in physical health. An app which maps out the best areas for picking Chanterelle mushrooms in autumn and records the places you visit and spend time. An app which based on your location displays the weather forecast for that place or recommends the best restaurants in the area. A program that checks which of your friends are in the area, and provides a way for you to contact them. These are all examples of possible apps for iPhones, Android phones, and phones with similar capabilities.

As users usually carry their smartphones around with them, the locational data will often track the user's geographical movements, thus becoming the source of sensitive personal data. That is why, even if the potential benefits and useful areas are huge, such information pose threats to user privacy. An Android phone user can download a Location-aware (LA) app from Android Market, a place where anyone can upload their code. How can that user protect its (the phone's) geographical movements from unlawful disclosure to other users, the developers of the app or other interested parties, and why can this be undesirable?

This chapter will first investigate the nuts and bolts of location awareness in mobile devices in general, then address how such a feature can be used in company with SNSs. Following this, we will look into examples of LASNSs. All the mentioned subsections will contain a part where privacy in relation to the relevant subject is discussed.

5.1 Location-aware mobile services

LA mobile services are services provided through a mobile device, in this case a smartphone, which uses the device's location to provide some useful information or a feature. To be able to fully understand LASNSs, we will first dig in to the underlying technology and privacy implications of mobile location awareness in general.

5.1.1 Underlying technology overview

For mobile producers or third party developers to be able to develop and provide LA services for a device, it requires the underlying technologies to support such services. A device has to contain the two following (abstract) components:

1. A GPS receiver (or components facilitating other positioning techniques like Wi-Fi or cellular tower triangulation information).

2. A processing component.

According to Wikipedia, ([Wikd]) GPS is

”...a space-based global navigation satellite system (GNSS) that provides location and time information in all weather and at all times and anywhere on or near the Earth when and where there is an unobstructed line of sight to four or more GPS satellites. It is maintained by the United States government and is freely accessible by anyone with a GPS receiver.”

As mentioned before, most new smartphones contain a GPS receiver, and can thus, when in line of sight to four or more of these satellites, receive sufficient information to calculate their geographical position. The three basic components of GPS are: absolute location, relative movement and time transfer. Absolute location is the most relevant one for LASNSs and this master thesis. If a smartphone can not connect to enough satellites, most phones today can derive their location, although less accurately, by using information from cell towers or a Wireless Fidelity (Wi-Fi) connection. This is at least true for both iPhone OS (iOS) and the Android platform, as discussed below. These phones thus have three location-aware hardware options:

1. GPS receiver
2. Wi-Fi connections
3. Cell tower triangulation

Whether the phone has a GPS receiver, Wi-Fi or cell tower information, the device has to contain logic which will, based on some or all of this information, calculate or process the device’s location as accurately as possible. If the location is to be made accessible to third party developers, this hardware or software also has to provide a standard format and interface for these application to get the location and to use it in their services. This processing component will thus have to first figure out the most accurate source of location information at the time (GPS, cell towers or Wi-Fi), then calculate the location, and lastly provide the location in a format readable by LA service programs or apps.

Usually the geographical location will be presented to programs and apps as latitude and longitude coordinates, which will uniquely identify any location on earth through a pair of numbers, one for latitude and one for longitude ([Wikc]). Together they make up a sort of coordinate system, based on

the equator and the chosen prime meridian through the Royal Observatory, Greenwich, UK.

Take the iPhone, which is currently one of the most popular smartphones on the market, as an example of such technology. Apple has made available to external developers location information from the iPhone's hardware through their iOS framework called "Core Location". According to Apple's own description of the framework ([Lib]) it

"... lets you determine the current location or heading associated with a device. The framework uses the available hardware to determine the users position and heading."

This framework is open to iOS app developers, and these apps will later reach the users through the Apple app store. This post: [Fle08] explains how iPhone in 2008 was upgraded so that it was able to use Wi-Fi signals to determine the phone's location in addition to cell tower triangulation. iPhone 3G was released in July that same year, and this new model came equipped with a built-in GPS receiver, making tracking the phone's location much more accurate ([Gre08]). Later iPhone models thus have the ability to determine the phone's location using all three types of location-aware hardware (GPS, cell tower triangulation or Wi-Fi connections) based on what is best at the time.



Figure 5.1: This figure shows an example of a location-aware app developed for both iOS and the Android Platform, called ViewRanger. The figure is taken from [nav10]

Another example of a smartphone OS which is widely used these days is the Android platform. It is an open source project led by Google, and the core is an open-source software stack for mobile devices. Android also has a developer framework which is accessible to external developers. These developers will release their apps on Android Market, where users can download and take the apps into use. The Android framework, like the iOS framework, contains a package which provides the developer with locational data from the phone's hardware and software. This package provides data from a multitude of locational sources, just like the iPhone's OS. These include the GPS receiver, Cell-IDs and Wi-Fi information ([Deva]).

5.1.2 Privacy implications

The thought of having to wear a device around your neck which tracks your every move and stores it, can seem more like some sort of punishment than technological progress. Locational data is very personal, as much more personal data can be derived from it. Imagine tracking a person's every step for a year, and try to think about which questions regarding that person's life you would be able to answer by investigating that data. Where does she work? Most weekdays she spends her time at a Bank of America branch office, so she probably works there. Does she have a car? Yes, she is always on the highway. No, she spends time in bus-stops, so probably takes the bus. Does she take vacations? Yes, she goes to very expensive resorts and probably has a lot of money. No, she spends time in the mountains, and is probably a sporty person. Does she have many friends? Yes, she is always at restaurants and other people's houses. Maybe not, as she spends most days in her apartment. Does she cook a lot? No, she always goes and gets take-away from the chinese food place around the corner. What kind of clothes does she buy? She often spends time in cheap clothing stores, so probably buys them there. Does she go to an expensive hairdresser? Has she been to the doctor lately (a fact most insurance companies would love to know). Now imagine you gain access to all her friends' and colleagues' locational data for that year. Now you can answer many of these questions with a much higher probability, plus many, many more. Does she visit her mom often? Does she have a boyfriend? Which of her friends does she hang out with the most? Now, if that person was you, try to imagine how you would feel about disclosing all this information.

Privacy means different things to different people, and is often relative to how private a person likes to be. Regardless of this, most these questions are highly personal and sensitive, and the author certainly would very much dislike if she could not control who knew such information about her.

As was mentioned in the introduction to this chapter, as people strive to be available and "online" at all times, they end up carrying around their smartphones wherever they go. This means that the location of your phone, most often will coincide with your location. Already most smartphones contain equipment to track themselves in multiple ways, by GPS data mostly. Thus, the only difference between us wearing a device around our neck tracking and storing our every move, is whether or not this data is stored.

Lately many articles have been posted regarding news that iPhones contain an unencrypted secret file that logs and stores locations plus corresponding time-stamps. One example is this article from the Guardian: "iPhone keeps record of everywhere you go", [Art11]. The file contains the latitude and longitude coordinates plus a time-stamp of all the phone's recorded locations. With this file it is possible to get a good view of the owner's movements since the phone was taken into use (after the alleged start of the recording with Apple's iOS 4 update in June 2010). This file will also be transferred to one's computer when synchronized with an iPhone. Figure 5.2 shows the result of a program called iPhone Tracker, available at <http://petewarden.github.com/iPhoneTracker/>, which uses this log file to create a map of all the recorded locations. This file certainly holds private

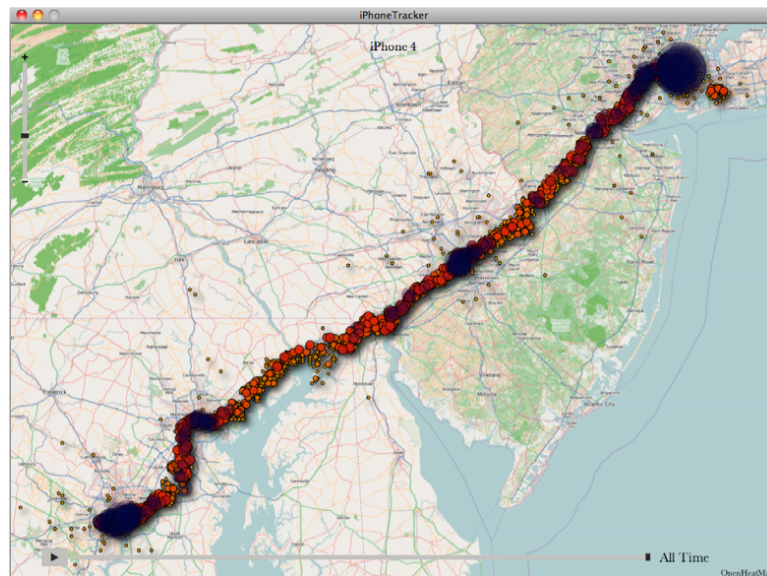


Figure 5.2: This figure shows an example of a map generated by the program iPhone Tracker, using some iPhone's secret location log file. The figure is taken from [Bod11]

information. It could possibly be used or abused by the police, phone or

computer thieves, suspicious spouses, etc., and this raises privacy concerns. The biggest concern probably has to do with the fact that iPhone users were unaware of the existence of such a file, and that this example is probably not the only case where too much information is gathered or stored without the knowledge or agreement of the user. After a month of massive critique, Apple has now launched an iOS update, only seven days after they promised to fix the errors relating to this file. This update makes sure the location log data is stored for no longer than seven days and, most importantly, the user can choose for this information not to be logged at all through "Settings" on their iPhone([Clu11]).

This example still shows that data will sometimes be stored with or without the user knowing about it or is able to control it. Combined with the sensitive nature of locational data, it shows there are reasons to be concerned with privacy when dealing with locational features in smartphones.

5.2 Location-awareness with SNSs

Section 4.1.1 in Chapter 4 explains how SNSs have emerged and been developing the last decade. Section 5.1.1, above, describes the technology behind location-aware smartphones and that location awareness has the last few years become a standard in the most popular phone types (e.g. iPhones and Android phones). It was only a matter of time before these two technologies were combined to form LASNSs. The rest of this chapter will be dedicated to the task of explaining how LASNSs work in general, privacy implications of using such services, plus a description of a few example LASNSs.

5.2.1 Integration: how it works

A LASNS will combine the information contained in a social network structure; profile information, relationships, interests, age, etc. with the geographical information provided by the underlying technology for location awareness. This could be combined in different ways depending on the idea of the developers. Following is an incomplete list of possible ideas for such LASNSs, to illustrate the possibilities for this technology.

- Broadcast your location to friends.
- Post information related to a location for others to find.
- Recommend places nearby based on your stated interests and other peoples' recommendations.

- Broadcast to, or targeted advertising from, businesses nearby based on your profile information.
- Mark all pictures taken with location information, for your friends or others to find based on location.
- Find possible dates (through dating SNSs) that are nearby or usually spend time in the same locations as you.

The technology required to integrate SNSs with LA is first of all a location-aware device one can carry around (usually a smartphone or another phone with location awareness) as the location of the human being carrying it around is what these services are based on. This device has to be able to connect to the Internet or some network. Being able to communicate data to and from other users is after all the idea behind most SNSs. Secondly, this device has to contain, or most often be open for downloading and installing, a piece of software that the user and the service servers can interact with. For smartphones like iPhones or Android phones this happens through special app stores, where users can easily search and find an app, press download, and it will be installed in seconds. These apps use the OS based framework for development on these platforms, and the developers will host their own servers, which the user will interact with through the app and the network.

When developing a LASNS there are roughly three ways to go:

1. Decide to extend your already established SNS with a LA component.
2. Use new LA ideas plus existing social network information (through APIs) to make a new LASNS app.
3. Develop a new LA idea and collect and create your own social network information base for the resulting LASNS system.

Some LASNSs are born from already existing and usually web based SNSs. This is true for Facebook Places, which is an addition to the rest of Facebook, where Facebook users can download an app on their phone, and use this to post their locations. Other LASNSs are developed purely for location awareness, but relies on established SNSs APIs to connect users to their friends. One example of this is Foursquare, which is based on people checking in at different locations, sharing, learning and earning points. They use an import of your Twitter and Facebook connections to base the social networking component of the service on. Others, on the other hand, will develop services with LA in mind, but aims to establish a new social networking information base. Starting from scratch with no social network

information can be pretty difficult. You will often have the "only fax machine in the world" problem, as the value of the network lies in the number of connections, and someone has to be first to get the ball rolling. That is why most new LASNSs will rely on importing connections from existing social networks like Twitter, Facebook or E-mail contacts, etc.

5.2.2 Privacy implications

When taking LASNSs into use, certain privacy issues will arise because of the private nature of the information gathered. Section 5.1.2 sheds light on how locational information is not only personal because of the revelation of one's location, but also because of all the other clues it provides about a person's life. The new LASNSs are mixing social networks and location information, and this will add to the privacy concerns of location awareness and SNSs put together.

Knowing where a person moves provides some information, and knowing how a group of people travel around and interacts provides even more information. Coupling the location of your friends with the relationships between them, their profile information, their interests, age, posts, activities, etc. can provide the holder of such information with an elaborate picture of these individuals' personal lives. Location awareness poses its own privacy issues, as do the use of social network services, and combined they will produce information which the users should be careful who they reveal to.

Another privacy concern regarding LASNSs which is becoming more and more present is the openness of the smartphone and SNS platforms. As of today, anyone with moderate programming skills can develop an Android app and post it to Android Market, and have it be downloaded instantly by users, with no filtering or control. As mentioned before, the APIs of e.g. Facebook and Twitter enables developers of new LASNSs to use the the existing social network information, making it even easier for anyone with a useful LA tool to quickly deploy and spread such a service. This means that people with a smart LA idea and possibly impure intentions can develop an app for it, and start to gather and store all those incoming locations people share as a part of the service use. As they control this information in their servers, and might legally own it (depending on formulations in the terms of agreement every user must accept) they might start selling personal information to whomever might be interested.

