

# Preface

This report is the result of the work with a Master's thesis in computer science at the Department of Computer and Information Science (IDI) at the Norwegian University of Science and Technology (NTNU) in Trondheim. The thesis has been developed in Spring 2005 as a contribution to the UbiCollab platform. UbiCollab is an on-going project in the field of Computer Supported Cooperative Work done in collaboration between IDI and Telenor Research and Development (Telenor FoU) in Trondheim.

We would like to thank Monica Divitini for very valuable guidance and support on writing the report. It has been very helpful to have someone with such insight in the problem area to help us focus our thoughts and refine our ideas into a presentable solution. Thanks is also due to Alan J Munro for supplying us with ideas taken from his vast reservoir of information in the borderline between computers and humans. We would also like to thank Babak Amin Farschian at Telenor FoU for helping us with the technical and architectural work on UbiCollab. Finally we would like to thank Carsten Heitmann and Børge Jensen for sharing ideas on how to enhance the UbiCollab platform.

Trondheim, June 21, 2005

Hans Steien Rasmussen  
Anders Magnus Braathen

# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Problem domain	8
1.1.1	Basic design issues for privacy in ubiquitous solutions	9
1.2	Project goal	10
1.3	Building on previous work on UbiCollab	11
1.3.1	Preserving privacy in a ubiquitous collaborative environment	11
1.4	Relation to the thesis by Heitmann and Jensen	12
1.5	UbiCollab - Conceptualized	13
1.5.1	Clients	13
1.5.2	Collaboration server	14
1.5.3	Services	14
1.5.4	Summary	14
1.6	Research method	14
1.7	Challenges and limitations	16
1.8	Structure	16
<b>2</b>	<b>Problem elaboration</b>	<b>18</b>
2.1	Scenarios and privacy in UbiCollab	18
2.2	Privacy management	19
2.3	Base scenario	21
2.4	Issues of privacy in UbiCollab	23
2.4.1	Anonymity	23
2.4.2	Identities and collaboration	24
2.4.3	Managing multiple user representations in UbiCollab	25
2.4.4	Reputation	28
2.4.5	Management of sensitive information	30
2.5	Conceptual requirements	31
2.6	Conclusion	32
<b>3</b>	<b>Related work</b>	<b>33</b>
3.1	Platform for Privacy Preferences (P3P) Project	33
3.1.1	P3P user agents	34
3.1.2	P3P specification	35
3.1.3	P3P Base data schema	37
3.1.4	P3P beyond HTTP	37
3.1.5	Evaluation	38

3.2	APPEL . . . . .	39
3.2.1	Rule processing and evaluation . . . . .	39
3.2.2	Evaluation . . . . .	40
3.3	A Privacy Awareness System for Ubiquitous Computing Environments (pawS) . . . . .	40
3.3.1	Privacy policies . . . . .	41
3.3.2	Policy announcement mechanisms . . . . .	41
3.3.3	Privacy proxies . . . . .	41
3.3.4	Evaluation . . . . .	42
3.4	WASP - Web Architectures for Services Platform . . . . .	42
3.4.1	Using P3P in WASP . . . . .	43
3.4.2	Evaluation . . . . .	44
<b>4</b>	<b>Analysis</b>	<b>45</b>
4.1	UbiCollab users . . . . .	45
4.2	Management of privacy and identity in UbiCollab . . . . .	46
4.3	Analysis of conceptual requirements . . . . .	49
4.4	Conclusion . . . . .	50
<b>5</b>	<b>Designing for privacy</b>	<b>52</b>
5.1	Introduction . . . . .	52
5.2	UbiCollab privacy review . . . . .	52
5.2.1	Components in the previous design . . . . .	53
5.2.2	Conclusion . . . . .	54
5.3	Privacy design issues . . . . .	54
5.4	Conceptual Model . . . . .	56
5.4.1	Low level conceptual model of UbiCollab . . . . .	57
5.4.2	Client . . . . .	57
5.4.3	Privacy proxy . . . . .	58
5.4.4	Trusted 3 <sup>rd</sup> party service . . . . .	59
5.4.5	Collaboration Server . . . . .	59
5.4.6	Services . . . . .	59
5.5	Redesigning UbiCollab - architectural overview . . . . .	59
5.5.1	Privacy Proxy . . . . .	60
5.5.2	P3P and APPEL . . . . .	62
5.5.3	Reputation . . . . .	63
5.5.4	Highly sensitive information . . . . .	65
5.6	UbiCollab interactions . . . . .	67
5.6.1	UbiCollab login . . . . .	67
5.6.2	User-initiated request . . . . .	68
5.6.3	Service initiated request . . . . .	69
<b>6</b>	<b>Prototype - UbiCollab privacy extension</b>	<b>71</b>
6.1	Scope of the prototype . . . . .	71
6.2	Platform overview . . . . .	72
6.3	Technology . . . . .	73
6.4	Privacy proxy . . . . .	73
6.5	User management . . . . .	74
6.6	User APPEL rulesets . . . . .	75

6.7	Service P3P policies . . . . .	76
6.8	Policy and preference evaluation . . . . .	76
6.9	Prototype limitations . . . . .	76
<b>7</b>	<b>Demonstration</b>	<b>78</b>
7.1	UbiClient . . . . .	78
7.2	User management . . . . .	79
7.2.1	Authenticate and login user . . . . .	80
7.2.2	Create a new user . . . . .	81
7.2.3	Logout . . . . .	82
7.3	Collaboration . . . . .	83
7.3.1	Create a collaboration instance . . . . .	83
7.3.2	Adding people to a collaboration instance . . . . .	84
7.4	P3P-enabled Demonstration Service . . . . .	85
<b>8</b>	<b>Conclusion</b>	<b>89</b>
8.1	Contribution . . . . .	89
8.2	Evaluation . . . . .	91
8.3	Future work . . . . .	93
	<b>References</b>	<b>94</b>
<b>A</b>	<b>Common Scenario</b>	<b>98</b>
<b>B</b>	<b>Glossary</b>	<b>100</b>
<b>C</b>	<b>UbiCollab API</b>	<b>103</b>
<b>D</b>	<b>Previous version of UbiCollab</b>	<b>107</b>
D.1	Conceptual Model . . . . .	107
D.1.1	Client . . . . .	108
D.1.2	Trusted AAA-server(Authentication, Authorization and Accounting) . . . . .	108
D.1.3	Collaboration server . . . . .	108
D.1.4	Directory Service . . . . .	109
D.1.5	Presence Service . . . . .	109
D.1.6	Privacy Service . . . . .	109
D.1.7	Location Service . . . . .	110
D.1.8	Resource Collector . . . . .	110

# List of Figures

1.1	UbiCollab . . . . .	13
1.2	Model Development [24] . . . . .	15
2.1	Collaboration instance with connected entities . . . . .	19
2.2	Different needs for multiple identities [29] . . . . .	27
3.1	P3P Transactions [16] . . . . .	34
3.2	P3P policy example . . . . .	36
3.3	Base data schema table: <code>postal</code> . . . . .	37
3.4	The pawS architecture [26] . . . . .	42
4.1	User representation . . . . .	46
4.2	Identity and privacy management in UbiCollab . . . . .	47
5.1	Privacy components in the previous UbiCollab design . . . . .	53
5.2	High level design of privacy in UbiCollab . . . . .	57
5.3	Conceptual model of the UbiCollab platform . . . . .	58
5.4	Architectural overview of the UbiCollab platform . . . . .	60
5.5	User identity data model . . . . .	61
5.6	P3P evaluation flowchart . . . . .	62
5.7	UbiCollab platform with reputation support . . . . .	64
5.8	Interaction and functionality in UbiCollab trusted 3 <sup>rd</sup> party service . . . . .	66
5.9	Login sequence on the UbiCollab platform . . . . .	68
5.10	User-initiated request of a UbiCollab service . . . . .	69
5.11	Service-initiated request in UbiCollab . . . . .	70
6.1	Platform prototype overview . . . . .	72
6.2	Privacy proxy overview . . . . .	74
6.3	Proxy user databases . . . . .	74
6.4	Collaboration server user database . . . . .	75
6.5	AppelEvaluator overview . . . . .	77
7.1	UbiClient user interface . . . . .	79
7.2	Table <code>ubicollabprivacy.userdata</code> . . . . .	80
7.3	Create new user . . . . .	81
7.4	Table <code>ubicollabprivacy.userdata</code> . . . . .	82
7.5	Table <code>ubicollabprivacy.useridentity</code> . . . . .	82
7.6	Table <code>ubicollab.user</code> . . . . .	82

7.7	Table <code>ubicollab.collaborationinstance</code> . . . . .	84
7.8	Table <code>ubicollabprivacy.usercollaborationinstance</code> . . . . .	85
7.9	Sample APPEL preference rule for user <code>andy@strict.example.com</code> . . . . .	86
7.10	Demo service P3P policy . . . . .	88
D.1	Conceptual model of UbiCollab . . . . .	107

# Chapter 1

## Introduction

Privacy has become an increasingly important area of focus when developing computer systems. Privacy has been a concern for people also before the age of computers and Brandeis and Warren defined privacy as "*The right to select what personal information about me is known to what people*" [25] as early as in the 19th century. In computer systems privacy mainly concerns how to control information about people. This form of privacy is called **information privacy** and it's this kind of privacy that will be discussed in this thesis.

The interest in privacy has in many ways grown due to the explosion in the number of services offered on the Internet. E-trade, banking services and other sites that store sensitive information about their users have traditionally focused mainly on security, but privacy is now becoming an equally important issue when ensuring that users trust the system enough to use it. Other systems dealing with less sensitive information also need to focus on privacy, as users are becoming aware of the risks of sharing personal information, and have started to worry about how this information might be misused.

Privacy in ubiquitous computing is an issue that has received much attention. The idea behind ubiquitous computing is to conceal the technology for the common user in order to integrate the systems and gadgets into everyday life. To be able to make users feel comfortable with using equipment that operates beyond their apparent functions, mechanisms for controlling the gathering and sharing of personal information have to be developed. For instance, if a coffee mug has the concealed functionality of a voice recorder for note-keeping reasons, users should be made aware of this fact, and be able to express consent to the recording of a conversation [25].

UbiCollab [13] is a platform that provides users with collaborative tools and services in a ubiquitous environment. The platform offers a set of basic functionality through its services and provides an API for developing new services on top of the platform. To be able to provide the users with an adequate amount of trust and the sense that the information they give away is being treated properly, the UbiCollab platform needs to respect the basic principles of privacy. UbiCollab will also be offering services based on location and positioning information, which makes it even more important to have a set of mechanisms for protecting the privacy of the individual. Location privacy is one of the main

areas of concern, as GPS and other tracking technologies now can be incorporated in a wide range of electronic devices. The idea of letting other people know your whereabouts at any given time is not a desired prospect and must be avoided or at least only be done on the request from the users themselves.

At the same time it's important in a collaborative environment that the users have a way of communicating with each other and share data. People that are working on the same project should have the possibility to share more personal information than what is normal with people that are unknown to each other. In a collaborative environment the users should also be able to take on different characteristics according to whom they are communicating with, to allow flexibility in cooperation and be able to give more information away to the people they know and trust. To be able to support collaboration on the platform, anonymity is not a good solution. Further mechanisms need to be studied to find a way to protect the users privacy, while at the same time keeping the collaboration on the platform flexible and ubiquitous.

This report will show the importance of preserving privacy in ubiquitous systems and present a solution to how this can be done on a service based platform like UbiCollab. The main focus will be on supplying users with the means to control the gathering and use of personal information, without inhibiting the functionality or making the system too advanced for its users.

The remaining of this chapter will describe the project idea in detail and the approach chosen for the work with privacy on the UbiCollab platform. Further it will offer insight into the problem domain and the way UbiCollab is today. Finally, a overview of the rest of the report is presented as a conclusion to introducing the thesis.

## 1.1 Problem domain

The work done on this project is in the field of ubiquitous and pervasive computing. The idea of ubiquitous computing was first introduced by Mark Weiser [40] at the Computer Science Lab at Xerox PARC in 1988, and has since received an increasing amount of attention as computers have come of age. The ambition of ubiquitous computing is to advance from the era of personal computers and introduce embedded processing capabilities in common placed devices used in everyday life. This could be described as the opposite of virtual reality, which aims to bring the world into the computer. The term pervasive computing is often mentioned in the same context, and involves using small, easy-to-use devices like handhelds to get access to information about anything and everything.

Ubiquitous computing poses a threat to the less technically advanced users who will have problems understanding what actions the environment or system performs when using it. These implications have received more attention lately, due to the increasing number of devices with embedded processing abilities and concealed functionality.

This project has its main focus on the privacy considerations encountered in this area. Privacy in ubiquitous computing is an important field of research with its many implications on the technical part of a system, but also concerning the way users perceive a system and the willingness to use it. Insight into what makes a user trust such a system is of great importance when designing for



privacy in ubiquitous environments.

### 1.1.1 Basic design issues for privacy in ubiquitous solutions

In our research project from Fall 2004, **”Preserving Privacy in a Ubiquitous Collaborative Environment: Extending the UbiCollab Platform”** [6], we stated the most important aspects which have to be covered when designing for privacy in ubiquitous solutions. Most important is the idea of designing for the users and preserving their needs and demands. The following aspects are presented as the most important issues for the users when reasoning about privacy in a system like UbiCollab, that handles personal information. The list is taken from Richard Beckwith’s article **”Designing for Ubiquity: The Perception of Privacy”** [4].

- Information receiver - Who uses/has access to the data
- Information usage - How will the data be used
- Information sensitivity - How sensitive is the data

Consequently these issues have been used as a basis for the previous suggestions to a UbiCollab privacy architecture. In accordance with Richard Beckwith [4], this has been done to ensure that the common user will trust and accept the actions of the system. In this thesis, these issues will be kept in mind while designing the system, to preserve the general trust in the system and to be able to build on previous work.

Further we pointed at the six most important principles of privacy design as presented by Langheinrich [25]. These principles are the basic building blocks for a privacy management system in ubiquitous computing. The principles are presented along with a description of the implications each issue has on the platform.

- Notice (Principle of openness) - The idea behind notice is to give the user information about what actions the system will perform. This principle should be handled by mechanisms that allows the system, and its services, to present its data practices to the user before the system performs an action.
- Choice and Consent - Choice and consent is the user’s response to the notice given by the system. The user will be able to decide whether or not to accept the actions performed by the system.
- Anonymity and pseudonymity - This principle promotes the user’s privacy by allowing the user to stay anonymous when using the services.
- Adequate security - Refined mechanisms for adequate security is outside the scope of this project, but the principle of security will be preserved when designing the architecture. The existing UbiCollab architecture supports this issue by having an authorization service for the users when they log in.

- Access and recourse - This principle covers the usage and access to information collected in a system. This is important for providing a trustworthy relationship between user and system, but will not be the focus of this thesis.
- Proximity and locality - Proximity and locality are effective mechanisms for handling user privacy. They can be implemented on the application side of the solution by allowing access to certain devices only when the user is in the proximity. This will not be the focus for this work, but they are interesting notions for future work on UbiCollab.

In order to build acceptable privacy mechanisms in a platform such as UbiCollab, all of these principles will have to be covered when designing the privacy architecture. Still, we have decided to focus on what we find most suitable and important for UbiCollab at this time. The issues we will focus on in this report are building privacy mechanisms based on the principle of **Notice, Choice and Consent** and **Anonymity and Pseudonymity**. These principles are most important for the basic privacy needs in UbiCollab, since they all cover the user's control over his own personal information. They are therefore very connected to the three aspects that users find most important regarding privacy as mentioned above. At the same time these principles offer mechanisms that are possible to implement in the existing platform without redesigning the whole platform.

Further information about privacy in general and its implications on pervasive computing can be found in our previous work [6], where we presented the general idea of privacy, the different types of privacy and different privacy legislation.

## 1.2 Project goal

The main focus of UbiCollab has been to develop a platform that provides the possibility for users to share mobile workspaces and allow both formal and informal collaboration in a ubiquitous environment. Formal communication has traditionally been the easiest way of communication to support, since it includes setting up meetings, sharing documents and other formal tasks. In ubiquitous computing, informal communication is equally important, since it is based on flexibility, the possibility to work from different locations and with different devices. This makes ubiquitous environments very suitable for supporting informal communication. The basic ideas of UbiCollab are presented in "UbiCollab: Collaboration support for mobile users" by Divitini, Farschian and Samset [13]. The platform is meant to provide an API that allows different services to be developed to support collaboration on top of a set of basic functionality supplied by the platform. The work on the platform in this thesis will lead to extensions that preserves the privacy of the users of the platform, while maintaining the existing functionality.

The main ambition of this thesis is to develop an architecture that supports the privacy principles presented in our research project from Fall 2004 [6]. The idea is to be able to give the user better control of what information is gathered and how it is used. This is thought done by introducing a way for the platform to give notice of what it will need from the user and by providing the user with the means necessary to agree to this or not. Further the thesis will discuss the representation of the users in UbiCollab and how previous actions on the platform may affect the way other users look at you.

Another area of importance is to keep the resulting solution consistent with the existing architecture and prototype. Even if this will take time and forcing us to focus on other things than privacy, the importance of having a consistent platform is essential.

The goal is to keep the platform as simple as possible and offer the basic functionality. On top of this there should be the possibility to develop solutions that preserves the notion of privacy, and thereby encourage users to trust that there's no threats to personal privacy when logging on to UbiCollab.

## 1.3 Building on previous work on UbiCollab

This thesis is based on the existing UbiCollab platform developed through different project work at IDI (NTNU) with guidance from Telenor FoU. One important contribution to UbiCollab have been made by Christian Schwarz with his Master's thesis from spring 2004; "**UbiCollab - Platform for supporting collaboration in a ubiquitous computing environment**" [37] In this work a prototype has been implemented that realizes a subset of the services required to fulfill the demands that have been stated in the model of UbiCollab. The main services of the UbiCollab platform consist of management of collaborative efforts, people, resources, presence, privacy and location.

### 1.3.1 Preserving privacy in a ubiquitous collaborative environment

The earlier work has been a good foundation for our work on the platform, by presenting a prototype of the platform and discussing important issues concerning further development of UbiCollab. The work by Schwarz was used as a foundation for our work on privacy during the research project of fall 2004 - "**Preserving Privacy in a Ubiquitous Collaborative Environment: Extending the UbiCollab Platform**" [6]. The report includes a discussion on relevant aspects of privacy, and identifies the main challenges of incorporating these aspects in the design of a platform such as UbiCollab. The report contains a literature review of the research field, with analysis of technological solutions used in similar projects. An important privacy contribution is a redesign of the UbiCollab architecture, which enables anonymous use of UbiCollab services.

The research project covers the fundamentals of privacy, and addresses the issues that we felt required immediate attention. As a starting point for this Master's thesis, it has provided a good insight on UbiCollab and the area of research, and offered the opportunity for us to focus on more advanced issues of privacy and collaboration in this thesis. The results from the research paper have been studied to find the areas to focus on this year. By using the **Future work** section from the research report we have found the most important areas that should be taken into consideration in this thesis. In the research paper the following issues were presented as future work:

- Adapting a *privacy policy* standard to suit the needs of UbiCollab and its web-services.
- Research the possibility of applying a trusted 3<sup>rd</sup> party service to the platform that can handle highly sensitive information
- Development of a more detailed privacy design

- Development of a more extensive privacy service

Last year's work included the use of privacy policies as mechanisms for allowing the services to post their data practises to the users. The privacy policies are XML-data sheets in machine readable form that are used to easily state what data the system is going to collect and what it is intending to do with the collected data. The previous report suggested the use of a standard called P3P to deal with this. The issues of policies and the P3P standard are presented in the Related work chapter in section 3.1.

Another idea presented last year was the use of a trusted 3<sup>rd</sup> party service to handle highly sensitive information. A trusted 3<sup>rd</sup> party service is a service that acts outside the platform, but has been trusted by the platform. Accordingly, this service is also trusted by the users and is allowed to perform sensitive operations, like money transaction, on the behalf of the user. Further research on such a service is presented later in the thesis (chapter 5) along with a discussion on the definition of sensitive data (2.4.5).

Two other master's thesis have also been studied during this and the previous project; "**A shared display system for a ubiquitous computing environment**" by Anders Bakkevold [3] and "**UbiClient: a mobile client for a ubiquitous collaborative environment**" by Pedro Gonçalves [11]. These two projects have not been directly applied to the project. Instead they have been used as a tool to show what the platform is capable of, and demonstrating what types of interaction this platform can be used for.

## 1.4 Relation to the thesis by Heitmann and Jensen

This thesis has been developed in parallel with the work done by Heitmann's "**DISCOLab: a toolkit for the development of shared display systems in UbiCollab**" [20] and Jensen's "**Location-aware service for the UbiCollab platform**" [22]. The work done on these two projects have been based on the same architecture as our thesis and are both valuable contributions to the UbiCollab platform. A short description of the two reports is given below, to show the potential of UbiCollab and the different areas the platform is able to support.

### Location-aware service for the UbiCollab platform

Since privacy and location are closely intervened concepts in ubiquitous computing, it has been natural to maintain a collaboration with Jensen throughout the project. Jensen's thesis aims at developing a location aware service on the platform. This contribution consists of two services; **Position service** and **Location service**. The position service provides the location service with raw positioning data. This allows the location service to show the relative location of an entity in the granularity chosen by the application used. A location-aware service is an interesting contribution to the platform in general and also in connection with our work on privacy, since location privacy is one of the most discussed issues of privacy.

Our thesis will not focus on location-privacy, since the location service is a work in progress, but the mechanisms developed by our thesis are applicable to all UbiCollab services. This means that the

privacy mechanisms we develop can be integrated with the location service in future editions of the platform.

### **DISCOLab: a toolkit for the development of shared display systems in UbiCollab**

The DISCOLab project aims at creating a toolkit for the development of shared displays on UbiCollab. This thesis builds on the previous contribution by Bakkevold [3] and improves the collaborative efforts on the platform by introducing ways of developing shared display systems. Shared displays are a valuable mechanism in collaborative environments by allowing real-time sharing of resources to people that are not co-located.

## **1.5 UbiCollab - Conceptualized**

The existing UbiCollab platform has been the subject of several additions and extensions over the last couple of years and has by now turned into a project consisting of numerous individual contributions. This section gives an outline of the basic entities of the platform, and describes the starting point for this thesis. Figure 1.1 shows the different parts of UbiCollab and how the individual contributions have been connected to form a consistent platform. This figure sums up the conceptual model that was used to create the existing UbiCollab design and is incorporated in this report to show the foundation on which we are building our work.

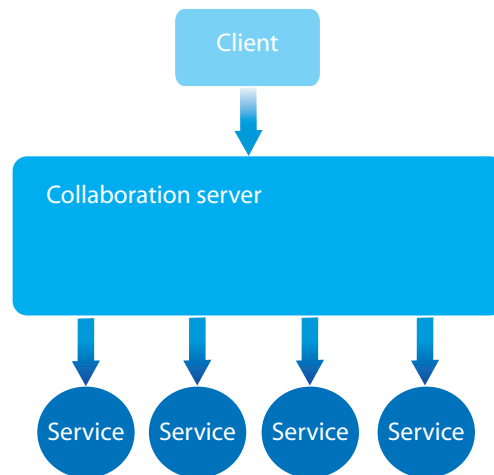


Figure 1.1: UbiCollab

### **1.5.1 Clients**

The existing UbiCollab platform supports both PDA-clients and Web-clients. The current prototype provides separate APIs for the two clients. The UbiClient [11] developed by Gonçalves runs on a PDA and uses web services for accessing UbiCollab. A test-purpose Web-client has also been developed,

accessing UbiCollab through Java-servlets and an API for the Web. At the same level as the clients there has also been developed an application that enables users to share a common display [3].

### 1.5.2 Collaboration server

The Collaboration server defines an API for the UbiCollab services. The server is the core of the UbiCollab platform and manages the services offered by the system for the users. After the changes made last year in our privacy project, the collaboration server handles anonymous users.

### 1.5.3 Services

UbiCollab presently includes a total of six different services, which together make up the basic functionality offered by the platform. The *Presence service*, *Resource collector* and *Directory service* was developed in spring 2004, and is described by Schwarz in his Master's thesis [37].

The *Location service* and *Position service* were developed by Heitmann and Jensen during the research project in fall 2004, "**Location-aware service for the UbiCollab platform**" [21]. The Position service collects raw position data about the resources, and the Location service interprets this information. The location information could be used to e.g. pinpoint resources to a map, calculate the proximity between devices or offer the user information about distance to nearby devices.

The *Privacy service* currently acts as a proxy between the users and the different services. It ensures that the users stay anonymous and that sensitive information about the users is not accessible by other services. This report focuses on privacy in UbiCollab and the main contributions will be made to this service and its interactions with the users and the system in general.

### 1.5.4 Summary

Many of the entities presented here are due for changes during the project. This section is meant to give a quick introduction into the existing architecture and make it easier to understand the choices made and the changes done to the platform during this project.

## 1.6 Research method

The approach chosen for the work with this thesis is based on scenarios. The work is driven by the scenarios and the ideas presented in the different scenarios will be used to create the design, on which we will base the prototype implementation. We have chosen to simplify an existing model development strategy to fit our needs as they appear when working with the UbiCollab platform.

The model is adapted from work done by Pradip Lamsal in connection with modeling trust for ubiquitous systems [24]. The figure below shows the steps in the adapted modeling phase.

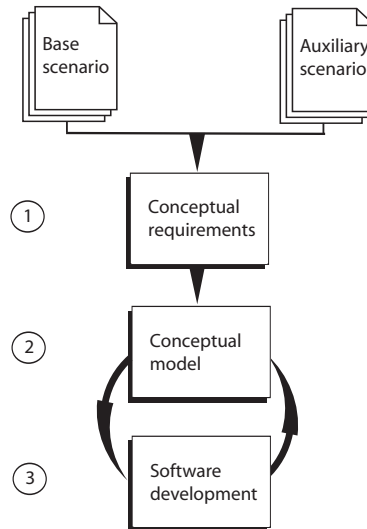


Figure 1.2: Model Development [24]

- 1. The first phase consists of analyzing the scenarios and coming up with the **Conceptual Requirements**. There are two types of scenarios; **Base scenarios** and **Auxiliary scenarios**. The base scenario presents the fundamental ideas behind the system and the basic functionality of the platform. When designing for privacy, a base scenario should include the basic mechanisms for handling the personal information of the users. The Auxiliary scenarios describe the special cases encountered in the system. The Auxiliary scenarios typically include situations that don't occur every time the user enters the system and situations that are more specific for the privacy issues that this thesis will focus on. The analysis of the scenarios will lead to the conceptual requirements, which show what is required from the model on a conceptual level.
- 2. This phase includes the **Development of a conceptual model**. The development is done using the requirements from the previous phase. The model should fulfill all the requirements presented to be able to support the functionality presented in the scenarios. The model might change over time according to new ideas that can be discovered during implementation. The conceptual model is therefore not final, but will reflect the conceptual requirements and the ideas presented in the scenarios.
- 3. Based on the conceptual model it's possible to start the **Software Development**. The conceptual model describes the ideal system and the development should be done according to this model. During software development it's possible to go back and change the model if new ideas are encountered that better suit the system. The process of continuously changing the model according to the progress in development, is an important issue of making the system both conceptually correct, but also ensuring that the ideas presented are technically and logically possible to develop.

