D5.3 Experiences and Feedback of the Pilots

Souheil Bcheri, Kasper L. Damgård, Daniel Deibler, Norbert Götze,
Hans G. Knudsen, Maksym Moneta, Apostolos Pyrgelis,
Eva Schlehahn, Michael B. Stausholm, Harald Zwingelberg

Editors:	*Norbert Götze (Nokia Solutions and Networks)*
Reviewers:	*Jonas Lindstrøm Jensen (Alexandra Institute A/S),* *Ioannis Krontiris (Goethe University Frankfurt)*
Identifier:	*D5.3*
Type:	*Deliverable*
Version:	*1.0*
Date:	*05/05/2014*
Status:	*Final*
Class:	*Public*

Abstract

This document is a public deliverable of the ABC4Trust project. Its main focus is to provide an internal feedback to the technical work packages of ABC4Trust after the execution of both pilots. On top of this, audiences like project leaders and developers will find this document a valuable input if they intend to incorporate Privacy-ABC technologies into their applications. Towards the end of this document, a chapter is dedicated to 'lessons-learned' to address pitfalls and bottlenecks and to provide some development hints for potential adopters of these technologies. All sections contain chapters on legal topics which are typically required by public projects designed to enhance and enforce the User's privacy. Especially in the context of minors being Users, ABC4Trust mandates a legal framework to be in place before real personal data can be processed.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 257782 for the project Attribute-based Credentials for Trust (ABC4Trust) as part of the "ICT Trust and Security Research" theme.

Members of the ABC4TRUST consortium

1.	Alexandra Institute A/S	ALX	Denmark
2.	CryptoExperts SAS	CRX	France
3.	Eurodocs AB	EDOC	Sweden
4.	IBM Research – Zurich	IBM	Switzerland
5.	Johann Wolfgang Goethe – Universität Frankfurt	GUF	Germany
6.	Microsoft Research and Development	MS	France
7.	Miracle A/S	MCL	Denmark
8.	Nokia Solutions and Networks GmbH & Co. KG	NSN	Germany
9.	Computer Technologies Institute and Press	CTI	Greece
10.	Söderhamn Kommun	SK	Sweden
11.	Technische Universität Darmstadt	TUD	Germany
12.	Unabhängiges Landeszentrum für Datenschutz	ULD	Germany

***Disclaimer**: The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.*

List of Contributors

Chapter	**Author(s)**
Foreword Executive Summary	Norbert Götze (NSN)
1st Chapter	Norbert Götze (NSN)
2nd Chapter	Maksym Moneta (EDOC), Souheil Bcheri (EDOC), Michael Stausholm (ALX), Kasper Damgård (ALX), Hans Knudsen (MCL), Eva Schlehahn (ULD), Daniel Deibler (ULD), Norbert Götze (NSN)
3rd Chapter	Apostolos Pyrgelis (CTI), Michael Stausholm (ALX), Kasper Damgård (ALX), Hans Knudsen (MCL), Eva Schlehahn (ULD), Daniel Deibler (ULD), Norbert Götze (NSN)
4th Chapter	Maksym Moneta (EDOC), Souheil Bcheri (EDOC), Apostolos Pyrgelis (CTI), Michael Stausholm (ALX), Kasper Damgård (ALX), Hans Knudsen (MCL), Eva Schlehahn (ULD), Daniel Deibler (ULD), Norbert Götze (NSN)
Appendix A	Eva Schlehahn (ULD), Harald Zwingelberg (ULD)
Glossary	Norbert Götze (NSN)
Acronyms	Norbert Götze (NSN)
Bibliography	Norbert Götze (NSN)

Foreword

In the final year of the ABC4Trust project, all scheduled pilots have been executed and the parties which have contributed to developing and operating the applications on top of the Privacy-ABC technologies, have a last chance to provide their feedback and experiences to the parties which originally developed these technologies. Even though no further pilot will be launched in the final phase of this project, the technical work package can still use this feedback to correct and/or enhance the publicly available code embedded into the following ABC4Trust repository:

https://abc4trust.eu/download/source/ABCEngine-source.zip

This repository contains the sources of the Privacy-ABC Engine (ABCE). Next to the plain engine itself, separate links point to the crypto engines (CEs) which represent the 2 hearts of the technologies. As proven in the pilots, both hearts can co-exist, like in a hybrid car. It is a choice of the web service, which technologies meets best its needs.

In the ABC4Trust project, *Microsoft Research and Development* provided the U-Prove CE and *IBM Research – Zurich* provided the Idemix CE. Both CEs have been adapted to interface to the ABCE. The reference implementation and the sample applications have been mainly developed by *Miracle A/S*, the *Alexandra Institute A/S* and *CryptoExperts SAS*. The consortium members which developed the applications for the pilots on top of the ABCE were *Eurodocs AB* (Söderhamn pilot), *Research Academic Computer Technologies Institute* (Patras pilot) and *Nokia Solutions and Networks GmbH & Co. KG* (both pilots). The legal frameworks for both pilots were crafted by *Unabhängiges Landeszentrum für Datenschutz*. So basically, the members who developed the applications and defined the legal frameworks will share their feedback and their experiences in this document with the members who developed the crypto engines, the reference implementation and the sample applications.

The ABC4Trust project partners encourage all decision makers and developers to adopt these technologies and thereby to make more Users aware of the value-add these technologies bring into their daily interaction with the Internet. If enough Users prefer to use web services which support Privacy-ABC technologies, other web services which don’t support them are more likely to add these technologies to their applications.

Executive Summary

In accordance with the ‘Description of Work’, this deliverable will provide feedback obtained in the practical execution of the Patras and Söderhamn pilots to the technical work packages of ABC4Trust. Since the deliverables [D63] and [D73] will focus on providing user feedback to the same audience, this deliverable will concentrate on the technical feedback.

Technical feedback includes information on which Java methods provided by WP4 (Reference Implementation) have been utilized by the application developers in order to integrate Privacy-ABC technologies into their design. Technical feedback is also information on which generic services implemented by WP4 have been customized and instantiated by the system administrators in the pilot architectures.

So what did the developers from WP5 (Application Requirements), WP6 (Community Interaction Among Pupils) and WP7 (Course Rating by Certified Students) use in order to provide the functionalities depicted in the final high level architecture figures and how easy was it to embed Privacy-ABC technologies into their design? The answer is rather surprising. The two IdM ABC Systems, the Course Evaluation ABC System and the Tombola ABC System are the only applications containing Privacy-ABC technologies which have not been written by the developers of WP4 themselves. These applications make use of wrappers, so-called ‘service-helper’ methods. Using software development kits, integration of these service-helper methods is an easy task of an experienced Java developer. In the ABC4Trust project, the application developers found examples on how to use the service-helper methods in the ‘sample applications’ (i.e. the pilot-patras and pilot-soederhamn Java projects) uploaded by developers of WP4 into a project-internal repository. This repository is not publicly accessible, but adopters of Privacy-ABC technologies can alternatively find usage examples of the service-helper methods in the public ABC4Trust repository in the abce-services directories. Contrary to that, the Inspector Application, the Restricted Area ABC System, the Revocation Authority, the Smart Card Initialization Tool and the User Application were provided by WP4 itself. All the pilots had to do here was to deploy these applications and operate them.

This document also provides some insight in the functionalities of the applications which do not contain Privacy-ABC technologies. The technical feedback given here concentrates on the pilot ecosystem and how the entities within the final high level architecture depend on each other. Several entities were not taken into account during the analysis phase of the ABC4Trust project. More specifically, several administrative applications were deemed necessary at a late point in time for setting up and maintaining the pilots. And so in e.g. the Söderhamn case, the applications ‘IdM Mass Provisioning Tool’, ‘IdM Admin GUI’, ‘Smart Card Registrar’, ‘Syntax Checker’, ‘Tray Application’ and the ‘Restricted Area Admin’ had to be developed on top of the original requirement catalogue defined in [D51].

The situation in the Patras pilot was significantly different. In order to lure more students into participating in the online course evaluation, a new use-case was defined for the 2nd round of this pilot. This use-case allowed students to participate in an anonymous tombola after casting their vote in the course evaluation. The impacts in the high level architecture and in the implementation of the Course Evaluation System were significant: the former required an Inspector next to a Tombola System and the latter had to be enhanced to perform advanced issuance (see [H22], Section 3.4.2). What the 2nd round of Patras has proven was that generating proofs based on credentials of different crypto engine types (see *Figure 42*) and that ‘advanced issuance’ (i.e. carry-over attributes) using Idemix (see *Figure 43*) is feasible with Privacy-ABC technologies. This document will elaborate in detail on these new applications and reveal that the effort for using these powerful Privacy-ABC features is just a matter of providing the correct XMLs.

An important aspect this document uncovers is the legal framework imposed on the pilots. Since the pilots handle personal data of pupils and students, only the pilot owners were allowed to administer the applications during the operation phase of the pilots. This underlines the necessity for readable and understandable printouts in the log files and in the command line windows. This document has come to the conclusion that there is room for improvement in the logging. The main issue that has been identified is that especially the crypto engines generate their printouts in the command line windows without dumping the same content into log files too. This has a number of disadvantages. The most important one is that the buffer of the command line window will be cyclically overwritten.

Setting up the pilots is not trivial, especially if a revocation authority is part of the ecosystem. In the integration and deployment phases of the pilots specific tasks have to be executed in a specific order. Data generated by the Issuer has to be copied to the Revocation Authority. And then, data generated by the Revocation Authority has to be copied to the Issuer. If a 2nd Issuer is in the same ecosystem, the global parameters of the 1st issuer must be copied to the 2nd Issuer. Finally, the data of the Issuer(s) has/have to be provided to the Inspector which then generates data which must be provided to the Users and to the Verifiers. In the end, if a credential specification changes, the whole process has to be repeated. The key message here is that setting up a Privacy-ABC System is not trivial but it would be more acceptable if flexibility is embedded into the design.

Finally, this document includes a 'Lessons Learned' chapter which provides pilot-specific sections, a 'Conclusions on legal Topics' section and a main section covering 'Technical Recommendations for future Pilots'. As the adopters of Privacy-ABC technologies will have access to the ABC4Trust public repository, it is very likely that they will be tempted to use the ready-to-deploy generic abce-services. This 'Lessons Learned' chapter contains a 'Security' section which points to a pitfall when deploying these generic abce-services and shows how to circumvent the security issues.

Table of Contents

Index of Figures

Index of Tables

1 Introduction

The ABC4Trust project's goal is to compare 2 specific privacy enhancing attribute-based credential technologies; one technology is developed by Microsoft (U-Prove) and the other by IBM (Idemix). In order to be able to make an objective comparison, a generic container was developed to which both Privacy-ABC technologies can connect to. This container is called the ABC engine (ABCE). The Privacy-ABC technologies themselves are referred to as 'Crypto-Engines' (CEs).

Contrary to state-of-the-art attribute-based credentials, Privacy-ABCs allow the User to selectively reveal attributes by transforming the credentials holding these attributes without interaction of the Issuer.

The ABC4Trust project has launched 2 pilots with different use-cases to prove the Privacy-ABC technologies. The Söderhamn pilot was aiming for implementing and operating a social platform for the Norrtullskolan primary school in Sweden. The goal was to offer this platform to pupils, their guardians and their teachers for exchanging private data. And the Patras pilot implemented and operated a course evaluation system for the University of Patras. The goal of this pilot was to offer students attending a specific course a comfortable and anonymous alternative to the paper-based course evaluation. Both pilots have been split into 2 rounds allowing enhancements from the 1st round to be incorporated into the 2nd round.

This document was written during and after the 2nd (and final) rounds of both pilots.

1.1 ABC4Trust Project and current Status

As depicted in Figure 1, the 1st rounds of both pilots were executed between M25 and M32 of the ABC4Trust project. More precisely, the 1st round of the Patras pilot started on Nov. 23rd 2012 and ended on Feb. 15th 2013. The 1st round of the Söderhamn pilot started on May 13th 2013 and ended on June 10th 2013.

The reader can easily see that the duration of the 1st round of Söderhamn pilot was rather short. Originally, this round of the pilot was to be launched end of Jan. 2013. But because of delays and obstacles in the smart card area and since this pilot was the first to make use of many new ABC4Trust features (specifically revocation, reIssuance and inspection), the start of this round had to be delayed by over 3 months. The 1st round of Patras deployed the ABCE version 1.0.0. Before the start of the 1st round of Söderhamn, the ABCE was significantly refined and enhanced so that Söderhamn deployed the ABCE version 1.0.8. So in effect one can say that the findings from the 1st round of the Patras pilot were already taken into account by the 1st round of the Söderhamn pilot.

During the kick-off of this deliverable, both the Söderhamn and the Patras pilots were in their 2nd round. The Söderhamn pilot started its 2nd round on October 14th with over 380 participants. And the Patras pilot started its 2nd round on November 4th 2013 with about 50 participants.

The 2nd rounds of both pilots ended before end of February (Söderhamn: Feb. 28th 2014 / Patras: Feb. 26th 2014). Due to the fact, that the 2nd round of the pilots are the final round of the ABC4Trust project, a major goal of this deliverable was to incorporate the technical experiences and feedback resulting from these final rounds into this document.

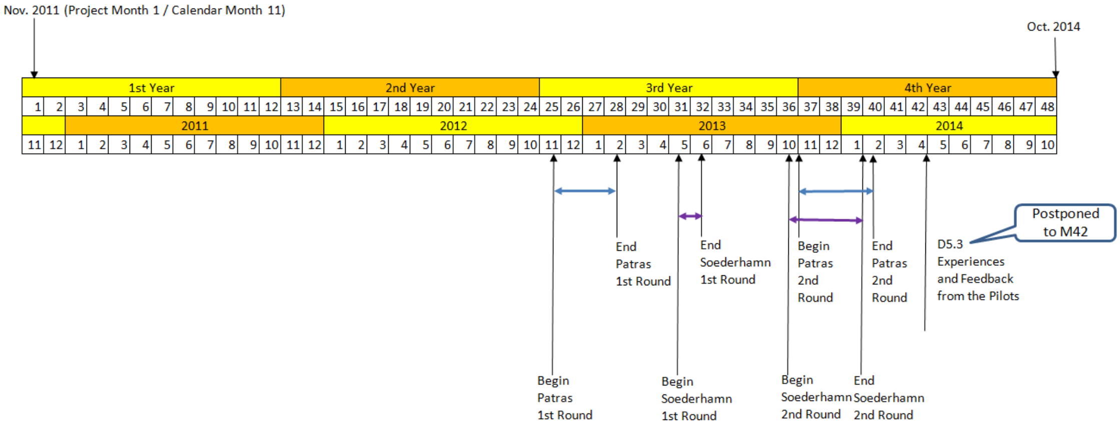


Figure 1: Pilot Execution in Relation to Deliverable D5.3

1.2 Scope of this Deliverable

The goal of this deliverable is to provide feedback resulting from the practical execution of the pilots to the technical work package, especially WP4. On top of that, in the early discussions of this deliverable, the consortium decided to include a 'lessons learned' section which should help an imaginative reader, who is new in this topic, to get started in using Privacy-ABC technologies. This document in effect should lower the threshold all decision-makers need to cross when analyzing these technologies and should help to accelerate the development process after a decision to use these technologies has been made,

1.2.1 Development-Phase Focus

In simple projects with no significant dependencies, a waterfall model containing the elements would describe the following development life-cycle:

- Requirements
- Design
- Implementation
- Verification
- Maintenance

But the ABC4Trust project has proven that a more detailed model is required (see Figure 2). This lies especially in the fact that the ABCE and the CEs are not being developed by the teams which are developing the pilot applications. So an additional integration step (6) is required. On top of that, NSN which is developing the Registration Systems cannot easily perform regression tests on the target platforms hosted by the pilots themselves. The target platforms would have destabilized and other applications, depending on the Registration Systems, would malfunction. So there was the requirement to set up a duplicate system on NSNs premises for testing the enhancements and bug-fixes before shifting the applications to the pilots' premises. Shifting the Registration Systems to their target platforms required an additional customization step. But the additional efforts for customization were definitely lower that the speed gain achieved in the test cycles executed on NSNs premises.

As in every software development process, a thorough analysis (1) needs to be performed before stepping into the design. The outcome of the analysis is a complete catalogue of requirements. After agreeing on this catalogue, the design phase (2) will generate an architecture which is able to meet all requirements. Since the ABC4Trust project launched two independent pilots, every pilot had its own requirement catalogue and its own architecture. So in effect, when looking at Figure 2, the reader must be aware that these development steps have been executed four times: twice for each pilot since every pilot executed two rounds. In the early phase of the ABC4Trust project, the requirement catalogue for the Söderhamn pilot and the requirement catalogue for the Patras pilot were documented in [D51]. The first drafts of the 'High Level Architectures' were presented in [D51]. The detailed use-cases of the pilots were discussed in [D61] and [D71] which paved the path to finally refine the architectures in [D62] and [D72]. The latter 2 documents describe only the 1st rounds of the pilots. The 2nd rounds of the pilots impacted the high level architectures. The pilot with the most significant changes in the 2nd round was the Patras pilot which decided to introduce a Tombola System and to make use of the inspection feature the Privacy-ABC technologies offer. The high level architectures of the 2nd rounds will be introduced later on in this deliverable so that the contributors of this document can refer to specific building blocks within the architectures when providing their feedback and their experiences from the practical execution of the pilots.

After the definition of the high-level architectures, the implementation (3) can begin. The result of this step is the application code. What every experienced developer knows is that during the development of code it is highly advantageous to check its functionality regularly. So typically, smaller implementation steps are verified in a module test (4) before adding new complexity to the applications in (3). As soon as the applications are ready to pick up the ABCE (together with the CEs) in (5), an integration test (6) must be performed. The integration test is typically not performed on the target servers i.e. the servers which will host the applications for the operation phase (9) of the pilot. Errors detected in (6) can lead to corrections in (3) and perhaps even corrections in the ABCE itself. Concept errors might lead to a revising of the high-level architectures (2) or even in adapting the underlying requirements (1). When 'code-complete' is declared by the team of (3) and if the integration test was successful (6), the applications can be installed on the target sites. Typically an additional customization step must be performed in order to reach (7). From then on, a final system test will verify the correct function of the applications in (8). In the ABC4Trust project, the final phase was when 'real' Users were allowed to access the system in (9). Before this, the pilot owners (EDOC and CTI) 'cut the lines' in order to enforce the legal framework setup by ULD. This resulted in that those developers from (3), who were not the pilot 'owners', could not access the servers remotely any more.

Figure 2 shows very clearly that any changes in the requirements will lead to cycling through the entire chain again. Late requirements coming from the pilots (especially requirements coming in when the development phase is beyond the integration test) are extremely expensive with respect to time and can impact the resource planning significantly.

The development phases this deliverable will focus on are:

(3 and 4) Implementation

(5 and 6) Integration

(7 and 8) Deployment

(9) Operation and Maintenance

The development phases of the ABCE and the CEs are out-of-focus of this document, as they have been provided by the technical work package (i.e. WP4) itself.

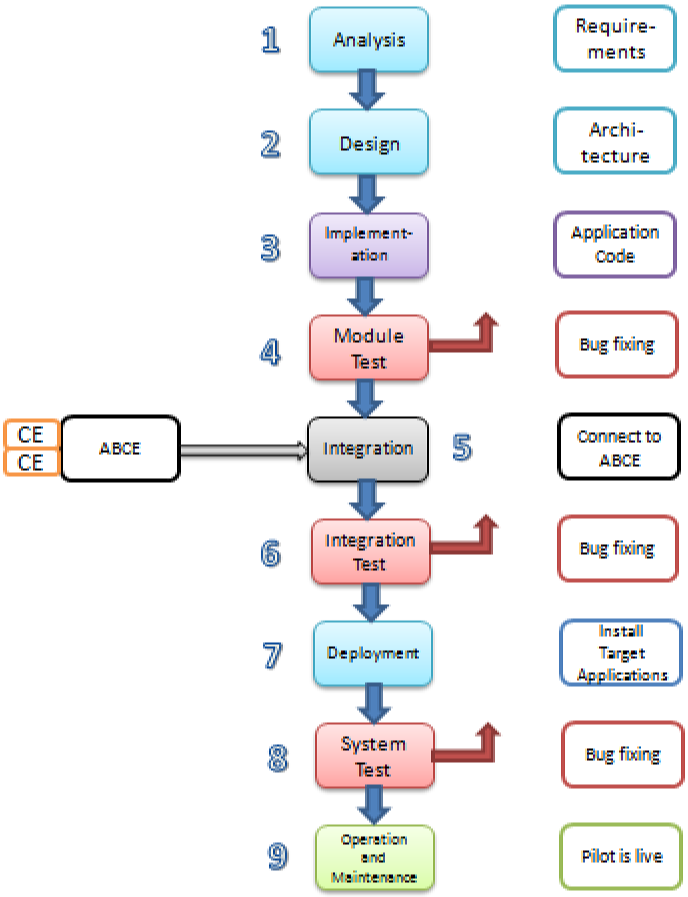


Figure 2: Development Life-Cycle of Pilot Applications

1.2.2 Personnel Focus

Technical feedback is best given by the programmers of the applications which make use of Privacy-ABC technologies. But, unfortunately for this project, most of the programmers were not directly involved in the practical execution of the pilots (see Figure 2, (9)). One reason for this was the privacy restrictions imposed by local laws and directives when personal information is being handled. Both NSN and ALX, as data processors in the legal sense, have contacts with the companies which were actually executing the pilots (i.e. the data controllers). But the data processors were only consulted during the execution of the pilots when the data controllers were not able to solve the problems themselves. So the experience that NSN and ALX could get during the practical execution of the pilot was very limited. As consequence, the experience obtained from the practical execution of the pilots had to be given mainly by the institutions hosting the pilots i.e. EDOC for the Söderhamn pilot and CTI for the Patras pilot.