For these reasons, LASNSs, even more than SNSs alone, should provide the user with sufficient tools to control access to the information gathered in these system, to ensure a decent amount of privacy.

5.3 Facebook Places

Facebook Places is an extension to the already existing features, and developed by Facebook themselves. It allows you to broadcast your location through a Facebook mobile app whenever you please. Facebook is one of the biggest SNSs in the world, and thus have a huge existing user base. Most of these users have been introduced to Facebook Places (FBP) either through downloading the Facebook app for Android or iPhone, or by watching how their friends post their location in the news feed from their smartphones. Because of this valuable introduction to the usage, FBP has great potential for widespread usage, as more and more people acquire smartphones. Another advantage with this service is that users have some form of privacy control through the regular Facebook web interface, meaning we get the chance to evaluate existing privacy control for such a service. For these reasons we are going to use FBP as the main LASNS example.

5.3.1 Overview

FBP relies on users to actively push their information to the Facebook (FB) system. This is done through a FB app on their smartphone. The leftmost screenshot in Figure 5.3 shows how the FB iPhone app contains a feature called "Places". The first time one enters the FBP feature menu one is asked whether to allow FB to know one's location. After accepting this term, you will be taken to the screenshot to the far right. Here you will see a list of people nearby, and a list of friends who have checked in elsewhere. To share your own location you press the "Check In" button, and enter the menu seen in the middle screenshot of Figure 5.3. This menu provides the user with three main options:

- You can add places.
- Check in to existing places.
- Tag people who are with you.

When you and your friends check into a place, this is by default posted to your profile, in the news feed and the activity stream for that place.

Facebook has opened certain data that will allow any and all developers to access parts of Places. Initially external developers could only get read access to check-ins for specific users, pages or places or search the check-ins of a user's friends ([DuV10]). At first, only a few of their FBP launch partners, Gowalla, Foursquare, Yelp and Booyah, was able to push location



Figure 5.3: This figure shows three screenshots taken from the usage of the current Facebook Places feature on an iPhone. The figure is taken from [Dig]

information (check-ins) to the FB system on behalf of users ([O'D10]). Now however, FB provides the developers with a read/write API for check-ins in FBP ([Par10]). Check-in information can now be accessed or posted through the Graph API described in Section 4.2.2 and accessed in the same way as other social graph objects (users, pages, groups, notes, etc).

5.3.2 Privacy control

FBP' main feature is for users to broadcast their location. The privacy control panel allows users to control when and who will be able to see their check-ins.

When One would assume the users are able to control **when** they choose to share their location as the act of checking in is a conscious choice. Yet, the default privacy settings allow a user's friends to check him or her in. The regular FB user account privacy settings panel, explained in Section 4.2.4, provides the tool to disable this default setting as depicted in Figure

5.4. Yet, this means that unless you explicitly prohibit this, you or all your friends choose whether to reveal your location or not at all times. The write API for Facebook allows developers to request permission to perform check-ins on behalf of a user, meaning that the user is able to relinquish the control of check-ins not only to all their friends but also an external application.

Friends can check me in to Places

Edit Settings

Figure 5.4: This figure shows a screenshot of how Facebook users can toggle whether friends can check them in to places.

Who Through the regular account FB privacy settings the user can also control **who** they allow to see their check-ins. The privacy panel provides control over two aspects of sharing check-ins, which can both be viewed in Figure 5.5.

Places you check in to

Only Me ▼

Include me in "People Here Now" after I check in

Visible to friends and people checked in nearby (See an example)

☒ **Enable**

Figure 5.5: This figure depicts the options Facebook users have to control who can see their check-ins.

The one called "Include me in 'People Here Now' after I check in" has to do with the FBP feature where a user who just checked in somewhere can see who else has checked in to the same place regardless of whether they are friends or not. Whether users wish to share their location with potential strangers at the places they check into can be set by enabling or disabling this option in the privacy panel (Figure 5.5). The option is enabled as a default for all users with one or more other privacy setting(s) allowing access to "Everyone" ([dot10]).

The one called "Places you check in to" is where you control who can see your check-ins through their news feed, FBP app and your profile. This is by default set to "Friends Only" ([dot10]). Figure 5.6 shows the possible subjects one can choose to receive the right to see your check-ins. As with the other privacy decisions in Facebook, these are limited to a drop-down menu of coarse-grained choices plus a customization choice.

The customization panel that will open when choosing "Custom" is identical to the one explained in Section 4.2.4, "Facebook user privacy control", and depicted there as Figure 4.4. As explained there, the user has the choice to make visible to certain people, and to hide from certain people.

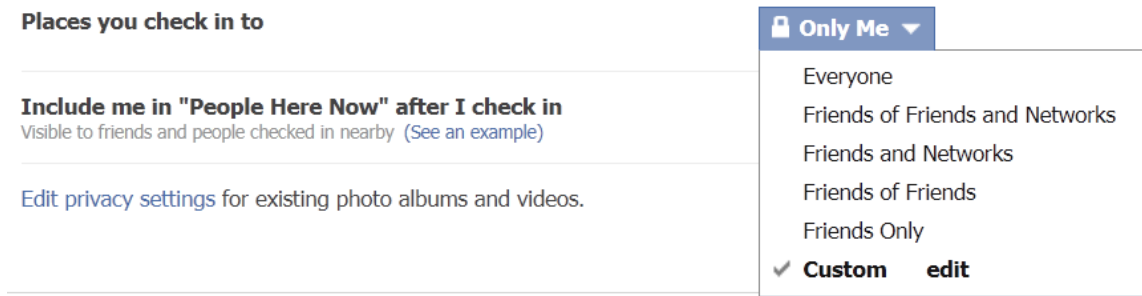


Figure 5.6: This figure depicts which types of subjects the Facebook users can give the right to see where they check in.

5.4 Other LASNS examples

In addition to Facebook Places, we will shortly present a few other example LASNSs. This is to get a better understanding of the different types of services currently offered using LA, and how the issue of privacy control are generally solved for these. What all these services have in common is a core; sharing locations with friends, plus additional functionality to differentiate themselves from competitors. They all have their users "check in" to their current location using a mobile device, and are integrated with Facebook, Twitter or similar SNSs to find a user's connections to be able to connect the user with his or her friends. On top, most of them have developed different kinds of incentives for the user to check in as often as possible, such as points, badges, promotional offers, etc.

5.4.1 Foursquare

Foursquare is a LASNS available to users with access to a GPS-enabled mobile device, such as a smartphone. The service is similar to Facebook as it expects the users to "check in" at venues by selecting from a list of venues based on the user's location. The users can choose to have their check in activity automatically posted to their Twitter account, Facebook account or to both. Figure 5.7 shows how the Foursquare app appears to iPhone users.

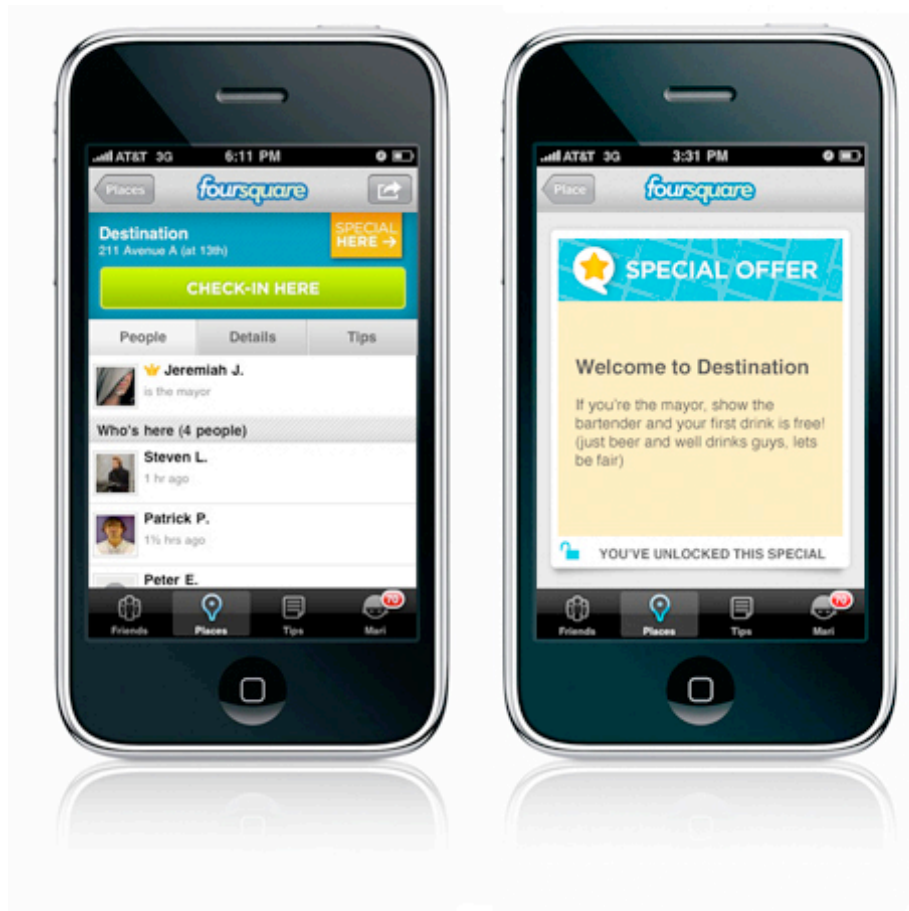


Figure 5.7: This figure shows screenshots from the Foursquare app on an iPhone. The figure is taken from [Hen10]

They have a point system where points are earned when checking in to venues. It is possible to earn mayorship at a venue meaning that you are the one who checked in at that venue most times during the last 60 days (the check-ins must be valid, only one counts per day, and they must have a profile photo). The second image in Figure 5.7 shows a special offer retrieved after achieving mayor status. When someone else beats your record they will take over mayor status. The users can also earn different badges by checking into special places, at special times or with high frequencies. Contrary to mayor status, once a badge is earned the user will have it indefinitely. It is also possible to achieve three different levels of "Superuser Status" based on different qualifications such as frequent check-ins and adding new information. All these features create incentives for users to check in as much as possible, spurring growth and usage of the service. As check-ins often are posted to

Facebook and Twitter, frequent Foursquare check-ins will reach many users, and serve as publicity for the service.

The website <http://pleaserobme.com/> is an example of how location information can be abused. It would crawl public Foursquare posts on Twitter, and retrieve locational information. It would then allow potential burglars to search through it for empty homes (the homes of people who through Foursquare have announced they are somewhere else). It should be mentioned that this site was created to shed light on the privacy issues with people being urged to share their location on Foursquare, not to get people robbed. Still, according to Wikipedia ([Wikb]), Foursquare had passed 7 million registered users on February 21, 2011.

Your contact information

Let my **friends** see my:

- ☒ phone number
- ☒ email address

Let **everyone** see the links to:

- ☒ my Twitter profile and Facebook profiles (only if I've connected them below)

Your privacy profile

Information related to your location is sensitive, so we give you control over how and when your personal information is published to friends, the foursquare community, and beyond. The settings below allow you to adjust how you share your foursquare data.

- ☒ Participate in foursquare Mayorships ([tell me more](#))
- ☒ Show me in the 'Who's here' list in the mobile app ([tell me more](#))
- ☒ Include me in my friends' status updates to Twitter and Facebook when I'm in checked into the same place as they are ([tell me more](#))
- ☒ Let local businesses know when I'm a loyal customer ([tell me more](#))

Tip: If you'd like to share your location information exclusively with foursquare friends, simply keep all of the boxes unchecked, and also choose not to publish your check-ins to Twitter or Facebook

Figure 5.8: This figure shows a screenshot of Foursquare's privacy control panel.

Privacy control The Foursquare mobile app does not provide the user with any tools to control their privacy. However, the web interface when a user logs in at <https://foursquare.com/settings> provides some amount of privacy control. You can toggle whether friends see your phone number and address, and whether everyone can see the links to your Twitter and/or Facebook accounts, as depicted in Figure 5.8. You can also choose not to participate in Mayorships, not to be shown in the 'Who's here' list in peoples' mobile apps, not to be included in friends' Twitter and Facebook status updates when you are checked into the the same place as them and to not let local businesses know when you are being a loyal customer (checking in there frequently). Other than these options, the users have no further control of who (which of their friends) are able to view their location or when.