The testing of the system is also a part of this phase, which will be performed in parallel to the development. Major logical flaws found while developing can also be reasons to go back and change the conceptual model. This leads to a cyclical process where all the phases contribute to the others. This will decrease the need for testing later on and improve the development by always having the possibility to go back and change what is discovered to be unrealistic expectations or errors and flaws in the architecture.

## 1.7 Challenges and limitations

The main challenge of this thesis will be to integrate the high level ideas of privacy and their consequences into the UbiCollab platform and develop mechanisms for controlling the users' personal information. The existing privacy architecture is based on total anonymity, which has resulted in difficulties when collaborating. When you meet face-to-face, there's no point in having an anonymous identity on the platform, since others know you already.

There should also be a way to contact other people that are logged in, which seems pointless at this time since everybody are anonymous. The transitions from being anonymous to enabling some sort of identity and thereby achieving trust between the users will be one of the most interesting and challenging aspects of this thesis. This also includes being able to list the users that are logged in at any given time and allow them to be represented by the identity they want others to see. The further challenges include managing the privacy policies of the system and achieving a greater amount of control over the personal information for the users.

Considering the technical part of the thesis, the challenges include being able to maintain a generic platform, while at the same time improving the functionality and increasing the ubiquity of it. It seems that the most challenging part will be developing a new layer on top of the existing Collaboration Server, which will handle user requests and provide the new API. The new ideas presented in this thesis will be implemented in the existing prototype and involves some risks since the consistency between the conceptual ideas and the prototype must be preserved.

Some work will have to be put into fixing bugs in the existing prototype and reengineering the platform to suit the new mechanisms that will be introduced during this thesis. This work is not directly connected to privacy, but needs to be done in order to have an extendable working platform prototype. Consistency between the different work done last semester is also a concern, since some of the interactions between the previously developed components have not yet been made consistent in the platform prototype.

## 1.8 Structure

This section presents an outline of the report and provides information about each of the next chapters:



- In chapter 2, **Problem Elaboration**, the main ideas behind the thesis are presented. This chapter includes the scenarios used for developing the system and a high level discussion concerning the different aspects of privacy encountered in connection with the UbiCollab platform. At the end of the chapter the conceptual requirements are presented along with the research questions for the study of related work.
- Chapter 3 presents the **Related Work** studied during the work with the thesis. This chapter covers similar work done in the same area, both on a technical and a theoretical level. Along with the presentation of the relevant articles a discussion concerning the value for UbiCollab is presented.
- Chapter 4 bridges the research part with the design and implementation phase. This chapter includes a discussion on the choices of technology to use and provides an **Analysis** of the conceptual requirements that were presented in the problem elaboration chapter.
- The **Design** chapter (5) presents the architecture for the previous UbiCollab platform to give the reader an overview of the issues that are due to change in this thesis. Further, the developed conceptual model is presented along with the design based on the new conceptual ideas. At the end of the chapter we describe in more detail the entities that are most important for the privacy mechanisms of UbiCollab and how these need to be modified to suit the model.
- Chapter 6, **Prototype**, includes the details of the prototype platform that has been developed as a proof of concept of the architecture.
- The **Demonstration** chapter (7) shows the main functionality of the implemented prototype. The demonstrator presents the most important parts of the work done with privacy in this thesis.
- Chapter 8 is the **Conclusion** to the thesis. The sections of this chapter are; *Contributions* made by this thesis, *Evaluation* of the work done and a discussion on what has been left for *Future work*.

## Chapter 2

# Problem elaboration

This chapter will present the motivation behind the work done in this thesis and give detailed information about the process of the research and how we will deal with the problems encountered. This chapter will explain which areas the research will need to focus on to be able to enhance the privacy mechanisms of the platform. The chapter starts with the scenarios used for developing the design along with an analysis and discussion. By developing the auxiliary scenarios we introduce the ideas we have found to be of value to the platform while aiming at making the basis for the design phase through the conceptual requirements.

At the end of the chapter, the high level requirements are presented together with the project's problem definition and the aim of the project. These issues will be the foundation for deciding on what related work that will have to be studied and for designing a suitable architecture for privacy support.

### 2.1 Scenarios and privacy in UbiCollab

The work method used during the project is based on scenarios, where the progress has been driven by the ideas presented in the different scenarios. Two types of scenarios have been used during this project; The **base scenario** and the **auxiliary scenarios**. The base scenario describes the core functionality of the platform and how the user can access the services and accordingly how the privacy mechanisms handles the interaction between the users and the system. There is only one base scenario, which handles all the basic mechanisms and the most common interactions between user and system. The base scenario has been preserved from our research project, since it demonstrates the basic ideas behind UbiCollab and shows the functionality that has been the goal from the start of the project. This is an adaptation of the scenario that was first presented by Schwarz et al. in Spring 2004 [37].

The auxiliary scenarios will describe the special cases, which can help support the conceptual model by highlighting the more specific parts of the platform and user interactions, especially in connection

with privacy. Several auxiliary scenarios have been developed to ensure that as many situations as possible on the prototype platform are covered. The base scenario and a discussion on its content is presented in the following section, while the auxiliary scenarios are presented along with the specific issues that they cover in section 2.4. Some issues are introduced without a scenario to support it. These areas of research are discussed on a general basis and introduce ideas that will be handled in the conceptual requirements.

The main contribution from the previous work with UbiCollab is a platform that supports collaboration through the use of abstract objects called **Collaboration Instances**. These objects are connected to the user identities of the people involved in a collaboration and the devices and resources used in the collaboration. Figure 2.1 shows how different people and different entities are connected to a collaboration instance.

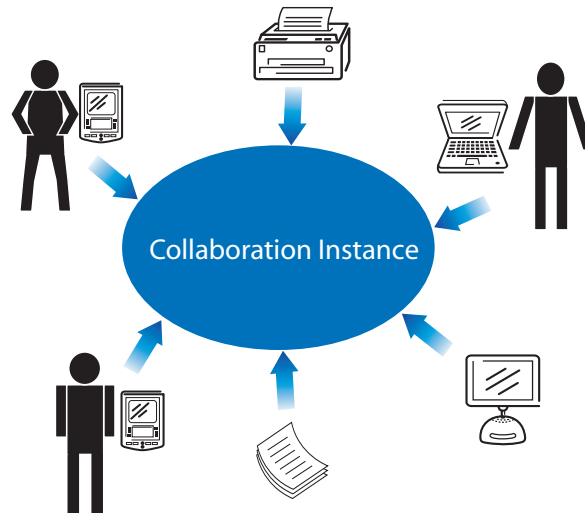


Figure 2.1: Collaboration instance with connected entities

As a result of the importance of collaboration on the platform, the main focus will be on preserving the privacy in connection with the collaborative services offered by the platform and investigate how users can control their appearance when collaborating.

## 2.2 Privacy management

Before presenting the scenarios we take a look at the general management of privacy in a platform like UbiCollab and what issues that should be taken into account when working on this thesis. **Privacy management** involves a lot of loosely intervened concepts that together make up the privacy concerns on the UbiCollab platform. Some of the issues that will be discussed in the scenarios are not necessarily functionality that need to be covered by the platform, but rather

more high level concepts that need to be supported to be able to develop a dynamic and flexible platform that handles privacy concerns of users and services in a satisfying way. The design of the platform privacy management should be able to take future situations into account in such a way that the amount of reengineering is decreased to a minimum in future contributions. Accordingly, it's interesting to look into the concept of privacy management and the implications resulting from implementing a privacy management system.

*"Privacy management is not about setting rules and enforcing them; rather it's the continual management of boundaries between different spheres of action and degrees of disclosure within those spheres"* [32]. The quote is from Palen and Dourish's article **"Unpacking "privacy" for a networked world"** and gives an idea of the vagueness of privacy and the difficulties that arises when trying to transfer the idea of privacy into a digital world. The authors claim that the boundaries between the spheres and the level of disclosure move dynamically as the context changes. In information technology the regulation of these boundaries are disrupted and destabilized frequently. To be able to manage privacy in computer systems, an understanding of how to deal with these boundaries is needed. The article lists three different types of boundaries that are of interest to privacy management:

- The disclosure boundary
- The identity boundary
- Temporal boundaries

### **The disclosure boundary**

The management of privacy is not just a matter of avoiding information to get out or just avoiding disclosure of information; It's a matter of keeping a dynamic boundary between these and allowing the user to be able to maintain what he feels is a suitable solution for him. A good management system of privacy allows for selective information disclosure. *"We seek to maintain not just a personal life, but also a public face"*. This calls for mechanisms to hide information, but also mechanisms for distributing information about yourself and the characteristics you would like others to see.

To handle the concept of allowing a dynamic boundary between what is public and personal we need to develop a design that allows the system to support mechanisms for controlling the identities. These mechanisms can be building of a reputation, having multiple identities, staying anonymous etc. At the same time it's important to allow the user to decide which of the concepts he wants to make use of. It's important to allow the user to flexibly control the tension between publicity and privacy to support the user's desires in connection with privacy.

Chapter 4 and 5 will describe a design that allows for this flexibility through the use of a privacy management system that is possible to implement in UbiCollab. In UbiCollab this issue calls for a mechanism that allows the services to state what information they will be needing from the users, and the possibility for users to deny this gathering if they want to.

### **The identity boundary**

The identity boundary is described as another important issue in privacy management. The boundary of identity concerns how your identity is perceived by others and how this can be controlled by the user. The problem with information technology in contrast to real world situations is that there exists a layer between yourself and others. This layer can consist of a great variety of technology, which may distort the appearance you like to present to others, without available means to control the appearance. This issue of mediation is a problem that needs to be taken seriously when designing a privacy management system, to allow the users the desired control over their public appearance when using the system.

### **Temporal boundaries**

The temporal boundaries concern the retention of information. Information that is meant to be ephemeral can in a digital world easily be stored and used against you at later occasions. The article mentions that privacy management should not be a static ruleset, but be able to change according to the present situation. The temporal issue is not the focus of our work with privacy, but when developing privacy management mechanisms it's important to take this into account as well.

The article states that management of privacy is a *"dynamic response to circumstances rather than a static enforcement of rules"*. It also states that the three issues mentioned are not to be resolved independently, since they are all connected and intervened concepts that need to be preserved to develop a privacy management system that both users and services are satisfied with. It is also important to mention that these concepts are meant to be mechanisms that support the development of applications on top of UbiCollab in future contributions. The aim for us is to make it possible to integrate these future applications with the privacy concerns presented here. The user is not the main focus of this thesis, but we need to support the basic mechanisms that makes it possible to focus on flexible user privacy mechanisms in the applications.

The ideas presented here will be introduced to UbiCollab through the scenarios and later included in the design of the new UbiCollab architecture. The auxiliary scenarios describe more special cases and the adaptations that need to be made to make the requirements suit the UbiCollab platform.

## **2.3 Base scenario**

The base scenario is included in appendix A. The following section presents the parts that are most relevant for this thesis and a discussion on the issues we want to pursue in this report. The scenario describes situations that will be possible after the thesis, but also situations we feel need to be supported in future work, even if we are not going to focus on designing for them at this time.

The sections are sorted in order of appearance.

*Brian suggests that they check with John, who is an expert on that technology, and Sylvia agrees.*

Collaborations in UbiCollab may include people that are known primarily by previous actions, which over time builds a reputation that represents a user's competence, social skills or area of interest. If some users wanted to get in touch with an expert in a branch office, the actual identity

of that person would be less important than the skills or position he is holding. To support this in UbiCollab would mean that persons should be able to build a reputation for their identities. This is not a main concern in our work with UbiCollab, but it's an area of interest regarding trust and identities in a collaborative settings. Looking at the scenario, the coworkers can know John from earlier cooperations or find him with a tool developed to search for knowledge in the different fields of work that the company deals with.

*[Sylvia] can see on her UbiClient that John is unavailable (picking up children in kindergarten).*

John has set his profile to display his availability for his coworkers during work hours. Other people is not getting any information about him, since he feels that his time at work is reserved only for the people he works with. Similarly, he has decided that the preferences change when he leaves work so that his boss cannot gather information about his leisure activities, while his personal friends can get the information they want. Concerning work hours the setting could vary depending on the situation. Since Sylvia isn't in the same collaboration instance or has a relation to John, she doesn't get any further information. Other people that work with John, or perhaps his boss, would get more information and possibly be able to get in contact with him. John has also set up his profile so that he is not anonymous to other users who would like to contact him.

*They still need some help and decide to try contacting Steve the project manager for the project.*

Steve is already in the same collaboration instance as the others, and they can therefore see if he's logged on or not. When they check the other people in the collaboration instance, they can see the users' UbiCollab identities if they are logged in.

*Sylvia can now see that Steve has joined thanks to a new presence widget which just showed up. Checking his available devices, she can see that he has a display application similar to the one in the projector.*

The system allows a user to share information on his available resources with a user connected to the same collaboration instance. (If specified in the settings of the given user). The identity of Steve is visible to the participants of the same collaboration instance and to other people he has added manually to his list of people that he wants to see him.

*At the same time Alice, which is working on a different project with Sylvia, is trying to find Sylvia. She opens her UbiClient and see that Sylvia is logged in but busy in a meeting in room S. Alice is able to get this information because Sylvia has configured her profile for Alice to be able to see her location.*

Sylvia is available to other co-workers even when she is active in another collaboration. The users of UbiCollab should be able to configure their personal setting in such a way that their positions are available to some trusted user, but not to everybody. Sylvia and Alice are not members of the same collaboration instance, but the users also have the possibility to specify what users they want appear as more than anonymous entities.

*Just when Alice is finished typing out the message to Sylvia she gets a message on her UbiClient telling her that George, one of her co-workers on the project, is passing by in the hallway. She opens her office door and says hi.*

This issue concerns the preservation of location privacy. George has configured his device to enable tracking while he's at work, so that other co-workers may get a notification when he's in the proximity of them. George has given his consent to allowing the system to gather and broadcast his location to the participants of the given collaboration instance. The collected location data is also accessible to other users, but they will just be able to see him as an anonymous user. This issue is of great concern when working with privacy, since it concerns the issue of location information, which is potentially the most harmful information to allow gathered. The mechanism for managing these situations should give the services a way of posting what actions or data collection they will perform and also allow the user to deny the collection of personal information if he doesn't agree to it.

### **Base scenario discussion**

The base scenario shows the interactions between the privacy architecture and the other parts of the platform. The different situations give a certain insight into the areas where a more refined privacy design is necessary in order to achieve the functionality described. To achieve this we need to extend the functionality of the platform. It's also necessary to pinpoint what the platform will need to support in order to cover the demands that future applications might state.

The base scenario gives a certain overview of the privacy architecture, but more detailed situations will have to be generated to fully cover the possible interactions that might happen on the platform.

## **2.4 Issues of privacy in UbiCollab**

The research project from fall 2004 identifies some issues of high concern regarding user privacy in the UbiCollab platform. Some of these issues were addressed in the development and implementation of the previous platform prototype. This section presents the actions that have been taken and what will have to be done to further improve the privacy on the platform and enhance the user-friendliness concerning privacy in the system. The following sections describe different privacy issues and ideas on how to deal with them. Some of the issues are presented along with a scenario to help describe the importance and to give a better impression of the privacy issues we will be dealing with later in the thesis.

### **2.4.1 Anonymity**

The most important contribution to privacy on the previous platform design is the possibility of anonymous access to the platform and its services. This idea will be preserved in the new design, since anonymity is an effective mechanism for supporting privacy. The following scenario demonstrates the necessity of handling anonymity in UbiCollab:

#### **Auxiliary scenario No. 1**

*Company X is using UbiCollab to allow the employees to set up meetings and manage the shared devices that are located around their offices. Users can access printers, projectors and shared displays from their computers and mobile devices, and exchange documents with other co-workers. Sylvia is using a PDA with a wireless network connection to keep in touch with her workgroup and check for personal messages. The positioning-functionality of the system is handy for accessing nearby printers, but she wouldn't want her co-workers or superiors to follow her every movement around the building. She has set up her privacy preferences to allow her co-workers to check if she's in the proximity, but has not allowed the system to give away her exact position. People she has no knowledge of are not able to find her position at all, not even to check if she's nearby.*

This example shows how the introduction of location aware technology can increase the usability and allow advanced functionality. But it can also pose a threat to user privacy. An effective way to solve this problem is to conceal the users' identity in the system. Anonymity or pseudonymity is recognized as one of the most important principles of privacy, as discussed in section 1.1.1. In short, data that is gathered about a user is rendered harmless when separated from the users identity. This idea was explored in the research project, which resulted in a design that supports anonymous user representation at platform level. This has made it possible to implement safe-to-use advanced location-aware features in UbiCollab applications.

## 2.4.2 Identities and collaboration

Anonymity is certainly a part of the solution to privacy in UbiCollab, but as this scenario shows, total anonymity is not always desirable or possible to achieve:

### Auxiliary scenario No. 2

*Sylvia's boss has asked her to publish some information about her department on the company Web-page. She decides to get in contact with the webmaster for technical guidance. She uses the UbiClient to find out when he is available, and invites him to a meeting where they can discuss the details.*

People who work together are rarely, if ever, anonymous. What information about us is known to others depends very much on the collaborative setting. People in a workplace environment often know each other by name and visual appearance, and may share contact information like e-mail address, phone number or even home address. Just by being close to other people we give away bits of information that are tied to our identity. People may know you as the person who rides a bicycle to work, wears fancy shirts, or always sits alone in the cafeteria. People may recognize you even if they don't "know" you, so you are never totally anonymous [29].

When collaborating with others, we often share more about ourselves than our general characteristics and personal information. Contextual information like where we are, whether or not we are available and when, etc. could be valuable to the people we work with in order to have a smooth cooperation. How much we are willing to reveal and trust other people, depends to a certain extent on how well we know them. In some situations we might even hide the truth by giving away false information about ourselves. At most times a general public appearance might be most appealing, where we only distribute enough information to be categorized as a user of the system and nothing more.



Giving the users the ability to display and control the use of personal information in UbiCollab is certainly desirable. During the Autumn project, this functionality was severely limited, in favor of allowing anonymous use of services. Mechanisms for sharing personal information will be further explored in this thesis. Further on, the notion of reputation will be explored. Building reputations over time in connection with your identity in UbiCollab, will make it possible to share information with persons you don't necessarily know in person, but only by having checked what the person is known to be interested in. The notion of reputation will therefore improve the collaboration in the platform, while at the same time being an important mechanism for enhancing the privacy by building trusted relationships between the users.

### 2.4.3 Managing multiple user representations in UbiCollab

People need information about the persons they collaborate with. But how we portray ourselves depends to a large extent on the people we interact with. It should be possible for the users to switch between their different identities according to which people they are dealing with. This scenario shows how a person is perceived differently depending on the context of the collaboration.

#### Auxiliary scenario No. 3

*John works in the IT department, and is the company's webmaster. He is also on the company's football team, who get together in the weekends for league matches. John is in a meeting with Sylvia when Terry from accounting, who is the team's coach, calls him up to have a chat and check if he is available for the upcoming match. John briefly replies that he is on for the match, and that they can discuss the tactics over a beer after work hours. He then gets back to the meeting with Sylvia.*

This shows how John is switching between different situations and thereby changing how he is perceived by others. Sylvia knows him as a tech-person at work, while Terry knows him as a key player on the team. Sylvia has no interest in knowing how many goals he scored last season, but he is happy to share this information with his fellow teammates in the company.

Individuals behave quite differently in different social situations and might well have quite different ways of behaving with different people. Your friends, family and co-workers all have different perceptions of your identity, and communication technology makes it possible to switch seamlessly between these facets of identity. Mobile phones and the Internet ensures that we are connected at all time, and hence being able to take on even more identities than before.

Identity-switching in UbiCollab is an exciting idea, and will be further explored in this thesis. Implementation of this idea would require mechanisms for setting up and maintaining different identities or profiles. It also requires support for a seamless transition between such identities. Some of this involve user-interface design at application level, which is out of the scope of this project, but the platform must be redesigned to allow multiple representations of the users.

#### Identity Management

When introducing multiple identities in a collaborative environment it's important that it's possible to identify the users and also allow the system and other users to find the characteristics of their

co-workers. This is necessary in order to allow a flexible cooperation where persons can act as resources based on their fields of expertise. It's also of importance that the users have mechanisms for managing the transitions between different identities. To do this we need to have a clear idea of how to represent the users and how the transitions between the identities will be experienced by other users. To better understand these issues we have looked into different ways of representing users and ways of switching between identities.

## Identities and personae

Users have traditionally been represented in a very static way in computer systems. The user most often gets one identity to use when logged in to a system, which represents him and all his actions when present in this environment. When looking at the World Wide Web, it's clear that most people would like to appear differently depending on the actions they perform. Not all statements you make in a chat room is something you want connected to your true identity. At the same time there are a lot of things you do that you would like people to know about you. For instance when searching for your achievements, CV or field of expertise. This tells us that a ubiquitous computer system should have support for letting the users change appearance according to their own desires and the context of the collaborative situation.

User representation in computer systems is often supported by the use of an identity that contains the personal information about the user which others can access if wanted. When dealing with multiple identities the user is not interested in displaying the same information in connection with each of the identities. Since identity is a concept that is strongly connected to yourself, the use of the term persona has become popular to describe the representation of the users. The term persona is defined as *"The role that one assumes or displays in public or society; one's public image or personality, as distinguished from the inner self"* [12]. The persona term is considered less static than identities and will allow more flexibility by letting the representation evolve over time. At the same time, the persona concept is very much connected to design issues, where the term represents imaginary archetypical users that help developing a system to suit the needs of different types of users [23]. This thesis will therefore use the term identities to describe the user representation in UbiCollab.

The aim of this thesis is to support an easy user representation, without making it difficult to expand the design to include a more refined identity management in future extensions of the platform. The representation of the users should allow the users to decide for themselves whether or not they would like to have different identities in the system, since some feel that it's more than enough to handle one representation of themselves in the platform. It is an important challenge for the design to satisfy the different needs of the different users. Figure 2.2 shows how different users have the need for a great variety in the number of different identities or personae when using computer systems [29].

The figure shows a typical variety between very active users of the web and users who mainly perform the bare necessity of actions online. The circles represent the need for different representations in the variety of actions performed by the user. People who use computer systems only to read work mail and pay bills would probably need just one account on the system, while users that use chat rooms, pay bills, keep blogs and host web conferences online may want to use multiple identities. Another aspect of this is the need for different identities connected to different actions. One person might want to perform various actions with separate identities, while others feel comfortable with using

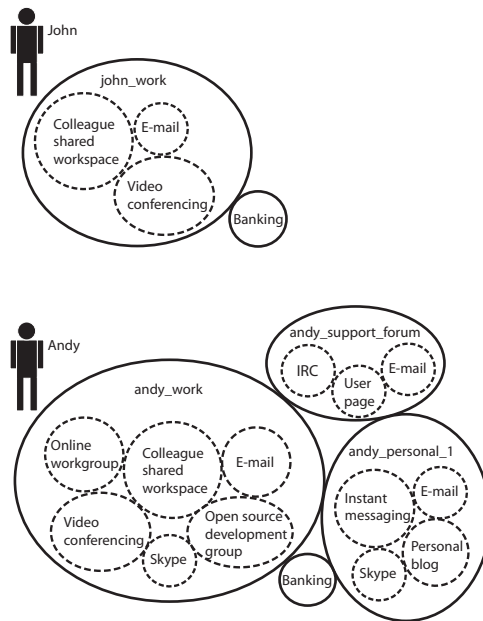


Figure 2.2: Different needs for multiple identities [29]

their real name or just one identity to perform all actions. The figure shows that John has a general work-identity that he applies to all his online activity. Andy, on the other hand, is using multiple identities to separate the activities from each other. This allows him to change his appearance accordingly, and also to use the same application with different identities.

Ubiquitous computing is offering increased availability and user flexibility, and a ubiquitous environment should try to empower the users and give them what they desire in form of flexibility of how they want to portray themselves. When introducing multiple user representation in a computer system, the system will require a flexible and user-controlled way to manage these identities.

### ”Switching and Stitching” identities

To be able to improve the collaborative environment that exists in UbiCollab the users will need mechanisms for changing their appearance towards others on their own request. This means that the users will have to have a way to ‘appear’ differently towards different people. This is an issue that calls for having multiple identities to choose from, according to whom you are working with. This is a common situation in ubiquitous systems, and accordingly the switching between the different identities has been the focus for research in the field of ubiquitous computing.

The article ”**Switching and stitching in the digital age**” by Smyth, Raijmakers and Munro [38] discusses the importance of allowing users to be able to maintain a connection between their real identities and the ones in the digital world. The article states that ubiquitous computing has lead to a situation where we have the opportunity to undertake a large amount and a great diversity of roles. This diversity and the increasing number of roles has lead to a situation where the transitions between the roles has become difficult to perform. We have a situation where you might be available

to your employer all the time, since you are digitally online 24 hours a day. This is of course a great threat to privacy. Along with the issue of switching between the different identities the problem of stitching the identities into place is raised. An identity is basically used to represent yourself in the way you find most convenient in the given situation. It's therefore important that the identities can be stitched into place in the context you use the identity. The article discuss this issue and how it is possible to solve these problems with technology.

These issues are not the main focus for the further research in this thesis, but it's important to develop an architecture that is capable of dealing with future extensions of the platform. Stitching is also important to be able to build permanent representations over time, which will allow the users to build a reputation in connecting with their identity. This issue will be discussed in the following section.

#### 2.4.4 Reputation

The idea of stitching an identity into place makes it possible to build permanent connections between the users and their representations in a computer system. This will allow the user to be able to create a "history" in connection with his user-representation and the actions he has performed in the computer system. A history log of a user's previous actions, along with feedback from other users, are often referred to as reputation.

Reputation is an interesting notion of preserving trust, or the opposite, between people in computer systems as well as the real world. Reputation has been defined as *"An entity's reputation is some notion or report of its propensity to fulfill the trust placed in it (during a particular situation); its reputation is created through feedback from individuals who have previously interacted with the entity"* [18]. To be able to use reputations in a sensible way there exists a need for reputation management. Reputation management is a mechanism through which a person's actions and the opinion of others about those actions can be recorded and later published, in order to allow others to make informed decisions about whether to trust the person or not [41]. The idea about reputation management comes from real world communities, where the interactions and the trust between people are based on previous experiences or the reputation you have in the community. This has traditionally been ensured by coincidental parts of society in small communities. It's very common that people base their relationship with others, at least in the beginning, on what friends or people you trust have told about them. This idea is also intriguing in connection with ubiquitous collaborative environments, where mechanism for providing information about the other users are essential for finding suitable sources of information and get the support you need when encountering a problem.

Reputation has been a mechanism for providing trust in E-trade for a long while. eBay [14], which is the leading site world-wide on electronic trade, is highly dependant on a reputation system to provide the users the trust needed for purchasing items through the system. eBay provides a mechanism for selling and buying merchandise through the help of the Internet. The idea is based on a traditional auction, where the item is sold to the buyer that offers most money. Even though eBay acts as an auctioneer, they do not take on any responsibility for the money transactions that are being made after a trade has been agreed on. This clearly opens for money frauds, since there is no authority that controls that the product exists or that the seller is willing to send it after receiving the money. Surprisingly, there are very few such cases in comparison to the amounts of trades being made. eBay

reported in 1997 that only 27 out of 2 million auctions over a 2 months period involved criminal frauds [36]. One reason for this is the system of reputation. eBay allows the users to post their opinion of the sellers and buyers after a successful trade. The information is stored in connection with the user and future trades are also added to this growing resumé. Accordingly it's possible for buyers to check the sellers profiles before committing to a trade. eBay calls this service the Feedback Forum, which in their own words makes it possible to "... view their reputations, and express your opinions by leaving feedback on your transactions. Such member-to-member comments help the millions of buyers and sellers in the community build trust and share their trading experiences with others." As a way of demonstrating how useful the users find the feedback systems, it's worth mentioning that eBay has now passed 3 billion feedbacks over a period of 9 years.