Looking at the development life-cycle and mapping it to personnel, technical feedback and experiences of the pilots can be given by the **developers** from Figure 2, in phases 3 and 4, the **integrators** from Figure 2, in phases 5 and 6, and the **administrators** Figure 2, in phases7, 8 and 9.

Please note, that if the team responsible for phase 7 is not the pilot owner, the administrator teams of phases 7 and 9 are different.

Goal of this deliverable is to gather the technical feedback from each the above listed parties and thereby allow the reader of this document to gain access to different viewpoints of the Privacy-ABC technologies. Dedicated Sections 2 and 3 have been provided in this document to gather these contributions pilot-wise.

1.2.3 Pilot Round Focus

The experiences and the feedback on setting up the applications and executing the 1st round of the Söderhamn pilot and the 1st round of the Patras pilot have already been processed by WP4. The ABCE and the CEs have been adapted accordingly.
So this document will focus mainly on the 2nd rounds of both pilots. Unsolved issues, pitfalls and important hints resulting from the execution of the pilots will be included in the lessons-learned section (see Section 4) of this deliverable.

1.3 Other available ABC4Trust Deliverables

Contrary to the scope of this deliverable which provides the technical feedback, the deliverables [D63] and [D73] provide the non-technical feedback.
There is no task which matches the scope of D5.3. The tasks which map to [D63] and [D73] explicitly mention that the feedback must be given to the 'Comparison' and 'Reference Implementation' work packages. On top of that, [D63] and [D73] are also intended for audiences like government organizations and data protection officers.

Since D5.3 covering the technical feedback, the contributors giving this feedback must be the developers, the integrators and the administrators.

The group giving feedback in the deliverables [D63] and [D73] will be mainly the Users, i.e. the pupils, teachers, counselors in the Söderhamn pilot and the students in the Patras pilot. Since these persons will not edit these documents, the contributors to the deliverables [D63] and [D73] will be the pilot owners themselves.

1.4 Structure of the Document

Chapter 2 is focusing on the technical 'Experience and Feedback' of the Söderhamn pilot

Chapter 3 elaborates on the technical 'Experience and Feedback' of the Patras pilot

Chapter 4 discusses the technical 'Lessons Learned' of both pilots and gives hints to potential adopters of Privacy-ABC technologies

Annex A contains the data-processing contracts

Glossary

Acronyms

Bibliography

2 Experiences and Feedback of the Söderhamn Pilot

The Söderhamn pilot is launched in the Norrttullskolan primary (comprehensive) school in Söderhamn. The Users engaged in this pilot are pupils, teachers, counselors and guardians. But the primary group is the pupils, and therefore minors. The idea of this pilot is to offer a privacy-respecting social platform (i.e. the Restricted Area) to the pupils so that they can communicate with each other giving them the choice to be totally anonymous, or to use a linkable pseudonym (an alias), or to reveal parts of their identity or to reveal their entire identity. Using Privacy-ABC technologies, the Users of the Söderhamn pilot remain in full control of what they disclose.

The Söderhamn pilot was the first pilot to make use of inspection (see Figure 3). In order to protect the minors, this entity was introduced to lift the identity of Users if the 'inspection grounds' attached to a presentation policy hold. The inspection grounds are typically mapped in the Söderhamn pilot to emergency situations i.e. threat of life, excessive mobbing etc. Inspection itself is only possible, if the Inspector receives the presentation token which has been sent by the User whose identity must be lifted. Coupled to inspection is a process which defines how inspection is executed and a legal framework which defines when inspection is required (see [D61] for more details on the pilot scenarios and [D63] for more details on the inspection process).

2.1 Setup

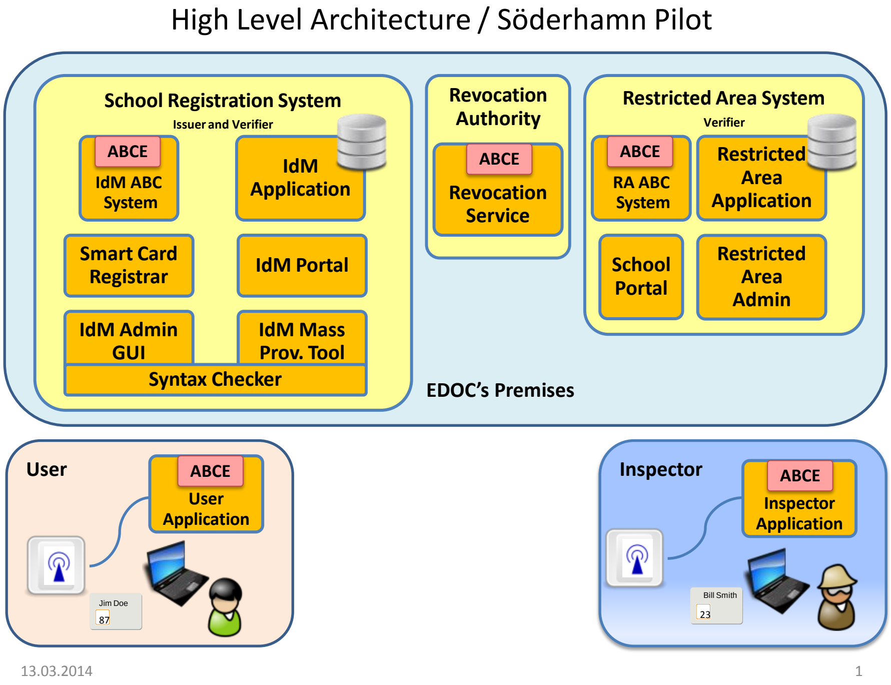


Figure 3: Final High Level Architecture of the Söderhamn Pilot

Three teams developed the applications for the Söderhamn pilot. WP4 of ABC4Trust developed the Revocation Authority (RevAuth), the Inspector Application, the 'IdM ABC System' of the Restricted

Area System and the User Application. EDOC developed the School Portal, the Restricted Area Admin, and the Restricted Area Application. And NSN developed all entities within the School Registration System.

The above mentioned teams have provided contents in the following chapters on the development phases until the Integration Test (see 3 to 6 of Figure 2). Contributions to the deployment phase (7) and the system test phases (8) were provided by EDOC and NSN. And finally, the operation phase (9) was handled by EDOC alone.

In the Söderhamn pilot, Users are provided with either a U-Prove or an Idemix smart card. Each card stores credentials of only one crypto-engine type. The key length of both Idemix and U-Prove are set to 2048. This pilot is the only pilot of this project that makes use of reIssuance since U-Prove smart cards need to fetch fresh batches of tokens from the Issuer in order to enforce un-linkability.

For the convenience of the pupils, customized smart cards with Privacy-ABC credentials are distributed to the Users. In order to increase the acceptance of this pilot, the GUIs of the School Registration System with which the pupils interact are translated to Swedish. Due to time restrictions, the 'new crypto architecture'[1] could not be deployed in this pilot. The Revocation Authority is not generic, meaning that it cannot be used in other pilots. Support of Microsoft's IE as browser is required by this pilot.

2.2 User

The User block, depicted in Figure 3, consists of the following main parts:

- Users (Pupil, guardian, teacher or other school personnel)
- Smart card with corresponding PIN/PUK codes and with Privacy-ABC credentials
- Smart card reader, installed on and connected to the PC
- Windows based PC (Connected to the Internet)
- Web browser (Firefox or Internet Explorer)
- User Application (Installed on the PC)

Every User participating (who has signed a consent form) in the Söderhamn pilot needs to have all the parts mentioned in the list above. The smart card is prepared (initialized and personalized) by EDOC before the card with PIN/PUK codes and a smart card reader are handed over to the User. In the 1st round of the pilot the User had to download her own credentials to the card. But in the 2nd round the credentials were downloaded to the smart card by EDOC before they were handed over to the User.

The smart card reader is easily installed on a Windows based PC. Windows automatically recognizes and installs the smart card drivers. Each User in the pilot receives a smart card reader from EDOC.

Windows based PCs were provided by the school. The school has a number of public laptops that the pupils can use. The pupils share those laptops with other pupils. The teachers have their own laptops which are different from the laptops used by the pupils. The laptops used by teachers and by pupils have different configurations and have access to different networks. The school has 2 networks for different users. The first network which is called "Skolnet4you.soderhamn.se" is publicly open and requires no passwords. This network is used mainly by pupils and visitors but of course also teachers can use it because it's public. And the second network which is called "admin.soderhamn.se" is protected by password because it's used by teachers, counsellors, nurses, administrators and the schoolmaster. As the teachers' computers required administrator permission to install programs EDOC

[1] The new crypto architecture was deployed in the 2nd round of the Patras pilot and introduced a set of enhancements.

had to ask the school administrators for permission to do the installation of all necessary components for the ABC4Trust pilot.

The web browsers Internet Explorer or Firefox were already installed on the school computers. If not, they were installed by EDOC.

In order to be able to make use of the Privacy-ABC technologies and participate in the pilot the User has to install the User Application. This is done in a simple way by clicking on the Installer link which can be found under the “Help” section at the School Portal (https://ra.abc4trust.se/Help). The browser plugins are also installed automatically by the same Installer.

EDOC did the installation of the User Application at all the school computers. Users did have to install this User Application at their computers at home.

All those details were explained for the Users in the User Manual that can be found under the “Help” section of the School Portal (https://ra.abc4trust.se/Help).

2.2.1 User Application

The User Application can be seen as a single entity that controls everything on the User side of the ABC4Trust architecture. It is contained in an installer, who handles both checking the Java version and downloading this if need be, as well as download and installation of the actual User Application. This includes setting up things correctly within the Windows registry and installing browser plugins with the appropriate security settings.

The User Application handles everything from webpage input to the ABCE to sending commands to the smart card. However, this architecture will be handled in more detail later in this document in order to get a deeper understanding of the problems during the different phases of development. Overall the challenge was to integrate the User Application with the other entities in the ABC4Trust ecosystem. A single mistake in either entity makes the entire system fail. This also holds for components in the User Application itself, but this was easier to debug as this can of course be done locally. However, one component was not easily debugged on the User side: namely the hardware smart card. More on this topic will follow in the sections to come. First, the different components in the User Application will be described. Then a closer look to the different phases of development will be made and finally, problems and issues during those phases will be mentioned.

2.2.1.1 Implementation

Given an ABCE with a crypto engine, which talks to the smart card, the project wants to develop a User Application that enables communication between the Restricted Area (RA) and the ABCE. For this, browser plugins had to be developed. This was of course written in JavaScript with the goal to reuse much of the code from one plugin for the other. The communication should then happen via a local web service to avoid cross-site scripting. This local web service then calls the ACBE properly. Jetty was used as the local web service, which transmitted messages using mostly XML as the policies were in this format.

On top of this, the User must be able to interact with the ABCE since she has to select which policy she wants to fulfill. Thus, a so-called ‘Identity Selector’ had to be developed that the User can interact with for selecting the wanted policy matching. This ‘Identity Selector’ was made generic such that it could be used for any use case, which includes the Patras pilot.

2.2.1.1.1 Browser Plugins

The browser plugins are necessary since a graphical interface is required to be able to communicate with the User ABCE. The User ABCE runs as a local web service which means that a normal website cannot communicate with it as a normal browser would consider it a cross-site scripting attack. Thus the browser plugins are required such that a foreign website is able to make requests to the User ACBE. It also has the advantage that communication to the User ABCE is strictly controlled which means that errors in the communication flow is less likely to happen compared to each website having to implement the correct protocol procedures. Aside from this, a plugin can be used as a utility feature, showing e.g. debug information and the credentials currently stored on the User's smart card. A plugin for both Firefox and Internet Explorer were made available. The plugins are the only UI's that the User can interact with, so it is also the browser plugin that shows the identity selector.

The plugins all have the same available menu points (example shown in Figure 4):

- **Manage credentials:** This function launches a window where you can browse your credentials and the contents of those as well as the revocation status. If enabled, you can also delete credentials manually. The view is shown in Figure 5
- **Check revocation status (optional):** This function checks for all of your credentials, if it is revoked, and deletes it if that is the case. This creates more space on the card and makes sure that no credentials are deleted which could still be of use. This is pilot specific (i.e. used in Söderhamn only).
- **Backup smartcard:** This function enables the User to take a backup of his or her smartcard. It stores encrypted content of the smartcard such as the device secret using a key known only to the smartcard and combine it with the User's password that he is prompted to enter. The information that only requires the User's pin code, such as the blob store, is encrypted using the User's password.
- **Restore smartcard:** Given that a backup file is present on the computer, this function will, given a password, try to restore the smart card to the state the card had when backup was performed. It can only be done if the smartcard has the same device ID as the card had when backup was performed. This ID is set at initialization time, which should prevent duplicate cards from being created.
- **Change pin:** This function changes the PIN of the card given that the User is able to enter the correct current PIN code.
- **Unlock card:** This function takes the PUK code as input as well as the new PIN, and can unlock a locked card. A card is locked if the PIN is entered incorrectly 3 times in a row.
- **Debug Info:** A helper method for easier debugging. It shows printouts which are only human readable by experts.

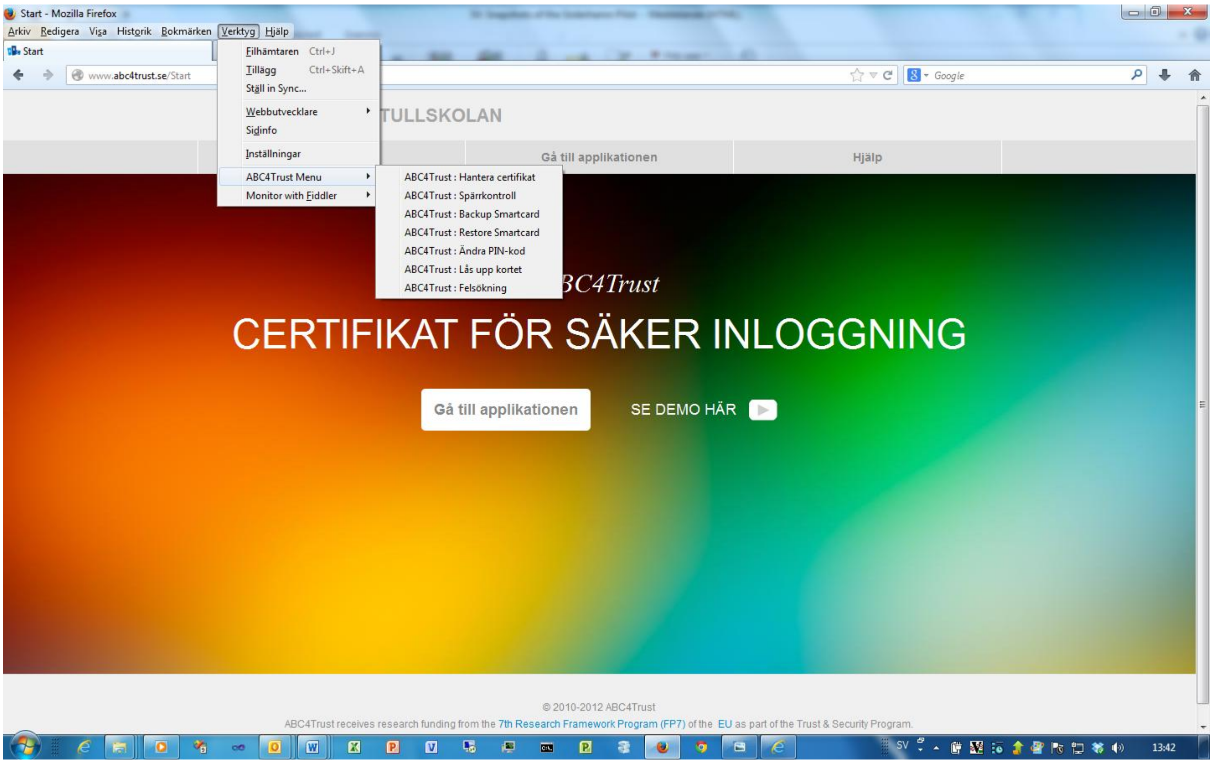
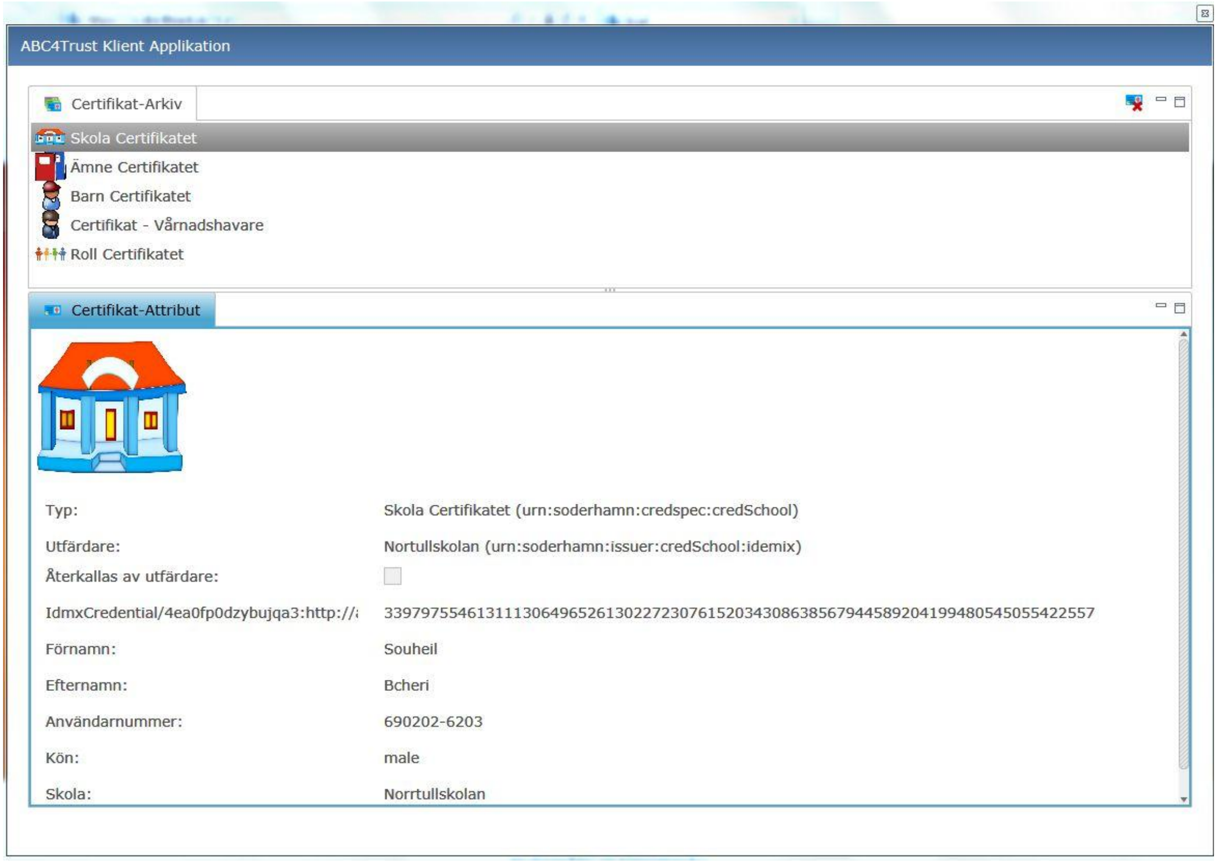

Start - Mozilla Firefox
Arkiv Redigera Visa Historik Bokmärken Verktyg Hjälp
Start
www.abc4trust.se/Start
Filhämtaren Ctrl+J
Tillägg Ctrl+Skift+A
Ställ in Sync...
Webbutvecklare
Sidinfo
Inställningar
ABC4Trust Menu
Monitor with Fiddler
ABC4Trust : Hantera certifikat
ABC4Trust : Spärrkontroll
ABC4Trust : Backup Smartcard
ABC4Trust : Restore Smartcard
ABC4Trust : Ändra PIN-kod
ABC4Trust : Lås upp kortet
ABC4Trust : Felsökning
TULLSKOLAN
Gå till applikationen
Hjälp
BC4Trust
CERTIFIKAT FÖR SÄKER INLOGGNING
Gå till applikationen
SE DEMO HÄR
© 2010-2012 ABC4Trust
ABC4Trust receives research funding from the 7th Research Framework Program (FP7) of the EU as part of the Trust & Security Program.
SV
13:42

Figure 4: The Internet Explorer Plugin Menu



Typ: Skola Certifikatet (urn:soderhamn:credspec:credSchool)

Utfärdare: Nortullskolan (urn:soderhamn:issuer:credSchool:idemix)

Återkallas av utfärdare: □

IdmxCredential/4ea0fp0dzybujqa3:http://: 33979755461311130649652613022723076152034308638567944589204199480545055422557

Förnamn: Souheil

Efternamn: Bcheri

Användarnummer: 690202-6203

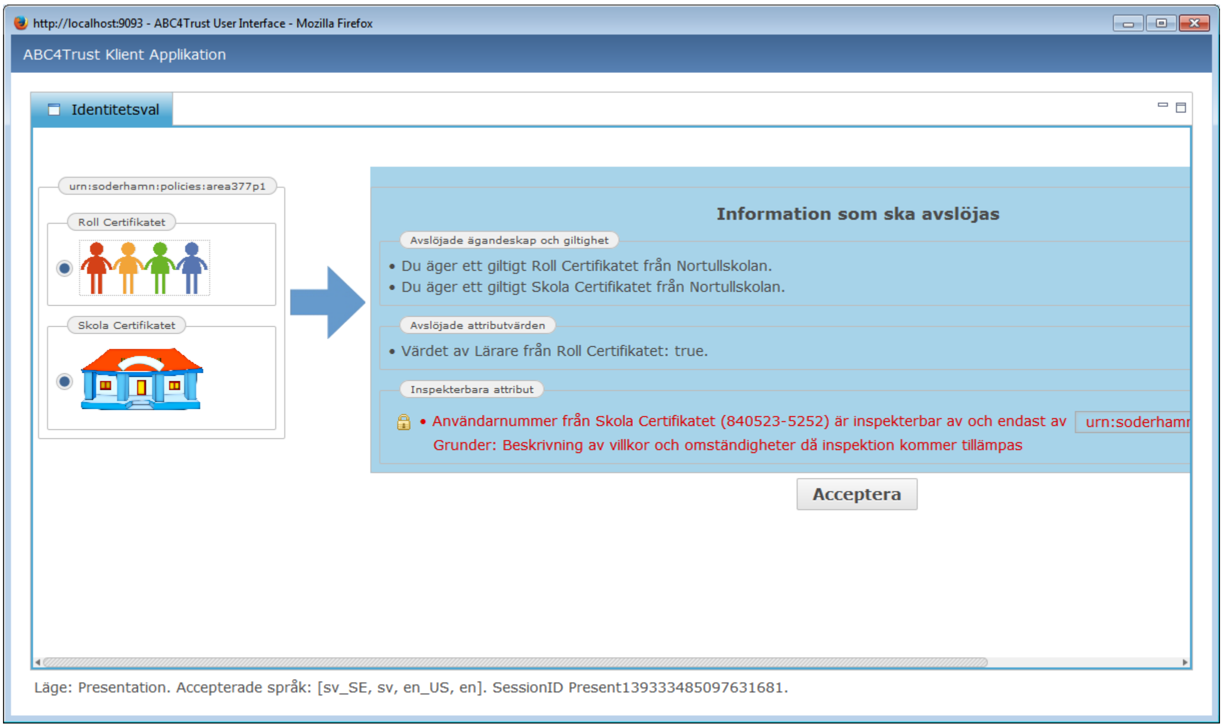
Kön: male

Skola: Norrtullskolan

Figure 5: Credential Viewer

2.2.1.1.2 *Identity Selector*

The Identity Selector is the only needed interaction that a User has to deal with in the User ABCE. It shows the issuance policies and the presentation policies sent to the User ABCE so the User can make an informed choice as to whether she wants to accept the conditions. It also is the only place a User can trust what she sees. A malicious website could show that the User only reveals possession of a credential, where in reality she might give away everything she knows. The Identity Selector prevents this as the content is known to come from a trusted source. An example of the Identity Selector is shown in Figure 6.


http://localhost:9093 - ABC4Trust User Interface - Mozilla Firefox
ABC4Trust Klient Applikation
Identitetsval
urn:soderhamn:policies:area377p1
Roll Certifikatet
Skola Certifikatet
Information som ska avslöjas
Avslöjade ägandeskap och giltighet
Avslöjade attributvärden
Inspekterbara attribut
urn:soderhamn
Acceptera

- Du äger ett giltigt Roll Certifikatet från Nortullskolan.
- Du äger ett giltigt Skola Certifikatet från Nortullskolan.