5.4.2 Gowalla

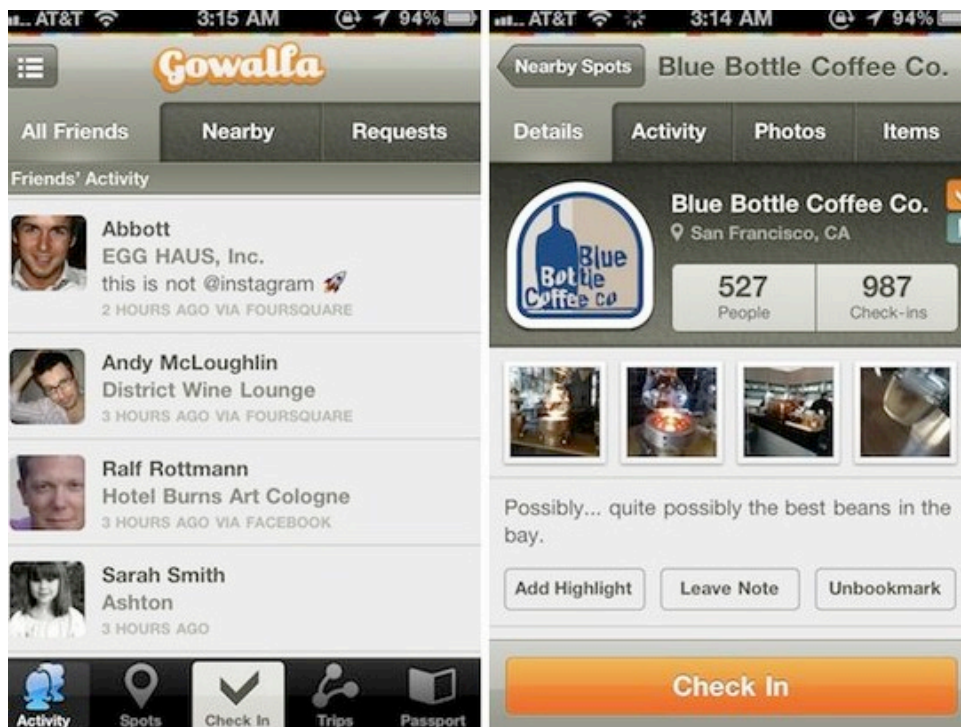


Figure 5.9: This figure shows screenshots from the a Gowalla app. The leftmost screenshot shows how you can monitor friends, while the rightmost screenshots depicts the check-in panel. The figure is taken from [Tip10]

This service is somewhat similar to Foursquare, and screenshots from the service is shown in Figure 5.9. You use a GPS-enabled mobile device to check

in to "spots", choosing a spot from a list of spots in the vicinity. Gowalla check-ins can also be pushed to a user's Facebook and/or Twitter account. The rightmost screenshot in Figure 5.9 shows how this service provides the user with check-in options. Users can leave their own opinions and notes about these spots and photos taken there. They can also receive digital "items" by checking in, creating incentives to do so ([Tin10]). These can be picked up, swapped or dropped, and can be linked to real-world items or may be promotional items linked to real-world prices. The spot's details, activities (people checked in, notes, etc.), photos and marketing items can be found in the check-in menu as seen in the screenshot to the left. Gowalla also provides the user with the ability to create and share "trips", linking 20 or less spots. These trips can represent nature hikes, sightseeing tours, a music festival's recommended spots, park highlights etc.

You can connect your Gowalla account to your Facebook and Foursquare accounts. This means that Gowalla use your connections from these services, and can display a friend activity feed as depicted in the leftmost screenshot in Figure 5.9. As of Gowalla 3, Gowalla supports checking into Facebook Places and Foursquare in addition to sharing with Twitter ([Wil]). Gowalla is thus a LASNS connecting to multiple SNSs and another LASNS.

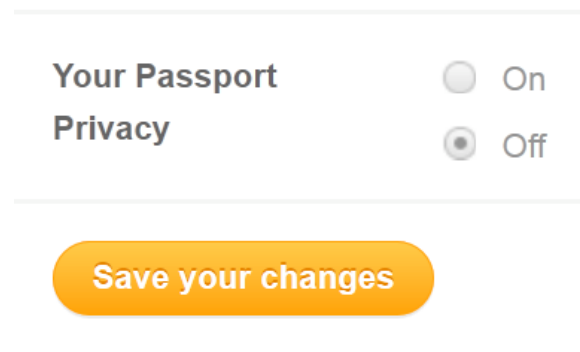


Figure 5.10: This figure shows a screenshot from the Gowalla user privacy control panel.

Privacy control Gowalla, like Foursquare, does not provide privacy control through their app, but through their web site where users log in: <http://gowalla.com/users/username/privacy>. They, however, only give the user the choice to turn privacy **on** or **off** as seen in Figure 5.10, with **off** as the default position. This setting reflects whether the user wishes to share

the stamps in their passport (the places they have checked into) and their items with: everyone (off) or only their friends (imported connections) (on). Even if you only wish to share with your friends, and place this setting at **on**, there are three exceptions where you will still share with everyone: 1) If you create a new spot, your name will be associated with the spot and visible to everyone, 2) if you drop an item at a spot, your name will be publicly associated with both the item and the spot, and 3) if you take a photo at a spot, the photo will be associated with the spot and available to everyone. Gowalla's privacy control leaves the user with no fine-grained control of what they share and to whom.

5.4.3 Loopt



Figure 5.11: This figure shows screenshots from the Android Loopt app. The leftmost screenshot show a map with people and places nearby .The rightmost screenshots show the rest of the Loopt menu. The figure is taken from [And]

Loopt is a LASNS with focus on showing users where their friends are and what they are doing through maps and other interfaces on their mobile device ([Wike]). It is currently only available on Android and iPhone devices. When registering as a user you can choose to connect it with your Facebook account, making available your Facebook friends. It is also possible to change the profile settings such that all your check-ins are posted on your Facebook account, Twitter account or both.

As seen in Figure 5.11, the main menu shows the different features of the service. "Map", "Friends", "Lately" (recent activity), "Check In", "Places" and "Events" (the menu choices "Invite" and "Settings" are less interesting). The map shows you and your surrounding area, places and friend's check-ins. The other menu options let you connect with friends, view recent activity, check in, view places, see events, invite other people to join or change the settings.

Privacy control The only privacy control available on your Loopt account when registering is whether Loopt should share your location automatically or only when you check in (shown in Figure 5.12). The default option is "Automatically", meaning every time you open the Loopt app on your mobile device. With this, Loopt offers the worst user privacy out of the LASNSs discussed in this chapter. This observation is based on the fact that in addition to not being able to make any fine-grained rules for controlling your data, they not only provide the user with a way to share their location automatically, this is in fact the default setting.

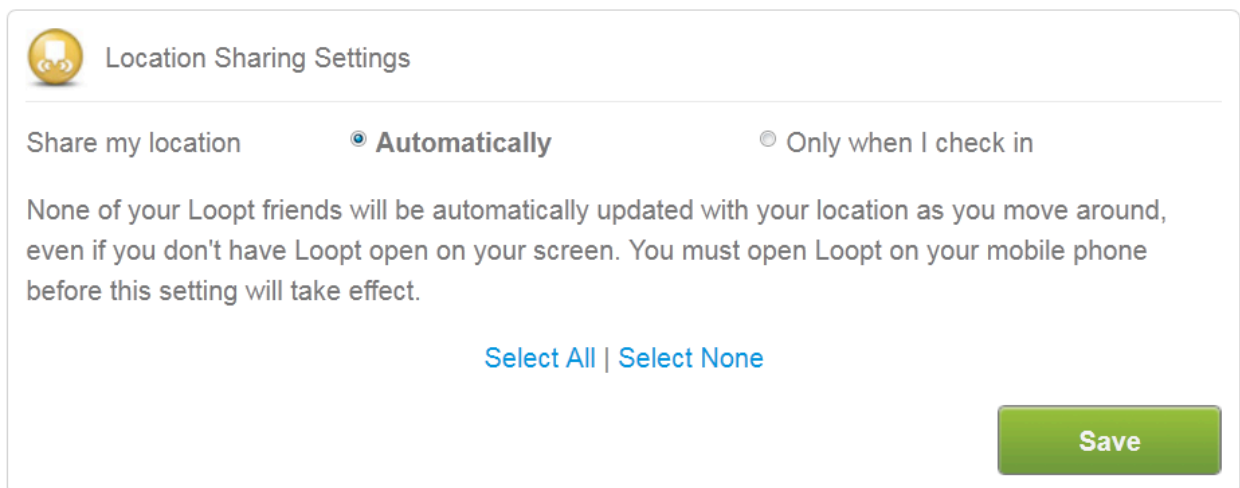


Figure 5.12: This figure shows the location-sharing control panel for Loopt.

5.4.4 Brightkite

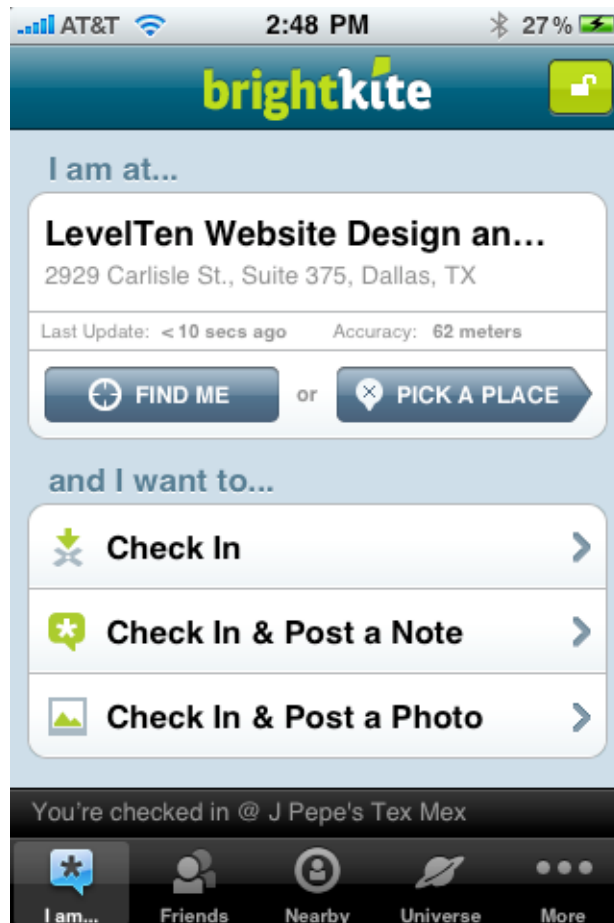


Figure 5.13: This figure shows a screenshot from a Brightkite app. The figure is taken from [Als09]

Brightkite is a service based on users connecting to each other, seeing who is nearby, who has been there before and being able to contact them. Like the other LASNSs, the user will "check in" to locations, in this case with an Android app, iPhone app or by visiting m.brightkite.com with a different mobile device. Users can choose to share their updates on their Facebook and Twitter accounts ([Wika]). They can also share their geotagged images (geographical location added as metadata) on the image sharing site Flickr. When a user has checked in to a location they can post notes, photos or comments to other user's notes for that place. Figure 5.13 shows a screenshot of the Brightkite app from 2009.

Privacy control The only privacy control users of Brightkite have is to block certain individuals from contacting them. You can choose to block people from the Brightkite app, and organize blocked people through the Brightkite web interface at <http://brightkite.com/> depicted in Figure 5.14.

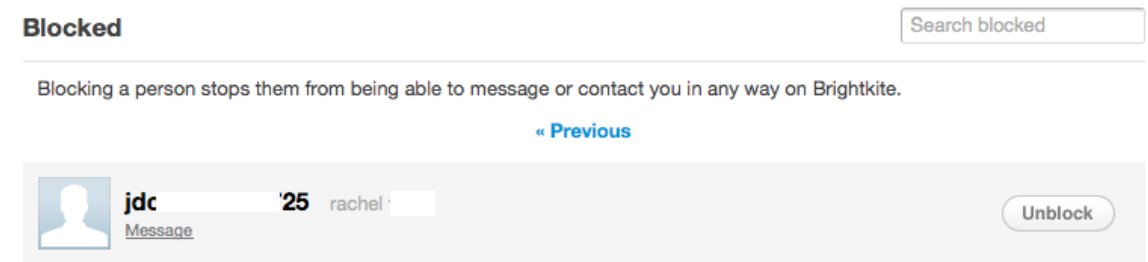


Figure 5.14: This figure shows a screenshot from the web-based Brightkite control panel for administrating blocked individuals

5.5 LASNS privacy control analysis

Without the LA component of these services, they are basically only SNSs, and these we have already had a privacy discussion of in Chapter 4. The privacy control of non-LA elements in these LASNS systems such as sharing photos, posting status updates, etc. should meet the general requirements as stated in Section 4.3, "Privacy requirements for SNSs in general". This is something all these services have to work with, as many of them lack it completely. What is more important is what kind of user privacy control of locational information these services have and might lack.

Protection of LA information When examining the LASNSs Facebook Places, Foursquare, Gowalla, Loopt and Brightkite it becomes clear that the users have little fine-grained control over who they share their location with and when. If the user has any control, it is usually limited to choosing to show your location to everyone or only friends. Only FBP provide any more fine-grained tuning of which subjects are allowed to see your check-ins, as they have the customization panel where you can specify individuals you would like to share with or hide the location from. In light of the list of privacy control requirements in Section 4.3, "Privacy requirements for SNSs in general", the privacy control for these services seems insufficient at the least. With the **location** being the object one wishes to control, none of them

meet requirements 2 to 6 for this, except for FBP with the customization option meeting the second requirement. As discussed previously, locational data is by itself (and can hold) extremely personal information. That is why the user should have **at least** as much control over how their location is shared as any other piece of information in SNSs, such as Facebook. To put it in perspective, Facebook offers their users more fine-grained privacy control over who can see their stated general interests, than what Gowalla offers for their users' locational information.

Based on our general SNS privacy requirements (Section 4.3) and our information sensitivity assessment of locational information, we have identified a few missing or inadequate privacy control elements for these kinds of LASNSs: Users should be able to control **when** (under which conditions) their locations (and other objects) are shared and **to whom** for different features. The user should be able to make some general rules for their locational sharing based on the following five variables. These rules will control how and when a user's location is displayed by different features to different individuals in the system. When users post their location these rules should dictate how it is shared, possibly hiding it entirely if the user has decided to display no check-ins after 10 pm and it is 10.15 pm. Even though this list is based on our requirements for privacy control of locational data, this list should also apply to all other types of objects.