Howard Rheingold presents the eBay community and the reputation system in his book **Smart Mobs** [36]. He states that one of the reasons for why the reputation system works is that a solid reputation is extremely valuable for eager users of the system. The temptation of performing frauds is overshadowed by the value of a clean reputation, which helps the system avoiding cheating of buyers. Rheingold also presents three important properties that a reputation system requires in order to function properly;

1. The identities must be long-lived in order to create an expectation of future interactions.
2. Feedback about interactions must be available for future inspection by others.
3. The users must pay enough attention to the reputation to allow themselves to base their interactions on it.

Rheingold is convinced that the reputation systems are going to be valuable knowledge in the future and that the importance of having a good and solid reputation online will be of great importance. In a ubiquitous collaborative environment a similar mechanism can be used to build relationships based on trust and information about each others knowledge and fields of interest. The following scenario describes a situation where a system of reputation and information about resources will be of interest.

#### **Auxiliary scenario No. 4**

*Dave is collaborating with John and Leland on a system development project. The project is in the field of quality assurance and estimation of the workflow and its effectiveness in the corporation. After their initial meetings they discover that their knowledge in this field are somewhat insufficient. Dave, who was appointed project leader, has decided that they need external input to make the project worth the participants effort. He uses the UbiCollab PDA client to search for registered users that has expertise in this field of work. UbiCollab comes up with the people that has the best references on quality assurance and provides John with a way to get in touch with them. John finds that the resources offered by the system is somewhat sparse concerning this field, and decides to perform a more extensive search using a WWW search engine that has been integrated in UbiCollab.*

By integrating a system for building reputations in connection with your identities when using UbiCollab, the functionality of the platform will be further enhanced. A system of building reputations also improves privacy, by providing the users with a mechanism for building relationships based on trust and previous cooperative experiences. Most importantly, a system like this improves the

collaborative environment by making the users able to control what information about them will be stored.

Implementation of a reputation system in UbiCollab is an interesting issue of improving the existing functionality. At the same time, the functionality of a system like this is not incorporated in the basic services provided by UbiCollab. To make this idea work, an extended client that can present the information in a proper way will have to be developed. Application level implementation is out of the scope for this thesis, but in chapter 5 (design), the necessary means for the platform level design for a system like this is presented. The scenario also opens for a more global idea of posting reputations in the different fields of expertise, so that users from all over the world can collaborate on issues that they are interested in.

## 2.4.5 Management of sensitive information

### Auxiliary scenario No. 5

*Susan wants to take advantage of one of the newest UbiCollab services; The money free kiosk. The kiosk provides different merchandise for the users of UbiCollab by charging their credit cards. The kiosk is operated with a PDA when being in the immediate proximity if the user has decided to accept such services. This service requires that the users have registered their credit card with a trusted 3<sup>rd</sup> party service connected to UbiCollab. When Susan enters the area her privacy preferences are checked and compared to the policy of the system. When the system discovers that she allows such services, the assortment of merchandise is offered to her on the PDA. She decides to purchase newspapers and coffee for her lunch break. The privacy service checks her profile and UbiCollab communicates with the external 3<sup>rd</sup> party service to charge her credit card with the purchase. The credit card is charged and Susan gets her coffee and newspaper from the kiosk.*

The most obvious demand raised by this scenario is the need for a way to handle location information in connection with being able to recognize who it is that enters the area with the money free kiosk. The system needs a way to recognize the user, even if the user is generally anonymous in UbiCollab. This suggests a proxy service that keeps track of the users real identity to be able to provide the services the users are requesting. In addition, this scenario shows the need for a trusted 3<sup>rd</sup> party service outside of UbiCollab that can handle the interactions between the users and the system in connection with e.g. money transactions. These are both interesting concepts, but mainly the mechanisms for supporting the protection of sensitive information will be researched in this thesis, since no purchasing services exist in UbiCollab today.

When defining mechanisms that provide security for personal information and other sensitive data, it's important to have an idea of how to classify data into different categories and to be less restrictive on information that should be viewed as public. The next section presents a discussion on the difficulties of deciding whether or not information should be regarded as sensitive.

## Sensitive information

Sensitive information is a central issue when working with privacy. Sensitive information can be defined as "*Sensitive information is knowledge that might give someone an advantage if revealed to persons not entitled to know it*" [42]. According to this there will be a great difference between what the different users might consider sensitive or not. One user might find his e-mail address to be too personal to allow everybody to see, while others don't care whether his birth number is freely distributed. Differences in what information is sensitive might also be determined by culture and legislation. In Norway you might get a lot of information from knowing a person's landline telephone number, since the number is stored along with other personal information. In other countries the regulations are stricter on issues like this, and the amount of information gotten by knowing a telephone number is sparse. Along with the differences in opinion of the users make it hard to classify the different personal information into categories and deciding what information is regarded too sensitive to allow others to see or not. There is also the issue of *highly* sensitive information. This information is typically data that most users will agree is too sensitive to allow public viewing. Still it's difficult to tell what data to classify as highly sensitive data, since this may vary from whether the users are friends, family, enemies or unknown to each other. As a result of this confusion regarding the different ways of perceiving information, this thesis will focus on building a system that considers all data to be sensitive unless the users says differently by changing the preferences of his profile. Regarding highly sensitive data, an external service will be included in the design, which can be used for money transactions etc.

## 2.5 Conceptual requirements

This thesis aims at developing more refined privacy mechanisms than the ones existing in UbiCollab today. At present, privacy is preserved mainly through anonymization of users, which denies the users and the system the possibility to find the users real identities and to be able to monitor their actions. The aim is to further examine the issues of privacy and to give the users more control of their personal information and the gathering of such information. Concerning the collaborative environment that UbiCollab should support, we have found that anonymity is not nearly enough, and in some cases not a good solution, for protecting the privacy of the users. Further research into identities, pseudonymity and anonymity will have to be carried out to be able to produce a solution that can handle the privacy concerns of the users on a platform like UbiCollab.

The main topics to be explored in this report are listed below. These ideas represents the best foundation for future contributions which was presented in section 1.3.1. These requirements are subject to discussion in the following chapters and will form the basis for the design and further development on the platform in this thesis. The requirements are not final, since ideas might appear during development that have not been presented here, but they should contain the most important aspects for developing a privacy management system on the UbiCollab platform.

1. Equip the users with the ability to have multiple identities to enhance user privacy and ubiquity to support the demands presented in the auxiliary scenarios 2-4 (presented in section 2.4). The users will also need mechanisms for making transitions between user identities depending on the

collaborative setting. This point also includes being able to choose between being anonymous, reveal partial information and to give away all information. Also, explore how identities can be used in building a reputation for the user over time.

2. In order to fulfill the previous point, the system must implement mechanisms that allow the users to express their consent to the gathering and use of personal information. This point supports the demand of choice and consent and satisfies the demands stated in section 2.2 on privacy management.
3. The users will need a mechanism that preserves their privacy when communicating with the platform. The mechanism should protect the user's interests while at the same time be able to control the actions performed by the services of the platform. This issue comes as a result of the discussion in section 2.2 where it's stated that the system will need a flexible way to allow users to decide what information to allow the services to gather. This problem is also presented in the last part of the base scenario.
4. To be able to develop a solution based on the auxiliary scenario number 5 the system needs to ensure that highly sensitive information is not accessed by unwanted parties. The highly sensitive information should be handled in another way than basic personal information to protect the privacy of the users. This point presupposes a discussion of what information is considered highly sensitive within ubiquitous collaboration, which was presented in section 2.4.5.
5. The platform needs to preserve the general privacy of the users and be able to empower the users through mechanisms that let the user be in charge of personal information. This is connected to how information is gathered, who has access to the information, how the information is stored etc. This issue is connected to the base scenario and the basic information handling on the platform.
6. The functionality and the API of the existing platform needs to be preserved. This is an important requirement since a lot of the issues of the base scenario are concepts that have already been integrated into the existing solution.

## 2.6 Conclusion

After looking into the problem area and the different possibilities for research we have decided on a set of problems which are most interesting for this thesis' further work in the area of privacy on the UbiCollab platform. The conceptual requirements need to be fulfilled in the design to be able to develop a system that preserves privacy in accordance with the scenarios and the general principles of privacy that exist in the area of ubiquitous computing. The requirements also specifies the area of research that will have to be studied in order to create a design that suits the demands stated in the scenarios.



## Chapter 3

# Related work

This chapter presents a selection of relevant projects and research work done in the area of collaboration and privacy in ubiquitous computing. The focus of this chapter is on the technical aspects of the thesis that will support the privacy issues discussed in previous chapters.

### 3.1 Platform for Privacy Preferences (P3P) Project

The World Wide Web Consortium (W3C) is behind the development of a standard for online privacy, the *Platform for Privacy Preferences (P3P) Project*. The background for this initiative is the increasing public concern on how personal data is used on the Internet. This concern is recognized as a major obstacle for further growth of Web-based commerce.

The idea behind P3P is to empower the users with more control over their online privacy in a simple and automated manner. It offers a way for organizations to express how they handle the various aspects of privacy. "*It does not attempt to ensure privacy by technology - for example, by cryptographic or anonymization techniques. Instead it relies on social and legal pressure to compel organizations to comply with their stated policies.*" [30].

When accessing a Web-site, the user is supplied with a P3P policy that contains information on what personal data is collected and how it is handled by the site. This policy is made available in a standard, machine-readable format. The transactions involved in a request is depicted in figure 3.1. A P3P-enabled browser will try to get the policy from the site (1) before requesting the Web-page (3), and display the page and policy to the user (5).

This example shows the mere basics of P3P, where the browser downloads and displays a human readable version of the policy for the user to inspect. The following section will explain how *user agents* and *user privacy preferences* can be implemented to offer advanced functionality in online privacy.

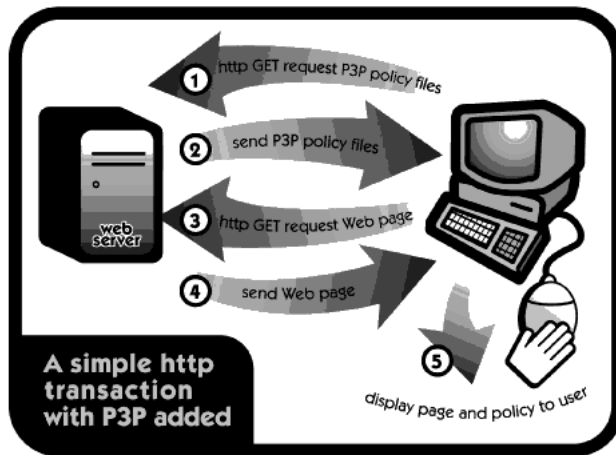


Figure 3.1: P3P Transactions [16]

### 3.1.1 P3P user agents

The user may define his own *privacy preferences* as a counterpart to the site's P3P policies. Here the user can specify what sort of privacy practices he finds to be acceptable, and let his *user agent* evaluate P3P policies against this. A user agent, like a P3P enabled browser, will retrieve and parse a site's policy before comparing it to the user's preferences. The user agent can then display symbols, play sounds or generate user prompts to alert the user of the site's privacy practices. The user agent may also act as a "gate keeper" that can store and authorize the release of data on behalf of the user. In case of inconsistencies, the user may be informed and given the opportunity to authorize release of data himself.

The creation of such policies and preference sets is comparable to filling out a standardized set of multiple-choice questions that covers the major aspects of online privacy. For example, a user may decide to give out personal information only if the site has a policy to never release data to a third party, or store data beyond a given period of time.

The P3P specification places few requirements on user agents, and different implementations have varying functionality. User agents can be built into Web browsers, browser plug-ins, or proxy servers. They can also be implemented as Java applets or JavaScript, and built into electronic wallets, automatic form fillers or other user data management tools. Microsoft Internet Explorer 6.0 is a well known Web browser that supports P3P. The user can apply his own privacy settings by moving a slider in the preferences setting. A higher privacy level means that more cookies will be blocked, and less information about the user is revealed to a site. Other browsers, like Mozilla and Netscape, also use P3P to filter and manage cookies.

### 3.1.2 P3P specification

The P3P 1.0 Specification [10] defines the syntax and semantics for expressing privacy policies. It also provides mechanisms for associating policies with Web resources. A policy is a collection of statements that expresses a site's privacy practices. The statements are made using the XML-encoded P3P vocabulary. The vocabulary covers various aspects of online privacy, including a detailed description of a site's tracking of data [16]:

- Who is collecting the data?
- Exactly what information is being collected?
- For what purposes?
- Which information is being shared with others?
- Who are these data recipients?

The following topics describe the internal privacy practices:

- Can users make changes in how their data is used?
- How are disputes resolved?
- What is the policy for retaining data?
- Where can the detailed policies be found in "human readable" form?

Based on the information provided on these points the user should be able to decide whether or not to trust the site with his personal data.

The following sample policy, figure 3.2, is included in the P3P 1.0 Specification. It shows the typical tags and layout for a site's P3P policy:

The **POLICIES** element (01) is the root element and gathers one or more P3P policies into one file. Each **POLICY** element (02) must have a distinct **name** to be able to reference the policy. The **discuri** attribute contains the URI of the natural language privacy statement, which should include information on how to contact the service with questions or concerns. The natural language privacy statement is used to provide the human readable form of the policy to the user. The **xml:lang** attribute is the language in which the policy is expressed, in this case English.

The **ENTITY** element (05) describes the legal entity behind the privacy statements. It consists of **DATA** elements for the name and contact information, e.g. postal address, telephone number, e-mail address and URI.

The **ACCESS** element (19) indicates whether the site provides access to various kinds of information or not. It describes the ability for users to view identified data and address questions or concerns

```

01 <POLICIES xmlns="http://www.w3.org/2002/01/P3Pv1">
02 <POLICY name="ForBrowsers"
03   discuri="http://www.catalog.example.com/PrivacyPracticeBrowsing.html"
04   xml:lang="en">
05 <ENTITY>
06 <DATA-GROUP>
07 <DATA ref="#business.name">CatalogExample</DATA>
08 <DATA ref="#business.contact-info.postal.street">4000 Lincoln Ave.</DATA>
09 <DATA ref="#business.contact-info.postal.city">Birmingham</DATA>
10 <DATA ref="#business.contact-info.postal.stateprov">MI</DATA>
11 <DATA ref="#business.contact-info.postal.postalcode">48009</DATA>
12 <DATA ref="#business.contact-info.postal.country">USA</DATA>
13 <DATA ref="#business.contact-info.online.email">catalog@example.com</DATA>
14 <DATA ref="#business.contact-info.telecom.telephone.intcode">1</DATA>
15 <DATA ref="#business.contact-info.telecom.telephone.loccode">248</DATA>
16 <DATA ref="#business.contact-info.telecom.telephone.number">3926753</DATA>
17 </DATA-GROUP>
18 </ENTITY>
19 <ACCESS><nonident/></ACCESS>
20 <DISPUTES-GROUP>
21 <DISPUTES resolution-type="independent"
22   service="http://www.PrivacySeal.example.org"
23   short-description="PrivacySeal.example.org">
24 <IMG src="http://www.PrivacySeal.example.org/Logo.gif" alt="PrivacySeal's logo"/>
25 <REMEDIES><correct/></REMEDIES>
26 </DISPUTES>
27 </DISPUTES-GROUP>
28 <STATEMENT>
29 <PURPOSE><admin/><develop/></PURPOSE>
30 <RECIPIENT><ours/></RECIPIENT>
31 <RETENTION><stated-purpose/></RETENTION>
32 <DATA-GROUP>
33 <DATA ref="#dynamic.clickstream"/>
34 <DATA ref="#dynamic.http"/>
35 </DATA-GROUP>
36 </STATEMENT>
37 </POLICY>
38 </POLICIES>

```

Figure 3.2: P3P policy example

to the service provider. In this example, the `<nonident/>` element means that the site does not collect identified data. Other examples could be to give access to all identified data, no access to data, or access to identified contact information only.

The `DISPUTES-GROUP` element (20) contains one or more `DISPUTES` elements (21) that describes the dispute resolution procedures. The `resolution-type` in this example indicates that users may complain to a third party organization. Other examples could be to handle complaints via the site’s customer service representative or take it to court.

The `REMEDIES` element (25) defines the consequences of a policy breach. The example uses `<correct/>` to signalize that errors or wrongful actions will be remedied by the service. Other possible outcomes of a policy breach could be to pay money or determine this based on legislation.

The `STATEMENT` element (28) describes data practices that are applied to different types of data, and groups together certain elements. The `PURPOSE` (29) could have many different values depending on the reason for processing of data. In this example, `<admin/>` means that the information may be used for technical support of the Web site and computer system. `<develop/>` indicates that information may be used to enhance, evaluate, and review the site, product, or market.

The `RECIPIENT` element (30) describes how data may be distributed beyond the service provider. In this case, `<ours/>` means that data is available to the service provider, and possibly a third party that processes data for a stated purpose. (e.g. a printing bureau which prints address labels.) `RETENTION` (31) indicates the policy for retaining the data that is referenced in the statement.

<stated-purpose/> means that information is retained to meet the stated purpose, and discarded at the earliest time possible. Other possible values here could be e.g. to retain information indefinitely, or no time at all.

The **DATA-GROUP** element (32) can hold several **DATA** elements that each describe a set of data that is collected by the site. The site in this example is logging the clickstream. This is typically server log information which may include the IP-address of the user, URI of the resource requested, time of request, size of response etc. A wide range of relevant collectable data elements has been identified and categorized in the *P3P base data schema* which is described in the next section.

### 3.1.3 P3P Base data schema

The base data schema is a standard set of elements that is commonly used by services, and should be known to all user agents. With the base data schema, P3P identifies a standard set of uses, recipients and data categories. P3P policies can reference these data definitions to ensure a common understanding of the policy and prevent ambiguity. The different data element sets are organized in a hierarchical way. For instance, the policy in figure 3.2 contains certain information about the organization from the **business** data set. In addition to the **business.name** element, this set includes a sub-structure of **business.contact-info**, which again includes a sub-structure of **business.contact-info.postal**. The definition of the **postal** data element set is shown in figure 3.3.

POSTAL	CATEGORY	STRUCTURE	SHORT DISPLAY NAME
name	Physical Contact Information, Demographic and Socioec. Data	personname	Name
street	Physical Contact Information	unstructured	Street Address
city	Demographic and Socioeconomic Data	unstructured	City
stateprov	Demographic and Socioeconomic Data	unstructured	State or Province
postalcode	Demographic and Socioeconomic Data	unstructured	Postal Code
country	Demographic and Socioeconomic Data	unstructured	Country Name
organization	Demographic and Socioeconomic Data	unstructured	Organization Name

Figure 3.3: Base data schema table: postal

Each data element is described and categorized in the table, and the underlying structure is listed. The element **postal.name** has the sub-structure **personname**, which defines the different elements of a name (e.g. **firstname**, **lastname** and **nickname**).

The base data schema specifies four basic data element sets: **user**, **thirdparty**, **business** and **dynamic**. User agents may support different mechanisms for the user to store values for the elements in the **user** set (e.g. the different elements of **personname**). The P3P specification also suggests that such mechanisms may include support for multiple personae.

### 3.1.4 P3P beyond HTTP

The use of P3P throughout this report is based on the P3P 1.0 specifications [10]. The specification presented for the 1.0 version includes little support for using P3P on web-services. The recently

released P3P v1.1 [8] is based on the P3P 1.0 Recommendation and adds some features, along with some revised element-definitions and new guidelines for implementation. This improvements also include adaptations towards use in web-services. Still, the P3P standard is mainly intended for use on web-sites. This means that some adaptations will have to be made to make the standard better suit a service-based system.

In 2003 a P3P task force report was published, presenting the specifications for requirements and scenarios for using the P3P language in other contexts than HTTP. The report is entitled **P3P: Beyond HTTP** and aims at building the foundation for applying the P3P standard also outside the originally intended use. The report focuses on providing support for P3P in web-services and adaptations that must be done to the P3P specifications in future versions of the language. Most importantly for UbiCollab is the suggested use of referencing the privacy policies through the use of the web-services' WSDL-files (Web Service Definition Layer). These files are used for describing the web-services and could include tags for describing its policy. This way of referencing policies will be discussed in the design chapter (5).

Further suggestions to adaptations to P3P presented in this report is not of great importance to the UbiCollab platform, since the report is not a specification, but merely a proposition on how to deal with web-services in later versions of P3P. The interesting part about the report is that there is research being done in the area and that we in the future might see a P3P version that is better suited for web-services, than the one existing today.

### 3.1.5 Evaluation

The P3P standard is an important contribution to online privacy. Although initially developed as a standard for web sites, its characteristics makes it applicable beyond the World Wide Web. The UbiCollab platform is using web service technology, where information exchange follows a client-server pattern similar to transactions on the World Wide Web. P3P allows different services to announce their privacy policies, which is what we want to see in the UbiCollab platform as well. Privacy policies are expressed in XML, and integration with the XML-based web service technology should therefore be straightforward.

One thing that P3P does not provide is data-types that describes contextual information. This will have to be addressed when adapting the standard to a context-aware environment like UbiCollab. Anyhow, the P3P standard is flexible enough to allow additional data elements to be defined by implementors, and some research has been done in this area. In particular, the contributions by Langheinrich [26] and Myles et al. [30] are important. Also, the WASP (Web Architecture for Services Platform) project [45] has made a relevant adaptation of the standard, and is described in detail in chapter 3.4.

Other privacy specifications have also been considered, e.g. IBM's *Enterprise Privacy Authorization Language (EPAL)* [1] and The *Privacy Policy Profile of XACML* [28]. Both are privacy policy languages, and share many characteristics with P3P. There are, however, some advantages to P3P considering an adaptation to our project. Firstly, P3P has a more global approach to privacy than the other specifications. It provides mechanisms to express aspects of privacy that apply beyond the boundaries of a specific organization, and it is developed to comply with the international diversity

of privacy laws. Secondly, it is flexible and can be easily adapted, and is also compatible with other technology tools.

In conclusion, we are confident that the P3P standard can be adapted to address the privacy issues that have been identified in the UbiCollab platform. This idea is supported by results from similar research projects [45] [26].

## 3.2 APPEL

P3P has a standard language for encoding users' privacy preferences; *A P3P Preference Exchange Language* (APPEL) [9]. As the counterpart to P3P policies, APPEL allows the user to express his preferences in a set of privacy rules called a *ruleset*. This ruleset can be used by his user agent to make automated or semi-automated decisions on the acceptability of the privacy policies found on a P3P enabled site. APPEL rules are expressed in XML, with a syntax similar to what is specified in the P3P 1.0 standard for policies. Rulesets are not intended to be read by end-users, but it allows the user agent to parse and compare machine-readable P3P policies with the users preferences. As a result, the user agent may simply display information for the user, generate prompts or take additional actions.

### 3.2.1 Rule processing and evaluation

The APPEL `rule`-element is a formal expression of a user's preference. A *rule evaluator* compares this to a service's P3P policy. Depending on the outcome of this comparison, it returns one of the *behaviors* that is defined by the rule. APPEL specifies three behaviors that must be supported by all user agents: *Request*, *limited* and *block*, as well as an optional *prompt*. If the *request*-behavior is returned, it means that the service policy is acceptable, and the resource should be accessed. The *limited*-behavior signals that the service policy is somewhat acceptable. The resource should be accessed, but only allowing the most necessary details to be retrieved. *Block* means that the policy and preference do not correspond and the resource in question should not be accessed. These three behaviors decide the actions taken after requesting a service. In addition, a *prompt* attribute might be received. The prompt attribute can have two different values; **no** or **yes**. When the prompt-attribute is *no*, the behavior should be performed seamlessly. This means that the operations is performed without any interference or input from the user. When a *yes* is given, the user should be prompted to be able to give his consent, or not, to the operation. The prompt-attribute *yes* is only given in connection with either the request-behavior or the limited-behavior.

The developers of the standard has acknowledged that rulesets may be difficult for end users to specify. They suggest that organizations create a set of recommended preferences that users can accept if they trust the organization. *"Primarily, we envision this language will be used to allow users to import preference rulesets created by other parties and to transport their own rulesets files between multiple user agents. User agent implementers might also use this language (or some easily-derived variation) to encode user preferences for use by the rule evaluators that serve as the decision-making components of their user agent."* [9]. Accordingly, a lot of work is left for the developers to create easy-to-use applications that allows the users to post their preferences in an intuitive way.

The *persona* identifier defined in the APPEL language is also worth mentioning. This is a unique identifier for a set of data elements in the user's data repository that the user wants to use during the current session. Implementations could extend this identifier to allow the user to store multiple personae and choose the one he wants to use for the different actions during a session. Later versions, and adaptations made for use beyond HTTP, should include an extended version of this identifier to support the use of multiple identities through APPEL.

### 3.2.2 Evaluation

The APPEL language is a work in progress and there exists some minor problems related to using it as a preference language. It has been found to be too complicated and an implementation of a preference configurator would in its simplest form involve tens of buttons and options [45]. It has also been found that APPEL leaves room for some ambiguity [45]. The future of APPEL will most probably involve improvements in these fields and hopefully it will be better suited in future editions. Future implementations of APPEL in UbiCollab should include the use of a preference editor to provide the users with an easy to use system for setting up their preferences. This thesis will just demonstrate that the APPEL language works by implementing a small ruleset that can be negotiated with the policies configured in the services.

## 3.3 A Privacy Awareness System for Ubiquitous Computing Environments (pawS)

Several articles have been found concerning the use of P3P in a ubiquitous environment, but few are based on the same design principles that we introduced in chapter 1.1.1. These principles are taken from an article by Mark Langheinrich [25]. Further work has been done by Langheinrich where he describes a privacy awareness system for ubiquitous computing. The following section is based on the work in "*A privacy Awareness System for Ubiquitous Computing Environments*" [26].

This work presents a privacy awareness system called pawS, which tries to implement a balanced system that combines strict security for sensitive information with openness and mutual trust for less sensitive information. Based on the same privacy principles as our work, this article seems to be good guidance for developing a privacy awareness system in UbiCollab as well. In contrast to the previous work with UbiCollab, the implementation of pawS doesn't support anonymity, pseudonymity or security because these issues has been found to be useful tools and only treated as being a supportive part of the infrastructure. Still the article is a suitable research objective for us, giving insight into how to deal with the other issues of privacy in a ubiquitous environment. Especially interesting is the use of privacy proxies, which might be the best solution for UbiCollab when it comes to supporting the users' privacy.

The privacy principles that have been preserved in the pawS architecture are **Notice, Choice and consent, Access and recourse** and **Proximity and locality**. In connection with integrating the use of P3P in UbiCollab, the choice and consent and notice sections will be presented. Notice and Choice and consent are the principles that are possible to support with the use of the P3P privacy



policy standard. In addition, the principle of **Access and recourse** will be presented, since this principle is supported through the use of privacy proxies in pawS.

Each of these principles leads to a high-level requirement for the system. The requirements are presented below along with what pawS suggest for covering them. The last part of this section presents a discussion covering the suitability for UbiCollab.