- Värdet av Lärare från Roll Certifikatet: true.

- Användarnummer från Skola Certifikatet (840523-5252) är inspekterbar av och endast av
  Grunder: Beskrivning av villkor och omständigheter då inspektion kommer tillämpas

Läge: Presentation. Accepterade språk: [sv_SE, sv, en_US, en]. SessionID Present139333485097631681.

Figure 6: Identity Selector Example

2.2.1.1.3 *ABCE*

The ABCE is responsible for gathering everything it needs in order to create proofs. It has to receive something from the browser plugins, contact the smart card and crypto engine and the User herself in order to be able to create the XML needed to pass a proof on to the browser plugin, which then in turn passes this on to the asking website. The ABCE is written in Java.

2.2.1.1.4 *Crypto Engines*

The crypto engine can currently support either Idemix or U-Prove and is responsible for doing cryptographic proofs. Contrary to Idemix, which runs in Java, U-Prove for Söderhamn runs in .NET. This means that there is no direct way for Java to access the U-Prove API. This was remedied by using a local web service to communicate with it.

2.2.1.2 Integration

Once a new ABCE was released, all that had to be done was change to the correct new version in the pom.xml file, which in turn fetched the new ABCE from the repository and injected it into the installer. If no interfaces were changed, that was all the work for integration besides of course testing that it now works with the new features and/or bug fixes included in the new ABCE. If, however, the interfaces changed, adjustments were required. Usually though, it was a small matter of adding or removing a parameter to a method. The idea is that the switch of ABCE should have no visible effect other than addition of features or bug fixes for the Users.

The new ABCE was then extensively tested for all thought of use cases and corner cases. This includes testing with both plugins and all scenarios with both the Issuer and the RA. It also includes testing the hardware smart card initialization and functionality.

2.2.1.3 Deployment

The User Application which also includes the two browser plugins was provided to EDOC by WP4 as an executable installer file. The installer file (from now called Installer or User Application Installer) was uploaded and made publically accessible at the School Portal by EDOC as to make it easy to download it by Users. Each User participating in the pilot has to download and run the Installer. The download is performed by clicking on link (https://ra.abc4trust.se/Content/install-hardware.exe) that is located in the help section of the School Portal.

During the 1st and the 2nd rounds of the Söderhamn pilot EDOC did install the User Application on all the school computers. Users did also download and install the User Application to their computers at home. This was done by the Users themselves with support by the User Manual or by contacting EDOC.

One major problem encountered during the preparations for the 1st round of the pilot was the firewall settings of the school network. Those settings did not allow communication using some ports that are necessary for the pilot. This problem could not be easily solved by WP5 and the solution to fix the problems was to change the firewall settings. A request to open (allow) port 8443 and 8444 was sent to the school network administrator. The request was accepted and the firewall settings of the school network were changed accordingly. Table 1 shows the different URLs used by the pilot and which ports were previously opened at the school network and which ports were opened during the time while both rounds of the pilot were running.

FQDN	Previously allowed	Allowed for the pilot
www.abc4trust.se	80, 8080, 443	80, 8080, 443
portal.abc4trust.se	80, 8080, 443	80, 8080, 443
ra.abc4trust.se	80, 8080, 443	80, 8080, 443
idm.abc4trust.se	80, 8080, 443	80, 8080, 443, **8443, 8444**
revauth.abc4trust.se	80, 8080, 443	80, 8080, 443, **8443**
abce.abc4trust.se	80, 8080, 443	80, **8080**, 443

Table 1: Firewall Settings of the School Network

During the installation process the Installer checks all dependencies to Java and other necessary components and services and it also checks if there is an old version of the User Application already installed on the computer. If there is an older version installed the Installer will remove the old version before installing the latest version.

The User interface of the installation is performed in wizard-style and includes the following steps:

- Choose language;
- Confirm update if the User Application was previously installed;
- Progress of the download and installation process;
- The final screen (successful installation or not).

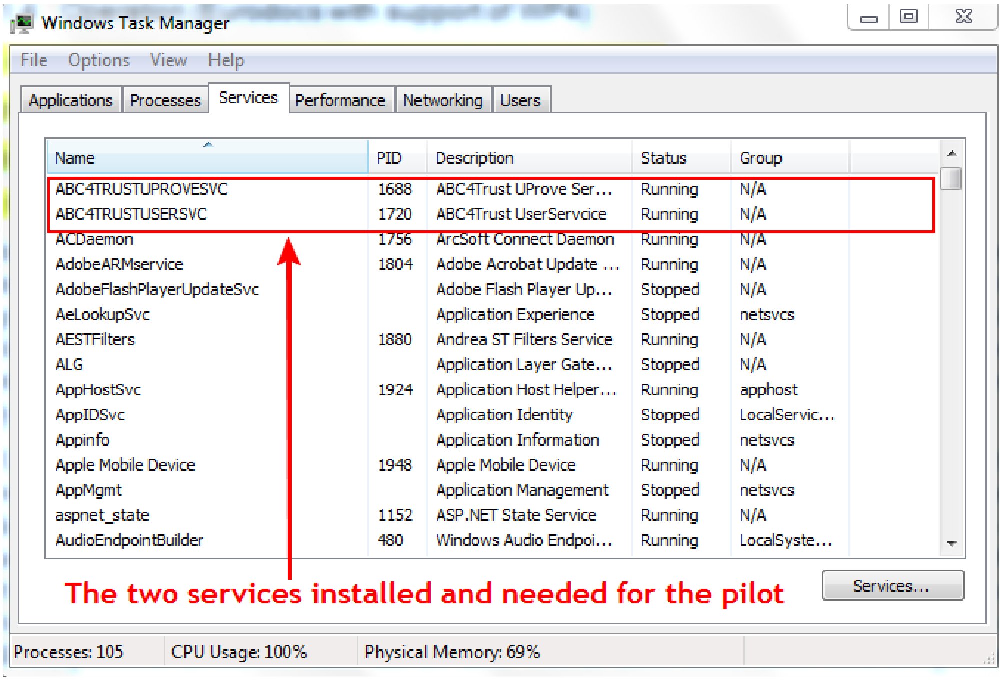
After that the User Application is installed on the computer the User service and the U-Prove Windows services appear in the Services section of the PCs control panel. The services are per default configured as auto-start.

Please note that the Installer installs a version of the User Application and of the browser plugins that are pilot-specific and can’t be used by other pilots or in any other project. They are created from the Söderhamn pilot’s branch of the source code and include many pilot-specific parameters. It’s of course possible to create new Installers which include parameters for other pilots, projects or customers.

2.2.1.4 Operation

During the operation of the 1st round of the pilot EDOC forwarded all reported problems and request for improvements to WP4. Some errors caused delays and other errors caused the computer to freeze completely which required a reboot of the computer. WP4 made all the bug fixes and after fixing all problems that were discovered during the 1st round of the pilot EDOC prepared for the 2nd round by reinstalling the latest and improved version of the User Application. The school computers were ready to be used by pupils and teachers. Users began to use the Restricted Area Application.

The User Application was very much improved and more stable during the 2nd round of the pilot. But we encountered a problem at the beginning of the 2nd round. This problem was caused by a random error inside the User Application which caused hanging of the services and required a reboot of the computer. The error was discovered and a fix was patched to the Installer binaries by WP4 very shortly after the incident. The reported error was related to locks of the U-Prove service. The solution was to implement U-Prove as a separate Windows service. Figure 7 shows the two services that are installed and are needed for the pilot. The User Application was once again updated on all computers (laptops) at the school. After this fix there were no more errors discovered or reported by the Users.



Windows Task Manager
File Options View Help
Applications Processes Services Performance Networking Users
Services...

Name	PID	Description	Status	Group
ABC4TRUSTUPROVESVC	1688	ABC4Trust UProve Ser...	Running	N/A
ABC4TRUSTUSERSVC	1720	ABC4Trust UserServcice	Running	N/A
ACDaemon	1756	ArcSoft Connect Daemon	Running	N/A
AdobeARMservice	1804	Adobe Acrobat Update ...	Running	N/A
AdobeFlashPlayerUpdateSvc		Adobe Flash Player Up...	Stopped	N/A
AeLookupSvc		Application Experience	Stopped	netsvcs
AESTFilters	1880	Andrea ST Filters Service	Running	N/A
ALG		Application Layer Gate...	Stopped	N/A
AppHostSvc	1924	Application Host Helper...	Running	apphost
AppIDSvc		Application Identity	Stopped	LocalServic...
Appinfo		Application Information	Stopped	netsvcs
Apple Mobile Device	1948	Apple Mobile Device	Running	N/A
AppMgmt		Application Management	Stopped	netsvcs
aspnet_state	1152	ASP.NET State Service	Running	N/A
AudioEndpointBuilder	480	Windows Audio Endpoi...	Running	LocalSyste...

The two services installed and needed for the pilot

Processes: 105 CPU Usage: 100% Physical Memory: 69%

Figure 7: Services required by the User Application

During the 2nd round of the pilot, an EDOC representative was available at the school every working day between 9 and 11 am to help Users with troubleshooting and usage of the system. In addition to that, a hotline phone number was also distributed in case the Users have trouble installing or using the User Application from home.

The default installation path of the User Application is

- “C:/Program Files (x86)/ABC4Trust/User Client”.

The following is a list of some important files that can be found under the installation path:

- abce.log – A log of all events and activities of the User Service
- run.log – An event log of starting the User Service under the User Application Windows service
- shutdown.log – An event log of shutting down the User Service under the User Application Windows service
- uprove-run.log – An event log of starting U-Prove service under the User Application Windows service
- uprove-shutdown.log – An event log of shutting down U-Prove under the User Application Windows service

The usability of the User interfaces of the User Application was significantly enhanced for the 2nd round of the pilot compared to the version used in the 1st round of the pilot. The new design was matching the needs of the pilot and was showing enough details to the Users. During the 2nd round of the pilot EDOC received some comments and suggestions to make some changes and improvements (as to make smaller icons, remove some text that is too technical or too complicated, to compress the size of the Identity Selector so that the Users do not have to scroll to the side). All those improvements are taken into consideration and some may be implemented in future releases.

2.2.2 User Smart Cards

The smart card component consists of either a software smart card, which can be used for debugging purposes only, or a hardware smart card along with a smart card reader. The smart card is responsible for storing User sensitive data, including a personal secret key, which only the card knows and cannot be extracted in readable form. The card is needed in order to complete the cryptographic protocols since this it is the only component that uniquely identifies the User and cannot be faked by others.

The hardware smart cards were probably the most problematic development area of all. While the software on the machines could be easily updated, this was not the case for the cards. Thus, it was impossible to distribute them before being somewhat sure that the cards worked as they should. New bugs were found often though, so it was not easily discernible when the time to release them was appropriate.

During the project two different kinds of cards were used. The first, ZeitControls BasicCard, had issues with the amount of space available. The Söderhamn pilot demanded that several credentials as well as several aliases must be stored. It also had problems with speed, and malfunctioned at seemingly random times. The quality of the support tools however, was quite good.

Because of these issues, it was decided to change to another type of card, i.e. MULTOS (see http://www.multos.com/), which had a larger storage capability and was faster. This was indeed the case, but it was not without issues of its own. First of all, the simulator for the card provided by the manufacturer and the card does not map to the API. Also, the outcomes of certain operations were handled differently, and the simulator was in general too buggy to be of any real use. Thus, debugging had to happen on the card, slowing down the process considerably. As if this was not enough, the underlying algorithms and available operations contained bugs as well. Also, the platform of MULTOS is in general of really poor quality. This includes the MUtil.exe tool that is used to load an application onto the card. The support for this was almost non-existent.

The process of loading an application onto the cards was somewhat simple. First the application should be loaded onto the smart card. This was done by the tools provided by the manufacturers. Then the card was to be initialized with the different parameters in the pilot ecosystem. That is, the Issuer parameters, system parameters etc. It takes a while to initialize a card, so this one should keep in mind if there are a lot of cards to initialize. The result of initializing a card is an entry in a file, which lists the card ID along with the PIN and PUK code of the card. These are to be handed over to the pupil with a reminder to change the PIN code using the browser plugin.

2.2.2.1 Implementation

In order to use the smart card, the card had to be initialized with the correct parameters such that it understands the 'ABC Systems'. For this, a Smart Card Initialization Tool was implemented, which loaded onto the card all the needed parameters. This also included whether the card should act as a U-Prove card or as an Idemix card[2].

The smart card handles storage of everything the User has to carry with him in order to prove she is who she claims she is. This includes, but is not limited to, the credentials of the User, the card secret that only the card knows and is unique for the card and Issuer parameters. When asked, the card will, with a correct PIN, produce its part of the proof. This part consists of the cryptographic evidence, which requires the card secret.

From the User Application point of view, there is no direct interaction with the smart card. You can view the credentials located on the card, but this call goes through the ABCE.

[2] The card needs to know this, since the computations are different depending on the CE.

The pilot required a ‘backup and restore’ feature in case of stolen, broken or lost cards. This posed several issues. The main problem is that if this is done incorrectly, an adversary could claim his card stolen, gets a new one and then is able to clone his card using the restore feature if he previously backed up his card (or someone else’s card).

For Söderhamn, the backup story is slightly more complicated than for Patras. Here, the pilot must ensure that pupils do not lose their aliases. This amounts to being able to backup the blob-store containing the aliases and the device secret. There was talk about backing up credentials as well, but since the device secret can be restored, a User can just obtain new credentials from the Issuer after completing the restore procedure.

The paradox the Söderhamn pilot imposed on the developers was to prevent someone from cloning a card and at the same time allowing the use of restored aliases. Say an adversary “lost” the card, and then got a new one with the same device ID enabling him to perform a restore. Since the device secret would be the same as the old card after restoration, he could then not only obtain new credentials but also successfully prove ownership of the restored aliases (recall this is the intention of backup and restore). To be able to block the old card (and therefore prevent cloning of the card), the use of the old device secret would have to be disabled, but this would have the side-effect, that the User cannot prove ownership of the restored aliases any longer. So essentially restoring the old device secret on the new smart card will always enable an adversary to clone cards, but only as many as the school gives him. This problem cannot be easily fixed apart from not doing backup at all which was rejected by the consortium. Instead, the model where clones are a possibility was finally chosen.

Another potential issue is that a tech-savvy pupil could copy the credentials and aliases of his card, as they are only protected by the Users PIN code, and give them to someone else. This means that i.e. a guardian could give access to a non-guardian for a room protected by a presentation policy requiring the guardian credential. However, exchanging credentials and aliases in order to impersonate other Users is not possible since the device secret must be the same in order to do proofs with the credential or prove ownership of an alias.

2.2.2.2 Integration

During integration a lot of bugs were found, which were corrected using a combination of debugging with the software smart card and looking at the logs produced by the ABCE. Of those mentionable, the CE’s did not agree on how to compute the hash function. The card was originally only designed to support the Idemix way of hashing, and it took a great effort to realize that the card needed a way to hash the same way as U-Prove does it. When using the ABCE with a U-Prove card, the card has to talk to both Java and .NET. This is done via two different drivers, which apparently causes the lock one driver has on the card to sometimes not be freed, causing the program to crash.

Another hard to locate bug was an “out-of-RAM” error, which happened on the BasicCard only. It happened at seemingly random points in computation, and was never fixed completely. This was part of the reason for switching to the MULTOS card. The MULTOS card, however, was not free of errors at all. In the end, a list was compiled with all the limits and errors made by MULTOS. The worst one was a problem that occurred in this project when a certain bit was set during a modular multiplication. This amounted to a seemingly random behavior from the card. Luckily, and strangely, the same mistake was not repeated if a multiplication was followed by a modular reduction call.

2.2.2.3 Deployment

For a successful start of the pilot EDOC was preparing smart cards to be issued to all Users that have signed the legal consent form. 34 smart cards were prepared for the 1st round and 381 for the 2nd round.

Preparation of smart cards includes the following steps:

- The customization of the smart card by printing was done using a special card printer. A smart card number and the ABC4Trust logotype was printed on all smart cards (see Figure 8)
- The installation of the ALU (adding the OS to the card using MUtil.exe)
- The initialization of the smart card to create a pseudonym and to generate the PIN/PUK codes
- The upload of User data to the IdM Database, match the card pseudonym with the IdM record and save the User credentials prepared by the IdM Portal to the smart card

Due to the large number of participants in the 2nd round of the Söderhamn pilot a decision was made that EDOC would download the credentials to the smart cards before handing out the cards to the Users. This required that NSN had to add more functionality to the School Registration System. In the 1st round all cards were prepared by EDOC except for the last step which is to download credentials which was done by the Users. In the 2nd round EDOC made all the preparations and did also download the credentials.

The teachers were the first group to receive their smart cards and card readers at the school. The teachers asked to have some time to get familiar with the RA System and to prepare some Restricted Areas to be used by the pupils. The next group to receive their smart cards was the pupils. EDOC and SK administrators distributed the smart cards, PIN/PUK codes and smart card readers to the pupils in each class separately. The last group to receive their cards was the guardians. The smart cards and the smart card readers of the guardians were given to the pupils to bring home and give to their parents. The guardians did receive their PIN/PUK codes with a letter sent directly to their home addresses. This process was done within 2 weeks.

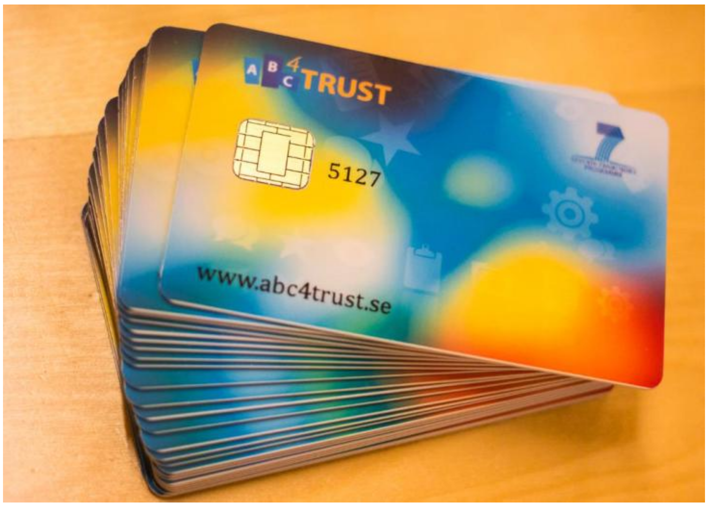


Figure 8: The final design of the smart cards

2.2.2.4 Operation

During operation the User enters her smart card into the smart card reader which is connected to the PC. No contactless smart card readers were used in the Söderhamn pilot. The User navigates to the School Portal and clicks on the “Go to the Restricted Area Application”. After this a window pops up asking the User to enter her PIN-code which allows the RA Application to communicate with the User’s smart card. If the User enters the correct PIN-code the RA Application can begin communicating and interacting with the smart card, the smart cards operating system and the Privacy-ABC components including the credentials which are stored on the smart card.

During normal operation of the smart card, which in this case means operations such as entering (log-in to) different Restricted Areas and conducting chat communication, counselling etc., non-revocable credentials that reside on the smart card do not change[3]. The credentials can only be changed using the School Registration System (the Issuer).