Privacy control elements that are missing or inadequate in privacy control panels in existent LASNSs:

1. **Object type** - The users should be able to control access to the objects related to them, in this case, especially their current location or location history.
2. **Owner relation to subject** - The choice between revealing your location to "Everyone" or "Only Friends", which is often the only choices (if there are any), is very limited. Connections ("Friends") imported through Facebook and Twitter for instance, can represent less close personal relationships than close friendships. The user should be able to make more fine-grained division between different types of connections and grant different access right to such groups.
3. **Location** - The users should be able to make access control rules based on their location or the access requesting subject's location. Also, every location should have a type such as "Bar", "Hotel", "Classroom", "Park", etc., and the user should be able to set different access rights

to different subjects based on the type of location one checks in to. A user might want colleagues to only see when they check into restaurants and sports arenas, but not personal homes or hotels.

4. **Time** - The user should be able to set a time constraint for each rule, stating when it is applicable (working hours, evenings, holidays, etc). In addition the user should be able to set a default expiration date on its check-ins. This should also be specifiable for every check in.
5. **Owner and subject attributes** - The user should be able to filter the subjects allowed to see their locations based on their own or the subject's attributes, such as a certain age group, people with similar interests, people who work at the place your are at, etc.

One feature that exemplifies all these is FBPs "Here Now" feature. Being included in "Here Now" after you check into a place can only be enabled or disabled, yet should be more tunable to different subjects with different attributes at different times and at different types of locations.

5.6 Conclusion

The location awareness of services, especially SNSs, demands adequate user privacy control. Knowing one's location not only tells you where they are, but gives you hints and indications about his or her personal life. Some people are very personal and protective of such information while others like to share everything. These preference differences should be reflected in users' privacy settings, and the developers of LASNSs should thus enable such control. These privacy control panels should give the users the chance, for those who wish to, to express fine-grained privacy preferences, especially when it comes to stating under which conditions and to whom their locations (or other objects) are shared.

Chapter 6

Enhanced Privacy Control Framework for LASNSs

Through elaborate scenarios, examples and discussions we have showed that users can have privacy preferences that current LASNS's coarse-grained control panels are not able to reflect. The analysis of existent privacy control features in LASNSs in Section 5.5, "LASNS privacy control analysis", has identified five missing or inadequate elements in current LASNSs, based on our requirements for end-user privacy access control (especially for locational data). Simply put, we require LASNS privacy control panels to have the tools for users to create access control rules based on the following variables:

- 1. Object type**
- 2. Owner relation to subject**
- 3. Location**
- 4. Time**
- 5. Owner and subject attributes**

Many users must currently compromise their sense of privacy in order to use these services. We will in this chapter propose a few access control enhancements that are implementable in these LASNS systems, which in turn will help the user fine-tune what they share and how. These enhancements will make sure the resulting system fulfills all of the six proposed requirements for SNSs from Section 4.3, "Privacy requirements for SNSs in general" plus all of the five proposed user control requirements for LASNS in Section 5.5, "Protection of LA information" (listed above).

6.1 Privacy-enhancing access control for LASNSs

The task of increasing user privacy control in LASNSs essentially means making sure the users can input their privacy preferences in the system, and have the system enforce them. Preferences can be expressed as privacy rules, which can be written as access control rules, which can be translated into some standard language and enforced in a system. Preferences can be based on subject roles, attributes, time, location or other factors, and access control rules should be able to contain these elements.

We propose two separate access control enhancements, which together will add all of the five currently missing elements to LASNS, satisfying our demand for end-user privacy control through access control rule specification. Both these proposals will be presented in detail in the sections below.

- More fine-grained subject separation.
- Fine-grained rule conditions.

6.1.1 More fine-grained subject separation

When a user makes an access control rule, the rule will have to at least contain the object in question (the location, an image or album, wall posts, etc.), the kind of access they wish to control (read, write, etc.) plus the **subject(s)** who are to receive such access to that object. For many of the services we have presented in this thesis the choices for subjects have been centered around "friends" and everyone else, where "friends" represent everyone you have made a connection with through Facebook, Twitter, e-mail contacts and other services. The main reason why we claim these services need more fine-grained subject separation is that these connections, often called "friends" does not necessarily correspond to real-world friendships in the traditional sense. A user's "friend"-base used in the LASNSs (often imported contacts from existing SNSs) can consist of many different types of relationships to the user such as; colleagues, family, classmates, acquaintances, fans, etc. People use LASNSs for different purposes for different kinds of relations, making new friends, maintaining close friendships, contacting colleagues, sharing content with friends and similar. This means that one might want to share different content with different types of "friends" to reflect this differentiated use. That is why the system should let the user be able to group these "friends" based on the user's relationship with them and their role to the user, and set different AC rules for different groups(roles).

RBAC

Granting access to "Friends", "Friends of friends" or "People in your network", can be classified as role-based AC, RBAC, as the user will grant access to a group of people based on their role or relationship with the user. Still, as explained above, the role-options provided by current end-user privacy control panels are too limited in light of our requirements. More fine-grained subject separation can be accomplished through the introduction of a more extensive version of Core RBAC (Section 2.1.4) in the system, with the user as the system administrator. Specifically, implementing an end-user control panel for user-role assignment, which further can be used when specifying access control rules based on the specified roles. The user will thus be the system administrator, administering roles and making AC rules based on these roles. As each user is the system administrator for their objects (images, posts, friendships, profile information etc.) and subjects (friends), the roles will be assigned according to the subjects role(relationship) to the user. These roles/relationships will thus only be relevant when access to that user's resources is requested, and the user's role-user assignment will be editable and accessible only to that user.

The point of RBAC is, instead of evaluating and making access rules for each individual subject, the subjects are divided into groups according to their role in the system, and access control rights are assigned based on different roles, not different subjects. This is efficient when the roles usually correspond to the different usages of the system and its resources, such that it is natural to assign AC rules to roles. As there are many more subjects than roles, RBAC becomes more efficient, in both assigning and changing access rights. In LASNSs the subject's role or relationship to the user often represents which access rights the user will want to grant the subject, and using this RBAC model might prove much more efficient than specifying individual "friends" for each access rule. The problem with the current, limited RBAC, with only a few system-defined roles, is that they provide a too coarse-grained separation of a user's connection. For RBAC to be efficient, roles have to be specialized enough for users to be able to reflect their privacy preferences through access control rules containing these roles. RBAC is not efficient when a user with certain privacy preferences has to list individuals for each access control rule to enforce them, because the system-defined roles contain the wrong people. This is why these systems should implement a control panel for role-assignment, and make these roles available as the subject element of access control rules.

The user will create their own roles/relationships reflecting the different relationships and thus usages with all their connections(friends). Every

friend can be assigned to one or more roles/relationships. As examples of this; a family member can also be a close friend, a classmate can end up as a colleague too and a colleague might end up as a close friend or a even a romantic partner. In these cases it is natural that the role/relationship with the highest level of access right in each access case prevails. An example of role assignment in a LASNS can be seen in Figure 6.1. When the user's

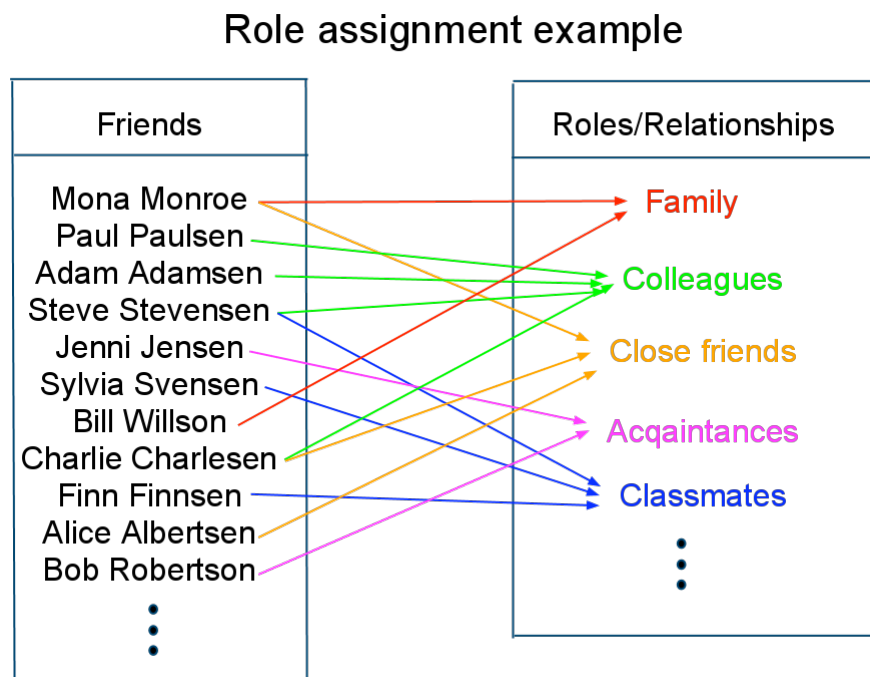


Figure 6.1: This figure shows a high-level view of how role assignment of a user's friend could possibly look like.

friends are assigned their correct roles/relationships, these can be specified (as choices in the drop-down menu) when creating access right rules in the control panel. A high-level overview of how these roles/relationships can be granted access to objects can be seen in Figure 6.2. Every connection (arrow) between role and object is accompanied by access types and possibly conditions under which the role's access to the object should be granted. A discussion of such conditions will be presented in Section 6.1.2, below. The current subject drop-down choices for the services we have seen are choices based on the notion of "friends", meaning all your connections or a list of specific individuals for each rule. Including the roles/relationship names in the drop-down menu will thus give the user more fine-grained control, plus make this more efficient than having to list individuals for each rule.

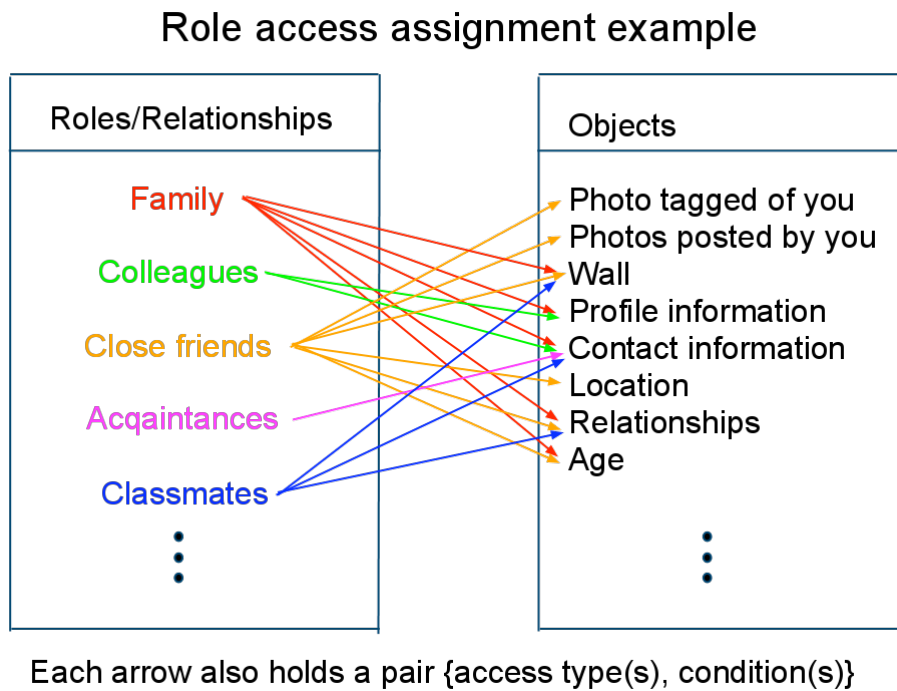


Figure 6.2: This figure shows a high-level view of how a user, after assigning subjects to roles, can assign those roles access to relevant objects in the system.

If a user A has no previous relationship with another user, user B, user B's role towards user A should be "Stranger" by default. If they become "friends", or create another type of connection to each other, the assignment of user B to the role "Stranger" should be automatically removed from the role list of A, and vice versa.

It is also important to note that it should be possible to also specify individuals as well as roles/relationships for each access right rule, so that introducing RBAC will not degrade user control compared to the current options in FBP for instance. It would be impractical if the user has to create a special role for one person, to be able to set access rights specifically for that individual. This is why we have chosen to call the "Subject" of the access control rules in this chapter "Subject/Role".

6.1.2 Rule conditions

SNS systems can contain a lot of data per user. Age, profile photo, interests, friendships, recommendations, romantic relationships, family relationships,

posted photos, achievements, etc, and for LASNSs specifically, at times, the user's location. A system will also know about external factors, such as time. Based on the scenarios in Chapter 3 plus our discussion and analysis of SNSs and LASNSs, we have identified a need for users to be able to create conditional access control rules in the privacy control panel of these services. These kinds of rules will only be enforced after the user stated condition(s), based on the value of the relevant attributes and logical operators, are evaluated to true. A user should be able to define multiple conditions, and pair them with "AND" and "OR" operators. It is probably possible to use most of the data these systems contain to make truly fine-grained privacy control through the use of conditions. An example of how conditions can be paired with roles/relationships(subjects), objects and access types to create a rule representation of user preferences can be seen in Figure 6.3. As can be seen,

Rule table example

Effect	Roles/relationships	Object	Access type(s)	Condition(s)
Allow	Family	Photos	read	Age < 30
Allow	Colleagues	Location	read	08.00 < Time < 16.00
Allow	Acquaintance	Profile photos	read	Has profile photo
Allow	Strangers (no connection)	Wall	read/write	Common friends

Figure 6.3: This figure shows a high-level view of a table of example rules based on rule effect, roles/relationship, object, access type(s) and condition(s).

each access control rule will consist of relevant rule effect, "Role/relationship" (subject), object, access type plus zero or more conditions

Condition format

To be able to translate user preferences into machine readable access control rules, it is important that the user can input the conditions for their access control rules in a standard format. A possible general format which can be used in most cases is depicted in Figure 6.4. We believe that most conditions consist of an attribute, an operator and a value. Together they create an expression which can be evaluated at run-time, deciding whether to grant the access defined in the access control rule the condition belongs to.