### 3.3.1 Privacy policies

The principle of notice supports the user's right to know what data collections are taking place. The system needs mechanisms for declaring this information. pawS suggests to use privacy policies for declaring the collection practices. The policies are machine-readable XML-files that state the data practice of the services and allow automated processes to read them and take actions on them. The policies will be able to specify data collection, storage and distribution parameters that can be processed by the users' clients to support choice and consent. Further details of the use of P3P was introduced in section 3.1.2

### 3.3.2 Policy announcement mechanisms

To announce the privacy policies Langheinrich suggests two different methods according to the different types of data collection that takes place. This concept supports the demand to notice in the system. The two methods suggested are **Implicit announcement** and **Active policy announcement**. When the clients are actively looking for and activating a service, the implicit announcement is used. In such environments the article suggests embedding links to the P3P policy into the service discovery protocol. When actively announcing the policy, Langheinrich suggests using a privacy beacon that constantly announces the privacy policies. This method is required in environments where the services work continuously in the background without any user interaction to activate them.

The issue of policy announcements is a very important concept concerning privacy since it is used to support the demand to consent, which is the best way to increase the users' control over the gathering and use of personal data.

### 3.3.3 Privacy proxies

The pawS architecture proposes the use of privacy proxies both for the services and the users in a ubiquitous environment. The proxies will handle relevant interactions between data subjects and the data collectors to preserve the privacy preferences stated by user and system. The proxies are continuously running services that are accessible to the data subjects anytime. The use of a *personal privacy proxy* is an interesting notion for UbiCollab, since this will satisfy the demands to access to the personal data gathered by the system and the system will be able to control how this data is gathered. The privacy proxy is configured using APPEL, which is the privacy preference language

chosen for pawS, and UbiCollab as well. The main functionality of such a proxy will be to agree on the data gathering practices of the services according to the preferences of the user.

### 3.3.4 Evaluation

The pawS architecture offers a lot of ideas to apply to the UbiCollab privacy architecture. All three concepts mentioned in the previous chapter are ideas we would like to pursue in UbiCollab. To give a better overview of how pawS suits the demands stated to Choice, consent, notice, data access etc. , figure 3.4 has been included.

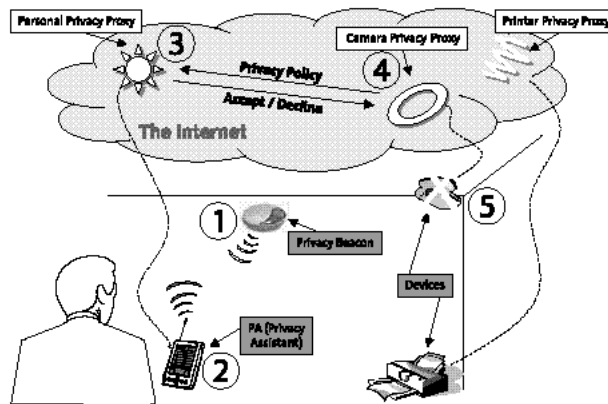


Figure 3.4: The pawS architecture [26]

The figure gives an overview of the main components that a privacy management system should include according to Langheinrich. The model is very similar to what we aim at with our work with UbiCollab. The core of the system is the use of P3P policies that are negotiated with the clients that either give consent to the requests or not based on the users' privacy preferences. The previous architecture of UbiCollab supports the use of P3P policies. In this edition we would like to take the platform one step further by developing a solution using a privacy proxy for the users.

## 3.4 WASP - Web Architectures for Services Platform

The WASP research project in the Netherlands is developing an application environment for mobile, context-aware services. The applications that are developed on top of this platform can be accessed via mobile phone or other devices over the new 3G mobile networks. The WASP platform gathers contextual information about the users, and lets the applications use this information to provide relevant, personalized services. The initial focus is on tourist applications that use location information to present museums, restaurants and other services.

WASP is a collaborative research project that involves participants from the Telematica Instituut, Ericsson and The University of Twente. One of the contributions is a proposal for a privacy control

architecture for WASP, based on the P3P privacy policy description standard. Martijn Zuidweg [44] [45] describes how this standard has been applied and adapted to suit a context-aware system based on web-services.

### 3.4.1 Using P3P in WASP

The author has identified several reasons why P3P is suitable in WASP:

- The P3P standard was developed for web-sites, but since the main purpose is to simply describe services, its applicability is much wider. Web services are based on much the same interaction as the WWW with a client-server interaction, which makes it easy to adapt the P3P to suit WASP. At the same time the web services and P3P are both using XML as the conversation language.
- The details on how and where contextual information is gathered and stored is irrelevant, P3P simply describes what data is collected by a service.
- The inquirer, or service provider, is an important determinant for people's privacy preferences. Users will normally have the same preferences for the same data collector. Privacy preferences are influenced by a description of the data collecting service, which is a key object of P3P.

The author also points to other successful implementations of P3P in context-aware systems. Langheinrich [26], Myles et al. [30] and Nilsson et al. [15] all propose P3P-based privacy control mechanisms for context- or location-based environments.

When building a P3P-based privacy architecture for WASP, one of the challenges was to extend the P3P and APPEL standards to be able to reason about contextual information: *"In a context-aware environment, we need two concepts for users to be able to fully evaluate a service's practices: The service's privacy policy, that describes the general, unchanging practices, and a user's context-dependent constraints, that restricts the service's behaviour when contextual conditions temporarily influence the user's willingness to be subject to the service."* [44] According to the article, a user's constraints could look like this:

*"I only want to use this service...  
... during work hours.  
... while I am in my office building.  
... on weekdays.  
... when I am not busy."*

The article describes the necessary extensions of P3P data elements and the proposition of a Context-Dependent Preference Language (CDPL) that is integratable with APPEL. A prototype has also been implemented that shows the use of the adapted standards.

### 3.4.2 Evaluation

The WASP privacy architecture shows a successful implementation of privacy mechanisms that are relevant to the UbiCollab platform. The designers use P3P to promote privacy in a context-aware environment, on a web-service based platform comparable to UbiCollab. This gives us confidence that we can take a similar approach when designing a privacy architecture for the UbiCollab platform. The results of their research on adapting the standards for context- and location-awareness could be applicable to UbiCollab as well. In this project we will concentrate more on the design of an identity management system based on P3P, but the suggested privacy control mechanisms could be integrated when UbiCollab has been redesigned to support P3P policy negotiation.

# Chapter 4

## Analysis

This chapter will bridge the ideas presented in the preliminary chapters with the research that has been done on the technical parts of the project. The development of the scenarios and the subsequent conceptual requirements is an important phase in order to produce an architecture that satisfies the privacy demands. The result of this chapter will be a presentation of the design issues we will have to consider and a description of how the problems of preserving privacy in UbiCollab will be solved by this project.

### 4.1 UbiCollab users

One of the main aspects of our work with UbiCollab consists of creating an architecture that allows the users to be able to control what information about them can be gathered by other users and the system. This also includes letting the users set up different profiles or identities according to what they are doing and with whom they are working. UbiCollab users are expected to register personal information like e-mail, name, etc., which is useful in collaboration and communication with others. In the previous UbiCollab design [6] the users are anonymous towards other users and the services in the platform. This has been done to increase the general level of privacy on the platform. To improve the functionality of the platform, the users need a way to perform transitions between being totally anonymous, letting some information be gathered and allowing others to see their real identity. This issue will be handled at the platform level and not at the client-side, which means that the platform will support the idea for future extensions to the platform. Figure 4.1 shows how the different user representations are connected in the conceptual idea of the UbiCollab users.

The figure shows the idea of how the different states of the user can be perceived by other people using the system. The figure shows the three different states of a user; **The users real identity**, **the multiple identities on the platform** and **the anonymous state** used in UbiCollab and how the user is perceived by different types of users in UbiCollab. Normally users will be perceived differently when collaborating with others compared to just using the services on their own. The figure shows that users outside of the collaboration instances will just see you as an anonymous

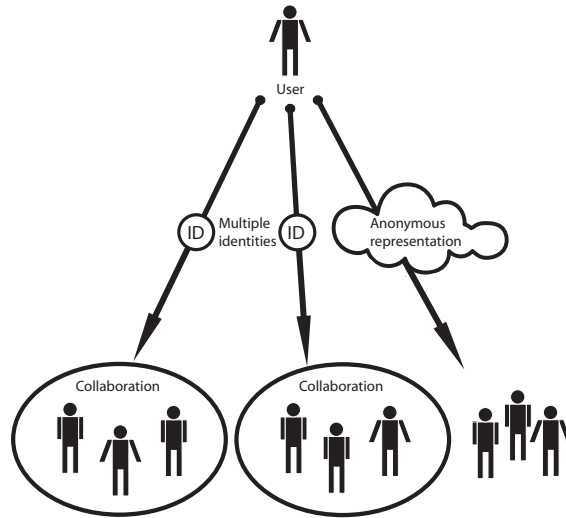


Figure 4.1: User representation

representation. The people you are working with, the ones inside the same collaboration instance as you are, will see one of your identities. The user representation might be the same to both collaboration instances, but it might also vary, as shown in the figure. This representation of the users is based on giving other users different ways of perceiving you, while you don't have to switch between identities yourself. The different states of the users and how the system and other users perceive them will be an important aspect of this thesis. Further detail and descriptions of how to design for these issues in UbiCollab will be presented in the design chapter (5).

## 4.2 Management of privacy and identity in UbiCollab

In the existing UbiCollab solution the user privacy management has been done in a very simple fashion. Figure 4.2 shows how the different components for handling the users identity and privacy are connected in the previous solution of UbiCollab.

The previous solution only used one identity for each user, who was represented anonymously in the platform. The anonymization was done by the AAA server, which authenticated users, but kept the real identity hidden from the platform. The privacy service in figure 4.2 was used to handle requests that needed the users' identities. This information was then gathered according to the privacy preferences and policies of users and services in the system. The management of privacy was designed to support the use of the P3P protocol on the service side, and APPEL on the user side. This concept is going to be preserved in the new solution, with additions to suit the use of several different identities for the users. The most important change will be to develop a proxy layer on top of UbiCollab, that handles the negotiation between the P3P policies and the APPEL preferences.

After studying related work on similar platforms and the use of P3P in general we have decided to develop an architecture based on a negotiating structure with P3P as a foundation also for handling

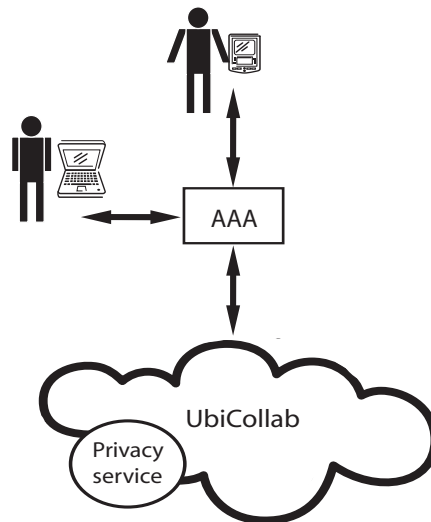


Figure 4.2: Identity and privacy management in UbiCollab

the management of different user identities. This will both satisfy the earlier design of UbiCollab and make it possible to enhance the solution by integrating multiple identities into UbiCollab. Other research has been done to ensure that P3P is suitable for doing identity management, which encourages us to try out the same on UbiCollab.

The article "**Identity Management Based On P3P**" by Olivier Berthold and Marit Köhntopp [5] shows the basic approach for developing an identity management system with the use of P3P. This article shows that P3P sufficiently supports the development of an identity management system. In this thesis the article will, together with the problem elaboration chapter (2), provide guidelines for our architecture of the basic mechanisms needed for identity management. *"Identity management is something we do in normal conversations everyday when we decide on what to tell one another about ourselves"*. This is the most important aspect of managing identities; to allow the user to be in charge of what the different identities are going to reveal about yourself. Further the authors claim that the empowering of users to be in control of the different identities makes the system appear similar to social behavior in real life, which should be the ultimate goal for ubiquitous systems. The article presents demands to an identity management system, and discusses how P3P complies with these demands. The following list shows the most important demands to an identity management system as stated by Berthold and Köhntopp;

- **Criteria C1** Privacy protection baseline
- **Criteria C2** Empowering the user
- **Criteria C3** Representation of pseudonyms/roles/identities cards with different properties
- **Criteria C4** Based on standardized protocols and open data structures
- **Criteria C5** Possibility for easy monitoring

- **Criteria C6** Compliance with legal framework

This list represents the demands they propose for a "powerful" identity management system, which they aim to develop. In this thesis we will focus mainly on the first three points, since one of the objectives of this task is to allow later contributions to build a more extensive identity management system on the application level: Our aim is to be able to provide the basic support for this on platform level.

### **A Privacy protection baseline - C1**

A privacy protection baseline includes the basics of privacy preservation. The following items list the most important issues of the privacy protection baseline on both platform and application level. Even if we intend to build the support on platform level it's important to take the considerations of the application level into account;

- Anonymous underlying communication network
- Data security concerning communication with other parties
- Trustworthy user devices - Client side
- Transparency; through user open source and an informative user interface
- Personally restricted access to the identity manager - Client side
- Validation of the data security level by independent experts

All these issues are important for the development of UbiCollab, but the client side issues are not the focus in this thesis, since our work aims at developing the basic platform mechanisms for privacy protection. Trustworthy user devices, anonymity in the communication and personally access to the identity manager through the log in process are issues that are already supported in the existing architecture, and will not receive more attention in this thesis.

### **Empowering the user - C2**

By empowering the user the authors list the following points as the most important objectives:

- Convenient user interface to manage different pseudonyms/roles/ identity cards
- Storage of personal data under user control
- Possibility of recording the communication in order to be able to reconstruct which party possesses which personal data



- Negotiation tool for the decision of which data the user wants to disclose and under what conditions
- Supporting user-controlled privacy facilities like grant of consent, data inspection, desire for change, desire for removal, and revocation of consent
- Negotiation tool for other aspects like personal reachability or security configuration
- Possibility of support from privacy protection authorities

Some of these demands are related to the ones discussed in the conceptual requirements 2.5, while others are demands to the applications.

### **Representation of pseudonyms/roles/identity - C3**

Representation of pseudonyms/roles/identity cards with different properties concerns having mechanisms for this through cryptographic means (like signatures, credentials etc.) and integration of public key infrastructures. In UbiCollab users are anonymously represented and identified by a sessionkey, that keeps track of the users actions on the platform. This key will stay the same even if the users decide to switch to another identity. This concept has been integrated in the existing architecture on a basic level, and the details about this and how it has been solved will be presented in the design chapter (5).

### **Compliance of P3P with the presented criteria**

The presented lists of demands are not realized in P3P as it is today, but the interesting part is that none of the criteria introduced contravenes the P3P specification. More details on P3P and its specification were presented in the **Related Work** chapter. According to our research, P3P is a good choice to build the basis for an identity management system on.

This analysis of the criteria for a identity management system shows that there are many aspects that will not be covered by this thesis since they are out of the scope. Some of the issues presented will not receive further attention in this report, but has helped to convince us that P3P can fulfill the needs for privacy, also when building a foundation for future contributions. The result will be a platform that makes it possible to build a more refined identity management system on top of the platform developed in this thesis.

## **4.3 Analysis of conceptual requirements**

The scenarios in chapter 2 resulted in the conceptual requirements for this thesis. Together with a study of related material we have been able to locate the focal points for the designing phase of the project.

Prior to the design phase the following list has been developed to specify the areas which need to be covered by the architecture in accordance with the research done in the field of privacy in ubiquitous systems. The points on the list correspond to the conceptual requirements presented in section 2.5 and show how to deal with the problems defined in the problem elaboration.

1. The system will support anonymity, pseudonymity and the possibility of real time switching between the different representations of a user. As presented in section 4.2, this is possible to support with the use of P3P. Further it's important to relate this to the collaboration instances that the user is connected to and design mechanisms that can be used for building a reputation based on a user's previous actions. P3P connected with the use of a privacy proxy will handle the demand to choice and consent and provide the user with improved control over the level of privacy he wants.
2. To be able to express the users' and the services' actions the P3P protocol will be integrated into the basic user and privacy management of the platform. On the client side the use of APPEL will let the users decide what actions they accept from the services and allow the proxy to perform a negotiation on the data practices presented in the services P3P policies.
3. Conceptual requirement number 3 is solved by developing a privacy proxy on top of UbiCollab that can handle the requests from the clients. This proxy needs to be in charge of the P3P/APPEL negotiation and be able to provide the existing API to developers that wants to build applications on top of UbiCollab.
4. Highly sensitive information, as stated in the conceptual requirement number 4, needs to be handled by an external and trusted service. The most suitable solution for this is to keep a trusted 3<sup>rd</sup> party service that will be used for services that deals with the exchange of information that will need a higher degree of security than provided by the UbiCollab platform.
5. P3P will handle the basic identity and privacy management of the system. As described in section 4.2, the control over personal information and the actions performed on this information is supported by the use of P3P. This will be further described in the design chapter.
6. When developing the new architecture, changes will be made to the basic communication internally in the platform. With the use of a privacy proxy on top of UbiCollab, changes will also occur to the communication between the applications and the platform. To cover the demand in conceptual requirement number 6, the API will have to be maintained as it is today when moved to the privacy proxy. Further, the reengineering process must be performed carefully to assure that the internal structure is preserved when moving functionality from inside the platform to the privacy proxy. Another important issue is the preservation of the clients. The existing clients will be using the new API, so some changes will have to be done concerning the log-in process and user authentication.

## 4.4 Conclusion

By analyzing the conceptual requirements and looking at the existing platform it's obvious that there will have to be made changes to the existing architecture and the existing platform prototype. The

following privacy issues sums up the most important technical aspects to focus on when designing the new UbiCollab platform.

<i>Focus Area No</i>	<i>Description of task</i>
FA1	Develop an architecture for using a <i>privacy proxy</i> for the users on top of UbiCollab, that deals with the communication with the services.
FA2	Develop support for <i>P3P and APPEL</i> in the architecture to handle negotiation between the users and services
FA3	Integrate <i>multiple identities and reputation</i> into UbiCollab Design ways to make transitions between the different identities and allow the user to control this switching.
FA4	Integrating a <i>3<sup>rd</sup> party service</i> for handling highly sensitive and personal information about the users into the architecture.

## Chapter 5

# Designing for privacy

### 5.1 Introduction

The overall goal for this work and our previous contribution to UbiCollab is to increase the level of privacy in the platform, while at the same time preserve the functionality and improve the users' control over the information gathered by the services. The new UbiCollab design maintains the focus on the users and the idea of allowing the user and system to negotiate what information is allowed to be gathered and shared. By preserving this idea as the basic privacy measure, improvements will be made both in the design and in the prototype implementation presented in chapter 7 (Demonstrator). To ensure compatibility with the research and the ideas presented in the Problem elaboration chapter, the design phase extends from the conceptual requirements in section 2.5 for developing a model for design and implementation.

This chapter presents a review of the previous privacy design to be able to state the necessary changes that will be made to UbiCollab. Further, the conceptual model and the new architectural design will be presented along with a more detailed description of the main components of the privacy architecture. Based on this chapter it will be possible to develop a platform prototype that preserves the conceptual ideas presented in the thesis.

### 5.2 UbiCollab privacy review

UbiCollab has experienced improvements concerning privacy during our research project last year [6], but the work done on the platform in this regards is far from enough to provide the users with an adequate amount of privacy. The developed design included a privacy service and an AAA server on top of UbiCollab. The existing platform also allows the users to stay anonymous in UbiCollab, which covers one of the main ideas of privacy in computer systems; Making it difficult to link any information to a real identity. At the same time this mechanism may hinder the functionality by

not allowing the users to find their co-workers or friends while being logged on to the platform. Accordingly, there exist possibilities to improve the privacy on the platform while preserving the main ideas and functionality of the platform.

In order to develop an extended privacy architecture on UbiCollab the previous privacy design needs to be analyzed to be able to locate the strengths and weaknesses. Figure 5.1 shows the connection between the main elements of the previous UbiCollab architecture.

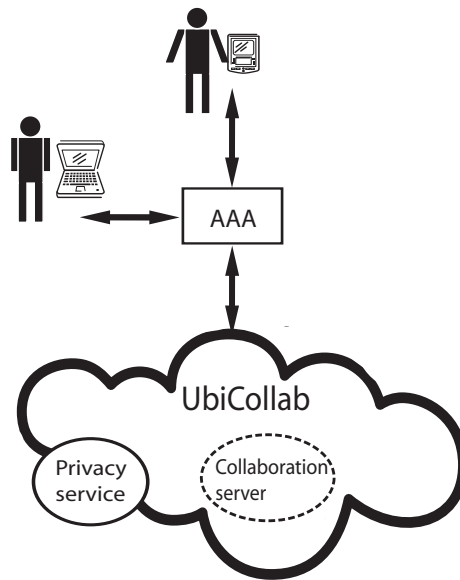


Figure 5.1: Privacy components in the previous UbiCollab design

The main components for privacy preservation here are the **AAA-server**, the **privacy service** and the **collaboration server**. The internal structure and the general data tables are not described in detail, since these entities are more a result from the architecture than mechanisms for achieving the desired level of privacy. The following sections present the most important components of the privacy architecture along with a discussion on their value and suitability and whether or not they are to be changed in the reengineering of UbiCollab.

### 5.2.1 Components in the previous design

#### AAA Server

The AAA server is a service that handles the communication between the users and the system interface. The use of the AAA server is described in Appendix D.1.2. The purpose of this server is to negotiate the users' privacy preferences with the policy description of the system through the use of the P3P standard. This service is also meant to keep the connection between the generated pseudonyms and the real identities of the users. The communication between this service and UbiCollab is done through web-services. Consequently, the AAA also provides the API for UbiCollab, which is somewhat out of the area of functionality originally offered by such a service.

The use of an AAA server is not the best solution for preserving privacy in UbiCollab. There exists some implementations of such services ( Radius [33], Diameter [35] etc.), but they are not very suited for use in UbiCollab. There might be a need for an AAA service for use with UbiCollab, but this service should not be the main entry-point into the platform. The reason for this is that the AAA has just a limited set of functionality. In our case it would need more functionality for handling users, resources, and collaboration instances and also provide the API, which makes the service very dissimilar to the idea of an AAA server. Because of this the AAA will be replaced in the new design. A similar service, without the access-point functionality, might be useful for UbiCollab to provide control over the sensitive information.

#### **UbiCollab - Collaboration server**

The collaboration server is the core of UbiCollab and distributes the requests of the users and the services offered by the system to the desired destinations. The collaboration server handles anonymous users. The existing collaboration server will keep its main functionality, but to improve the privacy design, changes will have to be made. One of the ideas presented earlier (section 2.4.3) concerns how to handle multiple identities on the platform, which is one of the reasons for changing the collaboration server. Further changes will also be performed, and will be presented in the new architecture.

#### **Privacy service**

The privacy service simulates a privacy proxy between the users and the services offered by the platform. In the previous privacy architecture the collaboration server negotiates the preferences of the user with the policy description of the system with the policies provided by the privacy service. This idea will be preserved in the new UbiCollab privacy design, but changes will have to be made to this service as well. In the previous architecture the privacy service is a basic service that is controlled by the collaboration server. To better protect the users privacy and improve the proxy functionality of the service, it must be moved outside the UbiCollab platform acting as a privacy proxy for the users and the main entry point for communicating with UbiCollab. With help from the collaboration server, the privacy proxy will found the basis for privacy management on the platform.

### **5.2.2 Conclusion**

The previous design is suitable for dealing with anonymous users as the main mechanism for preserving privacy, but when extending the privacy mechanisms further improvements will have to be made. Especially in connection with introducing multiple identities and a more effective policy description negotiation tool, extensions to the design will have to be made. The subsequent sections will introduce the issues that will be the focus for the new design and present a conceptual model developed from the conceptual requirements (2.5). The development of the design, and also the prototype implementation, will be based on these issues.

## **5.3 Privacy design issues**

The report has presented **notice, choice and consent** and **anonymity and pseudonymity** as the main principles of privacy considerations in ubiquitous systems (1.1.1). These issues are visualized

in the scenarios, where their importance for UbiCollab is presented. The conceptual requirements were developed to preserve these issues, and the following **Focus Areas** were deducted from the requirements stated;

- FA1 - Privacy Proxy
- FA2 - P3P/Appel
- FA3 - Multiple Identities and Reputation
- FA4 - Trusted 3<sup>rd</sup> party service that handles highly sensitive information

The following table summarizes what was supported in the previous privacy architecture and what will be developed by this thesis:

<i>Focus Area</i>	<i>Previous version</i>	<i>New version</i>
FA1	No privacy proxy. The AAA was intended to have much of the same functionality.	Privacy proxy will be developed to handle privacy of the users
FA2	Idea of P3P and APPEL was introduced, but mostly on a conceptual level.	Refined privacy management system based on P3P and APPEL will be developed.
FA3	No multiple identities or reputation. Only anonymous users.	Identity management system will be developed, that handles multiple users and makes it possible to build reputations based on previous actions.
FA4	AAA was intended to cover this area. Not really the functionality we need for this entity.	A trusted 3 <sup>rd</sup> party service will be integrated in the design to support handling of highly sensitive data.

These issues are connected to the technical aspects of the platform, but will provide mechanisms for solving the high level problems presented in the scenarios. The use of a privacy proxy will give the user better control over the gathering, use and distribution of personal information by taking responsibility for all communication between user and platform.

APPEL and P3P will provide the necessary means for allowing communication between the platform and the users. Accordingly, the use of these standards will be essential for developing the means of communication in the privacy proxy.

By introducing multiple identities in UbiCollab, the users will be offered the possibility to change appearance according to the actions they wish to perform. This will improve privacy by allowing a flexible variety of the amount and type of personal data the user gives away to the other entities in the system. This mechanism is also supported by introducing P3P and APPEL. The development of a reputation system will increase the collaborative efforts and thereby enhance the main functionality

of the platform. In connection to the privacy design, a reputation system will provide proof that the underlying privacy mechanisms increase the functionality of the platform.

The use of a 3<sup>rd</sup> party service is important for building a complete privacy solution. At the same time, this isn't the most important feature of the platform when designing for the basic privacy measures. The use of such a service will be incorporated in the solution, but will not receive any further attention than a description of its functionality and basic structure.

By developing mechanisms for handling these areas, the conceptual requirements will be covered and the platform will be taken another step closer to managing privacy and identities of the users in a powerful and fulfilling way and in accordance with what was presented in the scenarios. It has also become obvious that the development of a system for negotiating P3P policies with the APPEL preferences will lay the brickwork for the rest of the privacy mechanisms, since most of the other issues are dependent on the negotiation of policies to work properly in UbiCollab. Accordingly this will be the main priority of the architecture and prototype implementation.

## 5.4 Conceptual Model

Based on the research done in the previous chapters, a new conceptual model of UbiCollab with increased privacy support has been developed. The conceptual model will provide a "mental map" that helps us build the logical connection between the demands to functionality and the technical solutions presented in the design. The aim for the conceptual model is to be able to present the system with focus also on the user side of the platform. The hope is that the conceptual model is in alignment with how the users would like the system to be.

Figure 5.2 shows the main components of the model and how the platform will handle users and services. The main intention of this model is not to provide details for the specific parts of the platform, but to give an overview of what parts of the system that has been incorporated into the design to support high level privacy mechanisms.

Figure 5.2 is centered around the privacy proxy, which is the main entry-point to the platform. It presents the essentials of the privacy mechanisms that need to be incorporated into UbiCollab to support the conceptual requirements stated in chapter 2.5(Problem Elaboration). The model is based on the areas that were derived from these requirements and refined in the analysis (4). The figure involves both users and other entities to show that the model involves all the entities that can be connected to UbiCollab, as introduced in previous versions of the platform.