The RA Application is making use of aliases that are stored on the smart card as a way to avoid linkability. This means that this content does change during normal operation. This happens when the User creates a new alias. Even though there is a limit in the card memory space where the aliases are stored, there have not been any reported problems from any User related to the lack of space on the card.

During the 1st and the 2nd rounds of the pilot the smart cards were working with no fail reports. Some reports coming from the Users during the operation phase were mostly related to problems such as resetting of the PIN code and other functions not classified as failures of the smart cards.

2.2.3 Tray Application

The pupils in the Swedish pilot are mainly using the school computers (laptops) to log into the Restricted Area Application when they are at the school. As those computers may have many other programs installed, pupils sometimes complain about the school computers being too slow. This was also the situation before the pilot started. During the 1st round of the pilot, after the User Application was installed on the school computers it was difficult to know if the computers became even slower because of the User Application or because of other programs installed on the computer.

As to isolate this performance problem and in order to improve the user experience and make it easy to debug problems a decision was taken to develop a Tray Application. The Tray Application, introduced in the 2nd round of the pilot, is a Windows program that can be installed on the computer acting as a tray icon (indicator) that shows the status of the User Application and that offers additional functionalities such as starting, stopping or restarting the local services. The Tray Application was also designed to be used to launch updates of the User Application.

2.2.3.1 Implementation

The Tray Application was developed using C# programming language using .NET 4.0 Framework and works on both Windows 7 and 8 based computers. The icon of the Tray Application, which is shown in the tray, changes color depending on the status of the local services. If the local services are running and responding, the icon is white. If the services are running but not responding immediately, the icon turns yellow, which means that the services are busy. A red icon indicates that the local services are switched off or not responding because they are hanging. The Tray Application is using local HTTP REST requests to check the status of the User Application.

Some of the Tray Application functionality such as to start, restart or stop the services and the functionality to update the User Application Installer was developed but only used during a limited time period at the beginning of the 2nd round of the pilot. As those functionalities required administrative rights for the application to control other services on the computer, this was considered to be a potential security threat. The school could not give the administrator rights to pupils. Since a work-around was not found, those functions were not included in the final release version of the Tray

[3] In the Söderhamn pilot, only the school credential is revocable. So in case the User loads new revocation information, her school credential will be updated but the attribute values of this credential will not change. Non-revocable credentials will not change even if new revocation information is downloaded.

Application. The only functionality that could be used when EDOC was not present at the school was the ‘show status’ functionality of the User Application.

2.2.3.2 Integration

The integration of the Tray Application was done by interacting with and using some specific functions offered by the User Application. A specific function offered by the User Application is to check the status by calling the URL http://localhost:9500/pilot-soederhamn/issue/status. It returns the char value ‘0’ or ‘1’, depending on if the service is busy or not. The response status code is “200 OK” if the service is up and running. All other codes, including the code “404”, mean that the service is down.

To check if the User Application needs to be updated it was planned to add a function that would query for the current version of the User Application installed on the computer and compare it to the latest version on server, but this was not integrated in the version of the Tray Application that finally was released and used in the pilot.

2.2.3.3 Deployment

The Tray Application was deployed and installed on the school computers with the purpose of visualizing the status of the User Application during the start of the 2nd round of the pilot. Since it was used for testing only and internally, the application was launched on the computer with a preinstalled .NET framework 4.0 via executable file. A link to the Tray Application was added to the auto start list.

The Tray Application can also be started on the PC by double clicking on the executable file.

2.2.3.4 Operation

The Tray Application was installed on all laptops at the school and it was mainly used at the start of the 2nd round while testing and debugging the system at the school. The Tray Application in combination with the User Application logs helped in troubleshooting, localizing and solving some of the problems that were faced at the beginning, i.e. computer freeze and service hanging. The Users did not use the Tray Application on their private computers at home since debugging could be performed on the school machines.

2.3 Inspector

The Inspector block in the Söderhamn pilot consists of the following parts:

- A smart card reader (same as other Users had);
- A smart card which is prepared in a special way to contain the inspector key
- A PC which has the Inspector Application installed
- Internet connection for the Inspector Application to download presentation tokens

The Privacy-ABC inspection functionality was introduced in the Söderhamn pilot to guarantee the physical and mental safety of each participating pupil. Inspection means the revelation of the pupil’s identity in certain predefined emergency situations (called inspection grounds). Such inspection grounds can be:

- Situations implying a severe threat to the life, or the physical/mental integrity of a person
- Situations demanding an intervention according to the Norrtullskolan policy against discrimination and degrading treatment. This policy can be found at http://www.soderhamn.se/download/18.12494e5813c05809128e67/Norrtullskolans+plan+mot+diskriminering+och+kr%C3%A4nkande+behandling.pdf for further reading.
- An existing court order or other administrative request binding for Norrtullskolan or Söderhamn Kommun

In case a participant (pupil, legal guardian, or school staff) reports an emergency situation, an assigned School Inspection Board will investigate the matter. This board will decide if inspection is required by comparing the inspection grounds embedded into the presentation token with the current situation. In case the comparison is positive, the School Inspection Board triggers a formal inspection process, forwarding the request to an assigned Inspector. This Inspector will perform a double check and is equipped with the technical capability to reveal the identity of the pupil. The whole process will also be protocolled. This procedure guarantees that no single entity is able to arbitrarily spoil the privacy of the pupil and that the identity is revealed in emergency cases only. If the School Inspection Board decides that the situation does not require the identification of the User, it either closes the case or may decide to delete the content and/or write a warning to the respective Restricted Area.

2.3.1 Inspector Application and Inspector Smart Cards

2.3.1.1 Implementation

The Inspector Application is based command line tools supplied by WP4 wrapped into a custom GUI application. This custom GUI is called the ‘Inspector Wrapper’.

The tools supplied by WP4 consist of a Setup Tool and the actual Inspect Tool:

- The Setup Tool is used to generate the Inspector key pair and to store the private key on a smart card.
- The Inspect Tool which runs the actual inspection on a presentation token supplied by the RA Application after the Inspector confirmed that inspection grounds have been met.

Both tools are implemented in Java as executable jars and are both adopted directly from the WP4 sample without any changes.

The custom Inspector GUI application (Inspector Wrapper, see Figure 10) was developed to make it easier for a non-technical Inspector to retrieve presentation tokens, use the smart card and inspect messages, which were approved for inspection by the School Inspection Board.

Presentation tokens are downloaded from the RA Application after the Inspector logs in to the Inspector Wrapper via username and password. The Inspector Wrapper interacts thereafter with the Inspect Tool using the command line interface.

The Inspector Wrapper was developed for the .NET platform. The installation package contains both the Inspect Tool and the Inspector Wrapper.

2.3.1.2 Integration

Setting up the Inspector consists of a series of steps which could be handled by different roles – for example a developer who does the configuration and builds the tools and the Inspector who performs inspection on demand.

- First step is to setup the configuration of the Inspector by defining the URI of the Inspector key, receive the Credential Specifications which could be involved in inspections from the issuer and specify URI and XML files in a property filed included in the tools.
- Second the Setup Tool needs to be built and have configuration and XML files included.
- Hereafter the Setup Tool could be shipped to the Inspector who will then generate the Inspector key pair and store the Private Key on a smart card.
- The Inspector public key must then be handed over to the Verifiers plus the developers who are in charge of creating the User Application Installer.
- Next step is to build the Inspect Tool including configuration, XML files and the Inspector public key.

Naturally, the Inspector could also perform all these tasks without a developer being involved.

2.3.1.3 Deployment

In order to prepare the Inspector smart card EDOC has used the guide provided by WP4 and described in 2.2.2.3 plus the above mentioned tools, it included:

1. Installation of ALU of the same version as all other cards used in the pilot
2. Initialization of the smart card and the generation of PIN/PUK codes
3. Generation of the Inspector's private key using the Inspector Setup Tool
4. Copying the private key to the Inspector smart card (see Figure 9) also using the Setup Tool

One difference between the Inspector smart card and the smart cards of the Users is that the Inspector smart card does not contain any credentials.

During the process of preparation and integration of the smart card no errors or problems were discovered. The key was successfully generated and copied to the smart card and became ready for usage.

The Inspector key is written to the smart card using the Inspector Setup Tool before handing over the smart card to the Inspector. The Inspector in the Söderhamn pilot is the schoolmaster. Deployment of the Inspector Application in this case is a simple task of copying and unpacking a zip file to the computer that will be used by the Inspector (the schoolmaster).

Figure 9: Inspector Smart Card

If no content was reported and chosen for inspection by the School Inspection Board, the Inspector Wrapper will not display any inspection requests. Once content was reported and confirmed for inspection by the School Inspection Board, a line appears in the Inspector Wrapper and the Inspector is able to inspect the targeted token and reveal the PUN of the person who posted the reported content.

To inspect content approved by the School Inspection Board, the Inspector has to launch the Inspector Wrapper and type in login username and password as shown in Figure 10.

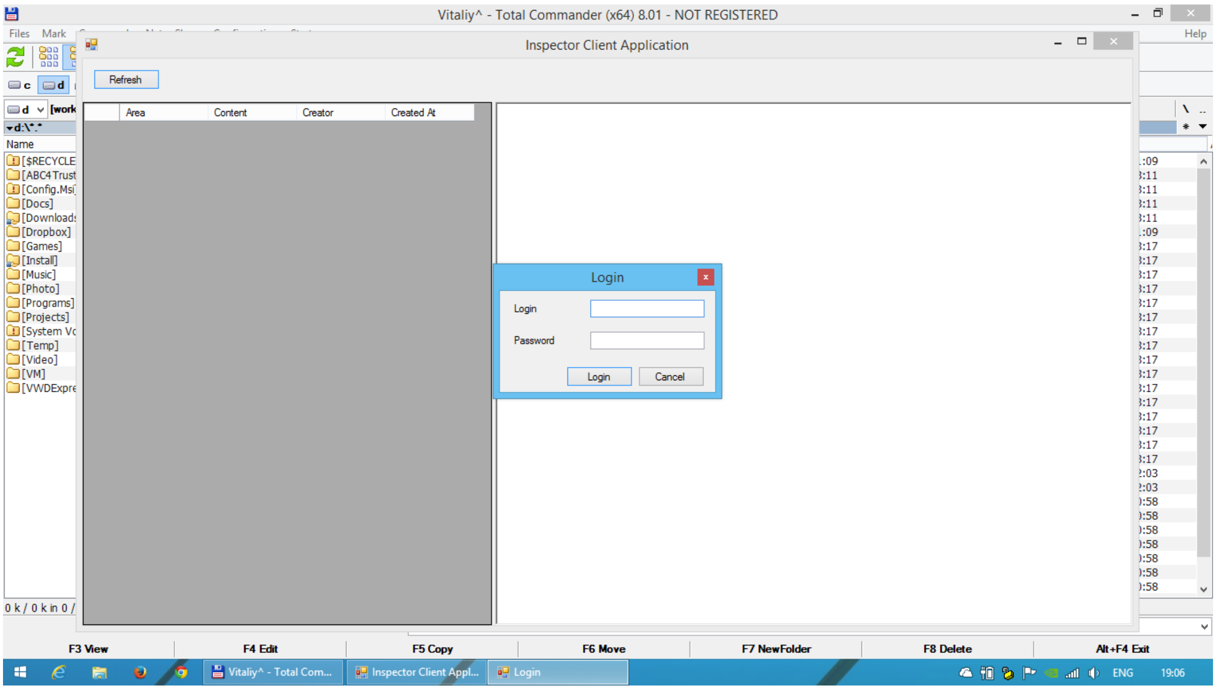
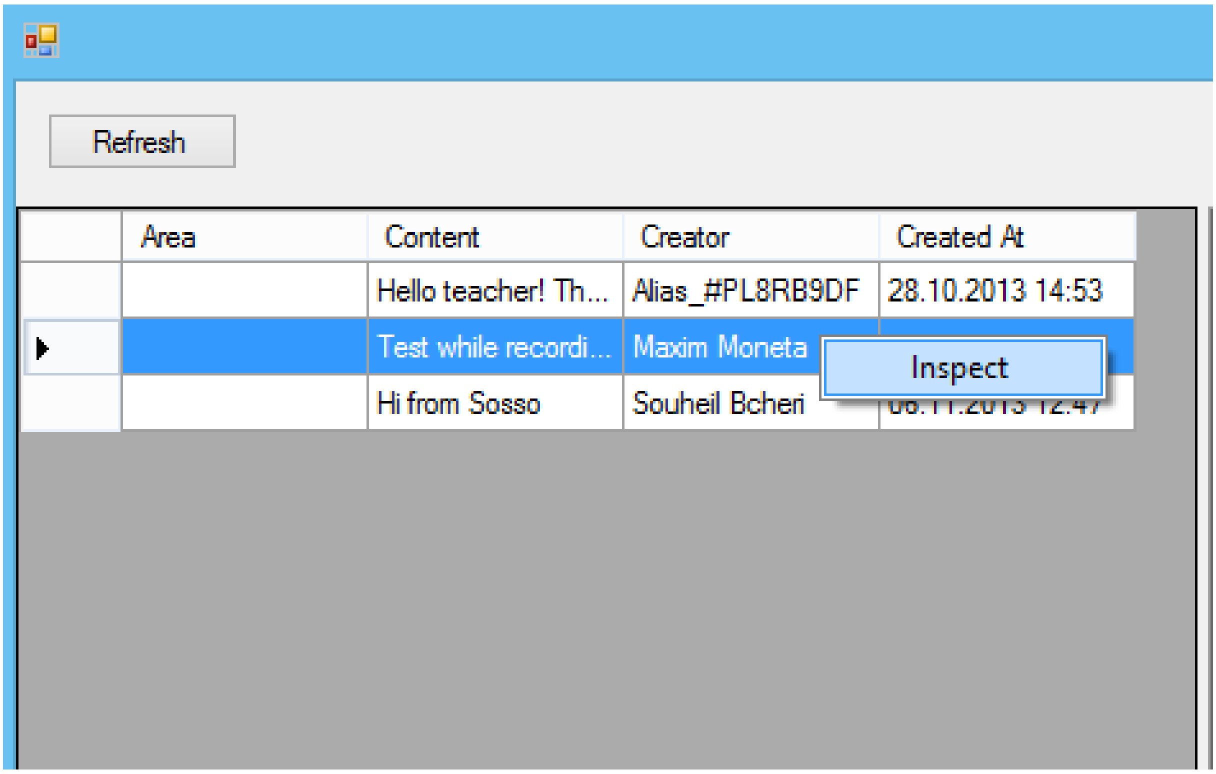


Figure 10: Inspector Wrapper

After successful login the Inspector will see a list of reported content which has been approved for inspection (see Figure 11).



	Area	Content	Creator	Created At
		Hello teacher! Th...	Alias_#PL8RB9DF	28.10.2013 14:53
▶		Test while recordi...	Maxim Moneta	
		Hi from Sosso	Souheil Bcheri	06.11.2013 12:47

Figure 11: Inspect Tool - Inspect reported Content approved by the School Inspection Board

The Inspect Tool and the Inspector Wrapper were deployed to the school Inspector computer and the Inspector smart card was handed over with generated secret key on it. Testing of smart card with key was successful. Information was decrypted correctly.

2.3.1.4 Operation

No failures or comments were reported during pilot time because of no real cases of usage of Inspection – no content was reported which required disclosing its author.

2.4 School Registration System

In the very beginning of this pilot, NSN decided to deploy the ABCE and the crypto engines on a Windows server. This entity was called ‘IdM ABC System’, an independent web service within the School Registration System as depicted in Figure 3. Hosting the ‘Identity Management’ applications and the ‘IdM ABC System’ on physically separated servers had several advantages. When this project started, the licensing issues were still unclear. In order not to risk any licensing impacts in NSN’s proprietary implementations of the IdM Application and of the IdM Portal, the libraries of the ABCE and of the crypto engines were not added to the builds of the IdM. The other main advantage of this loose coupling was that upgrading the ABCE did not lead to a downtime of the IdM. On top of that, a Windows server supports the installation of .NET, which is required by the U-Prove crypto engine. The IdM requires only Java, so it can be deployed in its native Linux environment on a separate server.

In the course of the ABC4Trust project, it became very clear that the School Registration System needs additional tools/applications in order to support the administrators of this system in their daily tasks. Due to the large number of participants, a tool of registering authorized smart cards (IdM Smart Card Registrar) and a tool for filling in the IdM Database (IdM Mass Provisioning Tool) were quickly identified as being necessary enhancements even though they were not taking into account in [D51]. On top of that, in the 2nd round of the Söderhamn pilot, the IdM Admin GUI was added to the architecture. This tool enabled administrators to modify attribute values of Users during the operation phase of the pilot. If a specific attribute was already certified in a credential, the revocation of this credential was automatically launched. Finally, since administrators were allowed to modify and add attribute values, a Syntax Checker was developed which compared the entries with a specific REGEX rule or with a list of allowed values.

Introduction of the message flow

The browser-specific JavaScript embedded into most GUIs of the School Registration System is downloaded by the User's browser. The browser plugin on the User-side receives therefore where to get the issuance policy, where to send the issuance message, where to get the presentation policy and finally, where to send the presentation token. On the School Registration side, the aforementioned locations point to web servlets. The data that is transported to the School Registration System is always an XML body transported via an HTTP POST or GET. In case the cookie forwarded by the User is a registered cookie, the School Registration system allows all issuance and verification services. If the cookie is not registered, reIssuance is only accepted and no prior login is required. The latter will only work for the User, if the User is able to meet the issuance policy mapped to the specific location she is visiting and to reveal all parameter values stored on that specific U-Prove credential. The XML body is extracted by the School Registration System using standard request and response methods. After extracting core information from the body, the XML data is forwarded to the 'IdM ABC System', a web service hosted by an Apache Tomcat container. The specification of these services offered by the 'IdM ABC System' are retrieved by pointing the IdM frontend services (e.g. the IdM Portal) to the WSDL file of the 'IdM ABC System' using state-of-the-art development tools (e.g. NetBeans).

2.4.1 IdM Application

2.4.1.1 Implementation

The IdM Application is a subset of NSN's 'IDM Solution' which had to be customized and integrated into the pilot scenarios (see Task 5.7 of [DoW]). This IdM Application is a SAML server which provides 'single sign on' functionality to Users requesting access to web services i.e. relying parties. In order for 'single sign on' to work, the relying parties must have federated with the IdM Application. This federation takes place offline. The IdM Application only accepts authentication from known and trusted relying parties.

The IdM Application is connected to the IdM Database which stores the private attribute values of Users. In the ABC4Trust project, the IdM Application authenticates Users via 2 methods (see also Figure 12):

a) standard username and password
b) via Privacy-ABC token

Method a) is allowed only when the User logs in to the IdM for the first time and when her scope-exclusive-pseudonym has not yet been mapped to her identity.

After logging in to the IdM via method a), the User is requested via the GUI to register her card. From then on, method a) is disabled and method b) is enabled.

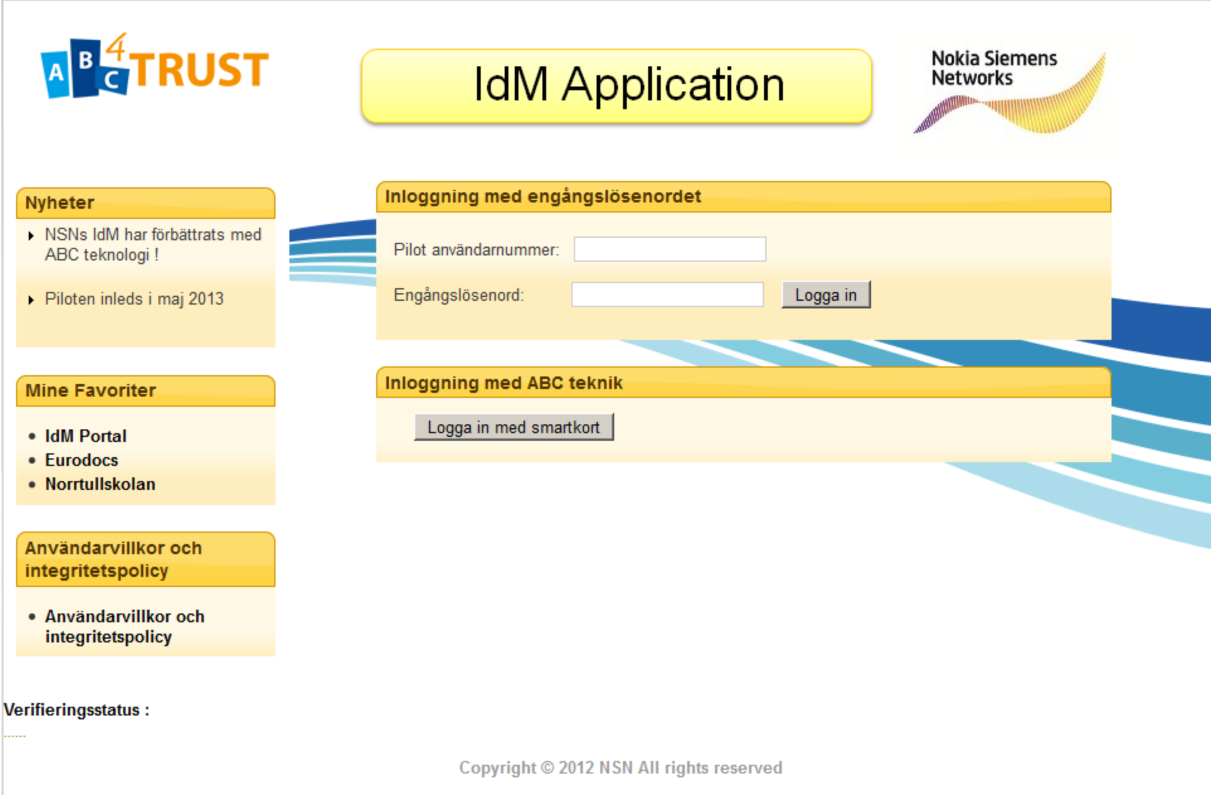
The main difficulties in the task of adapting the IdM Application to fit into the ABC4Trust pilot architecture were to

- delete or inactivate functionality of the ‘IDM Solution’ not required by the pilots
- adapt the IdM Database schemas to be able to store the new attributes required by the pilot (i.e. the attributes defined in the credential specifications and other attributes required by the applications themselves to cope with the new functionalities)
- connect to the ‘IdM ABC System’ to enable Privacy-ABC verification
- re-design the GUI, incorporate ABC4Trust logos, incorporate JavaScripts required for Privacy-ABC verification , support Microsoft’s Internet Explorer
- support Swedish

The IdM Application, combined with the ‘IdM ABC System’, can therefore be considered as Privacy-ABC Verifier as specified in [D51].