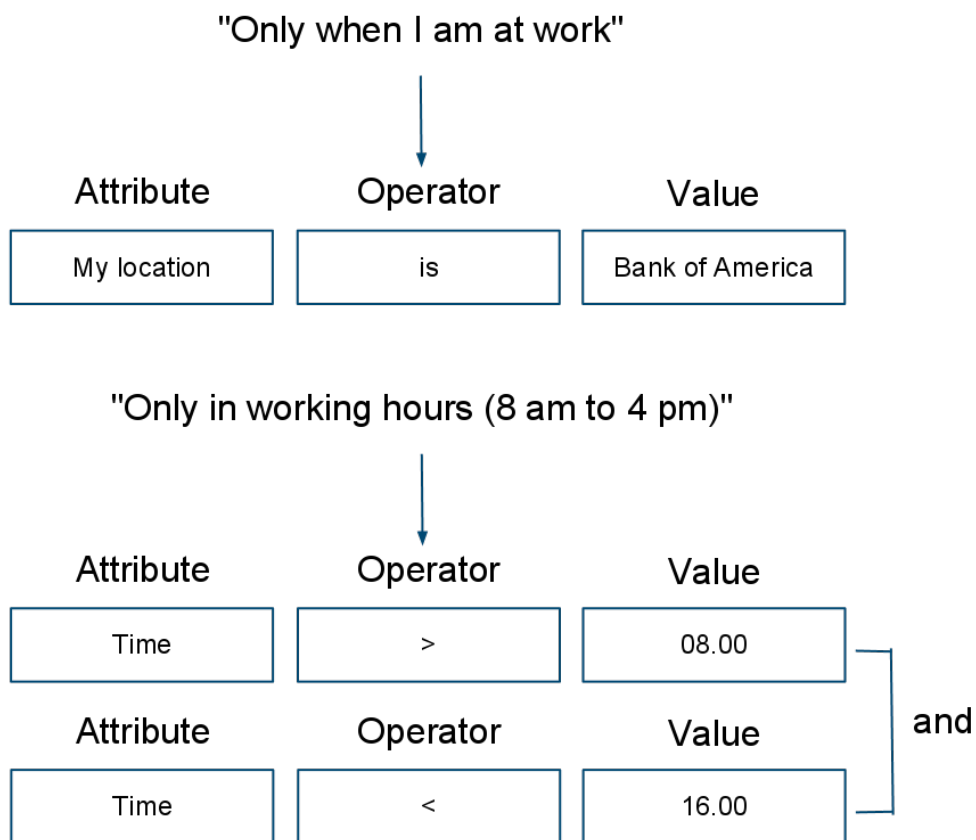


Figure 6.4: This figure shows a high-level view of how the conditions in access control rules reflecting user preferences can be translated into a general format, which in turn will be possible to translate into a standard machine-readable format.

These conditions will be built using different attribute types. Following is a list of possible system attributes. It will of course be up to system developers to choose which attributes it is possible to use, yet the more

choices the user has, the more fine-grained access control rules he/she can make, the more privacy control the user will have.

- Owner attributes (own location, age, education etc.)
- Subject attributes (subject location, interests, profile, age, etc.)
- External attributes (time)

The system should provide the user with a drop-down menu of possible attributes suitable for creating access control rule conditions. The operator choices will depend on which attribute is chosen. If it is a numerical value, the operators might be $<$ or $=$ for instance, while for locations the operators might be "is not", "is", "is nearby (within 1 mile, for instance)", or similar. The drop-down menu choices for the field "value" will depend on both the attribute and possibly the operator, depending on what the LASNS system developers have chosen to make available.

6.2 Implementation

The idea of being able to input user preferences into these LASNS system is a good one, yet there are two challenges that will arise when implementing this kind of user control. First of all, the interface where the users will input their rules has to be understandable and in a format that hopefully will capture most preferences, yet be simple enough for the users to bother to do it. Second of all, the rules defined by the user have to be translated into code that the system can understand, and the rules have to be enforced in the system.

6.2.1 User interface

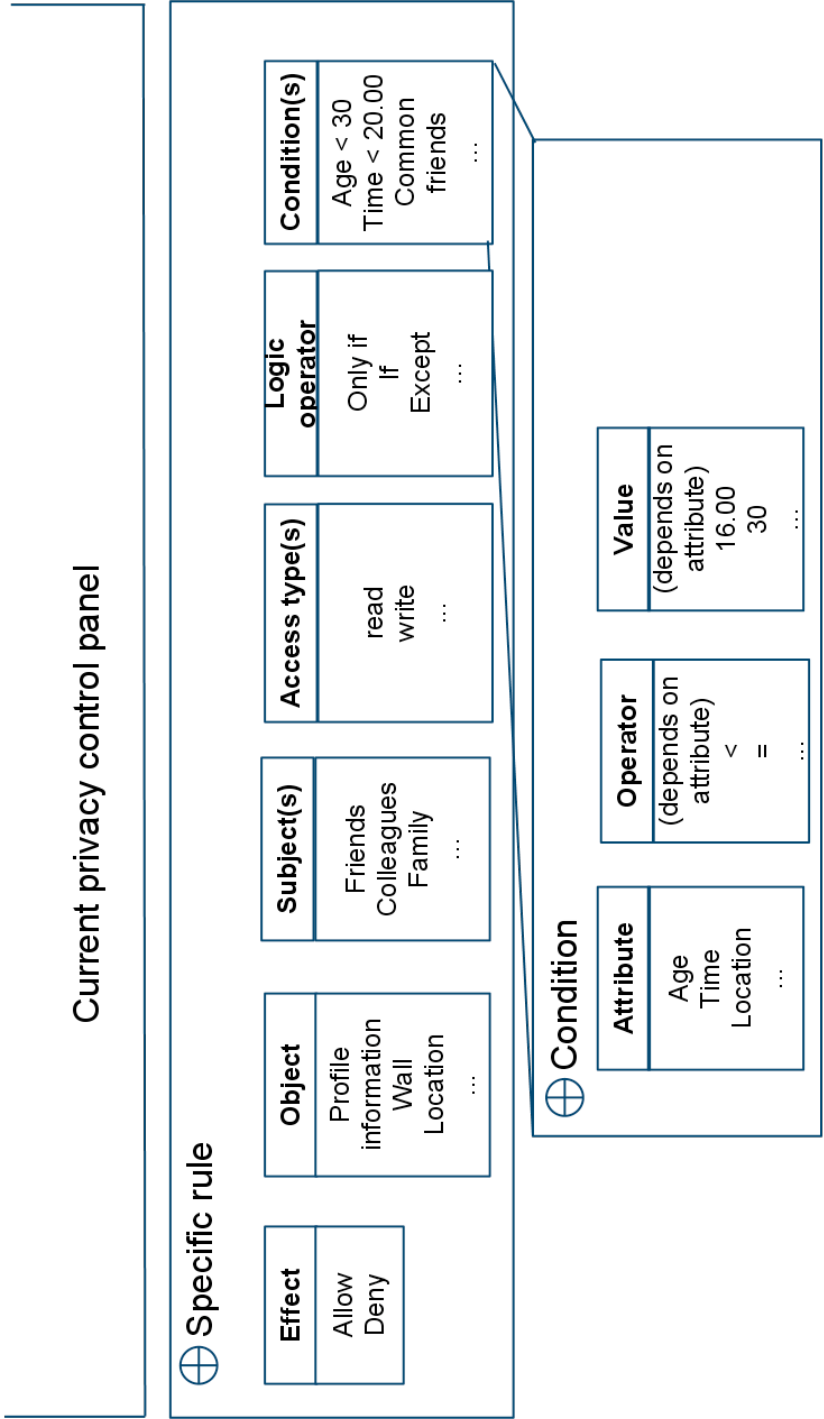
The article "Patient-Administered Access Control: a Usability Study", [sA09], by Lillian Røstad and Ole Andreas Alsos, sheds light on the importance of a well-designed user interface. This article discusses patient administered AC in a healthcare information system, where the patient is the system administrator. This case is somewhat similar to our case, where we aim to propose user administered AC in LASNSs. Information healthcare systems contain more sensitive data and different types of data than can be found in LASNSs. The research in [sA09] is therefore not exactly applicable to our case, yet the focus on interface design can be applied in this master thesis. Increased complexity of user privacy control can introduce weaker security,

as users usually are not security or privacy experts. In order to achieve better privacy, through more complex privacy control features, the users have to understand how to control it according to their preferences. This is why an understandable GUI has to be part of a well-designed user privacy control model, and attention should be brought to usability, as well as the underlying technology.

Different users have different privacy preferences. Some users might feel that current LASNS privacy control panels are adequate, and others might not even care much about privacy. Still, some people, the author included, have preferences which can not be reflected in the current control panels. The two enhancement techniques described above, RBAC and rule conditions, help users who wish to specify more fine-grained access control rules for their objects. Dealing with different user privacy control requirements calls for a privacy control GUI that is 1) simple and general enough to use for those who are satisfied with current privacy control, yet 2) customizable enough for users with special privacy preferences, such that they are able to create fine-grained access control rules. The goal when implementing the GUI for access control rule specification in the end-user privacy control panel should thus be to make it **efficient**, yet **simple** to use. This balance can be accomplished by continuing to offer the more general current privacy panels (as in Facebook), which are easily adjustable for general rules and easy to understand, yet adding a GUI part where users can input more specialized rules. The privacy control panel will thus consist of two parts:

- General privacy control.
- Special cases/preferences: Writing specific rules with object, subject, access type and condition.

Where a specialized rule conflicts with the general privacy settings an object is affected by, it might be fair to assume that the specialized rule for that object is more important (as the user has taken the time to specify it), and that the specialized rule should prevail and thus be enforced. An example of how a privacy control panel GUI can be shaped is depicted in Figure 6.5:



Users should be able to input multiple specialized AC rules to reflect their privacy preferences. Each specific rule consists of a rule effect ("Allow" or "Deny"), one Object, one or more Subjects (role/relationship or individuals), one or more access types plus zero or more conditions. When one subject, access type or condition is chosen, a new option panel will show up so that the user can specify more items (several access types, for instance).

Following is a description of the steps a user will take to create an access control rule through the privacy control panel depicted in Figure 6.5. Through these steps, the user will create an access control rule in a general and controlled format, which will be translatable into more formalized access control rule formats (XACML for instance) and later enforced in a system's access control component(s).

Effect First of all, the user will choose whether this rule will either "Allow" or "Deny" the specified subject(s) the type(s) of access to the object under the chosen conditions.

Object Secondly, the user will choose which object the rule will be relevant for, through a drop-down menu of all objects related to the user which he/or she is allowed to make access rules for. The reason why only one object can be chosen for each rule is that the drop-down menu choices for access types will depend on which object the user has chosen. One example of this is that it might not be appropriate to grant other users write access to your own profile picture, thus the access type "write" should not appear when writing an access rule for the user's profile photo. If multiple objects were possible, conflicts and confusion could arise from the access type choices.

Subject(s)/Role(s) Third, the user will choose which subject(s) will receive the right to access the object. The subject drop-down choices will consist of all the user-defined roles/relationships, more general groupings like "friends", "everyone", "my network", and similar, plus the option to specify individual users.

Access type(s) The fourth step is for the users to specify which access type(s) they would like to grant the subject(s) to the object.

Logic operator (optional) The fifth step depends on whether the user wishes to create conditions or not. If that is the case, the user is required to choose how the condition shall affect the rest of the rule. Whether access

should be granted "Only if", "If", "Except when" (or similar logic operators) the condition(s) are evaluated to true.

Condition(s) (optional) Last, the user can choose to create zero or more conditions under which the subject(s) will get the specific access type(s) to the object. The conditions are created in a separate panel with two drop-down menus and a value specification field, Attribute, Operator and Value, respectively. The drop-down menu choices for operator and the value field restrictions will depend on which attribute is chosen. The possible attribute value types for the chosen attribute, numbers, time, text, roles/relationship, location etc. will limit which operators are possible on that attribute, and naturally, which values the user is allowed to type in. If the user chooses to create multiple conditions, he or she has to put them together with operators such as "AND" and "OR". This kind of logic might be a little bit confusing to some users, and therefore it might be best to limit the ability to combine conditions to maximum two conditions, combined with **either** "AND" **or** "OR".

From input to textual representation

The order in which the different rule elements appear in the enhanced privacy control panel is based on which dependencies exist between them. That is why the order is: Rule effect, Object, Subject(s), Access type(s), and for conditions specification; Logic operator, Attribute, Operator and Value, where the choice between the logic operators "AND" and "OR" will appear after at least one condition is created. When a user has created an AC rule through this panel, it might be valuable for the user to be able to read the access control rule in a format closer to natural language, e.g English. Because the rule components are specified in a standard way, it will be possible to translate it into such a format. One can thus include a button or a display panel revealing the specified rule in a more understandable format for user inspection. With the rule elements specified through the panel, a sentence describing the rule will look similar to this:

"<Subject(s)> is/are <Rule effect> to/from <Access type(s)>
<Object> <Logic operator> <Attribute> <Operator> <Value>"

One example is:

Colleagues (role) are allowed to (rule effect) read (access type) my Profile information (object) only if (logic operator) their age (attribute) < (operator) 30 (value).