The most interesting part of the model is the privacy proxy, which is the most important part of the privacy architecture of UbiCollab. The privacy proxy is needed for all access to the platform and to be able to provide an API for the developers of future applications in UbiCollab. Further, the figure shows UbiCollab as a part of the Internet, to illustrate that the platform is part of a larger communication network and can be accessed from everywhere and with any device that has a UbiCollab client. Since the privacy proxy is based on web-service technology, it is accessible by every device that supports the use of web-services. This supports the ubiquity of the platform services as well as privacy concerns related to the use of such services.



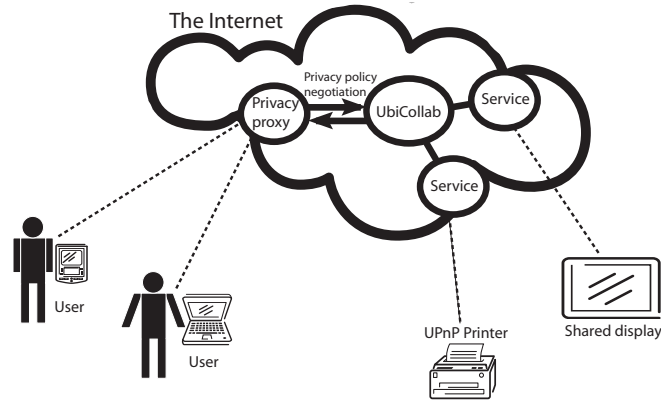


Figure 5.2: High level design of privacy in UbiCollab

### 5.4.1 Low level conceptual model of UbiCollab

Figure 5.4 is a conceptual model that shows the different parts of UbiCollab on a lower level than the previous model. This model shows the structure of the platform and how each component is connected to each other and gives an overview over the most important components in the architecture.

The following sections will describe the ideas behind each entity of the platform and how they will behave to be in accordance with the conceptual requirements and the new UbiCollab privacy design.

### 5.4.2 Client

UbiCollab clients have been developed to give users access to the functionality offered by the platform. At the moment, the clients are thin, to be able to be run by small devices, and perform only basic operations as a proof of concept for the developed architecture.

A client runs on the user's device, which could be e.g. a PDA or a laptop. Previously, a web-client and a PDA client have been developed. The clients need the privacy proxy to be able to make use of the platform. All users are registered with a unique username and Id in the proxy, which are provided by the user when the client logs him on to the platform. The clients use the proxy interface

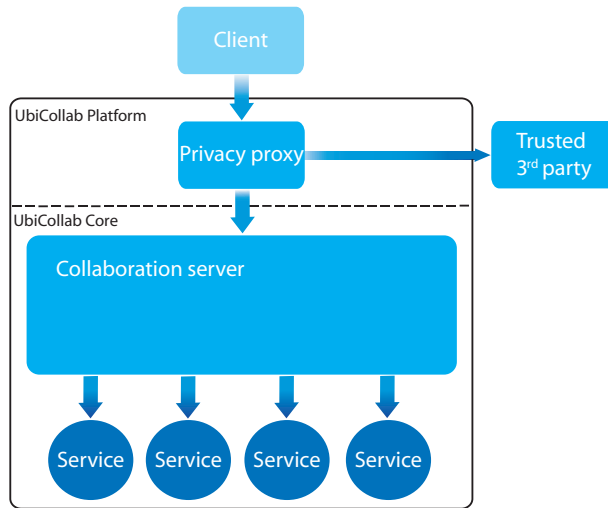


Figure 5.3: Conceptual model of the UbiCollab platform

to make requests to the platform by storing a sessionkey, that is received when logging in, which is used for validation. The client side of UbiCollab will still be a lightweight application and the development of new clients will be done in same fashion as before. The difference from previous versions is that the client will communicate with UbiCollab through a privacy proxy that preserves the users privacy in the platform.

Future contributions might enhance the functionality of the clients to allow the users to perform far more advanced operations with the help of UbiCollab. In connection with privacy, this will typically include advanced user management, preference configuration tools etc. This is out of the scope for this project and will not be given any attention in the architecture.

### 5.4.3 Privacy proxy

The privacy proxy takes over the responsibilities of user management that formerly resided in the Collaboration server. Adding a layer between the user and UbiCollab also enables implementation of the various privacy mechanisms that have been discussed. This is where the users' personal information is stored, and also where authentication is done when accessing UbiCollab. Anonymous use of services is made possible by representing the user by a random pseudonym in the platform. Only the proxy knows the real identity of the user.

One of the vital functions of the new proxy is the negotiation between service privacy policies and the privacy preferences of a user. This ensures user-controlled release of private data, and enables implementation of advanced user management functionality. The user can specify a set of multiple identities which are used to vary his appearance towards the system and collaborators.

#### 5.4.4 Trusted 3<sup>rd</sup> party service

In order to handle highly sensitive information the new design of UbiCollab will include a trusted 3<sup>rd</sup> party service. The trusted 3<sup>rd</sup> party service is based on the idea of using an AAA server as introduced in the previous design. This service will control and distribute information that is considered too sensitive to be allowed accessed and distributed through the same mechanisms as other personal information. The users will accept the use of such a service when registering their personal information in the proxy. To allow the service to make money transactions etc. the users will either be prompted by the platform or they have already agreed to this in their preferences.

#### 5.4.5 Collaboration Server

The Collaboration server will be accessed through the privacy proxy. The functionality of the platform is controlled by this component and every service in UbiCollab is connected to the collaboration server. The collaboration server provides the privacy proxy with the services' privacy policies (P3P) to allow a negotiation with the APPEL preferences of the users.

The collaboration server will provide the basics for privacy management by communicating with the services and providing the policies that the privacy proxy will evaluate. All requests from the services or the users will be processed by the collaboration server and delivered to their receiver. The collaboration server will still be the core of the UbiCollab platform and handle all the issues connected to collaboration instances and the other basic functionality of the platform.

#### 5.4.6 Services

The services will not be subject to any major changes in the new privacy design, since our work mainly consist of reengineering the platform. The only new addition to the services is that they will need a way to post their P3P privacy policies. In accordance with the requirements to the use of P3P policies, all services will include a reference to their privacy policies, so that they can be fetched when necessary.

### 5.5 Redesigning UbiCollab - architectural overview

This section offers a detailed description of the redesigned UbiCollab platform. The proposed architecture supports the ideas on privacy and identities discussed in the previous sections. The new design builds on the design proposed by Schwarz [37], which is briefly described in chapter 1. Figure 5.4 shows an overview of the main components of the system in connection with the privacy mechanisms on the platform.

The different components of the architecture will be presented in detail in the following sections.

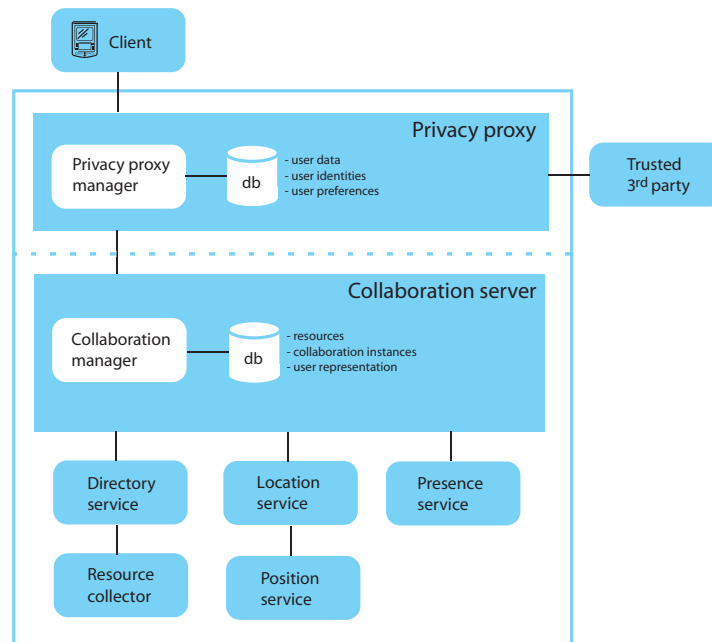


Figure 5.4: Architectural overview of the UbiCollab platform

### 5.5.1 Privacy Proxy

The privacy proxy is the access-point to the platform and will provide a new layer on top of the existing UbiCollab platform. This layer will serve the following purposes:

- Provide an **API** for UbiCollab application developers
- **Authenticate** users when logging in to UbiCollab, and **handle user data**
- Enable **anonymous** use of UbiCollab services
- Communicate with a **trusted 3<sup>rd</sup> party** service for sensitive user information
- **Evaluate** UbiCollab service privacy policies against users' preference rulesets

The proxy is the new interface towards the clients, and will provide an API for UbiCollab as well as the new functionality implemented in the proxy. All communication is done through the proxy, which is necessary in order to make possible the new mechanisms of privacy- and user-management. In contrast to the previous design, which relied on a platform-dependant privacy service, the new proxy is a standalone web-service. The communication is done by remote procedure calls over HTTP, in a similar fashion to the interactions between the client and system.

The user logs on to the proxy to establish a connection. The username and password are authenticated in the proxy, and a random pseudonym is generated and sent to the UbiCollab platform

to initiate a new session. The proxy holds the connection between the user’s real identity and the random identity. This ensures that the user is anonymous in UbiCollab, and his actions cannot be traced back to his real identity.

Developers may implement functionality that allows a user to create multiple identities in connection with his general user profile. The identities could be designated to certain actions, services or collaborations he is involved in. The general data model is illustrated in figure 5.5.

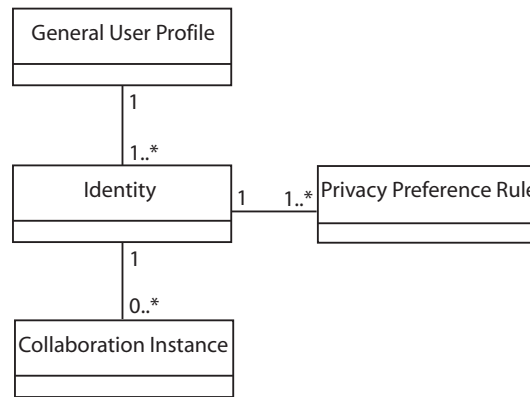


Figure 5.5: User identity data model

Because a collaboration instance is connected to a specific identity rather than directly to the users profile, his appearance can be varied accordingly. Each different identity has an individual set of privacy preferences connected to it. The scope, content and availability of personal information for each identity is defined in the preference set. The user must decide which of his generated identities he wants to make use of in a given situation, this will determine how he is perceived by the other parties. The user is free to decide how many such identities he needs, and may just as well have only one identity for use in all interactions. The flexibility of this model allows the developers to implement solutions that have a range of complexity and freedom of choice for the users. This also supports the idea presented in chapter 2 (section 2.4.3).

The model also specifies the use of a trusted  $3^{rd}$  party service to handle the users’ sensitive data. Developers may utilize this when designing services that will require e.g. credit card details and other information considered sensitive to the user. In this design, it is considered a generic service that will ensure secure storing and handling of data, and increase the level of trust and acceptance from the user. Additionally, this service could provide preconfigured preferences for the users to adjust and apply to their settings. This would ease the efforts of the users, since modifying the APPEL preferences manually can be a cumbersome process [26].

The proxy handles the negotiation of P3P policies and user preferences. As described, the user may define several preference sets, and appoint them to a range of purpose-specific or more general identities. The policies defined by the services are evaluated against these preferences when the user makes a request. The evaluation results in a behavior describing the further handling of the request, for example to block or complete the request. The user could also be prompted to give consent before further actions are taken. The same mechanisms apply to actions that are not user-initiated, for instance if another user requests information about a member in the same collaboration instance.

Evaluation is done by parsing and comparing XML-format policy and preference files, as described in the next section.

### 5.5.2 P3P and APPEL

The privacy architecture supports the use of the P3P and APPEL standards to specify policies and preferences. The functionality and syntax of these languages were presented in chapter 3. Originally developed by W3C for use on web sites, the standards have been adapted here to fit a service based platform. Figure 5.6 shows a flowchart demonstration of the evaluation process, based on a presentation by the Joint Research Center [7].

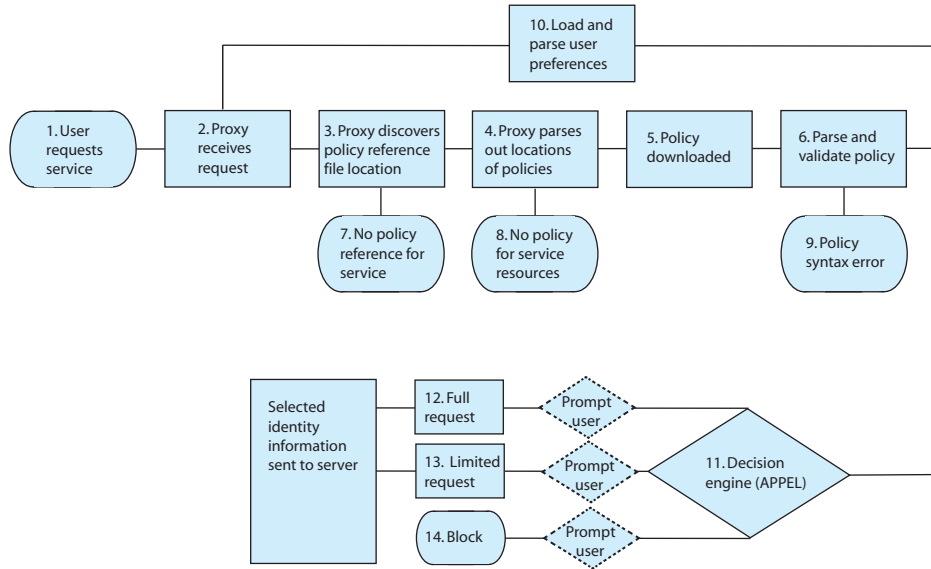


Figure 5.6: P3P evaluation flowchart

The process begins by a service request by the user (1). This request is handled by the proxy (2) which will try to locate the reference-file for the service's P3P-policy (3). The reference file points to the relevant policy on the service (4) which is then collected by the proxy (5). The proxy parses the policy and checks the validity of syntax etc (6).

The user-preferences are then loaded and parsed from the database (10), and evaluated against the policy in the APPEL decision engine (11). The evaluation loops through and compares statements in the two parsed documents, and decides on a behavior for handling of the request. If specified in the preference rule, the user may be prompted and notified of the outcome of this evaluation. The user might also be required to express his consent, based on a description of the current situation.

The behavior can result in one of the following actions. A full request (12) where the request is completed and all necessary user data transmitted to the service. A limited request (13) where some of the user data is transmitted, according to the response given by the user. Or a block (14) where further action is suspended.

It is up to the developers of applications to implement mechanisms to handle the outcome of this evaluation. The P3P and APPEL standards are fully supported in the design, and can be adapted according to the requirements of each service. The prototype (chapter 6) presents a sample set of APPEL and P3P documents that demonstrates the actual use of such policy/preferences sets. Please refer to the related work section 3 for a comprehensive walkthrough of the specifications and use of the standards.

### **P3P references in the services**

In order to apply the P3P policies on the platform the policies need to be referenced by the services so that the collaboration server is capable of providing them to the proxy when needed. The P3P specifications [10] requires that the policies of a service is referenced in one of the following ways:

1. At a well-known location
2. Referenced through an HTML <link>tag
3. Referenced through HTTP headers

The task force working on extensions to P3P beyond HTTP [19], offers an additional way of referencing the policy of a service. The task force suggests that it should be possible to reference the policies through the use of the **WSDL (Web Service Description Language)**, which is normally used to give exact descriptions of the web-service. The extension to the existing WSDL file needs to include an element named `privacy`, that can reference the policy. This is the most suitable way of referencing policies, but since not all web-services use WSDL this is not a reliable way to reference the policies. In UbiCollab, only the privacy proxy and the collaboration server supports WSDL, which makes the use of WSDL unsuitable for the services offered by the platform. It is possible to allow the services that use WSDL to reference through these files, but that would make the solution inconsistent.

The task force suggests using either method number (1) or (3) for referencing the services policies. The solution we have chosen for UbiCollab is to use a well-known location that is accessed by the collaboration server each time the proxy requests one of the services. This is the most suitable solution for UbiCollab, since it's the collaboration server that accesses the services and not the proxy. It's not necessary to reference the policy descriptions over HTTP, since the services are all a part of the platform and their actions are controlled by the collaboration server.

### **5.5.3 Reputation**

Reputation is a recognized mechanism for building trust between the users in computer systems. Reputation management is most often used in situations where the users are unknown to each other in order to provide some information about the people you don't know. In UbiCollab, the reputation system will be based on feedback from the users. The users can post feedback based on a collaboration with someone, or other actions that the user might have performed on the platform.

The idea is based on the solution used on eBay, where the users are encouraged to post a feedback notice after having made a purchase with a seller on the system. Some systems use trust to weight the opinions differently according to how the trust relationship is between the users. In UbiCollab the opinion of all users will be treated equally, to better reflect the users behaviour in collaboration with the other users.

The UbiCollab reputation system will be administered by a designated service, that can keep track of the users, the feedback given and in what areas they have been working in. The idea of reputation management is new to UbiCollab and the previous platform functionality was not developed to support such a system. With the redesign of the platform with support for multiple identities, the basics for developing a reputation system are in place. Figure 5.7 shows the UbiCollab platform with the **Reputation service** added.

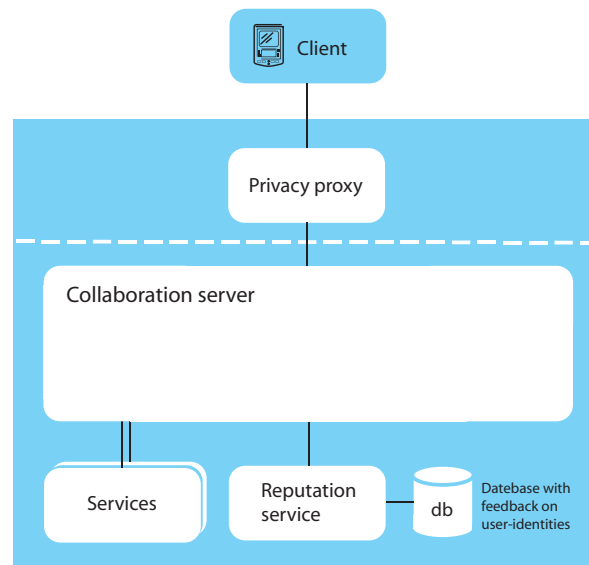


Figure 5.7: UbiCollab platform with reputation support

The reputation service will receive feedback from the users after a completed collaboration. This information will be stored connected to the identity the user is currently using on UbiCollab. This implies that the users can build a reputation in connection with each of their identities in the system.

The figure shows that the reputation service is controlled by the collaboration server, which also controls the basic services of UbiCollab. All services added to UbiCollab should be developed along the same lines as the existing services to keep consistency and avoid creating new methods for connecting to the services in the collaboration server. The reputation service will be a UPnP enabled service, that is accessible through the same mechanisms as used for the other services. The reputation service has access to a database containing the feedback registered in the system. Further functionality will be to present the feedback so that future applications can present the information about the users in sensible way. It will also contain methods for storing new feedback and general administration of the information.

The reputation service will support the requirements that were presented in section 2.4.4. The



following list presents the different demands to a reputation system and how this has been dealt with in UbiCollab:

- Long-lived identities - The identities in UbiCollab are long-lived and connected to the previous collaboration performed by the users. The identities are strongly connected to the user, since a lot of the collaboration is face-to-face.
- Feedback will be available for future inspection by others through the reputation service. The service will provide the wanted methods for developers in order to allow applications to present the reputations.
- The last requirement presented had demands to how much attention the users will pay to the reputation service. This is an issue that is difficult to predict, since the platform has not been properly tested. If the developers that build applications for UbiCollab makes this an important issue, the users will have to pay attention to it. It is difficult to force the reputation system on the users when developing the platform side of the solution.

The reputation service will need further research into what specific functionality it should provide for the applications, in order to present a complete solution. The design presented here only covers the basic ideas of a reputation system and presents a solution to the most important requirements stated in related research of the subject. Reputation services are considered to be important mechanisms for future computer systems and e-trade [36], and should be an important feature of further research on the UbiCollab platform.

#### 5.5.4 Highly sensitive information

A trusted 3<sup>rd</sup> party service (**TTP**) is used to handle highly sensitive information in UbiCollab. TTPs are defined as *"A TTP is an impartial organisation delivering business confidence, through commercial and technical security features, to an electronic transaction. It supplies technically and legally reliable means of carrying out, facilitating, producing independent evidence about and/or arbitrating on an electronic transaction. Its services are provided and underwritten by technical, legal, financial and/or structural means"* [27]. The use of TPPs are most typically connected to money exchange and purchases on computer systems. UbiCollab does not have any functionality that provides these kinds of services, but the development of a 3<sup>rd</sup> party service has been performed to support future extensions to the services of the platform.

The trusted 3<sup>rd</sup> party service integrated with UbiCollab is developed to handle the information that is too sensitive to be handled by the common mechanisms used in UbiCollab. This information is typically credit card numbers etc. The 3<sup>rd</sup> party service will require some encryption to conceal the information that it distributes. The security measures concerning the use of encryption and keys to allow access to the service are out of the scope of this thesis. Our service will communicate with the privacy proxy, that sends instructions of what the service is supposed to do. The trusted 3<sup>rd</sup> party will then already have been recognized by the user, who has allowed the service to perform the payment. Further functionality and other issues concerning security has not been discussed in this thesis. The service included in this thesis merely operates on command from the proxy, which has the authority to act on behalf of the user if this is stated in the preferences.

The trusted  $3^{rd}$  party will be a stand-alone web-service, that is trusted by both the system and the users. The service will ensure that the transaction is performed and perform the banking actions necessary to charge the user's account. There exists trusted  $3^{rd}$  party paying services for the web, but the service for UbiCollab will need further functionality. Future contributions to the platform should consider further research into this area of the platform, since our contribution only deals with the trusted  $3^{rd}$  party as a black-box that provides us with the wanted functionality for preserving privacy.

Figure 5.8 shows how the service is connected to the rest of the platform. The subsequent section will explain the functionality of the UbiCollab trusted  $3^{rd}$  party service and its interactions with the rest of the platform.

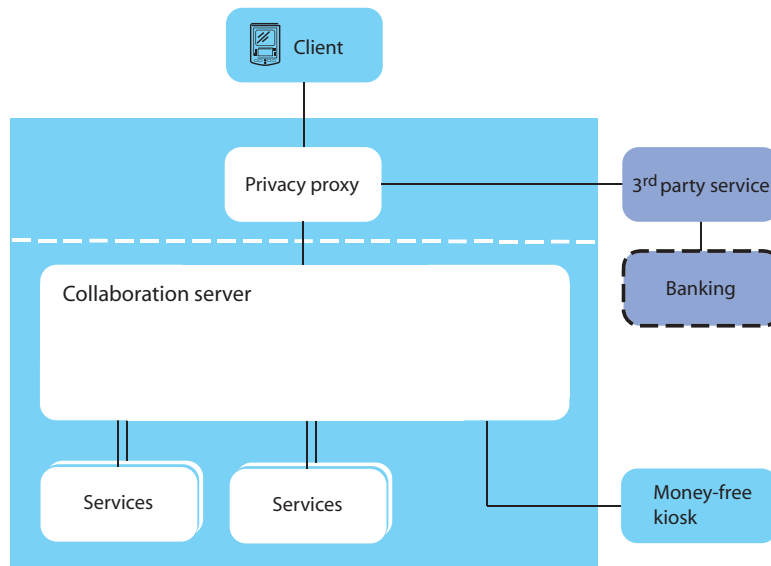


Figure 5.8: Interaction and functionality in UbiCollab trusted  $3^{rd}$  party service

The figure gives an overview of the platform with focus on the  $3^{rd}$  party service and the reasons for adding such a component. The *"money-free kiosk"* and the *"banking"* service are entities that are not part of the platform, but they have been added to demonstrate the need for handling highly sensitive data. The need for a  $3^{rd}$  party service was demonstrated in the auxiliary scenario number 5 where a user wants to take advantage of a money-free kiosk in UbiCollab. The kiosk might be location-aware and automatically ask the user if she wants to purchase something when she enters the proximity of the kiosk. It should also be possible for the user to activate the kiosk herself. The question of how to activate the kiosk is not an issue in this thesis, since the kiosk service is only integrated on a conceptual level to demonstrate the use of the  $3^{rd}$  party service.

If the kiosk initiates the purchase, the request is sent to the collaboration server and further to the privacy proxy, where the user-preferences are checked to see if the user has registered to accept these kinds of services. If so, the user is prompted and asked to describe what she wants to purchase. For this purpose an additional client needs to be developed that represents the kiosk and the available items. The user then sends the request with the desired items to the kiosk, and the kiosk checks for availability. After the purchase has been agreed upon, the kiosk needs to be paid before allowing the

user the items she has purchased. This is when the 3<sup>rd</sup> party service is inquired. The proxy receives a request from the kiosk with the amount the user must pay. To allow the transaction the privacy proxy must contact the 3<sup>rd</sup> party service, which handles the transaction with the user's bank. The user is again prompted for allowing the transaction to be performed. This last prompting of the user could include an interface provided by the bank, which requests a personal code from the user to enhance the security. After the money has been paid, the user receives her merchandise.

To make extensive use of the trusted 3<sup>rd</sup> party service, UbiCollab needs to develop more services that is based on this type of functionality. The design presented only supports the basic use of such a service. Future contributions to the platform might need an improved version of the service along with more support on the platform level. Due to a lot of issues that needs support on the client side, further research into this area of the solution has not been performed. The version presented here is enough to present a solution to the issue of highly sensitive information and to preserve the privacy of the users in connection with the transaction of such information on the UbiCollab platform.

## 5.6 UbiCollab interactions

To show how the design supports the conceptual requirements and preserves the privacy of the users, sequence diagrams have been developed to show the most common interactions on the UbiCollab platform and how the different components interact to perform the users requests. The sequence diagrams are based on the privacy architecture and will demonstrate the most important contributions to the UbiCollab platform added by this thesis.

### 5.6.1 UbiCollab login

Figure 5.9 shows the login sequence on the UbiCollab platform.

When a user logs in with a UbiCollab client, a username and password must be sent to the privacy proxy. As shown in the sequence diagram, the privacy proxy will handle the authentication and create a pseudonym for the user. The pseudonym is stored along with the username in order to keep the connection between the user and the identity he will use in the core of UbiCollab. Since the platform core doesn't know this connection, the user's privacy is protected inside the platform. The connection is stored in the proxy. When system requests further information about the user, the proxy will then know which user to fetch information about.

The user is then logged into the core of UbiCollab with the new pseudonym. When the user is registered with the Collaboration server, a sessionkey is created to be used for internal authentication. This key is used for authentication of the user in later requests to the platform during the session. After the sessionkey is returned to the client, the user has been authenticated and registered on the platform and can perform the actions he wants.

The logout sequence is a reversed version of the login sequence demonstrated in figure 5.9. When the user logs out the sessionkey is deleted and the pseudonym is removed from the list of active users.