The IdM Application incorporates a significant set of servlets, most of which handle the SAML call flow. But in effect only 2 servlets handle the Privacy-ABC traffic:

- verification/policy
- verification/verify



NSNs IdM har förbättrats med ABC teknologi !

Piloten inleds i maj 2013

Copyright © 2012 NSN All rights reserved

Figure 12: IdM Application Login GUI

2.4.1.2 Integration

Since the IdM Application itself does not host the ABCE libraries and CEs, the IdM Application has to forward presentation policy requests and presentation tokens to the 'IdM ABC System'. The IdM Application therefore basically acts as proxy when it comes to handling Privacy-ABC messages.

The IdM Application connects to the 'IdM ABC System' via WSDL file. The IdM Application is therefore a web service client in the scope of Privacy-ABC verification services. When the IdM Application accesses a web service of the 'IdM ABC System', an ExchangeData class containing one or more strings is forwarded in the request.

When a User tries to login to e.g. the IdM Portal, she is forwarded to the IdM Application GUI which requests the User to login via OTP or via ABC technologies. When the browser of the User displays the login page, the UI has already loaded the JavaScript and is therefore already aware of the policy URL and the verification URL. As soon as the User presses the button 'Logga in med smartkort', a specific JavaScript function is executed. The User now visits the policy URL which points to a servlet of the IdM Application. The task of this servlet is to query the 'IdM ABC System' for a presentation policy with a fresh nonce. Please note, that this presentation policy is static except for the nonce. The policy itself is very simple and does not require the User to possess any Privacy-ABC credential. The policy only requires the User to present a scope-exclusive pseudonym mapped to the scope 'urn:soderhamn:registration'. The IdM Applicaton forwards only the policy ID to the 'IdM ABC System' via the ExchangeData class. The policy ID is configurable via properties file. Important to note is that since the Swedish alphabet contains some special characters which are used in the 'FriendlyPolicyName' and 'FriendlyPolicyDescription' fields of the presentation policy, the HTTP response sent back to the User has to be in UTF-8 format. After receiving the presentation policy, the User generates a presentation token (containing the scope-exclusive pseudonym) and offers it to the verification URL. The servlet of this URL has as task to forward the presentation policy alternatives and the presentation token via the ExchangeData class to the 'IdM ABC System' for verification. If the token is valid, the 'IdM ABC System' returns 'SUCCESS' as string and the IdM returns 'HttpServletResponse.SC_OK' to the User. Finally, the IdM Application retrieves the identity of the User by making a lookup in the IdM Database using the verified scope-exclusive pseudonym as handle.

The servlets of the IdM Application are not session-aware and communicate with session-aware parts of this application using a lookup table.

Integration tests of the IdM Application alone are not possible as a SAML client (i.e. a relying party) is needed. The IdM Application requires a Linux server (preferably SuSe) as host. This server must be provided with an LDAP server (preferably OpenLDAP for the IdM Database) and an Apache-Tomcat installation. The SSH server and Java OpenJDK are typically included in Linux but need to be selected during the installation. Since the User will access the IdM Application directly via the Internet, the Apache-Tomcat container must be provided with an X.509 certificate so that the connector can enforce SSL encryption. Finally, NSN's firewall must be configured to allow Internet traffic to this connector.

After setting up the server within NSN's integration testing environment, the IdM Application can be copied into the Apache-Tomcat container. The only test that can be performed without a SAML client is checking if the application powers up without errors. If a SAML client is available (e.g. the IdM Portal) logging in to the IdM Portal via OTP can be verified. If the 'IdM ABC System' is online too, logging in to the IdM Portal with the HW or the SW smart card can be tested.

2.4.1.3 Deployment

In both rounds of the pilot, the School Registration System was installed in one go in the EDOC target site. This chapter will concentrate on the steps that had to be followed for customizing the IdM Application from the integration testing environment to fit into the target site.

EDOC provided an Ubuntu server for the IdM Application. As in the integration testing environment, LDAP and Apache-Tomcat had to be installed.

EDOC made use of the on-board firewall of Ubuntu (UFW). Port 8443 had to be opened in the UFW in order to allow Internet access to the HTTPS port of the IdM Application. The firewall must also allow traffic originating from the 'IdM ABC System'. Without this rule, the 'IdM ABC System' cannot terminate the TCP connections successfully. And finally, the UFW had to be configured to allow NSN's developers access to the SSH server (port 22000) and the LDAP server (port 389).

The IdM Application had to be customized to fit into EDOC target site. Next to adaption the pom.xml to point to the new wsdlLocation http://abce.abc4trust.se:8080/ABC4TrustSystem/abcHandler?wsdl, the abcHandler.wsdl file had to be modified to point to the new SOAP address http://abce.abc4trust.se:8080/ABC4TrustSystem/abcHandler.

With these changes, the IdM Application can connect to the 'IdM ABC System' web services.

Finally, the policy and verify URLs of web page of the IdM Application had to be modified to point to its servlets.

As stated previously, the IdM Application cannot be tested alone. But as soon as a SAML client, e.g. the IdM Portal, and the 'IdM ABC System' are up and running, the complete functionality of the IdM Application can be verified.

EDOC and NSN executed the system tests. Before the operation phase of the pilot, NSN invited to a formal handover of the School Registration System and the Revocation Authority. EDOC thereafter configured their firewall to block traffic coming from NSN. After 'cutting the lines', the pilot entered the operation state.

2.4.1.4 Operation

During the operation phase of the pilot, interaction with the IdM Application was performed using GUI tools i.e. the IdM Admin GUI and the IdM Portal. For these tools, the IdM Application authenticated the Users via SAML.

No IdM Application failures were reported during the operation of the pilot.

2.4.2 IdM Portal

2.4.2.1 Implementation

Both the IdM Portal and the IdM Application represent a subset of NSN's 'IDM Solution'. The IdM Portal is a SAML client and must have federated with the IdM Application to enable the 'single sign on' feature.

The IdM Portal itself does not authenticate Users. It redirects Users to the IdM Application for this task via a SAML '`AuthnRequest`'. If the authentication was successful, the SAML response will contain a unique '`nameid`' of the User. After evaluating this response, the IdM Portal sends a SAML

‘`AttributeQuery`’ directly to the IdM Application (without involving the User’s browser) and is able to receive all attributes stored about the User in accordance with the ‘attribute release policy’ agreed upon during federation.

The following main implementation tasks had to be performed to enhance and adapt the IdM Portal for the deployment in the Söderhamn pilot:

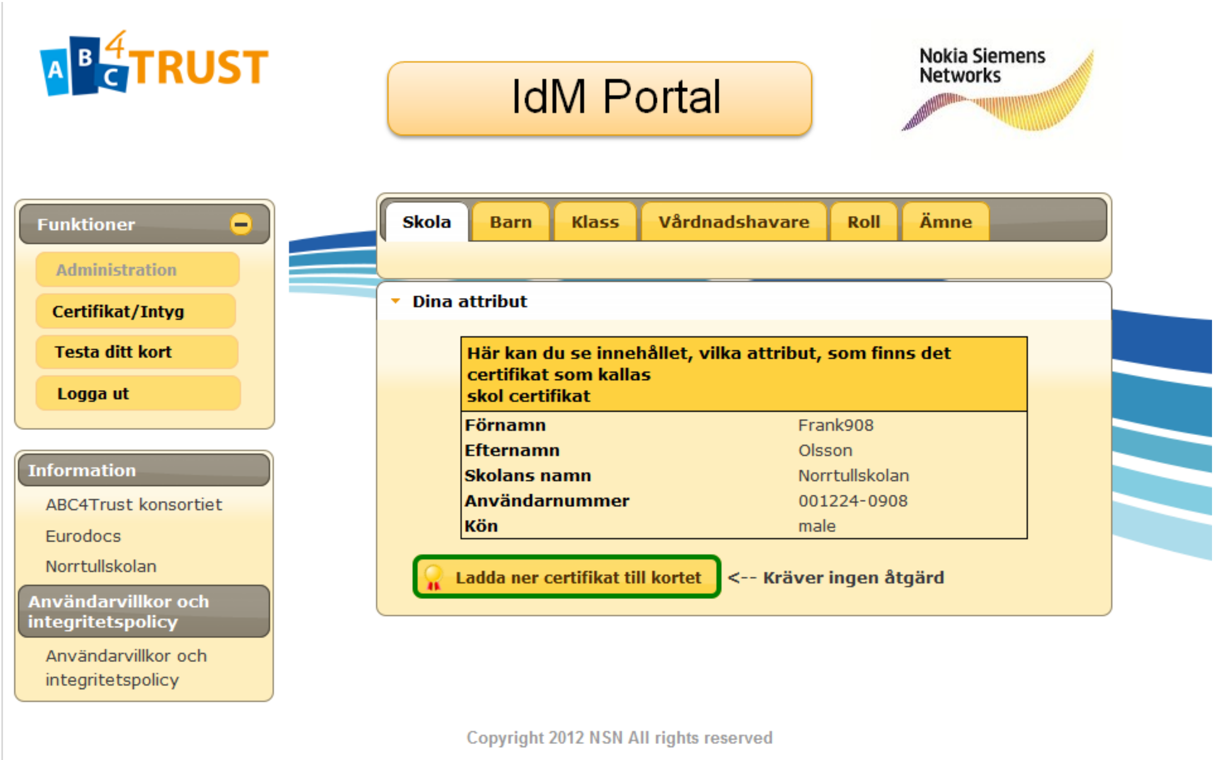
- delete or inactivate functionality of the ‘IDM Solution’ not required by the pilots
- connect to the ‘IdM ABC System’ to enable Privacy-ABC issuance, reIssuance, revocation and verification
- connect to the IdM Database to store the scope-exclusive pseudonym and the revocation handles and to set the oneTimePass flag to ‘disable’
- re-design and enhance the GUI, incorporate ABC4Trust logos, incorporate JavaScripts required for Privacy-ABC issuance and verification, support Microsoft’s Internet Explorer
- support Swedish

The IdM Portal, combined with the ‘IdM ABC System’ is both a Privacy-ABC Issuer, a Privacy-ABC Verifier and a ‘Revocation Requestor’ according to [D51].

The IdM Portal incorporates a significant set of Privacy-ABC servlets:

- issue/start*Cred*
- issue/step*Cred*
- verification/policy
- verification/verify
- test/policy
- test/verify

Cred is a placeholder for the 6 different credentials the IdM Portal can issue (see tabs in Figure 13). So in effect, the IdM Portal offer 16 servlets for handling Privacy-ABC related request. Please note, that the 2 ‘issue’ servlets process reIssuance too.



Här kan du se innehållet, vilka attribut, som finns det certifikat som kallas skol certifikat

Förnamn	Frank908
Efternamn	Olsson
Skolans namn	Norrtullskolan
Användarnummer	001224-0908
Kön	male

Copyright 2012 NSN All rights reserved

Figure 13: IdM Portal GUI

2.4.2.2 Integration

Identical to the IdM Application, the IdM Portal acts as proxy when it comes to handling Privacy-ABC messages. The IdM Application connects to the 'IdM ABC System' via WSDL file. When Privacy-ABC issuance or verification is requested by the User, the IdM Application calls the corresponding web service of the 'IdM ABC System'.

In the design phase of the ABC4Trust project, it was decided not to allow the User to launch revocation directly. The IdM Portal triggers revocation indirectly if the User requests for a credential which has already been issued to the User. In this case, the IdM Portal will offer the revocation handle of the old credential to 'IdM ABC System's revocation web service.

When the IdM Portal receives the attributes of a specific User, it will evaluate the oneTimePass flag. If this flag is set to 'enabled', the IdM Portal will only offer the User the possibility of registering her smart card in the GUI. The GUI that will be presented to the User contains JavaScript pointing to the policy URL and the verification URL. As soon as the User presses the button 'Registrera ditt kort'she will visit the corresponding servlet. The servlet will forward the policy ID to the 'IdM ABC System' in the ExchangeData class. Please note that this policy ID is identical to the policy ID the IdM Application uses for authenticating Users. Therefore, the User will be requested to generate a scope-exclusive pseudonym based 'urn:soderhamn:registration'. The rest of the Privacy-ABC handshake for registering the smart card is identical to the flow described in 2.4.1.1. Important to note is that the IdM Portal will become aware of the scope-exclusive pseudonym during this handshake. If verification is successful, the IdM Portal will check if the presented pseudonym is in the list of authorized pseudonyms stored in the IdM Database. If this is the case, the IdM Portal will mark this pseudonym as 'claimed', map it to the identity of the User and set the oneTimePass flag to 'disabled'.

Once the User's smart card is registered, more services are offered to the User in the GUI of the IdM Portal. From now on, the User can apply for Privacy-ABC credentials. When the User clicks on a specific tab mapped to a specific credential, the corresponding issue URL and step URL are loaded. Finally, when the User clicks on the 'Ladda ner certifikat till kortet' button, the servlet will return the corresponding issuance policy. For obtaining this policy, the IdM Portal forwards the relevant attribute values of the User via the ExchangeData class to the web service method of the 'IdM ABC System' responsible for starting issuance of the specific credential. All issuance policies of the IdM Portal require the User to present a scope-exclusive pseudonym based on 'urn:soderhamn:registration'. After receiving the issuance policy, the User forwards an issuance message to the servlet located at the step URL. The servlet responsible handling the 'step URL' requests is aware of which credential must be issued. And the servlet knows the identity of the User based on the cookie the User presents. So the servlet can check if the scope-exclusive pseudonym the User is presenting matches the scope-exclusive pseudonym mapped to her identity. Only if this is the case, the IdM Portal will forward the issuance message of the User to the 'IdM ABC System'. Issuance of Idemix credentials require in general 2 rounds of messages. In the case of U-Prove; 3 rounds are required.

If the User has already obtained a revocable credential and applies for the same one, the old credential will be automatically revoked by the IdM Portal before this application requests the 'IdM ABC System' to send an issuance policy with a fresh nonce.

In the case of U-Prove, automatic reIssuance is supported by the IdM Portal. During the issuance of a U-Prove credential, the User Application stores the issue URL and the step URL. If the first batch of credentials/tokens has already been consumed by the User, the User automatically visits the IdM Portal responsible for the issuance of this credential and applies for a new batch. This procedure is GUI-less. No User interaction (and no PIN) is required for reIssuance.

With a valid school credential (credSchool), the User has the possibility to test her smart card. Basically, Privacy-ABC verification will be launched using a special presentation policy requiring the possession of a valid credSchool. In the Söderhamn pilot, this policy requires the User to reveal her first and last name.

As in the IdM Application, the servlets have no session awareness. So the data exchanged between the servlets and the session-aware parts of the IdM Portal are managed via a lookup table.

The IdM Portal stores all images used by the IdM Portal itself and by the IdM Application. On top of that, the IdM Portal stores the images of the credentials. Each credential specification defines from where the User must download the corresponding image via the abc:DefaultImageReference field.

Integration testing of the IdM Portal requires the presence of the IdM Application. If not, all User requests will be rejected since authentication is not possible. Registering the scope-exclusive pseudonym, executing Privacy-ABC issuance and verification requires the presence of the 'IdM ABC System'. The native environment of the IdM Portal is a Linux-based OS. NSNs integration testing environment hosts the IdM Portal on a SuSe server. As stated in 2.4.1.2, the IdM Portal requires OpenJDK, an SSH server and an Apache-Tomcat installation. Since the IdM Portal needs to connect to the IdM Database, the IdM Portal must be located on the same machine as the IdM Database. This way, access to the database can be handled internally via localhost without involving a firewall. In both pilots, the IdM Portal and the IdM Applicaton are hosted in the same Apache-Tomcat container. Both the IdM Portal and the IdM Application must be directly accessible via the Internet. The connector of port 8443 of this container must be configured with an X.509 certificate and enforce the use of SSL. NSN's firewall must allow incoming Internet traffic to port 8443 and allow incoming traffic to the SSH port only from dedicated source IP addresses.

2.4.2.3 Deployment

The deployment of the IdM Portal is very similar to the IdM Application (please see Section 2.4.1.3) since both applications are hosted in the same Apache-Tomcat container. The IdM Portal requires a subset of the same firewall rules and connects to the 'IdM ABC System' during its build via WSDL file.

The only difference to the deployment of the IdM Application is that the IdM Portal's GUI need not be edited when shifting from one site to another: these URLs in the JavaScript of the GUI are configurable via a properties file. This feature was easily implementable since the IdM Portal GUI is designed using Java Servlet Faces (JSF) technology. Contrary to that, the IdM Application uses Java Server Pages (JSP) technology.

EDOC and NSN executed the system tests. Before the operation phase of the pilot, NSN invited to a formal handover of the School Registration System and the Revocation Authority. EDOC thereafter configured their firewall to block traffic coming from NSN. After 'cutting the lines', the pilot entered the operation state.

2.4.2.4 Operation

During the 1st round of the Söderhamn pilot EDOC assisted the Users to visit the IdM Portal in order to download their credentials to their smart cards. The Users had to download each of the credentials separately. Before the start of the 2nd round EDOC downloaded all the credentials to the smart cards before they were handed out to the Users.

During the 2nd round of the pilot, while the smart cards were prepared by EDOC, a recurrent failure was discovered which was blocking the process of downloading the credentials to the smart cards. This issue did not appear earlier, in the 1st round, because only small number of smart cards was prepared.

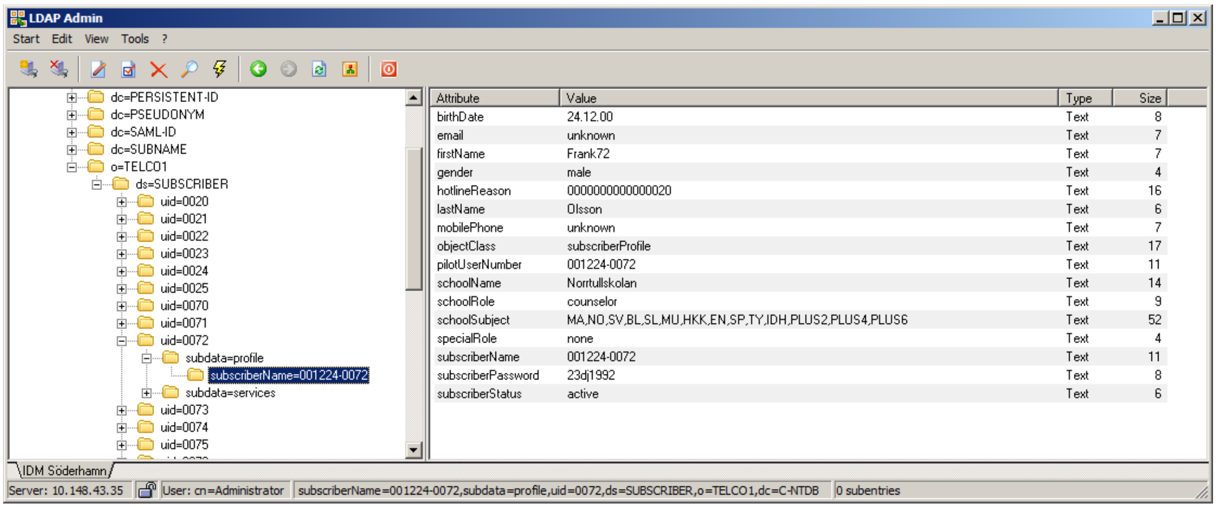
This failure appeared as a hanging of the User service during issuance and had quite sporadic occurrences – that made debugging more complicated. Enhanced for the 2nd round of the pilot, the logging of the User Application made WP4, WP5 and WP6 come to the assumption that the hanging was related to the lock of U-Prove under the User Application Windows service. After the fix was posted and the User Application was updated no more problems were reported. The conclusion was that the issue with the hanging of the User service was not caused by the IdM Portal.

Issuance of credentials and storing them on the smart cards via the IdM Portal interface was tested using up to 50 smart cards in a row, which allows positive conclusions about the reliability of the IdM Portal for mass usage (issuance) during the pilot, even though Users did not have to access the IdM Portal.

2.4.3 IdM Mass Provisioning Tool

2.4.3.1 Implementation

For provisioning single Users in the IdM Database, the developers of the IdM System used the 'LDAP Admin' Tool (http://www.ldapadmin.org/). On the one hand, this tool allows fine granular tuning of the database. On the other hand, this tool requires deep knowledge of its structure, including knowledge of the applications reading and writing into it, and knowledge on the syntax of the allowed values of its parameters and fields (see Figure 14). So it was clear, that this tool only had limited usefulness for foreign administrators i.e. administrators which were not involved in the development of the School Registration System. Another argument for the development of a provisioning tool was that the Söderhamn pilot required large numbers of Users to be provisioned.