There is a need for some formatting of the words to create the right sentences, such a translating "Allow" to "allowed to", "Deny" to "denied from",

use singular and plural form depending on the subject/role, bind multiple access types and subjects with "and" and maybe modifying attribute values such as "Subject Network" to "Their Network" depending on the variables. This can be done through programming, as the developers decide the drop-down menu choices for these elements, and thus can tailor translation algorithms to display the sentence in correct natural language. An example of such a translation, from data sent to the system through the proposed GUI to an understandable sentence, is illustrated in Figure 6.6:

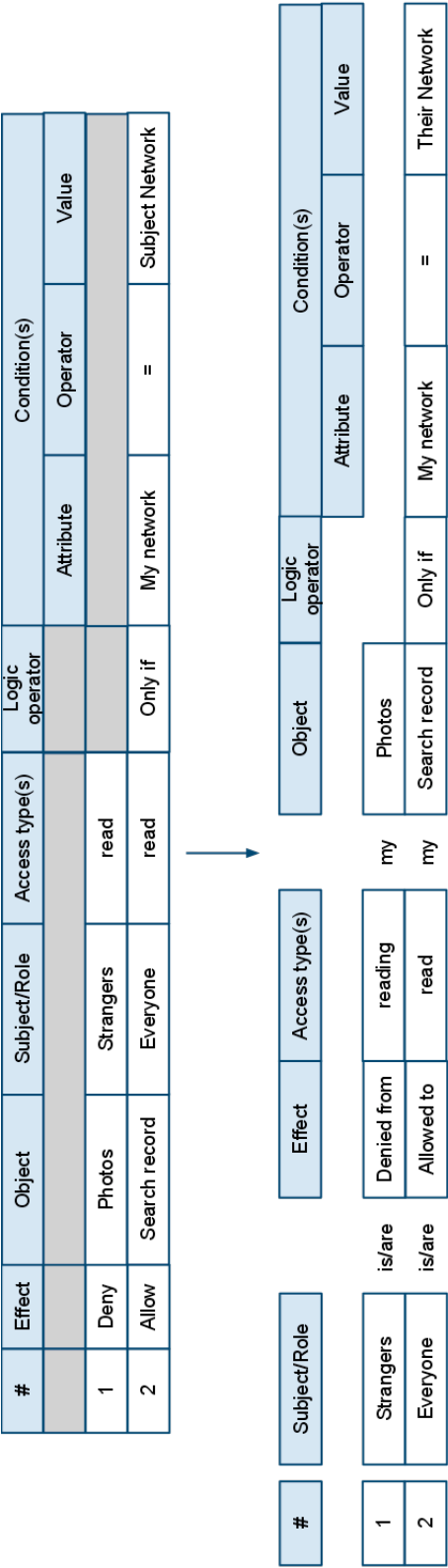


Figure 6.6: This figure shows an example of how access control rule data from the GUI can be displayed to the user in a more understandable format, closer to natural language.

6.2.2 From user scenario case to system-interpretable access control rule

When the users have reflected their privacy preferences through privacy control panels, the system has to receive it, interpret it and enforce it. A rule-structure understandable by users is not necessarily the best representation for a computer system. That is why the data received from the GUI has to be properly prepared before used in the system's access control decision and enforcement components. We are going to use a few of the user privacy preferences from Chapter 3, "Scenarios", listed below, to explain how such preferences can be translated into a more machine-readable format. We will translate the preferences into a high-level AC rule format, and further translate these into more formalized languages like Datalog and XACML, as explained below.

Each of the five preferences represents one of the five currently missing or inadequate privacy control elements stated in Section 5.5, "Protection of LA information" and in the introduction of this chapter. The reason for this is to show that our proposed access control rule format containing these elements are both understandable as a GUI (as discussed above), and translatable into a more formal machine-readable format. This will again show that our proposed AC rule enhancements will not only fulfill our requirements from a user perspective, but are implementable in a system.

1. **Object type** - Scenario 1, preference 16 - "As a regular user, I wish to hide **my photos** from strangers."
2. **User relation to subject** - Scenario 1, preference 11 - "As a 25 year old guy, I want to show the photo album from the guy-trip to London only to my **close friends**, except the close friends who are also in **my family**."
3. **Location** - Scenario 2, preference 5 - "As a Facebook Places user, I would like for my colleagues to be able to chat with me only when **I am at work**, at Bank of America."
4. **Time** - Scenario 1, preference 13 - "As a business woman, I only want my colleagues and acquaintances to see the places I have checked in to **during business hours**."
5. **Owner and subject attributes** - Scenario 1, preference 2 - "As a student, I want only the people in the **same network** as me to be able to find me in searches."

The first step is to extract the relevant data components from the textual preferences. Through the proposed user interface, or similar, the system will receive rules consisting of:

- The rule effect.
- The object in question.
- Roles/relationship or users (Subjects).
- The access type(s)
- Logic operator (optional)
- Condition(s) (optional), each consisting of the following components:
 - Attribute
 - Operator
 - Value

In the case of the chosen preferences the system will receive the following data:

#	Effect	Object	Subject/Role	Access type(s)	Logic operator	Condition(s)		
						Attribute	Operator	Value
1	Deny	Photos	Strangers	read				
2	Allow	Album: London	Close friends	read	Except	Subject's Roles	contains	Family
3	Allow	Chat	Colleagues	read,write	Only if	My Location	is/=	Bank of America
4	Allow	Locations	Colleagues	read	Only if	Time	>	08.00
			Acquaintances		And	Time	<	16.00
5	Allow	Search record	Everyone	read	Only if	My network	=	Subject Network

Figure 6.7: This figure shows how the data in the textual descriptions 1 through 5 in the preference listings above can be broken down into standard AC rule components.

The rules in the table in Figure 6.7 can be presented as structured natural language, as defined below. This language and structure is inspired by the syntax in [Wikg], and brings us closer to a machine-readable format:

1.

```
Deny access to resource Photos with attribute OwnerID="
y"
  if SubjectRoles contain "Stranger"
  and action is read
```

2.

```
Allow access to resource PhotoAlbum with attribute
  AlbumID="x" and OwnerID="y"
  if SubjectRoles contain "Close_friend"
  and if SubjectRoles do not contain "Family"
  and action is read
```

3.

```
Allow access to resource Chat with attribute OwnerID="x
"
  if SubjectRoles do contain "Colleague"
  and if OwnerLocation="Bank_of_America"
  and action is read or write
```

4.

```
Allow access to resource Locations with attribute
  OwnerID="x"
  if SubjectRoles do contain "Colleague" or "
  Acquaintances"
  and if Time > 08.00
  and if Time < 16.00
  and action is read
```

5.

```
Allow access to resource SearchRecord with attribute
  OwnerID="x"
  if SubjectNetwork = OwnerNetwork
  and action is read
```

Further these structured natural language rules can be translated into the language called Datalog, referenced in [BCFP03], and explained in [Cer90].

Datalog can be directly embedded into SQL-based DBMS, including Cassandra (used by Facebook).

1.

```
rule:: if
  Attribute ResourceType=PhotoAlbum
and
  Attribute OwnerID=x
and
  SubjectRoles isa Stranger
and
  Action=read
then
  may-not-access
```

2.

```
rule:: if
  Attribute ResourceType=PhotoAlbum
and
  Attribute AlbumID=x
and
  Attribute OwnerID=y
and
  SubjectRoles isa Close Friend
and
  SubjectRoles is not Family
and
  Action=read
then
  may-access
```

3.

```
rule:: if
  Attribute ResourceType=Chat
and
  Attribute OwnerID=x
and
  SubjectRoles isa Colleague
and
  OwnerLocation="Bank_of_America"
and
```

```
Action=read or Action=write
then
  may-access
```

4.

```
rule:: if
  Attribute ResourceType=Locations
and
  Attribute OwnerID=x
and
  SubjectRoles isa Colleague or Acquaintance
and
  Time > 08.00
and
  Time < 16.00
and
  Action=read
then
  may-access
```

5.

```
rule:: if
  Attribute ResourceType=SearchRecord
and
  Attribute OwnerID=x
and
  SubjectNetwork=OwnerNetwork
and
  Action=read
then
  may-access
```

These Datalog rules and their components can be translated into XACML rules. With XACML the rules are represented in a standard way, and one can reap the benefits from the standardization work done with the language. Benefits include using a standardized representation of rules instead of creating one themselves, being able to export privacy preferences to other systems, plus using the XACML enforcement architecture explained in Section 2.2.5.

XACML is an elaborate language with much syntax and many words to describe each rule. It can thus be challenging to understand the rules at first glance, because of the heavy syntax. Therefore, while we have trans-

lated all the five rules into XACML syntax, only one of them is included and thoroughly explained in this chapter as an example. A complete XACML representation of all the rules can be found in Appendix A: "XACML examples". The chosen rule will be presented in a simplified format for the sake of the explanation. To best present the rule elements through all the syntax, we will not include the complete paths of all the DataTypes, MatchIds, FunctionIDs, AttributeIDs and similar. One example of this is; instead of writing 'DataType="http://www.w3.org/2001/XMLSchema#string"' we will just write 'DataType="string"'. We have chosen to use rule number 4, based on the user preference "As a business woman, I only want my colleagues and acquaintances to see the places I have checked in to during business hours". This rule contains both of the enhancements we have proposed in this chapter, user-administered role-based access control and rule conditions, in this case based on the external attribute "Time".

The blue text are comments describing which resource/subject/condition variables the code below reflects. The red text highlights the most important elements of the different rule-components.

```
<Rule Effect="Permit">
  <Target>
    <Subjects>
      <AnySubject/>
    </Subjects>
    <Resource>
      Resource Type=Locations
      <ResourceMatch MatchId="...string-equal">
        <AttributeValue DataType="string">
          Locations
        </AttributeValue>
        <ResourceAttributeDesignator DataType="string"
          AttributeId="...resource-type" />
      </ResourceMatch>
      OwnerID=x
      <ResourceMatch MatchId="...string-equal">
        <AttributeValue DataType="string">
          x
        </AttributeValue>
        <ResourceAttributeDesignator DataType="string"
          AttributeId="...resource:owner-id" />
      </ResourceMatch>
    </Resource>
  </Target>
</Rule>
```

```

    <Action "read">
    </Target>
    <Condition>
    <Apply FunctionId="and">
    SubjectRole=Colleague
    <Apply FunctionId="...string-at-least-one-member-of">
    <SubjectAttributeDesignator DataType="string"
    AttributeId="...subject:roles" />
    <Apply FunctionId="...:string-bag">
    <AttributeValue DataType="string">
    Colleague
    </AttributeValue>
    <AttributeValue DataType="string">
    Acquaintance
    </AttributeValue>
    </Apply>
    </Apply>
    Time > 08:00
    <Apply FunctionId="...time-greater-than-or-equal">
    <Apply FunctionId="...:time-one-and-only">
    <EnvironmentAttributeSelector DataType="time"
    AttributeId="...environment:current-time" />
    </Apply>
    <AttributeValue DataType="time">
    08:00:00
    </AttributeValue>
    </Apply>
    Time < 16:00
    <Apply FunctionId="...time-less-than-or-equal">
    <Apply FunctionId="...:time-one-and-only">
    <EnvironmentAttributeSelector DataType="time"
    AttributeId="...environment:current-time" />
    </Apply>
    <AttributeValue DataType="time">
    16:00:00
    </AttributeValue>
    </Apply>
    </Apply>
    </Condition>
  </Rule>

```

Through this translation exercise we have shown that the enhancements we propose will not only cover the relevant requirements and make sense from a user perspective, but also they are representable in languages which have an existing enforcement architecture, like Datalog and XACML. The developers of existent and future LASNS systems wishing to implement the privacy control enhancements suggested in this thesis using the XACML framework will thus have to do the following:

- Create a role/relationship representation for connections ("friendships") in the social graph.
- Create a GUI for role/relationship management.
- Change/add to the back-bone such that it handles subjects, objects, attributes, access types and corresponding operators and values as rule elements.
- Create the user GUI, where drop-down menus and field restrictions reflect the possible rule elements, for the users to create enhances access control rules.
- Program the translation of the input from the GUI into the XACML format.
- Implement the necessary XACML rule enforcement architecture.

6.3 Comparison with existent solutions

In previous chapters we have identified privacy control requirements, based on various possible user preferences, which are missing or inadequate in current systems. What was lacking was a good way for users to control access to their objects based on the following variables; object types, owner relation to subject, location, time, plus owner and subject attributes. One of our goals have been to present user privacy control enhancements that will, when implemented in LASNS systems, fulfill these requirements. The enhancements we have proposed in this chapter; more fine-grained subject separation and the addition of fine-grained rule conditions, thus represents an improvement to existent solutions because it allows the users to make their own access control rules based on the variables just mentioned. This has been shown through five example preferences, one for each of the five variables, translated into a formal representation usable in a system. All of these five preferences

could not have been reflected in a system where our two proposed enhancements were not implemented, thus we can say that they are both essential for fulfilling our stated LASNS privacy control requirements.

Another important goal has been to make the enhancements effective and understandable from a user perspective, plus implementable in LASNSs from a developer perspective.

User-perspective We have proposed a possible GUI for an enhanced end-user privacy control panel, the goal being that both general privacy control, through current coarse-grained panels should be possible, yet more fine-grained access control rules should be specifiable. Different users have different preferences, and the goal has been to present a solution that will work for all kinds of user, with all kinds of different (realistic) preferences. In the proposed GUI we have therefore retained the more general privacy control, as in Facebook's privacy control panel, where users can specify high-level rules. This will create an easy way for people to create the more general rules based on preferences which current panels can reflect, plus satisfy the needs of people who do not want to or who are not confident creating new and possibly more demanding fine-grained rules. Yet, our proposed GUI will provide those who do wish to specify more fine-grained access control rules the option to do so. We created the GUI for these specialized rules with the focus on being both effective and understandable, creating a panel where drop-down menus allows users with no experience with AC rule logic to create such rules.