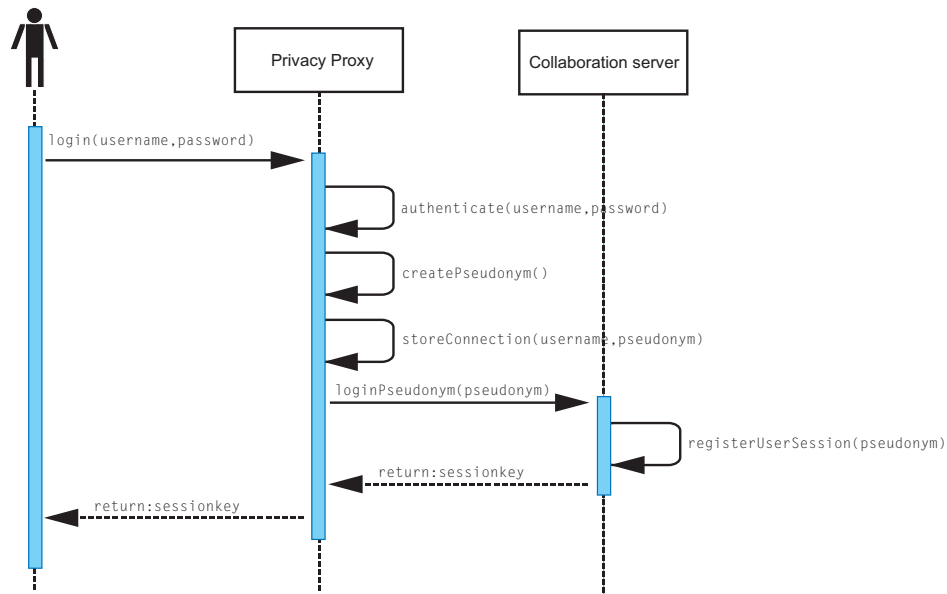


Figure 5.9: Login sequence on the UbiCollab platform

## 5.6.2 User-initiated request

Figure 5.10 shows the interactions on the platform when the user requests a service that needs further information about the user in order to perform the chosen action.

The user initiates a request by sending his useridentity to the privacy proxy. The privacy proxy then sends a request to the collaboration server asking for the privacy policy of the service in question. The collaboration server controls all the services integrated with the platform and locates the correct service and requests the policy from the correct service. The policy is returned from the service; first to the collaboration server and then to the privacy proxy.

The privacy proxy will then retrieve the user's privacy preferences in order to evaluate the policy. The privacy proxy will evaluate the P3P policies with the APPEL preferences through an evaluation engine integrated in the solution. The evaluation either results in a block or a request-behavior, which decides what will happen next.

If the policy is not in accordance with the preference of the user, the block behavior is returned to the user and the request is cancelled. In the case of a request-behavior, the user information is gathered and sent to the service. The service then performs the requested actions. This part of the request might lead to additional prompting of the user, where the service asks for additional information or needs permission to perform certain actions.

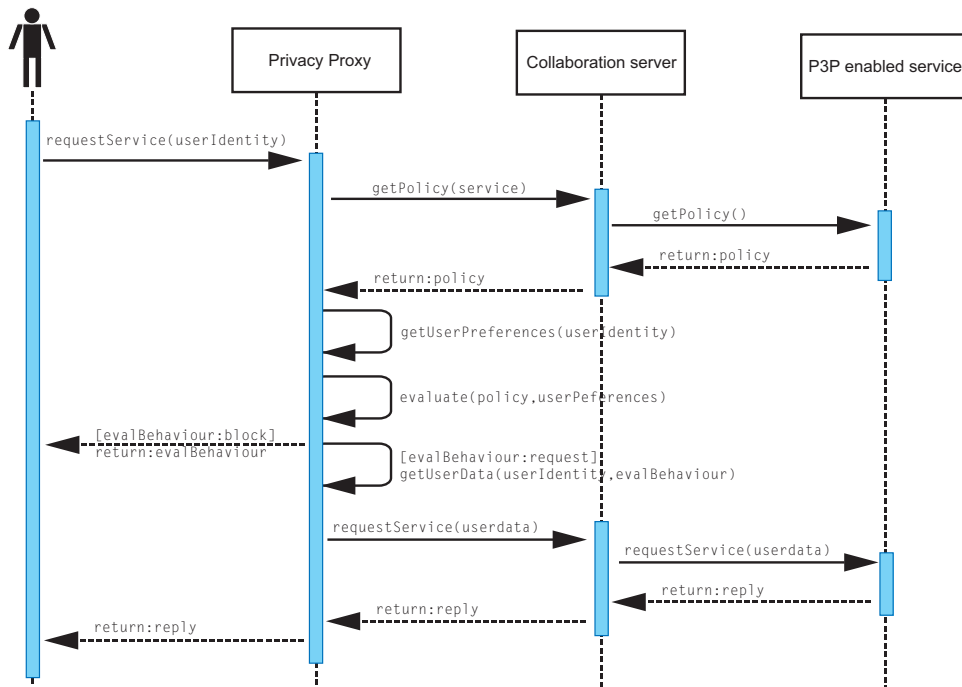


Figure 5.10: User-initiated request of a UbiCollab service

### 5.6.3 Service initiated request

Figure 5.11 demonstrates the interactions when a context-aware service initiates actions that requires information about a user.

When the service itself initiates a request for gathering personal user information, the service's policy is sent along with the request to the collaboration server. The service also provides the collaboration server with the pseudonym of the user it wants information about. Given a context-aware service that offers it's services to users in the proximity, the service will only manage to find the pseudonym of the user without communicating with the collaboration server.

The collaboration server can not access the the user information connected to the pseudonyms and will have to ask the privacy proxy to provide this information. The pseudonym and policy are therefore sent to the privacy proxy, where the real identity of the user and the connection to the different pseudonyms are stored.

First the privacy proxy identifies the user connected to the pseudonym. The privacy proxy fetches the preferences of this user and evaluates the preferences against the policy of the service. The evaluation either results in a block or a request-behavior as in the user-initiated request. Upon a block request, the behavior object is sent back to the service, and no further action is taken.

In the case of a request-behavior, the user is prompted and asked to give consent to the collection

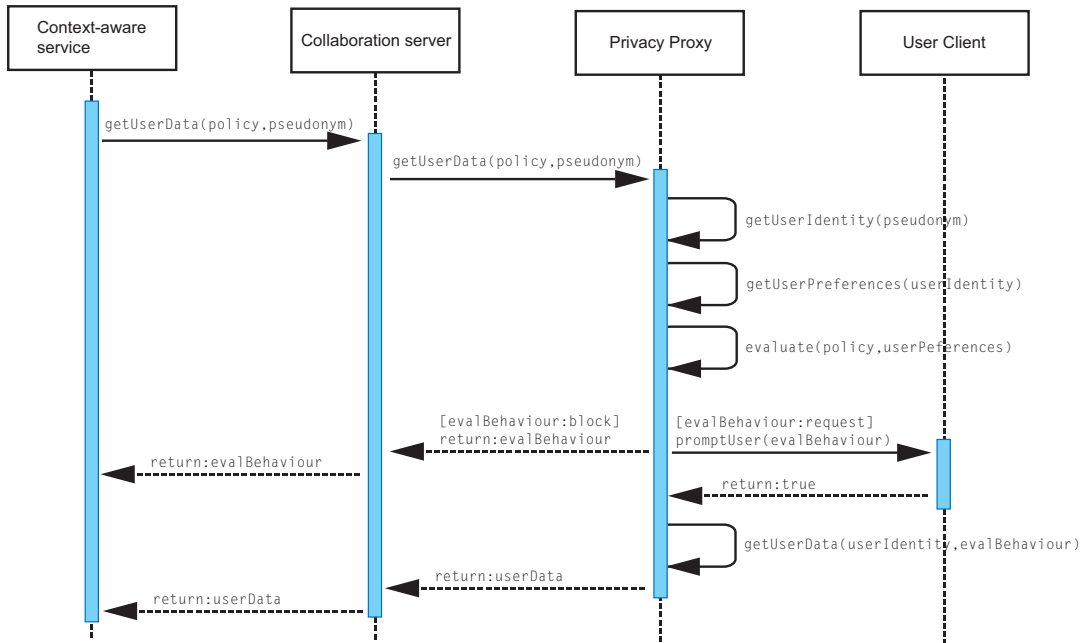


Figure 5.11: Service-initiated request in UbiCollab

of information. If the user agrees to this, the privacy proxy gathers the information requested from the service and returns it to the collaboration service. The collaboration service then provides the service with the information it first requested.

## Chapter 6

# Prototype - UbiCollab privacy extension

A prototype has been implemented as a "proof of concept" of the design and technology that is proposed in this thesis. The implementation builds on an existing prototype of the UbiCollab platform, developed by Schwarz in 2004 [37]. The realization of the privacy design through a prototype is one of the main contributions by this thesis. This chapter presents the platform structure and the new components that have been added, and explains how the main concepts have been realized in the platform prototype.

### 6.1 Scope of the prototype

The initial challenge will be to adapt and rewrite parts of the previous platform prototype to suit the new privacy design. The complexity and size of the platform makes this a time-consuming and difficult task. Because of this, the demonstration of the extended functionality suggested in this thesis will be somewhat limited. It is out of the scope of this project to implement a complete solution to privacy at prototype level. Rather, we will illustrate how the concepts presented in the privacy design can be incorporated in the platform. In order to demonstrate the more refined privacy-mechanisms described in the design, implementation at application level will be necessary, which is out of the scope of this project. The work on the platform prototype has been limited to the following tasks:

- Implement a privacy proxy that handles user data and allows anonymous use of services (6.4)
- Implement user management with multiple identities (6.5)
- Implement support for APPEL rulesets to express user preferences (6.6)
- Implement support for P3P privacy policies in services (6.7)

- Implement mechanisms to evaluate service policies with user preferences (6.8)

The resulting platform prototype and the various components are described in detail below, starting with an overview of the platform and description of the technology that is used. The parts of the design that are not handled in this prototype are discussed briefly in section 6.9.

## 6.2 Platform overview

The overall structure of the implemented prototype is fairly close to the architecture described in the design. Figure 6.1 shows the layout and relation between the different components.

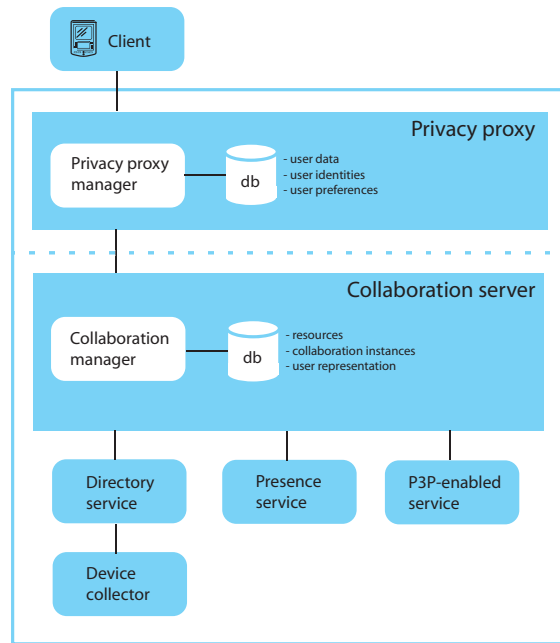


Figure 6.1: Platform prototype overview

The *privacy proxy* is implemented on top of the *UbiCollab core platform*, and handles user authentication and interaction with the core platform and services. The proxy stores the connection between the user’s real identity and his current representation in UbiCollab. Any personal data the user might register will be stored here. The *collaboration server* registers a session key when a user logs in, which is returned to the client and used to maintain the session. The collaboration server’s main responsibility is to handle and provide an API towards the different UbiCollab services. Also, the collaboration instances and connected resources are handled by the collaboration server.



## 6.3 Technology

The privacy proxy and the collaboration server are implemented as two individual web-services, and all communication between them are done over HTTP. In this way, they could be separated physically by setting them up on different servers. The prototype implements the **Apache Web Services Project - Axis** [2] for communication over SOAP (Simple Object Access Protocol), a lightweight, XML-based protocol for exchange of information in a decentralized, distributed environment. The protocol is used for describing the content and processing of messages by remote procedure calls and responses. It also allows expressing instances of application-defined datatypes within messages.

We have set up an **Apache Jakarta Tomcat** [39] standalone HTTP server, to host the Axis web-services. Communication between the collaboration server and the UbiCollab UPnP-services also relies on the Tomcat server, where all the services are assigned a designated portnumber. **UPnP** (Universal Plug and Play) [17] is a protocol specification that enables discovery and control of networked devices and services, such as network-attached printers, Internet gateways, and consumer electronics equipment. In UbiCollab, UPnP devices and services are automatically recognized and gathered by the *device collector*, and made available to the system in the *directory service*.

The privacy proxy and collaboration service each have their own database where they store information about users, resources and collaborations etc. The databases run on a **MySQL Database Server** [43]. All the prototype components are implemented on the **Java 2 Platform** [31], and built using **Apache Ant** [34], which is an XML-based software tool for automating software build processes.

## 6.4 Privacy proxy

The privacy proxy plays a major part in the new prototype, this is where all user management takes place, and all requests to the collaboration server goes through this. Figure 6.2 shows the layout of the privacy proxy.

The proxy is set up as a standalone web-service, and the platform API is defined in the **PrivacyProxyWebService**-class. The methods that involve authentication, login/logout, user-creation etc. are located in the **PrivacyProxyManager**-class. **PrivacyProxyManagerUtils** is a support-class that contains the functionality for e.g. reading and updating the proxy's database and parsing xml-documents.

The **RequestHandler**-class handles all the calls that are not internal to the proxy. This includes generic forwarding-mechanisms for the UbiCollab core platform methods, and dealing with requests to P3P-enabled services. The **RequestHandler** will request and parse the policy of the service, and fetch the user's privacy preferences. The evaluation is done in the **AppelEvaluator**, which returns a *behaviour* object that describes the actions that should be taken. The proxy can act as a *user agent* and fetch the user data that is necessary in order to complete the request.

The prototype supports the basic P3P user agent mechanisms, but user prompting is not fully implemented. In order to handle user-response to prompts, some mechanisms are necessary for

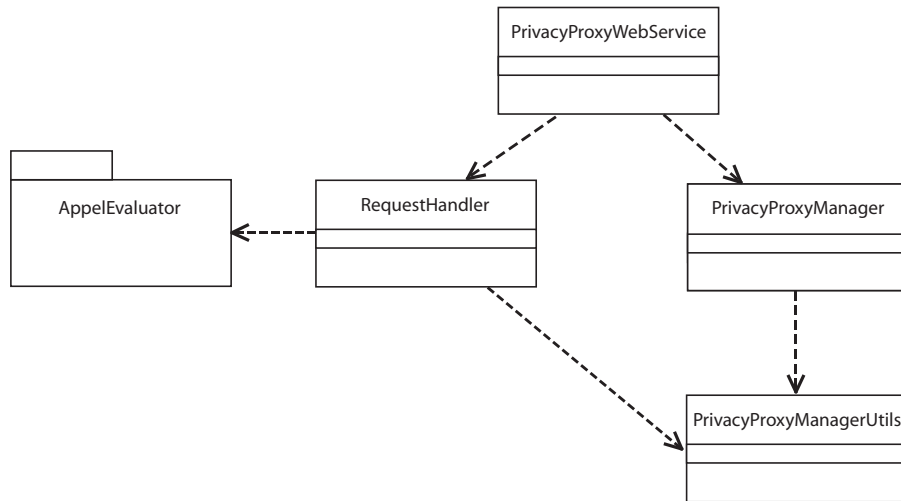


Figure 6.2: Privacy proxy overview

maintaining a request-handling over multiple response and replies between the client and proxy. This is possible to implement with the current design, but has been left out to be able to demonstrate a wider range of functionality in this project.

## 6.5 User management

The prototype has support for creating multiple identities for each user. Figure 6.3 shows the tables that concern the user's data and his different identities in the prototype.

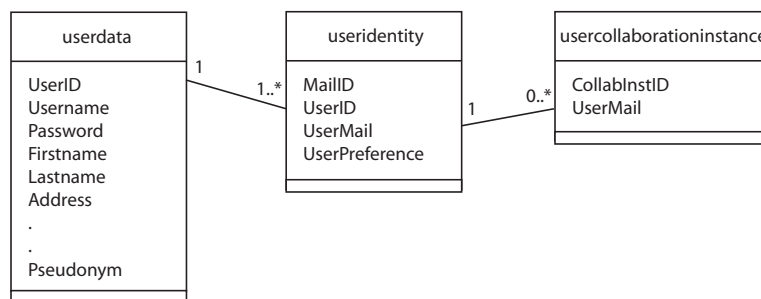


Figure 6.3: Proxy user databases

A user logs on to UbiCollab through the proxy by supplying a username and password, which is checked against the *userdata*-table. This table stores all personal information about the user, including name, address, contact information, etc. A one-time random pseudonym is generated and used for representing the user in the collaboration server. This pseudonym is temporarily stored with the users profile in the proxy. As in previous versions, the collaboration server supplies a session

key to maintain a user session with the client.

The user may register several identities to his user profile. All identities have their own privacy preference set, which will act as a "filter" for the data stored in his general profile. In this simplified model, a user's identity is connected to his e-mail address. So the user may register multiple e-mail addresses (and thereby multiple privacy settings) and thus be able to switch between identities depending on the task or situation. The user is connected to a collaboration instance through this identity (e-mail). But because the login is at person-level rather than identity-level, he can maintain several identities simultaneously, and in this manner appear "online" as several identities at the same time.



Figure 6.4: Collaboration server user database

The collaboration server has its own user database (fig 6.4), which acts as a directory of users that can be added to collaboration instances. A non-mandatory registration can be done here to be visible to other UbiCollab users. The relation to the user's identity is given by the e-mail address. In this way, users can post several of their identities in this directory. The user database in UbiCollab contains limited information about the user. Depending on the purpose of registration this could be restricted to e.g. a nickname or the user's name, but could also include a full set of user contact information. However, the functionality implemented in the prototype to update this database is somewhat limited at present, and includes only the e-mail and name of the user. This could easily be extended in future versions.

## 6.6 User APPEL rulesets

The user preferences mentioned in the previous sections are implemented in the form of APPEL rulesets connected to the users identities. The XML-format preference files each contain a set of rules that the actions of the user is evaluated against. When the user requests a service, the P3P policy of this service is fetched, and each statement is checked to comply with a corresponding rule. This process is described more in detail in section 6.8 "Policy and preference evaluation".

Although the use of preference files is supported in the prototype, the actual creation of such rulesets is not implemented. As a next step, the user should be able to create their own preferences or import standard rulesets suggested by UbiCollab or the services. Ruleset creation is considered to be performed at application-level, and outside the scope of this project. The APPEL project [9]

suggests how this can be implemented on a user agent.

## 6.7 Service P3P policies

A demonstration P3P-enabled UPnP-service has been implemented in the prototype. The service has an XML-format P3P-policy file that can be fetched via a `getPolicy()`-method in the service API. This is a simplified way of distributing the policy. The P3P standard has suggested the use of reference files at a well-known location to point to policies. This formal procedure has not been prioritized in the current prototype implementation.

The service policy contains privacy statements about what information will be gathered about the user and how it is handled etc. It is based on a standard P3P-policy example distributed by the Joint Research center [7], who are presented in more detail in the next section. The Service allows the user to invoke certain test-purpose methods if the policy is accepted by the user agent (privacy proxy). Information about the user can then be collected from the users personal profile.

The service is intended as a proof of concept, and a guideline for P3P-enabling of services in UbiCollab. A complete framework for creation of such P3P-policies is out of the scope of this project.

## 6.8 Policy and preference evaluation

The policy and preference evaluation engine is a key component in the proxy prototype. The heart of this engine is the `AppelEvaluator` (fig. 6.5), a free distribution of a P3P-policy and APPEL-ruleset evaluator. The `AppelEvaluator` has been integrated in the proxy, and is accessed from the `RequestHandler`. The evaluator takes an APPEL ruleset XML-document and a P3P policy XML-document as arguments, and performs a comparison in the `RuleLooper` of all the included statements. An object of type `EvalReturn` is built from the results of these comparisons, and returned to the `RequestHandler` upon completion. The `EvalReturn`-object contains information about the *behaviour* that should result from the evaluation, any *prompt messages* that should be displayed to the user, and other information about the APPEL-rule that has caused the current behaviour.

## 6.9 Prototype limitations

As a conclusion to this chapter, we present a summary of the issues that are left unhandled in the prototype. These are parts of the design that are either regarded as too comprehensive to implement at this point, or not vital in order to demonstrate the basics of the design.

- The integration of a 3<sup>rd</sup> party service to handle sensitive user data has been left out. Some research is still left in this area, in order to find a solution that suits UbiCollab. An interface

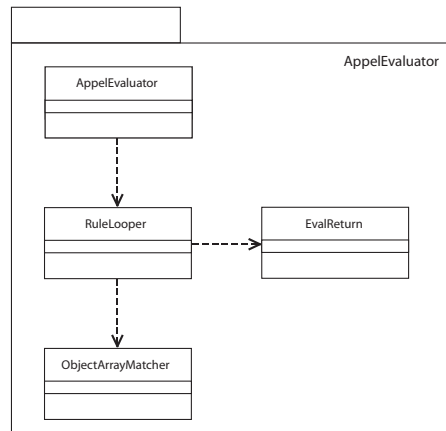


Figure 6.5: AppelEvaluator overview

towards such a service must also be implemented.

- The reputation service has not been implemented in this prototype. This is possible to implement as a stand-alone service, and should be easy to integrate with the platform prototype in the same manner as the other UPnP services.
- The prototype has no functionality for defining APPEL preference rulesets or P3P privacy policies. This should be supported at platform level, but is mainly a client/application GUI-issue. With some research, an existing solution may be found that is integratable with UbiCollab.
- There is no demonstration of service-initiated requests for user-data. This should be possible to implement with the existing mechanisms of policy negotiation, but has been left out due to time constraints.

## Chapter 7

# Demonstration

This chapter will demonstrate the parts of the architecture that were implemented in the prototype. This includes a walkthrough of the most common interactions with the platform, and a detailed description of the internal working of the platform prototype. Our contribution to the platform prototype is demonstrated at a lower level than what is depicted in the scenarios. We describe the basic platform mechanisms, and how they are implemented to support user privacy.

The UbiClient [11] has been modified to support basic requests via the proxy and is demonstrated in the first section.

The interactions are demonstrated with a server test-class, sending requests to the platform over HTTP in the same way as e.g. a PDA- or Web-client. The server test-class performs a sequential runthrough and simulation of functionality, including methods for user creation and login/logout, collaboration instance functionality and negotiation of P3P policies. This should give developers a clear idea of how these mechanisms can be supported in UbiCollab clients.

### 7.1 UbiClient

The UbiClient [11] was created by Pedro Gonçalves in Spring 2004. It has been rewritten during this project to suit the new API to demonstrate that the changes performed in this project are consistent with the previous contributions. The client works with the basic functionality of UbiCollab like logging in and setting up meetings.

Figure 7.1 shows the user interface of the client when working with a collaboration instance that represents a specific meeting. The client lists the different entities connected to the meeting and allows the user to perform the configuration he wants on the collaboration instance. This includes adding and removing users and devices connected to the meeting.

The UbiClient has not been an issue of importance in our prototype, and the functionality is the



Figure 7.1: UbiClient user interface

same as before. The only change made to the client is how it accesses the platform; This was previously done by using the *Collaboration server web-service*, but is now done through the *Privacy proxy web-service*.

The UbiClient's functionality and GUI are insufficient to show the range of mechanisms that is supported in the new platform prototype and the client is therefore just added to demonstrate consistency with the previous contributions.

## 7.2 User management

The prototype has all the functionality required for basic user management. The API includes methods for creating new users, logging in and out of the platform, searching for users and adding users to collaboration instances. This corresponds externally to the methods found in previous versions of the prototype. The demonstration will show how the internal structure has been modified in order to preserve the existing functionality and support the new mechanisms of privacy and identity management.

As described in the prototype chapter, the entry point to UbiCollab has been shifted to the new privacy proxy Web-service, and all method invocations are done through this. The proxy itself handles the typical user- and identity-methods, some of which are demonstrated in this section. Other calls are forwarded directly to the collaboration server. The demonstration shows how user data is handled in the different databases, and the communication between the different parts of the

platform.

### 7.2.1 Authenticate and login user

The user-login is done in two steps. First, the user is authenticated in the proxy by supplying a username and password. The proxy generates a unique pseudonym that the user will be known by in the collaboration server for the duration of the session. Second, the proxy performs a login to the collaboration server with the new pseudonym. The collaboration server registers a new session for the anonymous user, and returns a sessionkey. The procedure is shown below.

The client invokes the method `userLogin(username, password)` with the following result:

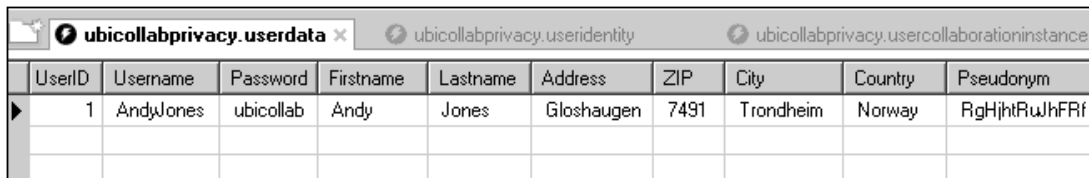
```
>> Invoking login...
<< Sent. Got [ok] [837098273904873214]
```

The method returns status confirmation that the user is authenticated and logged in to the collaboration server. A sessionkey string has been generated and associated with the users pseudonym in the collaboration server. The sessionkey is used in the subsequent interaction, in order to check the users authenticity and maintain the session.

The internal processes and interactions between the PrivacyProxy [PP] and CollaborationServer [CS] are visible in the server log:

```
[10:25:20] [PP] About to login user [AndyJones]
[PP] Authentication [true]
>>> Start invoke cs:login
[CS] About to register session with user [RgHjhtRuJhFRf]
[CS] Generated sessionKey [837098273904873214]
<<< End invoke cs:login
[PP] Return [ok] [837098273904873214]
```

The proxy and the collaboration server have separate databases. The user has registered a username and password in the proxy database table `ubicollabprivacy.userdata`, which he is authenticated against upon login. The table also contains personal information about the user. After the user is logged in, the table is updated with his current pseudonym. The entry is shown in figure 7.2.



UserID	Username	Password	Firstname	Lastname	Address	ZIP	City	Country	Pseudonym
1	AndyJones	ubicollab	Andy	Jones	Gloshaugen	7491	Trondheim	Norway	RgHjhtRuJhFRf

Figure 7.2: Table `ubicollabprivacy.userdata`



## 7.2.2 Create a new user

The proxy also handles the creation of new users. The process involves several steps, as depicted in figure 7.3.

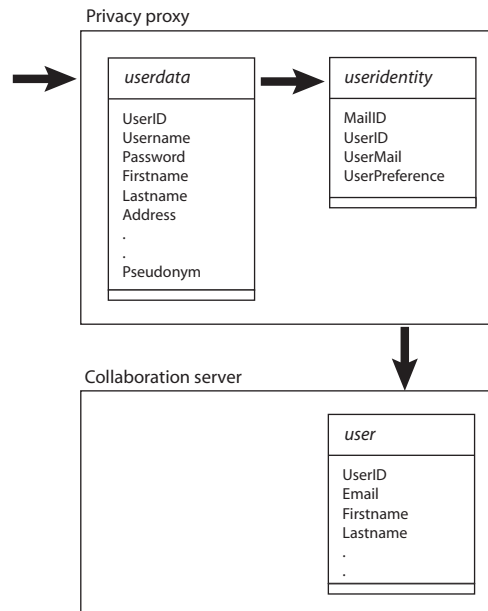


Figure 7.3: Create new user

- First, a new entry is created in `ubicollabprivacy.userdata` (fig. 7.4). This table contains all the information that is known about a user. This includes authentication details like username and password, and personal data such as name, phone-numbers, home-address, etc.
- Second, a new *useridentity* is created in `ubicollabprivacy.useridentity` (fig. 7.5), where the user's e-mail is stored, possibly along with a set of personal information that is unique to the identity. The user can have multiple such identities.
- Finally, the new identity is posted in the collaboration server's table `ubicollab.user` (fig. 7.6). This allows the user to be added to collaboration instances in UbiCollab.