Attribute	Value	Type	Size
birthDate	24.12.00	Text	8
email	unknown	Text	7
firstName	Frank72	Text	7
gender	male	Text	4
hotlineReason	0000000000000020	Text	16
lastName	Olsson	Text	6
mobilePhone	unknown	Text	7
objectClass	subscriberProfile	Text	17
pilotUserNumber	001224-0072	Text	11
schoolName	Norrtullskolan	Text	14
schoolRole	counselor	Text	9
schoolSubject	MA,NO,SV,BL,SL,MU,HKK,EN,SP,TY,IDH,PLUS2,PLUS4,PLUS6	Text	52
specialRole	none	Text	4
subscriberName	001224-0072	Text	11
subscriberPassword	23dj1992	Text	8
subscriberStatus	active	Text	6

Server: 10.148.43.35 | User: cn=Administrator | subscriberName=001224-0072,subdata=profile,uid=0072,ds=SUBSCRIBER,o=TELCO1,dc=C-NTDB | 0 subentries

Figure 14: Snapshot of IdM Database using LDAP Admin Tool

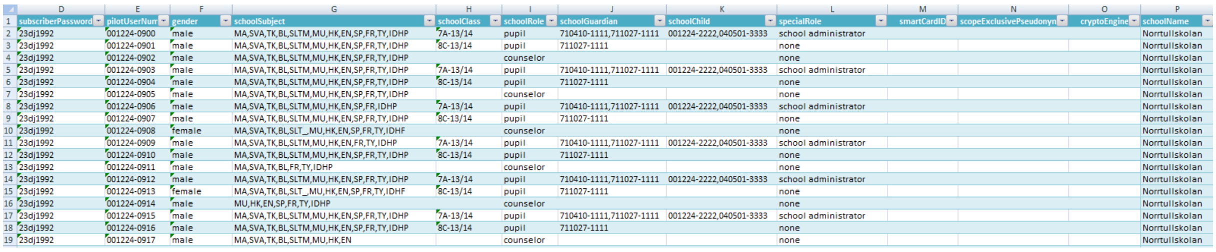
The IdM Mass Provisioning Tool was not taken into account in the architecture and in the requirements of [D51]. But the necessity of such a tool became clear in the course of the pilot. The following main implementation tasks had to be executed:

- provide an interface to read the data of a comma separated file as input source (see Table 2)
- allow as optional input data the output data supplied during the initialization of the smart cards
    - Scope Exclusive Pseudonym
    - smart card ID
    - Crypto Engine Type (Idemix or U-Prove)
- check the syntax of the input data before writing it into the IdM Database
- provide an interface to the IdM Database
- provide a self-explaining GUI for triggering the functionalities of the application
- provide safeguards so as to not overwrite already available User data or already ‘claimed’ smart card data
- support Swedish attribute values

Important to note, is that shortly before the 2nd round of the Söderhamn pilot, EDOC needed to shorten the installation time of the pupils’ smart cards. For the convenience of the pupils, EDOC decided to distribute the smart cards with a complete set of pre-installed Privacy-ABC credentials. In the original setup, this meant that the administrators responsible for the operation phase of the pilot would have to:

1. initialize the smart cards
2. deploy the Smart Card Registrar and thereby feed the IdM Database with data mapped to the pseudonyms
3. deploy the IdM Mass Provisioning Tool to feed the IdM Database with data mapped to the Users
4. register the smart cards via the IdM Portal using the OTP of the Users
5. use the IdM Portal to apply for Privacy-ABC credentials

In order to shorten the smart card preparation time, ALX added a feature to the initializer of the smart cards so that the Scope Exclusive Pseudonym, the smart card ID and the Cryto Engine Type are output to a new CSV file as a result of the initialization. And NSN added a feature to the IdM Mass Provisioning Tool to pick up these values next to the values mapped to the Users. So in the end, the administrator of the pilot just had to prepare the User data in a table (see Table 2) and copy/paste the smart card data into it before launching the tool to import the resulting table into the IdM Database. So in effect, the steps 2, 3 and 4 were merged into a single step.

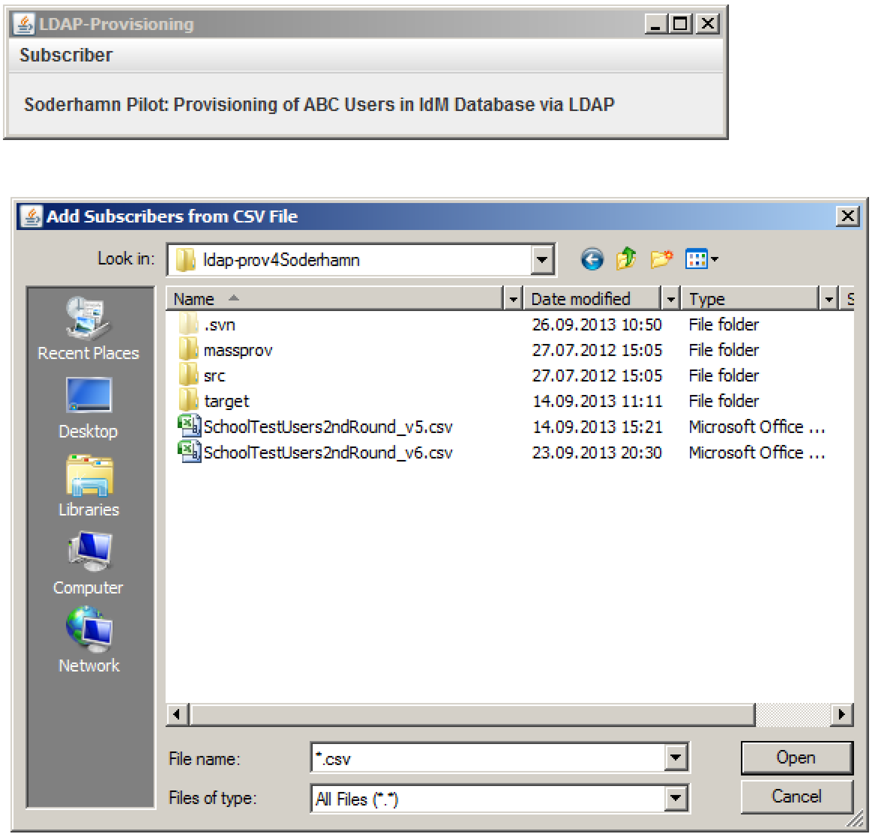


	D	E	F	G	H	I	J	K	L	M	N	O	P
1	subscriberPassword	pilotUserNum	gender	schoolSubject	schoolClass	schoolRole	schoolGuardian	schoolChild	specialRole	smartCardID	scopeExclusivePseudonym	cryptoEngine	schoolName
2	23dj1992	001224-0900	male	MA,SVA,TK,BL,SLTM,MU,HK,EN,SP,FR,TY,IDHP	7A-13/14	pupil	710410-1111,711027-1111	001224-2222,040501-3333	school administrator				Norrtullskolan
3	23dj1992	001224-0901	male	MA,SVA,TK,BL,SLTM,MU,HK,EN,SP,FR,TY,IDHP	8C-13/14	pupil	711027-1111		none				Norrtullskolan
4	23dj1992	001224-0902	male	MA,SVA,TK,BL,SLTM,MU,HK,EN,SP,FR,TY,IDHP		counselor			none				Norrtullskolan
5	23dj1992	001224-0903	male	MA,SVA,TK,BL,SLTM,MU,HK,EN,SP,FR,TY,IDHP	7A-13/14	pupil	710410-1111,711027-1111	001224-2222,040501-3333	school administrator				Norrtullskolan
6	23dj1992	001224-0904	male	MA,SVA,TK,BL,SLTM,MU,HK,EN,SP,FR,TY,IDHP	8C-13/14	pupil	711027-1111		none				Norrtullskolan
7	23dj1992	001224-0905	male	MA,SVA,TK,BL,SLTM,MU,HK,EN,SP,FR,TY,IDHP		counselor			none				Norrtullskolan
8	23dj1992	001224-0906	male	MA,SVA,TK,BL,SLTM,MU,HK,EN,SP,FR,IDHP	7A-13/14	pupil	710410-1111,711027-1111	001224-2222,040501-3333	school administrator				Norrtullskolan
9	23dj1992	001224-0907	male	MA,SVA,TK,BL,SLTM,MU,HK,EN,SP,FR,TY,IDHP	8C-13/14	pupil	711027-1111		none				Norrtullskolan
10	23dj1992	001224-0908	female	MA,SVA,TK,BL,SLT_,MU,HK,EN,SP,FR,TY,IDHF		counselor			none				Norrtullskolan
11	23dj1992	001224-0909	male	MA,SVA,TK,BL,SLTM,MU,HK,EN,FR,TY,IDHP	7A-13/14	pupil	710410-1111,711027-1111	001224-2222,040501-3333	school administrator				Norrtullskolan
12	23dj1992	001224-0910	male	MA,SVA,TK,BL,SLTM,MU,HK,EN,SP,FR,TY,IDHP	8C-13/14	pupil	711027-1111		none				Norrtullskolan
13	23dj1992	001224-0911	male	MA,SVA,TK,BL,FR,TY,IDHP		counselor			none				Norrtullskolan
14	23dj1992	001224-0912	male	MA,SVA,TK,BL,SLTM,MU,HK,EN,SP,FR,TY,IDHP	7A-13/14	pupil	710410-1111,711027-1111	001224-2222,040501-3333	school administrator				Norrtullskolan
15	23dj1992	001224-0913	female	MA,SVA,TK,BL,SLT_,MU,HK,EN,SP,FR,TY,IDHF	8C-13/14	pupil	711027-1111		none				Norrtullskolan
16	23dj1992	001224-0914	male	MU,HK,EN,SP,FR,TY,IDHP		counselor			none				Norrtullskolan
17	23dj1992	001224-0915	male	MA,SVA,TK,BL,SLTM,MU,HK,EN,SP,FR,TY,IDHP	7A-13/14	pupil	710410-1111,711027-1111	001224-2222,040501-3333	school administrator				Norrtullskolan
18	23dj1992	001224-0916	male	MA,SVA,TK,BL,SLTM,MU,HK,EN,SP,FR,TY,IDHP	8C-13/14	pupil	711027-1111		none				Norrtullskolan
19	23dj1992	001224-0917	male	MA,SVA,TK,BL,SLTM,MU,HK,EN		counselor			none				Norrtullskolan

Table 2: Input File of IdM Mass Provisioning Tool

2.4.3.2 Integration

The IdM Mass Provisioning Tool does not make use of Privacy-ABC technologies. But the tool can be considered as an ‘enabler’ for the School Registration System. The IdM Mass Provisioning Tool is a Java executable which allows provisioning of the IdM Database via graphical user interface (GUI). As prerequisite, the administrator must prepare a CSV file containing the User data (see Figure 15).



LDAP-Provisioning

Subscriber

Soderhamn Pilot: Provisioning of ABC Users in IdM Database via LDAP

Add Subscribers from CSV File

Look in: ldap-prov4Soderhamn

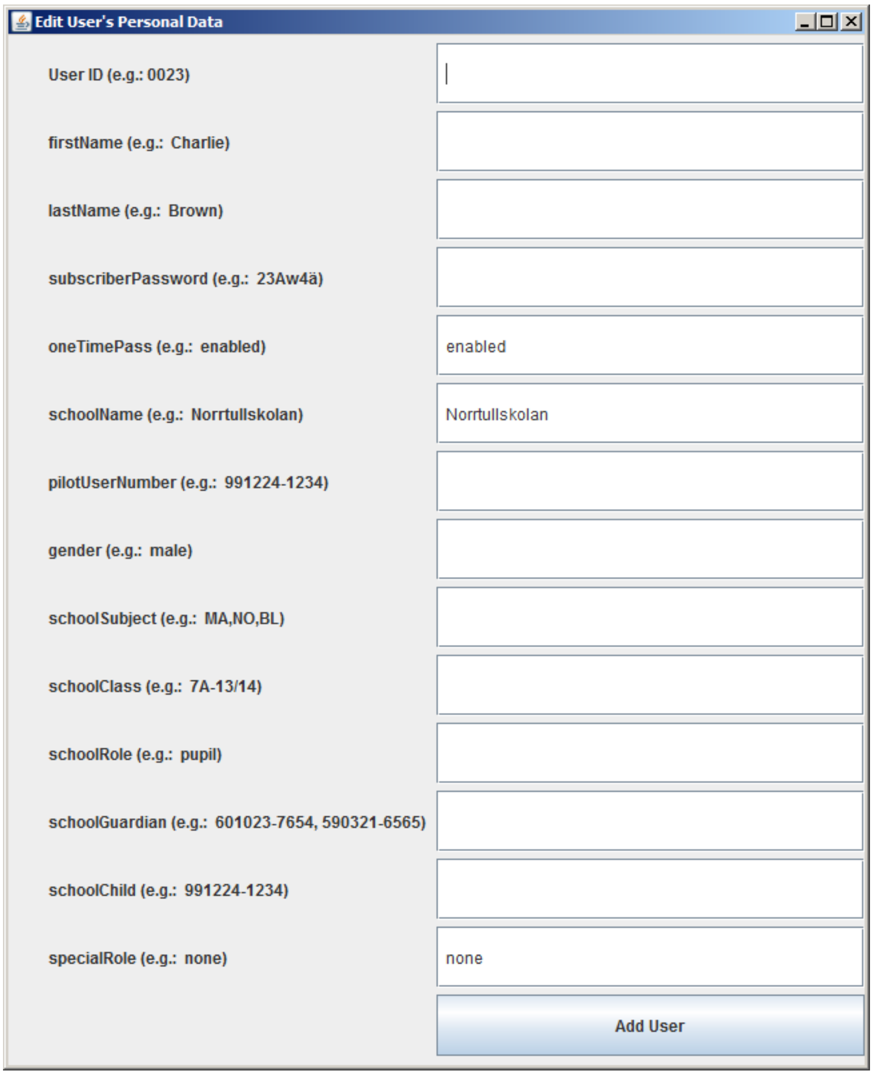
Name	Date modified	Type
.svn	26.09.2013 10:50	File folder
massprov	27.07.2012 15:05	File folder
src	27.07.2012 15:05	File folder
target	14.09.2013 11:11	File folder
SchoolTestUsers2ndRound_v5.csv	14.09.2013 15:21	Microsoft Office ...
SchoolTestUsers2ndRound_v6.csv	23.09.2013 20:30	Microsoft Office ...

File name: *.csv

Files of type: All Files (*.*)

Figure 15: IdM Mass Provisioning Tool: Add Subscribers from CSV Menu

In the integration phase of the pilot, only test User data is handled. For verification purposes, additional menus were developed which allowed provisioning data of a single User (see Figure 16). A closer look at this figure reveals which attribute values can be edited. The attribute specialRole is an important marker for the IdM Admin GUI. If the value is set to ‘school administrator’, special rights are mapped to this User allowing her full control when using the IdM Admin GUI.

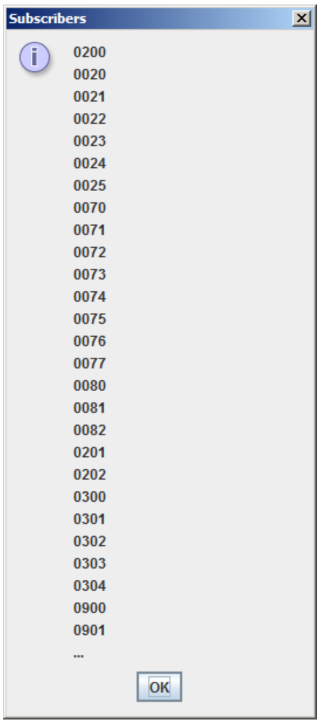


Edit User's Personal Data

User ID (e.g.: 0023)	
firstName (e.g.: Charlie)	
lastName (e.g.: Brown)	
subscriberPassword (e.g.: 23Aw4ä)	
oneTimePass (e.g.: enabled)	enabled
schoolName (e.g.: Norrtullskolan)	Norrtullskolan
pilotUserNumber (e.g.: 991224-1234)	
gender (e.g.: male)	
schoolSubject (e.g.: MA,NO,BL)	
schoolClass (e.g.: 7A-13/14)	
schoolRole (e.g.: pupil)	
schoolGuardian (e.g.: 601023-7654, 590321-6565)	
schoolChild (e.g.: 991224-1234)	
specialRole (e.g.: none)	none

Figure 16: IdM Mass Provisioning Tool: New Subscriber Menu

The ‘Show Subscribers’ menu (see Figure 17) gives an overview of which Users are already stored in the IdM Database. This menu was mainly developed for testing too.



```
0200
0020
0021
0022
0023
0024
0025
0070
0071
0072
0073
0074
0075
0076
0077
0080
0081
0082
0201
0202
0300
0301
0302
0303
0304
0900
0901
...
```

Figure 17: IdM Mass Provisioning Tool: Show Subscribers Menu

During integration testing, the IdM Mass Provisioning Tool was launched directly out of the NetBeans IDE. The check if all values were really copied into the IdM Database was performed using the LDAP Admin Tool mentioned previously. NSNs firewall had to be configured to allow incoming traffic to the LDAP port of the IdM Database from the company's Intranet only. Please note, that access to the IdM Database requires knowledge of a password.

Testing the IdM Mass Provisioning Tool included testing the 2 configuration files. The 'properties configuration' file is responsible for the tool location and the database location. And the 'categories configuration' file is responsible for the syntax checker rules.

2.4.3.3 Deployment

With the 'config properties' file, the IdM Mass Provisioning Tool can be moved to any target location. In the Söderhamn pilot, this tool was moved to the Windows server and therefore is co-located with the Apache-Tomcat application 'IdM ABC System'. Due to this fact, the Ubuntu firewall protecting the IdM Application with its IdM Database had to be configured to allow incoming LDAP traffic originating from the Windows system.

EDOC and NSN executed the system tests. Before the operation phase of the pilot, NSN invited to a formal handover of the School Registration System and the Revocation Authority. EDOC thereafter configured their firewall to block traffic coming from NSN. After 'cutting the lines', the pilot entered the operation state.

2.4.3.4 Operation

During the operation of both pilots the IdM Mass Provisioning Tool was used by EDOC as the main interface to upload the User data to the IdM Database. The input to the tool was a comma separated text file. This text file was exported from a filled-in Excel template that was provided by NSN. Excel provides the functionality to export tables into CSV (comma separated value) format. The IdM Mass Provisioning Tool is sensitive to text encoding (the text had to be in the Windows -1250 encoding) and to the CSV delimiter (the delimiter has to be semicolon ";").

While filling in the template with the User data EDOC had to be attentive and make sure that the User ID, the PUN and the pseudonym were not already uploaded before. This had to be manually checked by EDOC.

After the 1st round the IdM Database was wiped.

2.4.4 IdM Admin GUI

2.4.4.1 Implementation

The development of the IdM Admin GUI was requested by EDOC based on a need to allow provisioning of the User data during the operation phase when the Users were active.

The development of the IdM Admin GUI is based on the IdM Portal. The IdM Admin GUI does not authenticate the Users itself, but redirects the User to the IdM Application for this task. So the IdM Admin GUI is a SAML client. See Section 2.4.2.1 for details on SAML handling.

As with the IdM Mass Provisioning Tool, the IdM Admin GUI was not taken into account in the architecture and in the requirements of [D51]. NSN and EDOC agreed on the following development packet:

- authenticate the Users
- offer administration menus only to Users with the 'specialRole' attribute equal to 'school administrator'
- connect to the 'IdM ABC System' to enable Privacy-ABC revocation and verification
- connect to the IdM Database to store or delete attribute values
- support Microsoft's Internet Explorer
- support Swedish attribute values
- check the syntax of the input data before writing it into the IdM Database
- automatically launch revocation, if an attribute value in a revocable Privacy-ABC credential is being modified and if the User already obtained such a credential

In order to be able to deploy the IdM Admin GUI, the 'school administrator' must login via smart card. If the administrator has not yet registered his Smard Card, an integrated registration menu offers this service. As soon as the administrator is logged in via smart card, he can enter the PUN (Pilot User Number) of the User whose attribute(s) he needs to modify. After entering this PUN, the IdM Admin GUI will send an `AttributeQuery`' to the IdM Application in order to retrieve the attributes stored about this User. Please note, that the attribute release policy of the IdM Admin GUI is different to the attribute release policy of the IdM Portal. One main difference is that the only SAML client that can retrieve the 'specialRole' parameter value of Users is the IdM Admin GUI. If the administrator enters a PUN of a User who is himself a school administrator, he will be notified that this is not possible. Elevating Users' rights from specialRole='none' to specialRole='school administrator' will be prevented too. Degrading Users' rights from specialRole='school administrator' to specialRole='none' will also be prevented. In these specific cases, the LDAP Admin Tool must be used.

The IdM Admin GUI together with the ‘IdM ABC System’ is both a Privacy-ABC Verifier and a ‘Revocation Requestor’ according to [D51].