Developer perspective To show that our enhancements are implementable from a developer perspective, we have shown how the five preferences, provided to the system through the proposed GUI, can be translated into logic rules and to more machine readable formats (XACML or Datalog). This ties our proposed enhancements, developed from a user perspective, to representations in known and researched formats. There are multiple benefits to doing this; It might be possible to reuse the user preferences expressed in a platform-independent language to exchange or use/reuse privacy policies in other systems. If a user has a privacy policy profile in one LASNS, with a defined set of rules, these could possibly be transferred and reused in other systems where the same user has an account. Another benefit of translating these preferences into any formal logical representation is that it allows us to do security analysis of such rules. One example of this is to be able to check consistency and safety through formal analysis. Both Datalog and XACML contain a framework for making such analysis.

Chapter 7

Future work

The next natural step is to try out these enhancement ideas by mapping XACML into an example system with a fair amount of data, and develop the backbone and GUI such that the suggestions in Chapter 6 can be evaluated in practice.

The enhancements suggested in this thesis relies on the user to exploit the possibilities, making their own access control rules. Many users do not care about their privacy, and/or do not understand the consequences of sharing too much information. It is also possible that users have privacy preferences that conflicts with their LASNS account privacy control settings, because they are not aware of this or they do not know how to reflect them through these panels. Useful work can thus be directed to helping the users obtain a healthy amount of privacy awareness, and further have them understand how they can reflect their preferences through privacy control panels.

Through the usage of LASNSs, we can try to improve user privacy control panels, yet how can we trust the people behind these services not to abuse our personal information? They are the ones responsible for enforcing the access control, and the only way to protect your information from such abuse is to make sure only the information you wish to share with the system leaves your computer, phone or other device. To tighten privacy control even more, one can research ways to "move" the policy decision and enforcement points to the client side, where the clients can control exactly what information leaves their device. This point is even more important when it comes to apps developed by people who you have no reason to trust (third-party developers of iOS, Android, Facebook, Twitter -based apps and programs for instance). These people can, through open APIs develop popular apps with little resources, lowering the threshold for who are able to reach the users. It would be much more safe to use these kinds of LASNSs if the user could control which information leaves their device from the client side. Other

important work in relation to this problem is to get the providers of popular open APIs to create stricter rules as to how the developers using their framework handles user information.

The LASNSs used today are mostly made for fun and social interactions. Still, SNSs have made their way into corporate environments with systems such as Socialcast, and so could LASNSs. Creating a corporate environment SNS or LASNS is different than making a system open to whoever signs up and where each user is able to decide whether their information is public or not. A corporate setting is different as it is possible to control who the system is open to, because it is possible to limit who gets an account to only include employees or other people with some sort of connection to the company in a practical way. Another difference is that corporate settings often demand more privacy control and can also demand more centralized confidentiality control. There is a much bigger need for information flow control to protect sensitive business information. Future research can go into how access control can be used in such a LASNS setting, possibly using the proposed access control enhancement; RBAC, with a combination of the user as the system administrator to non-sensitive information objects, and a central system administrator to control access to sensitive information. In such a setting it might make sense to use more elements from the RBAC framework, introducing Hierarchical RBAC for instance.

In addition to this possible future work, topics related to this thesis are proposed as student project ideas for master students at the Institute of Telematics at NTNU, and a conference article extending this work is planned.

Chapter 8

Conclusions

In the introduction six research questions were defined.

”1. What kind of access control features exist in current LASNS to control end-user’s privacy?”

We have discovered, through analysis and discussion, that in our opinion, the users of existing LASNSs have very limited and inadequate tools for protecting how their data is shared and treated.

”2. What kind of privacy preferences may end-users have in LASNSs?”

Through imagined scenarios and experience in social networks, we have established that users may have many different fine-grained privacy preferences. Users wish to control who can access which data (related to them), in what way, and under which conditions. As LASNS contain sensitive personal data, users should demand such fine-grained access control.

”3. Are existent access control features in LASNSs able to satisfy end-user’s privacy requirements/preferences?”

Based on the different privacy preferences users might have, we created a set of end-user privacy control requirements. None of the examined LASNS control panels fulfilled all the requirements, some not fulfilling a single one.

”4. Which privacy-enhancing access control features in LASNSs should be added (or improved in which way) to satisfy end-user’s requirements/preferences? (Illustrated with examples.)”

Based on our established end-user privacy requirements, we have identified the need for two proposed enhancements. First, the need for more fine-grained subject separation. The ability to divide the potential subjects into

groups, based on their relationship/role towards the user, and make access control rules based on these roles/relationships. Second of all, make sure users can specify fine-grained conditions to these rules, reflecting under which conditions a rule should be applied. Together these two enhancements help fulfill all the identified requirements.

”5. How can the privacy-enhancing access control features be represented to end-users?”

Different users have different privacy preferences. A GUI for end-user privacy control should be both effective in reflecting different user’s different privacy preferences, yet as understandable and user-friendly as possible. We have therefore proposed an interface with both a more general way of representing access control rules (similar to Facebook’s current privacy control panel), with system-specified object types and access types, and a user-specifiable subject, yet with an option for specifying tailored access control rules with the combination of rule effect, object types, subject/roles, access types, and conditions through easily understandable drop-down menus and variable specification.

”6. How can the privacy-enhancing access control features be represented in terms of logic rules and in machine readable format (e.g. XACML)?”

With our proposed enhanced GUI the system will receive a defined set of data. We have shown that data from the GUI input of five example user preferences, one representing each of our end-user privacy requirements, can be translated into both Datalog and XACML logic rules. This shows that our two enhancements, developed from an end-user perspective are also implementable from a developer perspective, and could be further analyzed and extended through the use of formal languages.

LASNSs will usually demand a balance between trust and access control. Sometimes, in order for the service to be of value to the user, one might have to make sacrifices when it comes to privacy. Still, users should demand better privacy control than what is offered in existing LASNSs. Locational information, paired with information from SNSs, can represent sensitive personal information, and the user should be able to choose to whom, how and when it is shared. Our two proposed enhancements fulfill this requirement, and it is shown that they work from both a GUI and implementation point of view. The next step would be to examine the consistency of the logic rules from a security point of view, and naturally, to implement these enhancements to test them in practice.

Appendix A

XACML examples

This Appendix consists of a more complete translation from the following list of user preferences discussed in Chapter 6 into XACML rules.

1. "As a regular user, I wish to hide my photos from strangers."
 2. "As a 25 year old guy, I want to show the photo album from the guy-trip to London only to my close friends, except the close friends who are also in my family."
 3. "As a Facebook Places user, I would like for my colleagues to be able to chat with me only when I am at work, at Bank of America."
 4. "As a business woman, I only want my colleagues and acquaintances to see the places I have checked in to during business hours."
 5. "As a student, I want only the people in the same network as me to be able to find me in searches."
- 1.

```
<Rule Effect="Deny">
  <Target>
    <Subjects>
      <AnySubject />
    </Subjects>
    <Resource>
      <!-- ResourceType=PhotoAlbum -->
      <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0
        :function:string-equal">
        <AttributeValue DataType="http://www.w3.org/2001/
          XMLSchema#string">PhotoAlbum</AttributeValue>
```

```

    <ResourceAttributeDesignator DataType="http://www.
      w3.org/2001/XMLSchema#string" AttributeId="
        urn:variations2:resource-type" />
  </ResourceMatch>
  <!-- OwnerID=x -->
  <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0
    :function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/
      XMLSchema#string">x</AttributeValue>
    <ResourceAttributeDesignator DataType="http://www.
      w3.org/2001/XMLSchema#string" AttributeId="
        urn:xacml:2.0:interop:example:resource:owner-id"
      />
  </ResourceMatch>
</Resource>
<Action "read">
</Target>
<Condition>
  <!-- SubjectRole=Stranger -->
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
    :function:string-at-least-one-member-of">
    <SubjectAttributeDesignator DataType="http://www.w3.
      org/2001/XMLSchema#string" AttributeId="...
        :subject:roles" />
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:string-bag">
      <AttributeValue DataType="http://www.w3.org/2001/
        XMLSchema#string">Stranger</AttributeValue>
    </Apply>
  </Apply>
</Condition>
</Rule>

```

2.

```

<Rule Effect="Permit">
  <Target>
    <Subjects>
      <AnySubject />
    </Subjects>
    <Resource>
      <!-- ResourceType=PhotoAlbum -->

```

```

<ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0
:function:string-equal">
  <AttributeValue DataType="http://www.w3.org/2001/
XMLSchema#string">PhotoAlbum</AttributeValue>
  <ResourceAttributeDesignator DataType="http://www.
w3.org/2001/XMLSchema#string" AttributeId="
urn:variations2:resource-type"/>
</ResourceMatch>
<!-- AlbumID=x -->
<ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0
:function:string-equal">
  <AttributeValue DataType="http://www.w3.org/2001/
XMLSchema#string">x</AttributeValue>
  <ResourceAttributeDesignator DataType="http://www.
w3.org/2001/XMLSchema#string" AttributeId="...
:resource:album-id"/>
</ResourceMatch>
<!-- OwnerID=y -->
<ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0
:function:string-equal">
  <AttributeValue DataType="http://www.w3.org/2001/
XMLSchema#string">y</AttributeValue>
  <ResourceAttributeDesignator DataType="http://www.
w3.org/2001/XMLSchema#string" AttributeId="
urn:xacml:2.0:interop:example:resource:owner-id"
/>
</ResourceMatch>
</Resource>
<Action "read">
</Target>
<Condition>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
:function:and">
    <!-- SubjectRole=Close friend -->
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
:function:string-at-least-one-member-of">
      <SubjectAttributeDesignator DataType="http://www.w3
.org/2001/XMLSchema#string" AttributeId="...
:subject:roles"/>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
:function:string-bag">

```

```

    <AttributeValue DataType="http://www.w3.org/2001/
      XMLSchema#string">Close friend</AttributeValue>
  </Apply>
</Apply>
<!-- SubjectRole != Family -->
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0
  :function:not">
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
    :function:string-at-least-one-member-of">
    <SubjectAttributeDesignator DataType="http://www.
      w3.org/2001/XMLSchema#string" AttributeId="...
      :subject:roles"/>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:string-bag">
      <AttributeValue DataType="http://www.w3.org/2001/
        XMLSchema#string">Stranger</AttributeValue>
    </Apply>
  </Apply>
</Apply>
</Apply>
</Condition>
</Rule>

```

3.

```

<Rule Effect="Permit">
  <Target>
    <Subjects>
      <AnySubject />
    </Subjects>
    <Resource>
      <!-- ResourceType=Chat -->
      <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0
        :function:string-equal">
        <AttributeValue DataType="http://www.w3.org/2001/
          XMLSchema#string">Chat</AttributeValue>
        <ResourceAttributeDesignator DataType="http://www.
          w3.org/2001/XMLSchema#string" AttributeId="
          urn:variations2:resource-type"/>
      </ResourceMatch>
      <!-- OwnerID=x -->
    </Resource>
  </Target>

```



```

    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0
      :function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/
        XMLSchema#string">x</AttributeValue>
      <ResourceAttributeDesignator DataType="http://www.
        w3.org/2001/XMLSchema#string" AttributeId="
          urn:xacml:2.0:interop:example:resource:owner-id"
        />
    </ResourceMatch>
  </Resource>
  <Actions>
    <Action "read">
    <Action "write">
  </Actions>
</Target>
<Condition>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
    :function:and">
    <!-- SubjectRole=Colleague -->
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:string-at-least-one-member-of">
      <SubjectAttributeDesignator DataType="http://www.w3
        .org/2001/XMLSchema#string" AttributeId="...
          :subject:roles" />
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
        :function:string-bag">
        <AttributeValue DataType="http://www.w3.org/2001/
          XMLSchema#string">Stranger</AttributeValue>
      </Apply>
    </Apply>
    <!-- OwnerLocation=Bank of America -->
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/
        XMLSchema#string">Bank of America</
          AttributeValue>
      <ResourceAttributeDesignator DataType="http://www.
        w3.org/2001/XMLSchema#string" AttributeId="...
          :resource:owner:location" />
    </Apply>
  </Apply>

```

```
</Condition>
</Rule>
```

4.

```
<Rule Effect="Permit">
  <Target>
    <Subjects>
      <AnySubject />
    </Subjects>
    <Resource>
      <!--ResourceType=Locations-->
      <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0
        :function:string-equal">
        <AttributeValue DataType="http://www.w3.org/2001/
          XMLSchema#string">Locations</AttributeValue>
        <ResourceAttributeDesignator DataType="http://www.
          w3.org/2001/XMLSchema#string" AttributeId="
            urn:variations2:resource-type" />
      </ResourceMatch>
      <!--OwnerID=x-->
      <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0
        :function:string-equal">
        <AttributeValue DataType="http://www.w3.org/2001/
          XMLSchema#string">x</AttributeValue>
        <ResourceAttributeDesignator DataType="http://www.
          w3.org/2001/XMLSchema#string" AttributeId="
            urn:xacml:2.0:interop:example:resource:owner-id"
          />
      </ResourceMatch>
    </Resource>
    <Action "read">
  </Target>
  <Condition>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:and">
      <!--SubjectRole=Colleague-->
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
        :function:string-at-least-one-member-of">
        <SubjectAttributeDesignator DataType="http://www.w3
          .org/2001/XMLSchema#string" AttributeId="...
            :subject:roles" />
```

```

    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:string-bag">
      <AttributeValue DataType="http://www.w3.org/2001/
        XMLSchema#string">Colleague</AttributeValue>
      <AttributeValue DataType="http://www.w3.org/2001/
        XMLSchema#string">Acquaintance</AttributeValue>
    </Apply>
  </Apply>
  <!--Time > 08:00-->
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
    :function:time-greater-than-or-equal"
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:time-one-and-only">
      <EnvironmentAttributeSelector DataType="http://www
        .w3.org/2001/XMLSchema#time" AttributeId="
        urn:oasis:names:tc:xacml:1.0
        :environment:current-time"/>
    </Apply>
    <AttributeValue DataType="http://www.w3.org/2001/
      XMLSchema#time">08:00:00</AttributeValue>
  </Apply>
  <!--Time < 16:00-->
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
    :function:time-less-than-or-equal"
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:time-one-and-only">
      <EnvironmentAttributeSelector DataType="http://www
        .w3.org/2001/XMLSchema#time" AttributeId="
        urn:oasis:names:tc:xacml:1.0
        :environment:current-time"/>
    </Apply>
    <AttributeValue DataType="http://www.w3.org/2001/
      XMLSchema#time">16:00:00</AttributeValue>
  </Apply>
</Apply>
</Condition>
</Rule>