The client invokes the method `createUser(sessionKey, userEmail, password, firstName, lastName)` with the following result:

```
>> Testclass invoking createUser...
<< Sent. Got [JohnDoe]
```

An extract of the server log shows the internal server process:

```
[10:25:21] [PP] About to create userprofile for [Doe, John]
```

```

[PP] Successfully created userprofile [JohnDoe]
[PP] Successfully created useridentity [johndoe@example.com]
>>> Start invoke cs:createUser
[10:25:21][CS] About to create entry for [Doe, John]
[CS] Successfully created entry [johndoe@example.com]
[CS] Return [ok]
<<< End invoke cs:createUser
[PP] Return [JohnDoe]

```

The proxy reports a successful entry of a new user, and has generated a unique username. For the sake of this demonstration, the `createUser()` method is doing all the different updates in one operation. Other implementations might handle these three steps separately, and also support a wider range of user data to be stored in the different profiles.

UserID	Username	Password	Firstname	Lastname
1	AndyJones	ubicollab	Andy	Jones
2	JohnDoe	ubicollab	John	Doe

Figure 7.4: Table `ubicollabprivacy.userdata`

MailID	UserMail	UserID	UserPreference
1	andy@strict.example.com	1	C:/UbiCollab/Development/Eclipse ...
2	andy@loose.example.com	1	C:/UbiCollab/Development/Eclipse ...
3	johndoe@example.com	2	

Figure 7.5: Table `ubicollabprivacy.useridentity`

UserID	Email	Firstname	Lastname
1	johndoe@example.com	John	Doe

Figure 7.6: Table `ubicollab.user`

### 7.2.3 Logout

The logout closes the user-session and removes any temporary data that is registered about the user. The session-key is invalidated, and no further interaction with the platform is possible. The client invokes the method `userLogout(sessionKey, username)` with the following result:

```
>> Testclass invoking userLogout...
<< Sent. Got [ok]
```

The details of the operation are visible in the server log:

```
[10:25:25] [PP] About to logout user [AndyJones]
[PP] Found pseudonym [RgHjhtRuJhFRf] for user [AndyJones]
>>> Start invoke cs:logout
[10:25:25] [CS] About to logout user [RgHjhtRuJhFRf]
>>> Start invoke UPnP-Action:removeResourceFromUser
<<< End invoke UPnP-Action:removeResourceFromUser
[CS] Invalidating sessionKey [837098273904873214]
[CS] Return [ok]
<<< End invoke cs:logout
[PP] Return [ok]
```

The proxy finds the pseudonym of the user and invokes a logout in the collaboration server. All the temporary data, including resources connected to the user's pseudonym, are removed from the collaboration server. The sessionkey is invalidated and the connection between user and pseudonym is deleted in the proxy. The user is supplied a fresh pseudonym the next time he logs in to the platform.

## 7.3 Collaboration

Collaboration in form of e.g. meetings and resource sharing is supported through the use of *collaboration instances*. Collaboration instances are objects that contain information about the following: The *details* of the collaboration, a list of *users* connected to the collaboration, and a list of connected *resources*. The management of collaboration instances is done in the collaboration server, and some of the functionality will be demonstrated in this section.

### 7.3.1 Create a collaboration instance

UbiCollab users can set up e.g. a new meeting by invoking the method `createCollabInst(sessionKey, collabInstName, time, place, notificationType)`. This gives the following result:

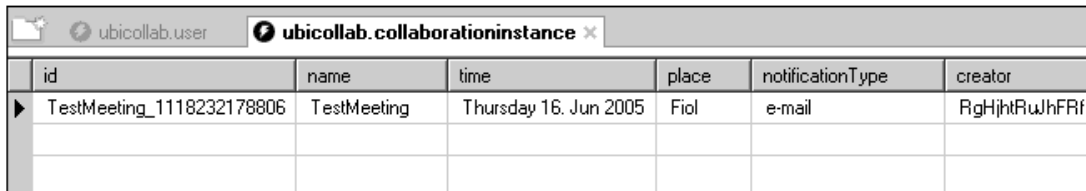
```
>> Testclass invoking createCollabInst...
<< Sent. Got [TestMeeting_1118232178806]
```

The client receives the id of the collaboration instance that has been created. An extract of the server log shows the internal server process:

```
[10:25:22] [CS] About to createCollabInst [TestMeeting]
```

```
[CS] Building new CollaborationInstance [TestMeeting_1118232178806]
[CS] Return [TestMeeting_1118232178806]
```

The collaboration server generates a unique identity and stores the instance in the table `ubicollab.collaborationinstance`, as shown in figure 7.7. The creator is anonymous, and represented by a pseudonym.



id	name	time	place	notificationType	creator
▶ TestMeeting_1118232178806	TestMeeting	Thursday 16. Jun 2005	Fiol	e-mail	RgHjhtRwHFRf

Figure 7.7: Table `ubicollab.collaborationinstance`

### 7.3.2 Adding people to a collaboration instance

The creator can now be added to the collaboration instance by invoking `addUserToCollabInst(userMail, username, collabInstID)`. The user can choose which of his identities he wants to use in the collaboration instance. The method also supports using a default identity for collaboration instances. Adding other people to a collaboration instance is done in a separate operation. The user specifies a list of e-mail addresses of the users to add, possibly by selecting from a list of registered UbiCollab users.

Invoking the method gives the following result:

```
>> Testclass invoking addUserToCollabInst...
<< Sent. Got [ok]
```

The client gets a confirmation that the user has been added to the collaboration instance. The following actions are performed by the server:

```
[10:25:22] [PP] About to add CollabInst [TestMeeting_1118232178806] to
_user [AndyJones]
[PP] No identity supplied. Using default identity [andy@strict.example.com]
[PP] Successfully added CollabInst [TestMeeting_1118232178806] to
_user [andy@strict.example.com]
[PP] Return [ok]
```

The identity `andy@strict.example.com` is registered with the collaboration instance as shown in figure 7.8.

This table in the proxy database holds the connection between all user-identities and collaboration instances. The methods for e.g. adding and removing people from collaboration instances have also been rewritten to fit the new design, but will not be demonstrated here. One of the key features is

CollabInstID	UserMail
TestMeeting_1118232178806	andy@strict.example.com

Figure 7.8: Table `ubicollabprivacy.usercollaborationinstance`

that the user can get information about all the collaborations that he is registered with, regardless of the identity they are connected to.

## 7.4 P3P-enabled Demonstration Service

The P3P-enabled demonstration service has been implemented to show how policies and preferences are evaluated in a user-initiated service request. It also shows how the proxy can act as a user-agent and authorize the release of personal data on behalf of the user. The service is invoked via the proxy, and then through the collaboration server, with the method `testP3PDemo(sessionKey, userEmail)`. The user indicates which of his identities he wants to use for the service call. The identity can also be decided from a default setting. Since the user can apply a different privacy setting to each identity, the result of the service invocation may vary. This example shows how a user tries to invoke the service with two different identities, with varying level of "strictness" on how personal data is allowed to be handled.

The service's P3P policy is shown in figure 7.10. The demonstration service allows users to make a purchase, and requires some personal information, including name and address details. The necessary data fields are visible in the policy on lines 29-35. According to the statement on line 42, the same information may also be used for sending "selected marketing solicitations" to the user.

The first invocation, using the identity `andy@strict.example.com` with a strict preference ruleset, gives the following result:

```
>> Testclass invoking testP3PDemo...
<< Sent. Got [Not executed. Blocked on user preference with the following
  _description: Personally identifiable info will be used beyond the stated
  purpose]
```

The proxy has evaluated the policy and decided to block the request. No further interaction with the service is done. The rule that has fired in the APPEL preference ruleset is shown in figure 7.9. The user has stated that he does not want information to be used beyond the purpose of the request.

The internal processes can be extracted from the server log:

```
[10:25:23] [PP] About to invoke P3P-enabled service [privacyDemoService]
>>> Start invoke cs:getServicePolicy
```

```

01 <appel:RULE behavior="block"
02   description="Personally identifiable info will be used beyond the stated purpose" prompt="no">
03   <p3p:POLICY>
04     <p3p:STATEMENT>
05       <p3p:DATA-GROUP appel:connective="and">
06         <p3p:DATA>
07           <p3p:CATEGORIES appel:connective="or">
08             <p3p:physical/>
09             <p3p:online/>
10           </p3p:CATEGORIES>
11         </p3p:DATA>
12       </p3p:DATA-GROUP>
13     <p3p:PURPOSE appel:connective="non-and">
14       <p3p:current/>
15     </p3p:PURPOSE>
16   </p3p:STATEMENT>
17 </p3p:POLICY>
18 </appel:RULE>

```

Figure 7.9: Sample APPEL preference rule for user `andy@strict.example.com`

```

>>> Start invoke UPnP-Action:getPolicy
<<< End invoke UPnP-Action:getPolicy
<<< End invoke cs:getServicePolicy
[PP] Successfully retrieved service P3P policy
[PP] Successfully retrieved preference ruleset for user
    _[andy@strict.example.com]
[PP] About to evaluate policy
[PP] Result [not ok]
[PP] Return [Not executed.  Blocked on user preference with the following
    _description:  Personally identifiable info will be used beyond the
    _stated purpose]

```

The service call is processed by the proxy's *RequestHandler*, which initiates a request to the collaboration server for the service's P3P policy. The collaboration server invokes a UPnP call to the service, which returns the policy. The user's APPEL preference ruleset is fetched by the proxy, and evaluated against the service's P3P policy. The result of the current evaluation is a *blocked* request. Further action is therefore suspended, and a description of the situation is returned to the client.

The user has a second identity, `andy@loose.example.com`, which has a ruleset that is less strict. He invokes the service with this identity, with the following result:

```

>> Testclass invoking testP3PDemo...
<< Sent.  Got [Welcome to privacyDemoService.  This is what we know about you:
    user.name.given:  Andy;
    user.name.family:  Jones;
    user.home-info.postal.street:  Glosaugen;
    user.home-info.postal.city:  Trondheim;
    user.home-info.postal.postalcode:  7491;
    user.home-info.postal.country:  Norway]

```

The demonstration service has received information about the user, and returns a welcome mes-

sage that demonstrates what it knows. The server log shows the following activity:

```
[10:25:24] [PP] About to invoke P3P-enabled service [privacyDemoService]
>>> Start invoke cs:getServicePolicy
>>> Start invoke UPnP-Action:getPolicy
<<< End invoke UPnP-Action:getPolicy
<<< End invoke cs:getServicePolicy
[PP] Successfully retrieved service P3P policy
[PP] Successfully retrieved preference ruleset for user
    _[andy@loose.example.com]
[PP] About to evaluate policy
[PP] Result [ok]
[PP] Fetching personal data for user [andy@loose.example.com]
>>> Start invoke cs:forwardServiceCall[privacyDemoService]
>>> Start invoke UPnP-Action:testP3PDemo
<<< End invoke UPnP-Action:testP3PDemo
<<< End invoke cs:forwardServiceCall[privacyDemoService]
[PP] Return [Welcome to privacyDemoService. This is what we know about you:
    user.name.given: Andy;
    user.name.family: Jones;
    user.home-info.postal.street: Glosaugen;
    user.home-info.postal.city: Trondheim;
    user.home-info.postal.postalcode: 7491;
    user.home-info.postal.country: Norway] ]
```

As with the previous example, the policy and preferences are fetched and evaluated. But this time it results in a *request*-behaviour, and the service is invoked. Also, the necessary data-entries are collected from the database and sent with the request. The data-definitions of the required information, e.g. `user.name.given`, correspond to the entries in the user's profile. The proxy, acting as a user agent, will authorize the release of this data. The demonstration service gets the required information about the user, and can perform the stated tasks. For the sake of demonstration, this information is returned to the user in this example.

```

01 <?xml version="1.0" encoding="UTF-8"?>
02 <POLICIES>
03 <POLICY discuri="http://www.UbiCollab.com/PrivacyDemo/PrivacyDem1.html"
04 name="policy1" opturi="http://www.UbiCollab.com/PrivacyDemo/preferences.html">
05 <ENTITY>
06 <DATA-GROUP>
07 <DATA ref="#business.name">UbiCollab PrivacyDemo Inc.</DATA>
08 <DATA ref="#business.contact-info.postal.city">Trondheim</DATA>
09 <DATA ref="#business.contact-info.postal.country">Norway</DATA>
10 <DATA ref="#business.contact-info.online.email">privacy.ubicollab@ntnu.no</DATA>
11 </DATA-GROUP>
12 </ENTITY>
13 <ACCESS><contact-and-other/></ACCESS>
14 <DISPUTES-GROUP>
15 <DISPUTES resolution-type="independent" service="http://www.PrivacySeal.example.org"
16 short-description="PrivacySeal.example.org">
17 <IMG alt="PrivacySeal's logo" src="http://www.PrivacySeal.example.org/Logo.gif"/>
18 <REMEDIES><correct/></REMEDIES>
19 </DISPUTES>
20 </DISPUTES-GROUP>
21 <STATEMENT>
22 <CONSEQUENCE>
23 We use this information when you make a purchase.
24 </CONSEQUENCE>
25 <PURPOSE><current/></PURPOSE>
26 <RECIPIENT><ours/></RECIPIENT>
27 <RETENTION><stated-purpose/></RETENTION>
28 <DATA-GROUP>
29 <DATA ref="#user.name.given"/>
30 <DATA ref="#user.name.family"/>
31 <DATA ref="#user.home-info.postal.street"/>
32 <DATA ref="#user.home-info.postal.city"/>
33 <DATA ref="#user.home-info.postal.postalcode"/>
34 <DATA ref="#user.home-info.postal.country"/>
35 <DATA ref="#dynamic.miscdata">
36 <CATEGORIES><purchase/></CATEGORIES>
37 </DATA>
38 </DATA-GROUP>
39 </STATEMENT>
40 <STATEMENT>
41 <CONSEQUENCE>
42 At your request, we will send you carefully selected marketing
43 solicitations that we think you will be interested in.
44 </CONSEQUENCE>
45 <PURPOSE>
46 <contact required="opt-in"/>
47 <individual-decision required="opt-in"/>
48 <tailoring required="opt-in"/>
49 </PURPOSE>
50 <RECIPIENT><ours/><same required="opt-in"/></RECIPIENT>
51 <RETENTION><stated-purpose/></RETENTION>
52 <DATA-GROUP>
53 <DATA ref="#user.name.given"/>
54 <DATA ref="#user.name.family"/>
55 <DATA ref="#user.home-info.postal.street"/>
56 <DATA ref="#user.home-info.postal.city"/>
57 <DATA ref="#user.home-info.postal.postalcode"/>
58 <DATA ref="#user.home-info.postal.country"/>
59 </DATA-GROUP>
60 </STATEMENT>
61 </POLICY></POLICIES>

```

Figure 7.10: Demo service P3P policy



## Chapter 8

# Conclusion

As a conclusion to the work done on redesigning the UbiCollab platform to support the privacy mechanisms described in this thesis, this chapter has been included to present the following:

- The *contribution* made to UbiCollab by doing research on how to improve privacy in a ubiquitous collaborative environment. Further, the section will present the contribution made by the redesign of UbiCollab in connection with privacy.
- An *evaluation* of the work done during the project.
- An overview of *future work* on the UbiCollab platform regarding further improvements on the issue of privacy and identity management, but also on the basic platform functionality.

### 8.1 Contribution

UbiCollab is a platform developed to support collaboration in a ubiquitous environment. The platform is inspired by work in the field of Computer Supported Cooperative Work (CSCW) and supports general and high level cooperation. Previous contributions have been focusing on developing the basic platform services and providing the users with the basic functionality for cooperation. Through our work we have developed an extension to this platform in the form of a basic privacy and identity management system, that will ensure the users' privacy in UbiCollab.

One of the contributions made by this thesis is the extensive research done in the field of privacy in ubiquitous computer systems. In order to define the areas in which we could improve the privacy mechanisms, a study of research done in related projects and similar solutions has been performed. The approach chosen for this thesis has been based on developing scenarios that defined the most important concepts of privacy in UbiCollab. The process and results from generating privacy specific scenarios and studying related research has been documented in the report and offers insight into the project's problem area.

Based on the research of related material and the theory of privacy in ubiquitous computing we were able to combine the studied material with our own ideas in the research field. The result from this work is the integration of privacy mechanisms, mainly developed for the web, into a service-based architecture. This work has resulted in contributions to UbiCollab, but also to the research field in general, by reusing existing models in a new environment.

In the Analysis (4) we defined the four most important areas to focus the design on. These areas of focus were defined based on the conceptual requirements and will provide the necessary privacy and identity management basics of UbiCollab. These contributions are valuable to UbiCollab both on a technical and a conceptual level, and are of great value for future extensions to the platform. The following list describes the most important contributions to the privacy architecture that have been supported in the design:

- Design of a **Privacy proxy** that handles all communication between the users and the platform has introduced the biggest change in the platform compared to previous designs. This is also the most important entity in the new privacy architecture and the most valuable contribution made by this thesis.
- Integration of the **P3P standard** in order to make it possible for the services to describe their data gathering practises in privacy policies. On the user-side the **APPEL language** has been integrated in UbiCollab to read and evaluate the policies of the services. An adaptation of the P3P standard has been done to suit web-services.
- The privacy architecture has been extended to make room for handling **multiple identities**. This improvement makes it possible for the users to define different identities for use on the platform and allows them to switch between these identities when they want to. In connection with the identities a **reputation system** has been designed, that allows the different identities to evolve over time and provide information about the users. The feedback is based on other users that post their opinions after having interacted with the user in question.
- Highly sensitive information will be handled by a **trusted 3<sup>rd</sup> party service**. The use of a trusted 3<sup>rd</sup> party service will cover the demand to mechanisms for dealing with the personal information that is considered too sensitive or important for distribution along the same channels as for instance name, phonenumber, email-addresses etc. This service has been incorporated in the design in order to support a wider spectre of additions in future versions of UbiCollab.

In the prototype implementation some issues from the design have been disregarded due to time constraints. The prototype implementation has focused on providing the basic mechanisms for privacy and being consistent with the previous version. The privacy proxy, basic P3P/APPEL evaluation and the possibility to use multiple identities have been implemented in the prototype. These concepts have been integrated in the platform prototype since they are the most essential part of the architecture, which make the foundation for the more refined mechanisms.

The previous platform prototype also needed changes before starting on the task of redesigning the architecture. To be able to implement the new design issues into the prototype we had to rewrite a lot of the existing code to suit the new demands to the privacy mechanisms. The prototype has been of great importance in order to check the design for flaws, which made the corrections of the

previous prototype a valuable contribution by the thesis. The implementation works as a proof of concept of the privacy architecture and shows that the redesign of UbiCollab offers valuable and realistic additions to the platform.

UbiCollab has experienced a series of contributions over the last years and has turned into a complex platform consisting of multiple services, applications and clients. This complexity has made the work with the platform complicated and a lot of time has been used to get to know the platform before starting to work on the new design. One of the contributions made by this thesis has been to provide an overview of the existing privacy architecture in order to define the areas which needed further attention in the new design.

With these contributions, especially from the design, we feel that we have offered the platform the basic mechanisms necessary to develop a powerful identity and privacy management system in UbiCollab. The use of P3P, APPEL, trusted 3<sup>rd</sup> party service and the privacy proxy are especially important for the privacy management system. The development of multiple identities and reputation has been a valuable contribution to the identity management.

## 8.2 Evaluation

This thesis is based on our previous work in the field of privacy on the UbiCollab platform and has taken the ideas presented there a step further. The thesis has also made new contributions to the platform, as summarized in the previous section. By using our previous study of basic privacy issues in ubiquitous environment as a basis gave us a flying start. We already had a good foundation for a privacy solution when we started and had time to perform a deeper research into the specific areas that needed attention. The scenarios were developed along the guidelines for privacy design defined in the previous report. The scenarios were refined into the conceptual requirements, which formed the most important areas of focus for the rest of the report. This way of approaching the problem has resulted in satisfying results both on the conceptual level and in the new design of the UbiCollab privacy architecture.

During the project a wide research in the field of ubiquitous computing and privacy has been performed, which has proved valuable for the work with identity and privacy management in UbiCollab. We are content with the research efforts, and the most important concepts of how to preserve privacy in a ubiquitous environment have been covered. The Problem Elaboration chapter along with the Related Work chapter present these concepts and their impact on the UbiCollab platform. We have by this work obtained a profound understanding of the area of privacy in ubiquitous environments and how to design for these issues. The only trouble connected to this work has been to relate the material to the UbiCollab platform. Even if some of the privacy mechanisms applied to the platform might appear as a bit of a "overkill" for the existing functionality offered by UbiCollab, the platform has been improved due to our research.

In the design of the improved UbiCollab platform we have covered the demands stated in the conceptual requirements (2.5) by designing the new privacy architecture and implementing the main concepts in a prototype platform. During the work on this thesis it has at some points become apparent that the privacy architecture we developed would be too extensive for the UbiCollab

platform the way it appears today. The privacy mechanisms applied to the platform will perhaps seem too complicated for the common user and developers. Still, the richness of detail in the privacy considerations are valuable for future extensions to the platform and will be necessary when the platform is taken further in future editions. Developers may not want to utilize all the privacy functionality offered by the new design, but still take advantage of the basic functionality and basic privacy considerations. The flexibility of the design allows the developers to choose what functionality to utilize, given the limitations of the API.

The overall goal has been to enhance the privacy mechanisms of the UbiCollab platform by providing the users with more refined ways of controlling the gathering and use of personal information. This ambition has been fulfilled in a satisfying way through the design and the privacy architecture we have presented in this thesis. The work has been based on the main principles of privacy design stated by Langheinrich [25], which we have covered in the new privacy architecture. Out of these principles, the most important issues have been to support choice, consent and notice. These principles are the ones that focus the most on increasing the users' control over their own personal information, which has been our main ambition from the start. By introducing P3P to the platform and providing users with a proxy to protect their privacy, these design principles are fully supported in the platform privacy architecture. The platform has been the subject of improvements both on a conceptual level and in the form of the parts that have been implemented in the prototype.

The most time consuming part of the project has been the implementation of the new design in the platform prototype. This part of the thesis proved more difficult than first assumed. The problems connected to the implementation has resulted in some weaknesses of the prototype. The consistency with the work done by the two parallel projects does not fully satisfy the ambitions we had to the prototype when we started. Conceptually the thesis are consistent, but the individual implementation of the different projects' prototypes do not correspond. This has happened due to shortage of time, but also because the cooperation between the different projects has been hard to maintain due to very different areas of interest.

Further, we have not been able to implement as much of the privacy architecture as we aimed at when creating the design. We have only been able to implement a small subset of the privacy architecture covering the most important concepts presented in the thesis. We have implemented the basic mechanisms for supporting identity and privacy management, but the more refined concepts have been left for future editions of the platform. The reputation system and the handling of highly sensitive information through a trusted 3<sup>rd</sup> party service have not been implemented in the prototype, since they demand work also on the client-side. The issues that have been implemented are demonstrated in chapter 7 to show how UbiCollab is working after our improvements. These concepts include a privacy proxy, negotiation of policies through an evaluator engine and support for multiple identities.

The shortage in the implementation is partly due to time constraints, but other issues have also been limiting factors in this regard. Some of the concepts from the privacy architecture are not easily defined without doing an analysis of the functionality an application using it would demand. A reputation system will not be a valuable contribution to the platform prototype if there's no application that allows the users to post feedbacks. These parts have therefore not been implemented, since it has been out of the scope of this project to develop suitable applications using the privacy mechanisms. It would also have been too time consuming to analyze what requirements such a system would have to privacy.

The scenarios presented different ideas on how to support privacy by refined privacy mechanisms and services. Through the design we have made the foundation for basic support of these concepts, but there are still issues that need further research before we have achieved full support in the platform. In the demonstrator we presented the main issues that have been realized in the prototype platform, which showed that the basic privacy support is present in the platform today. To fully realize the scenarios, more work will have to be done both on research of privacy mechanisms and on analyzing what demands future applications might state to the platform. There has been too little time for testing and analyzing the ramifications of implementing the privacy architecture in the platform. In order to fully understand the implications of performing the changes proposed in this thesis a more extensive prototype needs to be developed. This is one of the concerns at the end of the project, since not all the privacy issues applied to the architecture have been thoroughly tested and validated.

The size of the platform has been an issue of concern during the work. When redesigning the privacy mechanisms on the platform, a lot of the internal structure of the basic platform also had to be changed. This structure didn't have any direct consequences for the privacy preservation in the platform, but still had to be taken into account in order to develop a suitable privacy architecture. This resulted in a problem of focusing merely on the privacy architecture. Accordingly, our work has also had implications on the general flow of communication, internal structure and general user management on the platform. Another time consuming issue has been the bug-fixing on the previous prototype, which proved more extensive than first assumed. These issues have made us spend time on details that don't directly relate to privacy, but which has been important to preserve consistency between the privacy architecture and the rest of the platform.

Altogether we feel that our work has been a successful contribution to the UbiCollab platform and given us profound knowledge in the field of privacy in ubiquitous collaborative environments. The process and the work methods used during the project have been important tools for achieving the results presented in the report. We have fulfilled our demands to a privacy architecture and been able to provide the platform with an improved system of managing privacy and identities.

### 8.3 Future work

The contributions to UbiCollab over the last couple of years have introduced new services, improved functionality and an improved privacy solution to the platform. A lot of work has been done both conceptually and on a the technical level. The platform offers a collaborative ubiquitous environment, which has opened for flexible cooperation with different devices. Still we feel that there are areas that needs further attention. To make UbiCollab into a complete system with appeal both to users and developers we feel that more work is needed both on a general level and in the field of privacy.

In order to state further demands to the functionality of the platform, future contributions should include development of clients and applications that can be used to investigate and analyze the existing platform. Without a more specific research into what is expected from the platform, it will not be necessary to change or improve more details of the basic platform functionality.

Another area that needs further attention is the platform prototype. The existing prototype should be made consistent with all the projects that have been developed during the last year. This

issue has been neglected due to time constraints and the size of the task. Managing to get all the contributions to fully function together along with bug-fixing and extensive testing includes a large amount of work, which should not be treated as a side project to a master's thesis.

Apart from the future work on the platform in general, we have also discovered some issues related to privacy that would result in valuable contributions to the platform if incorporated into the platform. We have been working with privacy preservation on the platform and feel that the most important areas have been covered. Still, there are issues that have not been taken into consideration and concepts we have discovered that need further research to fully reach its potential in UbiCollab.

Concerning P3P and APPEL policy evaluation, further research should be executed to extend the languages to include tags that are specifically suited for UbiCollab. Further investigation into using P3P to support the *personae* attribute that is included in the APPEL specification needs to be performed. This would simplify the identity management system and provide the user with even more flexibility connected to his different identities.