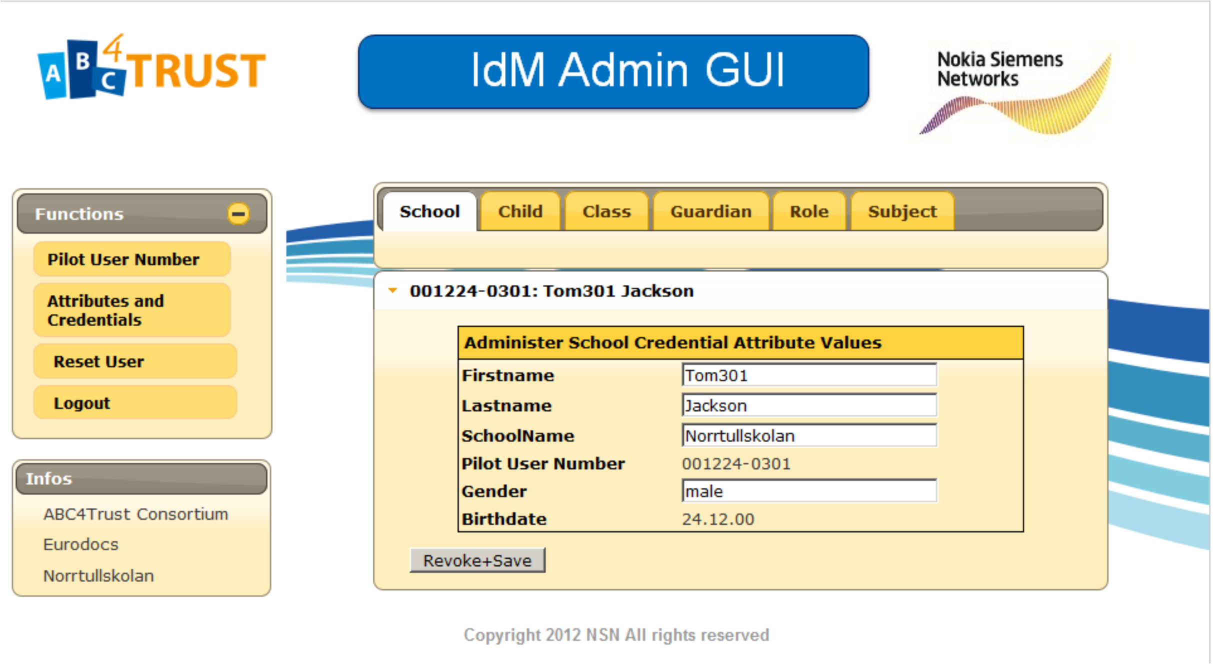
The IdM Portal incorporates a significant set of servlets, most of which handle the SAML call flow. But in effect only 2 servlets handle the Privacy-ABC traffic:

- verification/policy
- verification/verify

The handling of these servlets is identical to the policy servlet and the verify servlet required for registering the smart cards as described in 2.4.2.1.

The IdM Admin GUI offers one tab per Privacy-ABC credential (see Figure 18). When the administrator selects a specific tab, he will see all attribute values mapped to this credential. If the administrator changes one attribute value, the ‘Syntax Checker’ (see Figure 3) will verify this value against a list or a Regex rule. If the syntax is valid, the IdM Admin GUI will launch revocation (if applicable) and then modify the value.

Next to the tabs offering modifications in the attributes of the Privacy-ABC credentials, the application offers a ‘reset User’ service, which basically decouples the User’s data set from a specific scope-exclusive pseudonym, launches revocation of the school credential ‘credSchool’ (if applicable) and allows login of the User via OTP again.



001224-0301: Tom301 Jackson

Administer School Credential Attribute Values	
Firstname	Tom301
Lastname	Jackson
SchoolName	Norrtullskolan
Pilot User Number	001224-0301
Gender	male
Birthdate	24.12.00

Copyright 2012 NSN All rights reserved

Figure 18: IdM Admin GUI

2.4.4.2 Integration

In the case of registering the smart card of a User and thereby claiming a scope-exclusive pseudonym (based on the same scope the IdM Application, the Smart Card Registrar and the IdM Portal are using), the application acts as proxy. In the case of revocation, the IdM Admin GUI is the originator of the request. Please note, that launching revocation requires no interaction with the smart cards of the Users.

As in the IdM Application, the 2 servlets have no session awareness. So the data exchanged between the servlets and the session-aware parts of the IdM Admin GUI are managed via a lookup table.

The IdM Admin GUI stores all images required for the GUI itself and is not dependant on the IdM Portal.

As in the IdM Portal, the IdM Admin GUI connects to the 'IdM ABC System' via WSDL file. When Privacy-ABC verification is requested by the User or revocation is launched by the application itself, the corresponding web service of the 'IdM ABC System' is called.

All applications within the School Registration System except the IdM Mass Provisioning Tool require to be hosted in an Apache-Tomcat container. The IdM Admin GUI is only dependant on the presence of the 'IdM ABC System' (for registering the smart card of the administrator and for launching revocation) and the IdM Application (for authenticating the administrator). In the integration testing environment, the IdM Admin GUI is hosted in the same Apache-Tomcat container which hosts the IdM Application and the IdM Portal, but for security reasons, the application is designed to be easily shifted into a container hosting the administration web applications. As in the IdM Portal case, HTTPS is enforced by pointing the server.xml file to an X.509 certificate. NSN's firewall must allow incoming Internet traffic to port 8443 and allow incoming traffic to the SSH port only from dedicated source IP addresses. The former rule allows EDOC and WP4 to participate in the integration test.

2.4.4.3 Deployment

Contrary to the previous phases, beginning with the deployment phase, the IdM Admin GUI must not be accessible via the public Internet any more.

To enable this, the IdM Admin GUI was moved into an own Apache-Tomcat container. This allows configuration of a new HTTPS listening port, i.e. 8444. The IdM Application and the IdM Portal are located on the same server hosting the IdM Admin GUI. The UFW firewall of this server allows Internet access to port 8443. For protecting the IdM Admin GUI, the same firewall must be configured to allow traffic to port 8444 only from specific source IP addresses.

As described in 2.4.2.3, the IdM Admin GUI is configurable via properties file. So for customization, i.e. for adapting the application to work in the target environment, only the properties file needs to be adapted and the WSDL pointers in 2 locations (2.4.1.3) need to be adjusted.

EDOC and NSN executed the system tests. Before the operation phase of the pilot, NSN invited to a formal handover of the School Registration System and the Revocation Authority. EDOC thereafter configured their firewall to block traffic coming from NSN. After 'cutting the lines', the pilot entered the operation state.

2.4.4.4 Operation

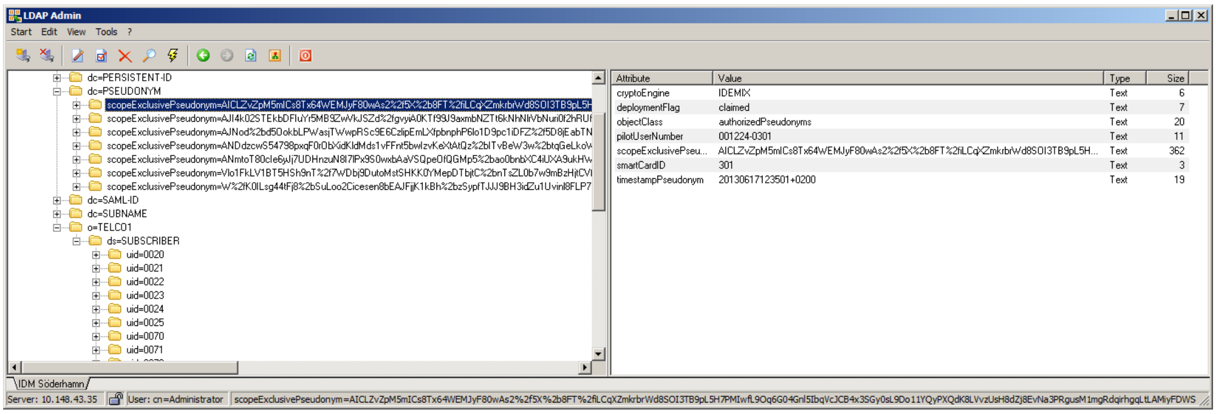
The IdM Admin GUI was used to edit the User data in the IdM Database. This tool was developed for the 2nd round of the pilot as to simplify the process of editing User data and prevents mistakes in data format and structure.

Tests before the pilot and further usage did not discover any malfunction cases.

2.4.5 Smart Card Registrar

2.4.5.1 Implementation

Complex scope-exclusive pseudonyms (see Figure 19) cannot be added to the IdM Database manually. This argument and the insight that the administrator must handle 2 kinds of data (User data and smart card data) in 2 independent storage areas of the IdM Database lead to the decision for developing the Smart Card Registrar.



```
dc=PERSISTENT-ID
dc=PSEUDONYM
    scopeExclusivePseudonym=AICLZvZpM5mICs8Tx64WEMJyF80wAs2%2f5X%2b8FT%2fiLCqXZmkrbrWd8SOI3TB9pL5H
    scopeExclusivePseudonym=AJI4k02STEkbDFluYr5MB9ZwVkJSZd%2fgvyiA0KTf99J9axmbNZTt6kNhNlrVbNuri0f2hRUf
    scopeExclusivePseudonym=AJNod%2bd50okbLPWasjTWwpRSc9E6CzlipEmLXfpbnphP6lo1D9pc1iDFZ%2f5D8jEabTN
    scopeExclusivePseudonym=ANDdzcwS54798pxqF0r0bXidKldMds1vFFnt5bwlzvKeXtAtQz%2blTvBeW3w%2btqGeLkoW
    scopeExclusivePseudonym=ANmtoT80cle6yJj7UDHnzuN8I7lPx9S0wxbAaVSQpeOfQGMp5%2bao0bnbXC4iUXA9ukHw
    scopeExclusivePseudonym=Vlo1FkLV1BT5HSh9nT%2f7WDbj9DutoMstSHKK0YMepDTbjtC%2bnTsZL0b7w9mBzHjtCVl
    scopeExclusivePseudonym=W%2fK0ILsg44tFj8%2bSuLoo2Cicesen8bEAJFjjK1kBh%2bzSypfTJJJ9BH3idZu1Uvinl8FLP7
dc=SAML-ID
dc=SUBNAME
o=TELCO1
    ds=SUBSCRIBER
        uid=0020
        uid=0021
        uid=0022
        uid=0023
        uid=0024
        uid=0025
        uid=0070
        uid=0071
```

Attribute	Value	Type	Size
cryptoEngine	IDEMIX	Text	6
deploymentFlag	claimed	Text	7
objectClass	authorizedPseudonyms	Text	20
pilotUserNumber	001224-0301	Text	11
scopeExclusivePseu...	AICLZvZpM5mICs8Tx64WEMJyF80wAs2%2f5X%2b8FT%2fiLCqXZmkrbrWd8SOI3TB9pL5H...	Text	362
smartCardID	301	Text	3
timestampPseudonym	20130617123501+0200	Text	19

IDM Söderhamn

Server: 10.148.43.35 | User: cn=Administrator | scopeExclusivePseudonym=AICLZvZpM5mICs8Tx64WEMJyF80wAs2%2f5X%2b8FT%2fiLCqXZmkrbrWd8SOI3TB9pL5H7PMIwfL9Oq6G04Gnl5IbqVcJCB4x3SGy0sL9Do11YQyPXQdK8LVvzUsH8dZj8EvNa3PRgusM1mgRdqirhgqLtLAMiyFDWS

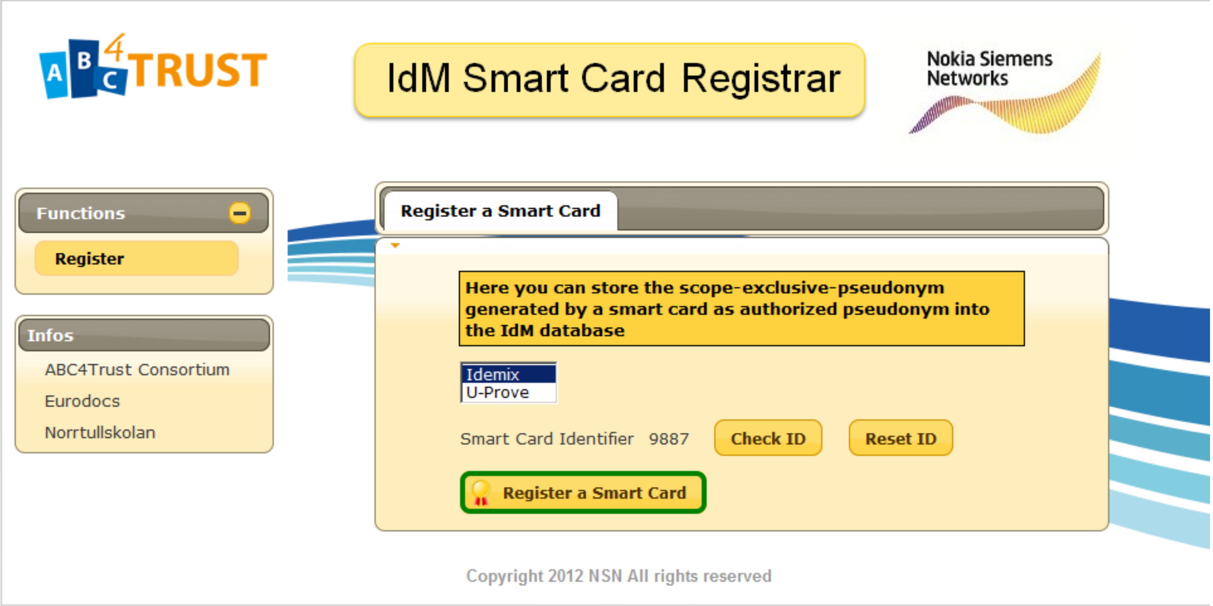
Figure 19: Smart Card Data in IdM Database

Contrary to the IdM Admin GUI, the Smart Card Registrar does not require authentication of the administrator at all. The Smart Card Registrar is also no SAML client and needs not retrieve any User data. The service for registering smart cards is accessible to anybody that can access this web application.

The Smart Card Registrar was not taken into account in the architecture and in the requirements of [D51]. This service includes the following development packet:

- offer administration menus to all administrators of the school
- connect to the 'IdM ABC System' to enable Privacy-ABC verification
- connect to the IdM Database to store new smart card data or to overwrite old 'unclaimed' smart card data
- support Microsoft's Internet Explorer
- allow input of crypto engine type and smart card ID (see Figure 20)
- provide a safeguard so as to not overwrite already 'claimed' smart card data
- provide a warning if a smart card ID is already used in the IdM Database

After initializing the smart cards, the administrator will use the Smart Card Registrar to input the crypto engine type together with the smart card ID and to extract the scope-exclusive pseudonym (based on the scope 'urn:soderhamn:registration') from the smart card. This smart card data is stored in the IdM Database using the pseudonym as handle.



Here you can store the scope-exclusive-pseudonym generated by a smart card as authorized pseudonym into the IdM database

Smart Card Identifier 9887

Copyright 2012 NSN All rights reserved

Figure 20: Smart Card Registrar GUI

The Smart Card Registrar together with the ‘IdM ABC System’ is a Privacy-ABC Verifier according to [D51].

The Smart Card Registrar incorporates only 2 servlets to handle the Privacy-ABC traffic:

- verification/policy
- verification/verify

The handling of these servlets is significantly different to the handling described in 2.4.2.1, as in the case of the Smart Card Registrar, the smart card data is being initialized in the IdM Database. Contrary to this, the registration services of IdM Admin GUI and the IdM Portal link the smart card data (provided by the Smart Card Registrar) to the User data (provided by the IdM Mass Provisioning Tool).

2.4.5.2 Integration

The Smart Card Registrar connects to the ‘IdM ABC System’ via a WSDL file. When the administrator presses the ‘Register a Smart Card’ button, the Smart Card Registrar acts as proxy between the User Application and the ‘IdM ABC System’. Since the registrar is able to extract the scope-exclusive pseudonym from the presentation token sent by the User, it can cache this value before forwarding the token to the ‘IdM ABC System’. When the ‘IdM ABC System’ acknowledges that the token is ok, the Smart Card Registrar will store the pseudonym along with other flags and information into the IdM Database.

As in the IdM Application, the 2 servlets have no session awareness. So the data exchanged between the servlets and the session-aware parts of the Smart Card Registrar are managed via a lookup table.

The Smart Card Registrar stores all images required for the GUI itself and is not dependant on the IdM Portal.

It is recommended to host the Smart Card Registrar in the same Apache-Tomcat container that hosts the IdM Admin GUI. In the integration testing environment, the Smart Card Registrar is hosted in the same container as the other web applications. But both the IdM Admin GUI and the Smart Card Registrar are designed to be hosted in a separate container accessible for administrators only.

NSN's firewall must allow incoming Internet traffic to port 8443 and allow incoming traffic to the SSH port only from dedicated source IP addresses. The former rule allows EDOC and WP4 to participate in the integration test.

2.4.5.3 Deployment

The deployment of the Smart Card Registrar is identical to the deployment of the IdM Admin GUI (see Section 2.4.4.3). The application must be shifted to a separate Apache-Tomcat container listening on port 8444. This way, the UFW can control access to these administration web services. The WSDL pointers and the properties file just need to be adjusted for the target site.

EDOC and NSN executed the system tests. Before the operation phase of the pilot, NSN invited to a formal handover of the School Registration System and the Revocation Authority. EDOC thereafter configured their firewall to block traffic coming from NSN. After 'cutting the lines', the pilot entered the operation state.

2.4.5.4 Operation

The Smart Card Registrar which was tested and handed over from NSN to EDOC was needed and used only in the 1st round of the pilot but not in the 2nd round. The main reason was that in the 2nd round administrators did not have to use this tool any more as the initialization of the smart card was enhanced to export the scope exclusive pseudonyms, the smart card ID and the CE type into a file. The administrators just had to copy/paste these contents into the Excel table of the IdM Mass Provisioning Tool which then inserted the contents directly to the IdM Database after checking the syntax.

During operation of the 1st round the administrators had to visit the Smart Card Registrar (https://idm.abc4trust.se:8444/idmSmartCardRegistrar/index.jsf) in order to register the cards before they were handed over to Users. After the cards were handed over to Users who have signed the consent forms, the Users visited the IdM Portal (https://idm.abc4trust.se:8443/idmPortal/index.jsf) and downloaded their own credentials. This process was successfully performed without any major problems. The Smart Card Registrar was only used by administrators and not by the Users.

During the 2nd round of the pilot the Smart Card Registrar was not used. EDOC uploaded card-specific data to the IdM Database via the enhanced IdM Mass Provisioning Tool which replaced the functionality of the Smart Card Registrar.

2.4.6 IdM ABC System – Issuer and Verifier

The 'IdM ABC System' is an independent web service located within the School Registration System. This web service is not accessible via the Internet. The infrastructure of the pilot owner protects this service via firewall rules allowing only the IdM Portal, the IdM Admin GUI, Smart Card Registrar and the IdM Application to forward ABCE related queries to its web methods. The 'IdM ABC System' embeds the ABCE and the U-Prove and Idemix CEs.

In order to facilitate the adoption of the new Privacy-ABC technologies, WP4 provided a Söderhamn 'sample application' in the beginning of the project. This sample application included Issuer and Verifier example code and was basis for the 'IdM ABC System' development. Towards the end of this project, WP4 split the sample application in ready-to-use RESTful 'abce-services'. Each abce-service can be mapped to an ABC role (Inspector, Issuer, Verifier, etc). The source code for these abce-services can be found here: https://abc4trust.eu/index.php/source.

Contrary to the publicly available implementations of the Issuer and the Verifier abce-services, the ‘IdM ABC System’ was implemented differently and does not offer its services via RESTful interfaces. Nevertheless, the methods used by the ‘IdM ABC System’ for processing the messages in the end are identical to those used by the abce-services.

2.4.6.1 Implementation

The ‘IdM ABC System’ was developed using NetBeans 7.3 IDE (Integration Development Environment).

The web clients within the School Registration System (IdM Portal, IdM Admin GUI, Smart Card Registrar and IdM Application) are connected to the ‘IdM ABC System’ via WSDL file during the build process. The communication between the web clients and the ‘IdM ABC System’ is therefore HTTP/SOAP. The SOAP body contains text-based XML.

The ‘IdM ABC System’ offers the following services:

- start*Cred*
- issueStep
- startReIssuance*Cred*
- reIssueStep
- revokeCredential
- requestPolicy
- verifyToken

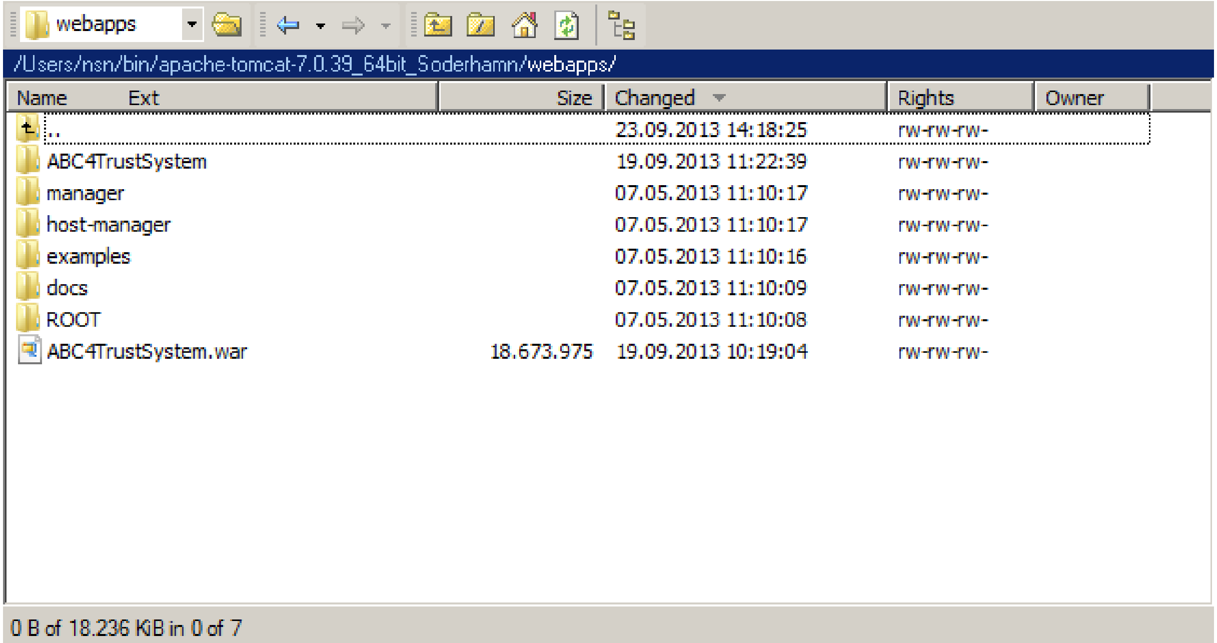
The italic placeholder *Cred* represents the 6 different credential types the pilot requires (i.e. School, Role, Subject, Guardian, Child and Class). So in effect, the ‘IdM ABC System’ handles 17 web service methods.