```

5.

```

<Rule Effect="Permit">
  <Target>

```

```

<Subjects>
  <AnySubject />
</Subjects>
<Resource>
  <!-- ResourceType=SearchRecord -->
  <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0
    :function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/
      XMLSchema#string">SearchRecord</AttributeValue>
    <ResourceAttributeDesignator DataType="http://www.
      w3.org/2001/XMLSchema#string" AttributeId="
        urn:variations2:resource-type" />
  </ResourceMatch>
  <!-- OwnerID=x -->
  <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0
    :function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/
      XMLSchema#string">x</AttributeValue>
    <ResourceAttributeDesignator DataType="http://www.
      w3.org/2001/XMLSchema#string" AttributeId="
        urn:xacml:2.0:interop:example:resource:owner-id"
      />
  </ResourceMatch>
</Resource>
<Action "read">
</Target>
<Condition>
  <!-- SubjectNetwork=OwnerNetwork -->
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
    :function:string-equal">
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:string-one-and-only">
      <SubjectAttributeDesignator DataType="http://www.w3
        .org/2001/XMLSchema#string" AttributeId="...
          :subject:network" />
    </Apply>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:string-one-and-only">
      <ResourceAttributeDesignator DataType="http://www.
        w3.org/2001/XMLSchema#string" AttributeId="...
          resource:owner:network" />
    </Apply>
  </Apply>
</Condition>

```

```
</Apply>  
</Apply>  
</Condition>  
</Rule>
```


Bibliography

- [Aba03] Martín Abadi. Logic in access control, 2003.
- [AHK⁺] Paul Ashley, Satoshi Hada, Gunter Karjoth, Calvin Powers, and Matthias Schunter (IBM). Enterprise privacy authorization language (epal 1.2). Submitted to The World Wide Web Consortium (W3C) 10 November 2003 <http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/>. Last modified at time of access: November 19. 2003.
- [Als09] Colin Alsheimer. 11 location based applications for your iphone. <http://www.leveltendesign.com/blog/colin/11-location-based-applications-your-iphone>, July 2009. Last modified at time of access: May 30. 2011.
- [And] Talk Android. Loopt for android: Mini review loopt. <http://www.talkandroid.com/20099-appbyte-loopt-for-android/loopt/>. Last modified at time of access: May 30. 2011.
- [And05] Anne Anderson. A comparison of two privacy policy languages: Epal and xacml. Technical report, Sun Microsystems, Inc., 2005.
- [Art11] Charles Arthur. iphone keeps record of everywhere you go. At [guardian.co.uk](http://www.guardian.co.uk) <http://www.guardian.co.uk/technology/2011/apr/20/iphone-tracking-prompts-privacy-fears>, April 2011. Last modified at time of access: May 30. 2011.
- [BBF01] Elisa Bertino, Piero Andrea Bonatti, and Elena Ferrari. Trbac: A temporal role-based access control model. *ACM Transactions on Information and System Security (TISSEC)*, 4:191–233, August 2001.

- [BCFP03] Elisa Bertino, Barbara Catania, Elena Ferrari, and Paolo Perlasca. A logical framework for reasoning about access control models. *ACM Transactions on Information Systems Security*, 6(1):71–127, 2003.
- [Bod11] Karl Bode. iphone keeps record of everywhere you go. At DSL reports <http://www.dslreports.com/shownews/iPhone-Location-File-Not-New-Not-Secret-113850>, April 2011. Last modified at time of access: April 21. 2011.
- [BP76] D. E. Bell and L. J. La Padula. Secure computer system: Unified exposition and multics interpretation, 1976.
- [Cas] Apache Cassandra. Welcome to apache cassandra. <http://cassandra.apache.org/>. Last modified at time of access: May 20. 2011.
- [Cer90] Stefano Ceri. *Logic programming and databases*. Springer-Verlag, 1990.
- [Clu11] Graham Cluley. Apple ios update quashes location tracking "bug". At naked security http://nakedsecurity.sophos.com/2011/05/05/apple-ios-update-quashes-location-tracking-bug/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+nakedsecurity+%28Naked+Security+++Sophos%29, May 2011. Last modified at time of access: May 30. 2011.
- [Cona] The Open Geospatial Consortium. Request for comment on geospatial extensible access control markup language (geoxacml). <http://www.opengeospatial.org/standards/requests/41>. Last modified at time of access: September 7. 2007.
- [Conb] The World Wide Web Consortium. Platform for privacy preferences (p3p) project. <http://www.w3.org/P3P/>. Last modified at time of access: November 20. 2007.
- [Conc] The World Wide Web Consortium. W3c workshop on the future of p3p. <http://www.w3.org/2002/p3p-ws/Overview.html>. Last modified at time of access: January 7. 2003.

- [Dat] Aster Data. More data, big insights. <http://www.asterdata.com/index.php>. Last modified at time of access: April 19. 2011.
- [DBCP05] Maria Luisa Damiani, Elisa Bertino, Barbara Catania, and Paolo Perlasca. Geo-rbac: a spatially aware rbac. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, SACMAT '05, pages 29–37. ACM, 2005.
- [Dee10] Deeplinks. Facebook’s eroding privacy policy: A timeline. At Electronic Frontier Foundation <http://www.eff.org/deeplinks/2010/04/facebook-timeline>, April 2010. Last modified at time of access: May 30. 2011.
- [Deva] Android Developers. Obtaining user location. <http://developer.android.com/guide/topics/location/obtaining-user-location.html>. Last modified at time of access: May 20. 2011.
- [Devb] Facebook Developers. Core concepts. <http://developers.facebook.com/docs/coreconcepts/>. Last modified at time of access: March 29. 2011.
- [Dig] Digitizor. ”facebook places unveiled foursquare in danger? <http://files.digitizor.com/wp-content/uploads/2010/08/facebook-places.jpg>. Last modified at time of access: August 19. 2010.
- [dot10] dotRIGHTS. Facebook places: Your friends are here, but what about your privacy? <http://www.dotrights.org/facebook-places-your-friends-are-here-what-about-your-privacy>, August 2010. Last modified at time of access: May 30. 2011.
- [DuV10] Adam DuVander. Facebook location api launches: Read only for now. At programmable web <http://blog.programmableweb.com/2010/08/18/facebook-location-api-launches-read-only-for-now/>, August 2010. Last modified at time of access: May 30. 2011.
- [Fac10] Facebook. Facebooks privacy policy. <http://www.facebook.com/policy.php>, December 2010. Last modified at time of access: April 14. 2011.

- [Fer10] Elena Ferrari. *Access Control in Data Management Systems*. Synthesis Lectures on Data Management. Morgan and Claypool Publishers, 2010.
- [fITS04] InterNational Committee for Information Technology Standards. Role based access control. Technical Report 359-2004, American National Standards Institute, February 2004.
- [Fle08] Glenn Fleishman. Apple adds iphone location over wi-fi, base station backup, macbook air. At Wi-Fi Net News (WNN) http://wifinetnews.com/archives/2008/01/apple_adds_iphone_location_over_wi-fi_base_station_backup_ma.html, January 2008. Last modified at time of access: February 7. 2011.
- [ftAoSISO] Organization for the Advancement of Structured Information Standards (OASIS). About oasis. <http://www.oasis-open.org/who/>. Last modified at time of access: March 16. 2011.
- [ftAoSISO03] Organization for the Advancement of Structured Information Standards (OASIS). extensible access control markup language (xacml) version 1.0. Standard, OASIS, February 2003.
- [ftAoSISO10] Organization for the Advancement of Structured Information Standards (OASIS). Xacml v3.0 privacy policy profile version 1.0. Standard, OASIS, August 2010.
- [FV03] Hansen F. and Oleshchuk V.A. Srbac: A spatial role-based access control model for mobile systems. In *Proceedings of the 8th Nordic Workshop on Secure IT Systems*,, NORDSEC03, pages 129–141, October 2003.
- [Gol09] Jay Goldman. *Facebook Cookbook*. O’Reilly Media Inc., 2009.
- [Gre08] Chris Green. Apple iphone 3g review. At Know your mobile http://www.knowyourmobile.com/appleiphone/iphone-reviews/92168/apple_iphone_3g_review.html, July 2008. Last modified at time of access: May 30. 2011.
- [Hen10] W.J. Hennigan. Foursquare signs deal to develop tv show. At L.A. Times blogs <http://latimesblogs.latimes.com/technology/2010/12/foursquare-signs-deal-to-develop-television-show>.

- html, December 2010. Last modified at time of access: May 30. 2011.
- [Ide08] Web Identity. Privacy policies of social networking sites: Facebook and myspace. <http://webidentity.wikidot.com/social-networking>, February 2008. Last modified at time of access: May 30. 2011.
- [Inc] The Open Geospatial Consortium Inc. Welcome to the ogc website. <http://www.opengeospatial.org/>. Last modified at time of access: March 25. 2011.
- [JFM10] Audun Josang, Lothar Fritsch, and Tobias Mahler. Privacy policy referencing. In *Trust, Privacy and Security in Digital Business*, Lecture Notes in Computer Science, pages 129–140. Springer Berlin / Heidelberg, 2010.
- [Lib] Mac OS X Developer Library. Core location framework reference. http://developer.apple.com/library/mac/#documentation/CoreLocation/Reference/CoreLocation_Framework/_index.html. Last modified at time of access: November 4. 2010.
- [nav10] navigadget.com. Viewranger now good for iphone and android. <http://www.navigadget.com/index.php/2010/05/01/viewranger-now-good-for-iphone-and-android>, May 2010. Last modified at time of access: May 30. 2011.
- [NB] Qun Ni and Elisa Bertino. Conditional privacy-aware role based access control.
- [oD85] Department of Defense. Trusted computer system evaluation criteria, 1985.
- [O’D10] Jolie O’Dell. A field guide to using facebook places. At Mashable <http://mashable.com/2010/08/19/facebook-places-guide/>, August 2010. Last modified at time of access: May 30. 2011.
- [Par10] Lucian Parfeni. Facebook opens up read/write places api to challenge google’s. At Softpedia <http://news.softpedia.com/news/Facebook-Opens-Up-Read-Write-Places-API-to-Challenge-Google-s-164741>.

- shtml, November 2010. Last modified at time of access: May 30. 2011.
- [sA09] Lillian Røstad and Ole Andreas Alsos. Patient-administered access control: A usability study. *Availability, Reliability and Security, International Conference on*, 0:877–881, 2009.
- [Shi10] Debra Littlejohn Shinder. Smartphone location and navigation services. At Tech Republic <http://www.techrepublic.com/blog/smartphones/smartphone-location-and-navigation-services/1750>, October 2010. Last modified at time of access: May 30. 2011.
- [Tin10] Roger Erik Tinch. Gowalla vs foursquare. <http://www.retinch.com/gowalla-vs-foursquare/>, January 2010. Last modified at time of access: May 30. 2011.
- [Tip10] Battery Tips. Gowalla3.0 compatible news. <http://www.batterytip.info/2010/12/03/gowalla3-0-compatible-news/>, December 2010. Last modified at time of access: May 30. 2011.
- [Ubi10] UbiCompForAll. Scenarios. <http://www.sintef.no/Projectweb/UbiCompForAll/Results/Scenarios/>, September 2010. Last modified at time of access: May 30. 2011.
- [UoC] Santa Barbara University of California. Privacy notification statement. <http://engineering.ucsb.edu/privacy>. Last modified at time of access: March 9. 2011.
- [War06] Mark Ward. How the web went world wide. <http://news.bbc.co.uk/2/hi/science/nature/5242252.stm>, 2006. Last modified at time of access: August 3. 2006.
- [Wika] Wikipedia. Brightkite. <http://en.wikipedia.org/wiki/Brightkite>. Last modified at time of access: May 4. 2011.
- [Wikb] Wikipedia. foursquare (social network). http://en.wikipedia.org/wiki/Foursquare_%28social_network%29#Privacy. Last modified at time of access: May 3. 2011.
- [Wikc] Wikipedia. Geographic coordinate system. http://en.wikipedia.org/wiki/Geographic_coordinate_

system#cite_note-OSGB-0. Last modified at time of access:
April 26. 2011.

[Wikd] Wikipedia. Global positioning system. http://en.wikipedia.org/wiki/Global_Positioning_System. Last modified at time of access: April 27. 2011.

[Wike] Wikipedia. Loopt. <http://en.wikipedia.org/wiki/Loopt>. Last modified at time of access: May 18. 2011.

[Wikf] Wikipedia. Social network services. http://en.wikipedia.org/wiki/Social_networking_service. Last modified at time of access: May 26. 2011.

[Wikg] Wikipedia. Xacml. <http://en.wikipedia.org/wiki/XACML>. Last modified at time of access: May 8. 2011.

[Wil] Josh Williams. Go time for gowalla 3! At Gowalla blogs <http://blog.gowalla.com/post/2070359374/gowalla3>. Last modified at time of access: May 30. 2011.