The privacy mechanisms also need support on the client-side to better show its potential. The APPEL language needs a way to allow the users to easily configure their preferences. An administrative tool is also needed to allow administrators of the system to configure the policies of the P3P policies of the services.

UbiCollab also needs improvements of the reputation system and identity management. Conceptually these issues have been dealt with, but they need to be better supported in the design. There should be performed further investigation into the idea of a reputation service and how to provide the users with the necessary means to post feedbacks. This issue needs further research both on the client and platform-side of the system. Concerning identity management, UbiCollab functions well today, but to explore the potential of allowing multiple identities, further work on "*switching and stitching*" should be performed.

There will always be areas that need further research on an experimental platform as UbiCollab. Especially important is the development of applications that can be used to test the existing architecture and functionality of the platform. With this thesis we feel that the issue of privacy has been thoroughly investigated and hopefully future developers will be able to utilize the privacy mechanisms integrated in the new UbiCollab design to cover the demands they might state to privacy support.

# Bibliography

- [1] Paul Ashley, Satoshi Hadaand Gnter Karjoth, Calvin Powers, and Matthias Schunter. Enterprise privacy authorization language (epal 1.1). <http://www.zurich.ibm.com/security/enterprise-privacy/epal/Specification/index.html>.
- [2] The Apache Web Services Project Axis. <http://ws.apache.org/axis/>.
- [3] Anders Bakkevold. A shared display system for a ubiquitous computing environment. Master's thesis, NTNU, June 2004.
- [4] Richard Beckwith. Designing for ubiquity: The perception of privacy. *Pervasive Computing, IEEE*, pages 36–41, April 2004.
- [5] Oliver Berthold and Marit Köhntopp. Identity management based on p3p. In *International workshop on Designing privacy enhancing technologies*, pages 141–160, New York, NY, USA, 2001. Springer-Verlag New York, Inc.
- [6] Anders Magnus Braathen and Hans Steien Rasmussen. Preserving privacy in a ubiquitous collaborative environment: Extending the ubicollab platform. ., November 2004.
- [7] JRC P3P Resource Center. <http://p3p.jrc.it/>.
- [8] Lorrie Cranor, Brooks Dobbs, Serge Egelman, Giles Hogben, Jack Humphery, Marc Langheinrich, Massimo Marchiori, Martin Presler-Marshall, Joseph Reagle, Matthias Schunter, David A. Stampley, and Rigo Wenning. The platform for privacy preferences 1.1 (p3p1.1) specification. <http://www.w3.org/TR/P3P11/>.
- [9] Lorrie Cranor, Marc Langheinrich, and Massimo Marchiori. A p3p preference exchange language 1.0 (appel1.0). <http://www.w3.org/TR/P3P-preferences/>.
- [10] Lorrie Cranor, Marc Langheinrich, Massimo Marchiori, Martin Presler-Marshall, and Joseph Reagle. The platform for privacy preferences 1.0 (p3p1.0) specification. <http://www.w3.org/TR/P3P>.
- [11] Pedro André Cravo da Silva Garcia Gonçalves. Ubiclient: A mobile client for an ubiquitous collaborative environment. Master's thesis, NTNU, June 2004.
- [12] Dictionary.com. Persona. <http://dictionary.reference.com/search?q=persona>.
- [13] Monica Divitini, Babak A. Farshchian, and Haldor Samset. Ubicollab: Collaboration support for mobile users. *Proceedings of ACM, SAC 2004*, March 2004.

- [14] eBay. <http://www.ebay.com>.
- [15] Nilsson M. et al. Privacy enhancements in the mobile internet. *Working Conf. on Security and Control of IT in Society, Bratislava*, June 2001.
- [16] W3C Platform for Privacy Preferences Initiative. P3p 1.0: A new standard in online privacy. <http://www.w3.org/P3P/brochure.html>.
- [17] UPnP Forum. <http://www.upnp.org/>.
- [18] Jeremy Goecks and Elizabeth Mynatt. Enabling privacy management in ubiquitous computing environments through trust and reputation systems. *Privacy in Digital Environments: Empowering Users: CSCW 2002 workshop*, November 2002.
- [19] Hugo Haas, Rigo Wenning, Lorrie Cranor, Joseph Reagle, and Patrick C. K. Hung. P3p: Beyond http, p3p task force report. <http://www.w3.org/P3P/2003/p3p-beyond-http/Overview.html>.
- [20] Carsten Heitmann. Discolab: a toolkit for the development of shared display systems in ubicollab. Master's thesis, NTNU, June 2005.
- [21] Carsten Heitmann and Børge Jensen. Location-aware service for the ubicollab platform. ., November 2004.
- [22] Børge Jensen. Location-aware service for the ubicollab platform. Master's thesis, NTNU, June 2005.
- [23] J Kay, R J Kummerfeld, and P Lauder. Manage private user models and shared persons. *School of information Technologies University of Sydney*, 2003.
- [24] Pradip Lamsal. Requirements for modeling trust in ubiquitous computing and ad hoc networks. *Research Seminar on Telecommunications Software*, Autumn 2002.
- [25] Marc Langheinrich. Privacy by design - principles of privacy-aware ubiquitous systems. *Proceedings of Ubicomp 2001*, Sept-Oct 2001.
- [26] Marc Langheinrich. A privacy awareness system for ubiquitous computing environments. *4th International Conference on Ubiquitous Computing*, September 2002.
- [27] Dimitrios Lekkas, Sokratis K. Katsikas, Diomidis D. Spinellis, Pavel Gladychyev, and Ahmed Patel. User requirements of trusted third parties in europe. *User Identification and Privacy Protection: Applications in Public Administration and Electronic Commerce*, pages 229–242, June 1999.
- [28] Tim Moses. Privacy policy profile of xacml. <http://docs.oasis-open.org/xacml/accesscontrolxacml20privacyprofilespeccd01.pdf>.
- [29] Alan J Munro. (personal communications) conversations on personae and privacy, January-June 2005.
- [30] Ginger Myles, Adrian Friday, and Nigel Davies. Preserving privacy in environments with location-based applications. *Pervasive Computing, IEEE*, pages 65–64, Jan-Mar 2003.
- [31] Sun Developer Network. <http://java.sun.com/j2se/>.



- [32] Leysia Palen and Paul Dourish. Unpacking privacy for a networked world. *CHI 2003*, April 2003.
- [33] FreeRADIUS Server Project. <http://www.freeradius.org/>.
- [34] The Apache Ant Project. <http://ant.apache.org/>.
- [35] Diameter protocol. <http://www.diameter.org>.
- [36] Howard Rheingold. *Smart Mobs - the next social revolution*. Perseus Publishing, first printing edition, 2002.
- [37] Christian Schwarz. Ubicollab - platform for supporting collaboration in an ubiquitous computing environment. Master's thesis, NTNU, June 2004.
- [38] M Smyth, B Raijmakers, and A J Munro. Who am i and where am i? switching and stitching in the digital age. *International design conference Design 2004*, May 2004.
- [39] The Apache Jakarta Project Tomcat. <http://jakarta.apache.org/tomcat/>.
- [40] Mark Weiser. The computer for the 21st century. *Scientific American*, pages 94–104, September 1991.
- [41] Wikipedia. Reputation management. <http://en.wikipedia.org/wiki>.
- [42] Wikipedia. Sensitive information definition. <http://en.wikipedia.org/wiki>.
- [43] MySQL Developer Zone. <http://dev.mysql.com/>.
- [44] Martijn Zuidweg. A p3p-based privacy architecture for a context-aware services platform. *W3C Workshop on the long term Future of P3P and Enterprise Privacy Languages World Wide Web Consortium 19 to 20 June 2003*, August 2003.
- [45] Martijn Zuidweg, José Gonçalves Pereira Filho, and Marten van Sinderen. Using p3p in a web services-based context-aware application platform. *9th EUNICE Open European Summer School and IFIP Workshop on Next Generation Networks*, September 2003.

## Appendix A

# Common Scenario

Brian and Sylvia are both working at a telecommunication company. They are currently working on the same project, and Sylvia has some ideas which she would like feedback on from Brian. Unfortunately he is busy for the rest of the day. She decides to organize a meeting the next day.

Picking up her PDA, she opens the software client, UbiClient. She then creates a meeting. The screen asks for time, place and people. She schedules the meeting with Brian at 12 o'clock the next day at her office (room S, Telenor). She prepares some slides for the meeting, and adds the relevant files to UbiCollab.

Brian is sent an email announcing the meeting. When he starts UbiClient, he can see the meeting already added on the display of his PDA. He notices that Sylvia already posted some topics for discussion and some slides are available too.

On Wednesday they meet as planned. They both use their UbiCollab clients on their PDAs. Sylvia notices that the meeting room is equipped with a projector. She searches for the device with the UbiClient, the client comes up with the closest alternatives and she adds the projector to the meeting. She activates the projector, and it lights up and displays a welcome-screen, and a representation of her and Brian. She uses the client to display the slides she has prepared on the projector. The projector shows the file on the display, her representation widget lights up and she uses the PDA to remotely control the presentation. A small snapshot of the current window shows up on her display while she taps through the remote control commands.

During their discussion Brian takes his turn of arguing. He accesses the remote control window on his PDA and jumps to some previous slides of the presentation. His representation widget lights up. But still some doubts remain about the feasibility of the project. Brian suggests that they check with John, who is an expert on that technology, and Sylvia agrees. She can see on her UbiClient that John is unavailable (picking up children in kindergarten). They still need some help and decide to try contacting Steve the projectmanager for the project. Brian uses his UbiClient to invite Steve to the meeting. At the time, Steve is in the company cafeteria, logged on with his PDA. A message pops up on his screen, asking him to join the meeting with Sylvia and Brian. He confirms.

Steve opens an audio connection to the meeting room and Brian fills him in on their problems. Steve then wants to look at the slides and searches for a shared display using his UbiCollab Client. The system finds a display in the cafeteria but it is unavailable for him. He then decides to startup his laptop to use its shared display capability.

By glancing at the meeting information on her PDA, Sylvia can now see that Steve has joined thanks to a new presence widget which just showed up. Checking his available devices, she can see that he has a display application similar to the one in the projector. She says to Steve that she'll synchronize their displays, so that he can also see the current slide. She selects the two devices in UbiClient, and synchronizes them. Steve sees the slide showing up on his screen. He then examines the page and comments on Sylvia's remarks. Brian sees that the display is now being shared by all the three participants.

Steve wants to print out the slides to have a hardcopy to look at and he tells Sylvia and Brian to hold on while he does so. He uses his PDA and selects the document containing the slides, and tells the PDA to print it on the nearest printer. The PDA tells Steve which printer was chosen and shows him a map over where he is located and where the printer can be found. The PDA asks him to confirm if the printing is ok. Steve confirms and uses the map to find the printer

At the same time Alice, which is working on a different project with Sylvia, is trying to find Sylvia. She opens her UbiClient and see that Sylvia is logged in but busy in a meeting in room S. Alice is able to get this information because Sylvia has configured her profile for Alice to be able to see her location. Alice sends a message to Sylvia telling her that she wants to meet during the afternoon for a discussion on their project. The message will be displayed on Sylvias screen once the meeting has ended.

Just when Alice is finished typing out the message to Sylvia she gets a message on her UbiClient telling her that George, one of her co-workers on the project, is passing by in the hallway. She opens her office door and says hi. They talk about how things are going and then move on to more work-related discussion. From this conversation Alice gets the information she was going to query Sylvia about and she decides to postpone her meeting with Sylvia. She uses her UbiClient to cancel the message she had sent to Sylvia.

Brian asks Steve what he thinks about the idea. When Brian jumps to the previous slide using his PDA, the same thing happens on Steves display. Steve thinks it is a good idea, and he sees no counter arguments. Sylvia thanks Steve for his opinion and valuable time, and Steve leaves the meeting. His presence widget changes, and the audio connection ends.

Excited about the new idea, Brian and Sylvia continue their discussion. They still need input from John on the feasibility of the project and they tell the system to notify them once John is back in the office and available for discussion. Brian writes down a minute of the meeting in a text editor, and saves it using the UbiCollab client.

The next day, Steve is interested in reviewing what happened in the meeting. Starting UbiClient he accesses his meetings and taps the one he had with Brian and Sylvia. Tapping on the meeting notes he discovers Brian's topic minutes and reads them.

# Appendix B

## Glossary

**3G** - Third-Generation Cell-Phone Technology.

**AAA** - Authentication, Authorization and Accounting. Framework for authenticating users, giving them authorization to use certain resources and finally measuring the resources a user consumes when using the system.

**Apache Ant**- Apache Ant is a Java-based build tool with XML-based configuration tools.

**Apache Jakarta Tomcat** - Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed by Sun under the Java Community Process (<http://jakarta.apache.org/tomcat/>).

**API** - Application Programming Interface.

**APPEL** - A P3P Preference Exchange Language. Used to describe the users' privacy preferences.

**Axis** - Apache Web Services Project. Apache Axis is an implementation of the SOAP submission to W3C.

**CDPL** - Context Dependent Preference Language

**EPAL** - Enterprise Privacy Authorization Language created by IBM

**Formal communication** - A presentation or written piece that strictly adheres to rules, conventions, and ceremony, and is free of colloquial expressions.  
([www.armour.k12.sd.us/Mary's%20Classes/literary\\_terms\\_glossary.htm](http://www.armour.k12.sd.us/Mary's%20Classes/literary_terms_glossary.htm))

**GPS** - Global Positioning System

**HTML** - Hyper Text Markup Language

**HTTP** - Hyper Text Transfer Protocol. Protocol used for communication over the World Wide Web.

**Informal Communication** - A casual discussion, verbal exchange, note, or memorandum that may adhere less strictly to rules and conventions (e.g. a short note to a friend).  
([www.armour.k12.sd.us/Mary's%20Classes/literary\\_terms\\_glossary.htm](http://www.armour.k12.sd.us/Mary's%20Classes/literary_terms_glossary.htm))

**JavaScript** - A popular scripting language that is widely supported in Web browsers and other Web tools. It adds interactive functions to HTML pages, which are otherwise static, since HTML is a display language, not a programming language. JavaScript is easier to use than Java, but not as powerful and deals mainly with the elements on the Web page.  
(<http://computing-dictionary.thefreedictionary.com/JavaScript>)

**MySQL Database Server** - MySQL is an open source relational database management system (RDBMS) that uses Structured Query Language (SQL). ([www.easehosting.com/support/glossary.php](http://www.easehosting.com/support/glossary.php))

**P3P** - Platform for Privacy Preference Project by the W3C

**PDA** - Personal Digital Assistant.

**Persona** - The role that one assumes or displays in public or society; one's public image or personality, as distinguished from the inner self [12]. Used in computer systems as the representation of one's characteristics inside the system.

**Pervasive Computing** - The idea that technology is moving beyond the personal computer to everyday devices with embedded technology and connectivity as computing devices become progressively smaller and more powerful.  
([http://www.webopedia.com/TERM/P/pervasive\\_computing.html](http://www.webopedia.com/TERM/P/pervasive_computing.html))

**Privacy policy** - A statement describing what information is being collected, how the collected information is being used, how to access the data collected about you and what security measures are being taken by the parties that collected the data.

**Privacy preference** - A statement that describes what kind of personal information a user allows to be collected.

**Proxy** - The term proxy originally means the authority to act for another. In connection with information technology the term refers to a software agent that acts on behalf of a user. The most typical use of a proxy is to give the user access through authentication and create a safe distance between the user and system.

**SOAP** - Simple Object Access Protocol. SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses.

(<http://ws.apache.org/axis/>)

**SQL** - Structured Query Language. The most popular language for adding, accessing, and processing data in a database.

**Trusted 3<sup>rd</sup> party** - trusted external entity that provides security between two other entities and their exchange of information. The trusted 3<sup>rd</sup> party often manages its responsibility through the use of cryptography.

**Ubiquitous Computing** - The concept of building computers into our everyday working and living environments to such an extent that data, rich media and network access become constantly, frictionlessly and transparently available.

(<http://www.nottingham.ac.uk/cyber/fullglos.html>)

**UpnP** - Universal Plug and Play is a protocol specification that enables discovery and control of networked devices and services, such as network-attached printers, Internet gateways, and consumer electronics equipment.

**W3C** - The World Wide Web Consortium. The main standards body for the World Wide Web. W3C works with the global community to establish international standards for client and server protocols that enable on-line commerce and communications on the Internet. It also produces reference software.

(<http://dictionary.reference.com/>)

**Web service** - Refers mostly to a modular application that can be invoked through the Internet. Web services typically communicate over HTTP and use XML standards including SOAP and WSDL.

**WSDL** - Web Service Definition Layer

**XML** - eXtensible Markup Language. An initiative from the W3C defining an "extremely simple" dialect of SGML suitable for use on the World Wide Web.

(<http://dictionary.reference.com/>)

## Appendix C

# UbiCollab API

The API of the new platform prototype is much the same as in previous versions, but the interface has been moved from the collaboration server to the privacy proxy. The API consists of the following methods:

**public String[] userLogin(String username, String password)**

Logs the user into UbiCollab and returns an array with first part either "OK" or "not OK". When login is successful the sessionkey is returned. If login fails, a error message is returned.

**public String userLogout(String sessionKey, String username)**

Logs out a user from the system and deletes all the temporary records that have been kept in the runtime environment.

**public String addUserToCollabInst(String userMail, String username, String collabInstID)**

Add the creator of a collaboration instance to the instance. Allows connection to a specific identity with userMail or default identity with username. Returns ok or error message.

**public String createUser(String userEmail, String password, String firstName, String lastName)**

Create a new user profile in the proxy database. Also creates an identity with the given userEmail, as well as a representation of the identity in the collaboration. (This is a simplification for more advanced methods of user creation). Returns a username or error message.

**public String createCollabInst(String sessionKey, String collabInstName, String time, String place, String notificationType)**

Creates a new Collaboration Instance with the supplied values. Returns the id of the new Collaboration Instance if all went well, or an error string describing the cause of failure.

**public String createUser(String userEmail, String password, String firstName, String lastName)**

Creates a new user in the user-database with the supplied values. Returns ok if all went well, or an error string describing the cause of failure.

**public String registerDevice(String deviceID, String password, String name)**

Registers information about a device in the collaboration server. Returns ok if all went well, or an error string describing the cause of failure.

**public String addPersonsToCollabInst(String sessionKey, String[] userEmails, String collabInstID)**

Adds a list of users to a given Collaboration Instance. Returns a list of the users that were successfully added, or a an error string.

**public String addResourceToCollabInst(String sessionKey, String url, String type, String friendlyName, String description, String collabInstID)**

Adds a resource to a given Collaboration Instance. Returns ok if all went well, or an error string describing the cause of failure.

**public String getCollabInstInfo(String sessionKey, String collabInstID, boolean force-Output, boolean listActions)**

Returns an XML-string containing information about a given Collaboration Instance. If error, returns string describing the cause of failure.

**public String getCollabInstInfo(String sessionKey, String collabInstID, boolean force-Output)**

Returns an XML-string containing information about a given Collaboration Instance. If error, returns string describing the cause of failure.

**public String getCollabInstInfo(String sessionKey, String collabInstID)**

Returns an XML-string containing information about a given Collaboration Instance. If error, returns string describing the cause of failure.

**public String getUserCollabData(String sessionKey, String userEmail)**

Returns an XML-string containing data about the Collaboration Instances registered to a given user. If error, returns string describing the cause of failure.



**public String getUserCollabData(String sessionKey, String userEmail, boolean forceOutput)**

Returns an XML-string containing data about the Collaboration Instances registered to a given user. If error, returns string describing the cause of failure.

**public String getUserProfile(String sessionKey, String userEmail)**

Returns an XML-string containing the details registered about a user identity in the collaboration server. If error, returns string describing the cause of failure.

**public String getUserResources(String sessionKey, String userEmail, String searchString, int maxResults, boolean forceOutput, boolean listActions)**

Returns an XML-string containing the resources registered to a given user. If error, returns string describing the cause of failure.

**public String getUserResources(String sessionKey, String userEmail, String searchString, int maxResults, boolean forceOutput)**

Returns an XML-string containing the resources registered to a given user. If error, returns string describing the cause of failure.

**public String getUserResources(String sessionKey, String userEmail, String searchString, int maxResults)**

Returns an XML-string containing the resources registered to a given user. If error, returns string describing the cause of failure.

**public String justKeepAlive(String sessionKey)**

Set a new timestamp on session key. Returns ok if all went well, or an error string describing the cause of failure.

**public String removeUserFromCollabInst(String sessionKey, String userEmail, String collabInstID)**

Removes a user from a given Collaboration Instance. Returns ok if all went well, or an error string describing the cause of failure.

**public String removeResourceFromCollabInst(String sessionKey, String url, String collabInstID)**

Removes a resource from a given Collaboration Instance. Returns ok if all went well, or an error string describing the cause of failure.

**public String searchPeople(String sessionKey, String searchString, int maxResults)**

Returns an XML-string containing the persons that match the searchstring. If error, returns string describing the cause of failure.

**public String searchResources(String sessionKey, String searchString, int maxResults, boolean listActions)**

Returns an XML-string containing the resources that match the searchstring. If error, returns string describing the cause of failure.

**public String searchResources(String sessionKey, String searchString, int maxResults)**

Returns an XML-string containing the resources that match the searchstring. If error, returns string describing the cause of failure.

**public String setPresenceToCollabInst(String sessionKey, String userEmail, String collabInstID, String presencePercentage)**

Sets a presencevalue for a user in i collaboration instance. Returns ok if all went well, or an error string describing the cause of failure.

**public String searchResourcesClientPosition(String sessionKey, String searchString, String clientPosition, String deviceType, int maxResults)**

Returns an XML-string containing the resources that match the searchstring, including position of the resource. If error, returns string describing the cause of failure.

# Appendix D

## Previous version of UbiCollab

### D.1 Conceptual Model

The conceptual model from the previous work on the platform has been included to give a description of each existing service in UbiCollab and how they were changed in the previous design of UbiCollab. The model presented in figure D.1 is a modified version of the model used by Schwarz in Spring 2004 [37].

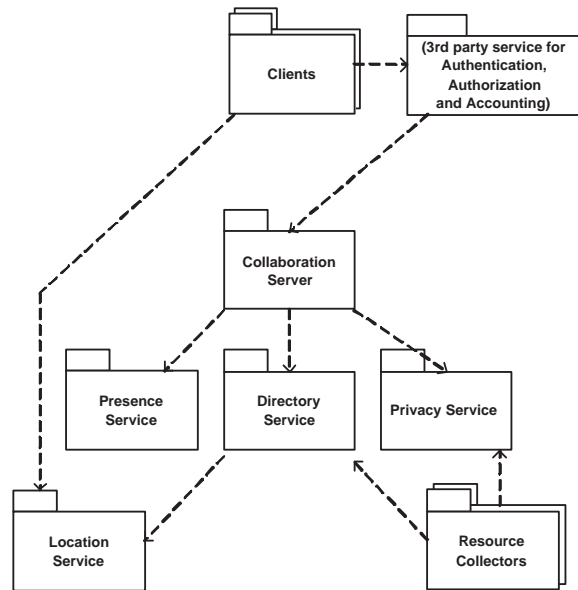


Figure D.1: Conceptual model of UbiCollab

### D.1.1 Client

The client is the user interface towards the UbiCollab server, that uses the trusted 3<sup>rd</sup> party to negotiate a user's rights and policies in connection with the system. The clients are of different sizes, according to the services they are supposed to offer and the technology they are built for. The UbiCollab platform only contains the native services, so the client must contain the extra services that are possible to add on top of the platform.

The user logs in on the AAA server with a username and password for authentication. Before he connects to the UbiCollab server, the user is supplied with a pseudonym from the AAA server. This pseudonym is used by the client until next time the user changes context or requests a service that needs more specific access rights. The pseudonym may also be changed during a session to protect the privacy for the user and making it harder for hostile observers to identify the real identity of the user.

### D.1.2 Trusted AAA-server(Authentication, Authorization and Accounting)

This service is an external service that handles the communication between users and the system interface. This service must have a way of negotiating the users privacy preferences with the policy description of the system. The use of P3P in a service based architecture like UbiCollab, with a policy description for each service, will satisfy the functionality we want for UbiCollab as described in ??.

This service will also keep the connection between the generated pseudonyms and the real identities of the users. The trusted external AAA server will act as a privacy proxy between user and system. General user information, like connections to collaboration instances, resources belonging to a specific user etc., must be kept here as well. This information will be labeled preferences through the remaining report. The preferences are kept in the policy repository, which is used during the authentication phase. This information must be kept here, since the AAA server is the only entity that knows the connection between the real user and their identities. During runtime the platform must have this information for each logged on pseudonym to provide the wanted functionality, but between sessions they are stored in the AAA server. The AAA server will communicate with UbiCollab through web services, and the communication will therefore be based on the XML standard for the transferred documents.

### D.1.3 Collaboration server

The collaboration server is the core of the platform and distributes the requests by the users and the services offered by the system, by communicating with the different services. The collaboration server will be changed to handle anonymous users. The collaboration server will also have to be designed to manage communication with the trusted AAA server and accordingly be able to store and interpret the user preferences that are sent from the AAA server.

#### D.1.4 Directory Service

The directory service lists the resources available in the system. This includes devices, documents, web-sites, users and other. The resources available are constructed by using the presence- and the resource collector services. The users will in our design also be included in the list of resources, since they may act as sources of information by for instance allowing other collaborators to track their position. The users may also be in possession of services available for other users, and since they're anonymous, the system will need to have the services belonging to each pseudonym listed during a session. The information about each users available resources will be sent to UbiCollab when logging in as an preference XML and will be updated during a session with support from the AAA server.

The directory service communicates with the location service when the location of a resource is needed for offering a service based on proximity or finding closest suitable device. Our contribution does not include any change to this service, except from making the listing of users as resources possible.

#### D.1.5 Presence Service

This service collects and stores the presence values of the different resources. The users may have different presences within the different contexts or pseudonyms they are operating with. Our architecture made no changes to this service.

#### D.1.6 Privacy Service

The privacy service acts as the systems privacy proxy. When a users logs on through the AAA server, the collaboration server negotiates the preferences of the user with the policy description of the system through the privacy service. The privacy service stores the properties found in the preferences of the user when being logged in. The property of a user describes the needed information during a single session. When the UbiCollab collaboration server has checked the *P3P* privacy preferences of the user with the privacy policy description of the system in the AAA server, the privacy service knows what services this specific user is interested in. The collaboration server then has the ability to offer the desired services to the user and also find the information about the preferences of the user through the XML that are stored during a session. The collaboration server also knows the actions the system is allowed to perform in relation to a user, which suits the demands to choice and consent.

The privacy service will also be able to manage the user preferences during a user session, so that when the user logs out the new additions to the preferences can be stored for later use. The privacy service also handles all registration of the users in connection with collaboration instances, change of identities and adding the user as a resource.

### **D.1.7 Location Service**

If the location service is allowed to track a user's location, it does so by gathering the position information from the user. It saves the position info, the pseudonym and the time the information was collected and sends this to the directory service. The information is then stored, and the devices that are allowed to track the user might find this listed as a resource. This service is also used by the directory service to collect the location information of other resources, when someone requests a service based on proximity or location. The location service does not know the true identity of the user, only the applied pseudonym. The right to collect user location is decided by checking the user preferences by sending a request to the AAA server. Further information on the design of the location service can be found in Heitmann and Jensen's work [21].

### **D.1.8 Resource Collector**

This service is collecting and sending information concerning about the available UPnP devices connected to the system. This service was not a subject to any changes in our work.