Every method requires a set of strings as input defined in a class (ExchangeData). And every method returns a string.

ExchangeData transports the attribute values required for generating the credentials. On top of that, this class has placeholders for a revocation handle, an ABC message, the presentation policy alternatives, a policy id, a presentation token and the crypto engine type.

When a web client accesses a specific web service method of the ‘IdM ABC System’, the latter will expect specific strings within the ExchangeData class to be defined. For instance, if the IdM Admin GUI launches revocation via the revokeCredential method, the ExchangeData field ‘revocationHandle’ must contain a valid value.

The ‘IdM ABC System’ is a GUI-less application which is not aware of the identity of the Users and which does not have access to the IdM Database. A Windows server hosts the ‘IdM ABC System’ as Apache Tomcat application (see Figure 21).



/Users/nsn/bin/apache-tomcat-7.0.39_64bit_Soderhamn/webapps/

Name	Ext	Size	Changed	Rights	Owner
..			23.09.2013 14:18:25	rw-rw-rw-	
ABC4TrustSystem			19.09.2013 11:22:39	rw-rw-rw-	
manager			07.05.2013 11:10:17	rw-rw-rw-	
host-manager			07.05.2013 11:10:17	rw-rw-rw-	
examples			07.05.2013 11:10:16	rw-rw-rw-	
docs			07.05.2013 11:10:09	rw-rw-rw-	
ROOT			07.05.2013 11:10:08	rw-rw-rw-	
ABC4TrustSystem.war		18.673.975	19.09.2013 10:19:04	rw-rw-rw-	

0 B of 18.236 KiB in 0 of 7

Figure 21: Söderhamn IdM ABC System Binary in Apache Tomcat Container

2.4.6.2 Integration

The integration of the ABCE and its CEs into the 'IdM ABC System' was performed on NSN's premises. An entire School Registration System was setup in a local integration testing environment. The goal was to perform end-to-end tests using HW and SW smart cards. Since the 'IdM ABC System' also contained Verifier functionality and since a copy of the Revocation Authority was installed on NSN's premises too, all tests from issuance to verification and finally to revocation could be performed. There was no dependency on a working and accessible Restricted Area. So bugs detected in this early phase could be fixed as soon as the root cause was identified. Swapping applications was very much faster on NSN's premises, as the developers had access to the company's Intranet.

Connecting to Privacy-ABC technologies was relatively easy. The binaries of new versions of the ABCE were made available in a project repository so that their compilation and build was not necessary. Swapping and updating the ABCEs and the CEs just meant modifying the version tag in the pom.xml file. The final version of the ABCE and its CEs in the 2nd round of the Söderhamn pilot was the soderhamn-1.0.16.

Extremely useful were the 'helper' methods supplied by WP4 for connecting to the Privacy-ABC technologies. Examples on how to use these 'helper' methods can be found in the 'sample application' source code and in the abce-services source code.

A servlet (i.e. the Init Servlet), which is automatically executed during startup, initializes the Issuer and the Verifier of the 'IdM ABC System'. Please note, that the credential specifications, the issuance policies and the presentation policies stored within the 'IdM ABC System' directories are static and will not change during the pilot operation. Therefore the Issuer and the Verifier only need to be initialized once during power up. There was no need to add new credential specifications or to add new Issuer parameters during operation. And there was no need to modify or add presentation policies. Nevertheless it must be stated here that the ABCE libraries include methods for modifying these values on-the-fly.

2.4.6.2.1 Deployed ABCE Methods

In the following paragraphs, an overview will be given on which methods the ‘IdM ABC System’ deployed in order to perform as Issuer and as Verifier concurrently.

All of the following methods have been provided by the technical work package WP4 of the ABC4Trust project. Since WP4 is the main audience of this deliverable, details on the functionality of these methods and their parameters will not be discussed in the following sections. These methods represent a snapshot of the ABC4Trust code which is not identical to the code available in GitHub. Adopters interested in integrating the ABCE into their design are recommended to download the newest version of the p2abcengine from GitHub and use software development kits like NetBeans or Eclipse to get acquainted with its methods.

2.4.6.2.1.1 Init Servlet (automatically executed during startup)

The IdM ABC System contains a set of servlets which offer Privacy-ABC services. Upon startup, the Init Servlet is automatically executed. The task of the Init Servlet is to prepare the Issuer and the Verifier for operation.

- eu.abc4trust.ri.servicehelper.issuer.SpecAndPolicy

  The SpecAndPolicy method maps a specific credential specification to a specific issuance policy and adds the URN of the revocation authority and one or more language-specific parameters to it.

  SpecAndPolicy school = new SpecAndPolicy (CommonDefinitions.SPEC_AND_POLICY_SCHOOL, "/xml/credspecs/credentialSpecificationSoderhamnSchool.xml", "/xml/issuance/issuancePolicySoderhamnSchool.xml", null, "urn:soderhamn:revocationauthority:default", "en", "Norrtullskolan", "sv", "Norrtullskolan");

  The last 2 pairs of parameters will be used as parameters in the ‘Identity Selector’ of the user interface (UI).

  > The UI generates, based on the SpecAndPolicy parameters and on the credential specification, the following sentence in a pop-up window:
  > "You own a valid {0} from {1}" whereas
  > {0} is the FriendlyCredentialName from the credential specification and
  > {1} is derived from a language-specific parameter of the SpecAndPolicy
  >
  > In the case of a browser set to ‘English’, the following sentence will be output:
  > "You own a valid School Credential from Norrtulskolan".
  >
  > The English friendly name in the credential specification {0} is "School Credential" and the English parameter in the SpecAndPolicy{1} is "Norrtullskolan".

  The credential specifications in XML (in sum 6) and the corresponding issuance policies in XML (in sum 12, because of reIssuance) are stored in directories within the ‘IdM ABC System’ application.

- eu.abc4trust.ri.servicehelper.issuer.IssuanceHelper

  IssuanceHelper.**initInstanceBridged**(2048, 2048, 50, systemAndIssuerParamsPrefix, fileStoragePrefix, new SpecAndPolicy[]{school, subject, role, guardian, child, grade, school_reIssuance, subject_reIssuance, role_reIssuance, guardian_reIssuance, child_reIssuance, grade_reIssuance}, revocationAuthorityParameters_resources);

  This is the main method for setting up the Issuer. If no Issuer parameters are available in the private and public storage areas of the 'IdM ABC System', this method triggers generating them. If the revocation authority parameters are available as a file in the parameters directory, they will be taken into account too. Please note, that in the Söderhamn pilot, the key lengths for the Idemix (first parameter) and U-Prove (second parameter) CEs are set to 2048. The amount of U-Prove tokens (third parameter), is fixed to 50. This means, that if a User already presented 50 U-Prove tokens, the User Application would automatically launch reIssuance. This method is fed with a pointer to the private storage area (systemAndIssuerParamsPrefix) and a public storage area (fileStoragePrefix). After that, a list of SpecAndPolicy follows finalized by a pointer to the revocation authority parameters.

- eu.abc4trust.ri.servicehelper.smartcard.PKIKeyTool

  PKIKeyTool.generateSignatureKeys(ISSUER_SYSTEM_FOLDER, "pki_keys");

  The pki keys are not required by the Issuer. These keys are necessary for setting up the smart cards. Since the Issuer and RevAuth parameters have to be distributed to the Users and Verifiers, the 'IdM ABC System' also generated the pki keys for the convenience of the project partners responsible for the operation phase.

- eu.abc4trust.ri.servicehelper.verifier.VerificationHelper

  VerificationHelper.**initInstance**(ProductionModuleFactory.CryptoEngine.BRIDGED, issuerParamsResourceList, credSpecResourceList, inspectorPublicKeyResourceList, revocationAuthorityParameters_resources, fileStoragePrefix, presentationPolicyResourceList);

  This is the main method for setting up the Verifier. The Verifier requires a directory containing the Issuer parameters (second parameter), a list of credential specifications in XML (third parameter), one or more Inspectors' public keys, the revocation authority parameters, a pointer to the private storage area and finally, a list of presentation policies in XML (last parameter),. Since the Verifier in the Söderhamn pilot needs to handle both Idemix and U-Prove presentation tokens, the first parameter must be set to 'BRIDGED'.

2.4.6.2.1.2 Issuance (executed on request during operation)

The IdM ABC System can launch issuance on the fly via dedicated servlets. For every credential, a set of servlets is defined whereas each servlet calls only one of the four (initIssuance, issueStep, initReIssuance and reIssueStep) methods described below. See also Section 2.4.6.1.

- eu.abc4trust.ri.servicehelper.issuer.IssuanceHelper
  eu.abc4trust.xml.IssuanceMessage
  eu.abc4trust.xml.IssuanceMessageAndBoolean

IssuanceMessage im_with_policy = IssuanceHelper.getInstance().**initIssuance**(cryptoEngine, specAndPolicyId, attributeValueMap);

This method launches the first step of issuance. Idemix requires 2 steps for issuance; U-Prove requires 3 steps. Important for the development of the 'IdM ABC System', is that for the initial issuance (not reIssuance) the Issuer must know in advance if an Idemix or a U-Prove credential must be issued (see first parameter). In order to solve this problem, the 'Smart Card Registrar' was enhanced. Before storing the smart card's scope-exclusive pseudonym in the IdM Database via the Smart Card Registrar, the administrator must select the CE type in the GUI. Please note that if the administrator selects by mistake the wrong CE type, issuance will always fail for this smart card.

The last parameter is a map of User attributes which will be inserted into the credential. The 'ABC-System' itself does not know any attribute values of Users. Contrary to this, the IdM Portal, who authenticated the User via the IdM Application, is aware of all attribute values of the User requesting for a credential and forwards those attributes, relevant for this specific credential, to the 'IdM ABC System'.

IssuanceMessageAndBoolean response_message = IssuanceHelper.getInstance().**issueStep**(cryptoEngine, issuanceMessage);

Even in the 2nd and last steps of issuance, the ABCE must be fed with the CE type (i.e. Idemix or U-Prove). This mandates an awareness of the School Registration System of which smart card the User is holding. In the case of reIssuance, the CE type needs not be provided.

The issueStep method returns a response_message of type IssuanceMessageAndBoolean. The IssuanceMessage can be extracted with the embedded method: response_message.getIssuanceMessage(). Using XmlUtils.toXml, the IssuanceMessage can be easily converted into a string.

With the embedded method response_message.isLastMessage(), the application has an indication if this message was the final message of the issuance handshake. The 'IdM ABC System' forwards this indication to the IdM Portal by appending this flag to the XML string.

IssuanceMessage im_with_policy = IssuanceHelper.getInstance().**initReIssuance**(specAndPolicyId);

This method launches the first step of reIssuance. If a User owns a U-Prove smart card and has already made use of all issued tokens (i.e. 50), her User Application will automatically launch reIssuance. Please note that the CE type does not need to be provided. The issuance policy in this case will require the User to reveal all attribute values of her old U-Prove credential (including, if relevant, the revocation handle).

IssuanceMessageAndBoolean response_message = IssuanceHelper.getInstance().**reIssueStep**(issuanceMessage);

This method launches the 2nd and last steps of reIssuance.

- eu.abc4trust.xml.ObjectFactory
  eu.abc4trust.xml.IssuanceMessage
  eu.abc4trust.xml.util.XmlUtils

ObjectFactory of = new ObjectFactory();

JAXBElement<IssuanceMessage> imJAXB = of.**createIssuanceMessage**(im_with_policy);

This method generates a JAXBElement<IssuanceMessage> from the issuance message class. A JAXBElement can be easily transformed into a string using XmlUtils.toXml. In the 'IdM ABC System', the JAXBElement formats are only used for convenience if classes needed to be transformed into strings.

IssuanceMessage issuanceMessage = (IssuanceMessage) XmlUtils.**getObjectFromXML**(new ByteArrayInputStream(exData.messageString.getBytes("UTF-8")), false);

The 'IdM ABC System' web service method responsible for issuance receives an issuance message stored in a sting. Using the above method, this string is converted into an object and then casted into the IssuanceMessage class. The IssuanceMessage class is required as input for the issueStep method.

2.4.6.2.1.3 Verification (executed on request during operation)

Two servlets of the IdM ABC System provide verification functionality. One servlet returns the presentation policy alternatives and the other verifies the presentation token. See also Section 2.4.6.1.

- eu.abc4trust.ri.servicehelper.verifier.VerificationHelper
  eu.abc4trust.xml.PresentationPolicyAlternatives

byte[] nonce = VerificationHelper.getInstance().**generateNonce**();

This method is used by the Verifier to generate a nonce which will replace the nonce placeholder in the presentation policy.

PresentationPolicyAlternatives ppa = VerificationHelper.getInstance().**createPresentationPolicy**(exData.policyId, nonce, null, null);

With the policy ID selected by the web client (e.g. the IdM Application), and with a fresh nonce generated by the 'IdM ABC System', a presentation policy alternatives class will be generated. This class will be transformed into a JAXBElement, then into string via XmlUtils.toXml and is finally forwarded to the web client.

- eu.abc4trust.xml.ObjectFactory
  eu.abc4trust.xml.PresentationPolicyAlternatives
  eu.abc4trust.xml.PresentationToken
  eu.abc4trust.xml.util.XmlUtils

PresentationPolicyAlternatives ppa = (PresentationPolicyAlternatives) XmlUtils.**getObjectFromXML**(new ByteArrayInputStream(exData.presentationPolicyAlternatives.getBytes("UTF-8")), false);

PresentationToken pT = (PresentationToken) XmlUtils.**getObjectFromXML**(new ByteArrayInputStream(exData.presentationToken.getBytes("UTF-8")), false);

In the verifyToken web service method, the presentation policy alternative and the presentation token, both presented by the User, will be converted to the corresponding classes.

boolean token_ok = VerificationHelper.getInstance().**verifyToken**(ppa, pT);

The verifyToken method analyses the two classes and returns a boolean true in case of success.

2.4.6.2.1.4 Revocation

One servlet of the IdM ABC System is dedicated to launching revocation.

- eu.abc4trust.xml.RevocationMessage

  Client client = Client.create();

  WebResource.Builder revokeResource = client.resource(revocationURL).accept(MediaType.APPLICATION_XML);

  RevocationMessage rmFromRevocation = revokeResource.**post**(RevocationMessage.class);

  The URL of this POST message contains revocation handle.

- eu.abc4trust.xml.RevocationInformation

  RevocationInformation riFromRevocation = ((JAXBElement<RevocationInformation>) rmFromRevocation.**getCryptoParams**().getAny().get(0)).getValue();

  The revocation message returned in the previous POST message contains the revocation information.

  String result = riFromRevocation.**getInformationUID**().toString();

  If the revocation information UID can be extracted from the revocation information, the revocation itself was successful.

2.4.6.2.2 Setting up the IdM ABC System

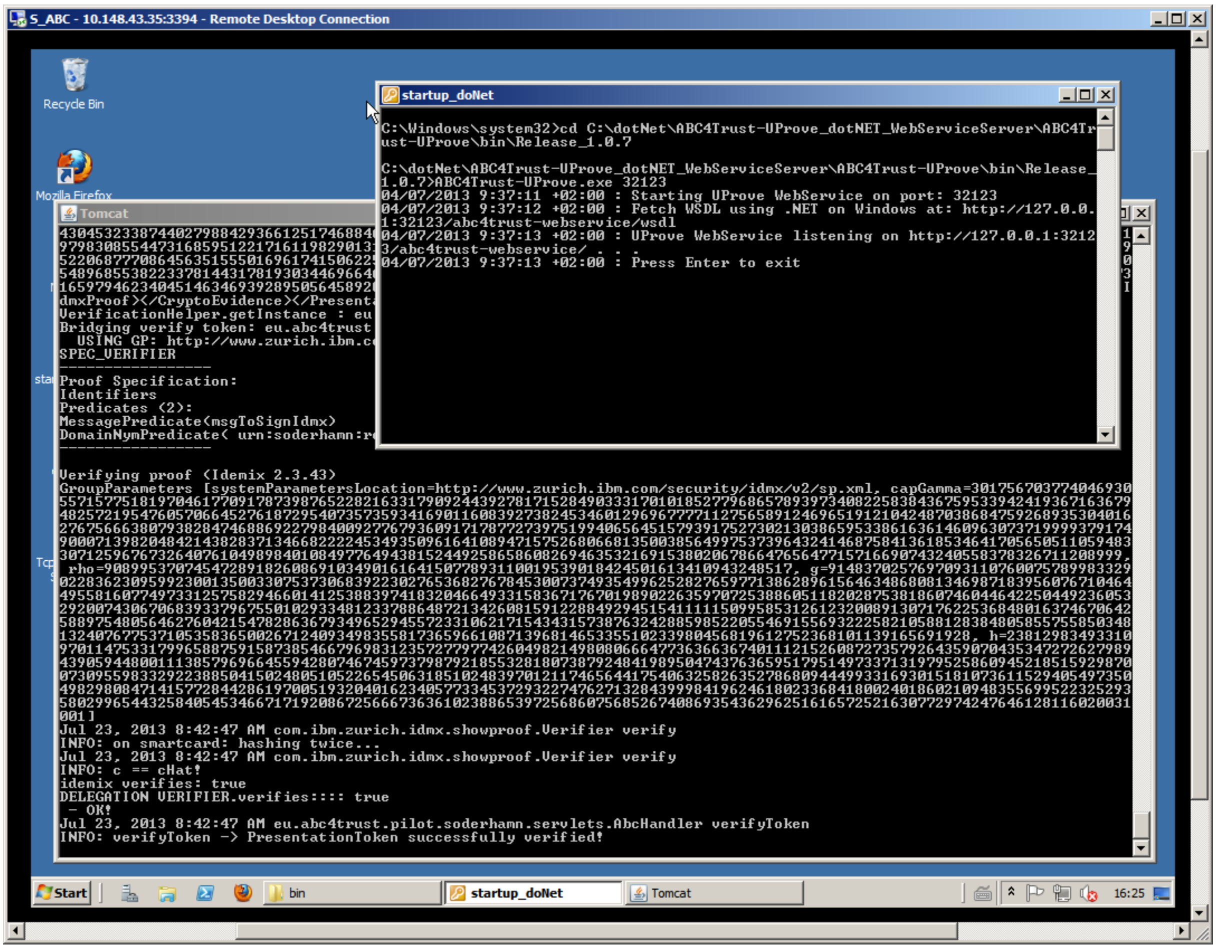
2.4.6.2.2.1 .NET

The U-Prove parts of the ABCE of the old crypto architecture require .NET so that the ABC4Trust-UProve executable file, supplied in the ABCE packet, can run. Next to the Microsoft .NET Framework 4.5, the U-Prove libraries and the executable were therefore installed on the Windows server hosting the ‘IdM ABC System’.

In the configuration file of ABC4Trust-UProve (ABC4Trust-UProve.exe.config) the timeprofile flag can be activated which leads to timing information being dumped in a ‘uprove-timeprofile-log.log’ file. On top of this, the logToFile flag can be activated, which leads to U-Prove logs being forwarded to a ‘uprove-log.log’ file. The aforementioned file names are default names and can be modified by the administrator.

The environment variable PathToUProveExe must be set to point to the directory storing the U-Prove binaries.

ABC4Trust-UProve is started in a command prompt window specifying inline the web service port (see Figure 22).



```
C:\Windows\system32>cd C:\dotNet\ABC4Trust-UProve_dotNET_WebServiceServer\ABC4Tr
ust-UProve\bin\Release_1.0.7

C:\dotNet\ABC4Trust-UProve_dotNET_WebServiceServer\ABC4Trust-UProve\bin\Release_
1.0.7>ABC4Trust-UProve.exe 32123
04/07/2013 9:37:11 +02:00 : Starting UProve WebService on port: 32123
04/07/2013 9:37:12 +02:00 : Fetch WSDL using .NET on Windows at: http://127.0.0.
1:32123/abc4trust-webservice/wsdl
04/07/2013 9:37:13 +02:00 : UProve WebService listening on http://127.0.0.1:3212
3/abc4trust-webservice/ . . .
04/07/2013 9:37:13 +02:00 : Press Enter to exit
```

```
4304532338744027988429366125174688
9798308554473168595122171611982901
5220687770864563515550169617415062
5489685538223378144317819303446966
1659794623404514634693928950564589
dmxProof></CryptoEvidence></Presenta
VerificationHelper.getInstance : eu
Bridging verify token: eu.abc4trust
  USING GP: http://www.zurich.ibm.c
SPEC_VERIFIER
------------------
Proof Specification:
Identifiers
Predicates (2):
MessagePredicate(msgToSignIdmx)
DomainNymPredicate( urn:soderhamn:r
------------------

Verifying proof (Idemix 2.3.43)
GroupParameters [systemParametersLocation=http://www.zurich.ibm.com/security/idmx/v2/sp.xml, capGamma=301756703774046930
557157751819704617709178739876522821633179092443927817152849033317010185277968657893973408225838436759533942419367163679
482572195476057066452761872954073573593416901160839273824534601296967777112756589124696519121042487038684759268935304016
276756663807938284746886922798400927767936091717877273975199406564515793917527302130386595338616361460963073719999379174
900071398204842143828371346682222453493509616410894715752680668135003856499753739643241468758413618534641705650511059483
30712596767326407610498984010849776494381524492586586082694635321691538020678664765647715716690743240558378326711208999,
 rho=90899537074547289182608691034901616415077893110019539018424501613410943248517, g=9148370257697093110760075789983329
022836230959923001350033075373068392230276536827678453007374935499625282765977138628961564634868081346987183956076710464
495581607749733125758294660141253883974183204664933158367176701989022635970725388605118202875381860746044642250449236053
292007430670683933796755010293348123378864872134260815912288492945154111115099585312612320089130717622536848016374670642
588975480564627604215478286367934965294557233106217154343157387632428859852205546915569322258210588128384805855755850348
132407677537105358365002671240934983558173659661087139681465335510233980456819612752368101139165691928, h=23812983493310
970114753317996588759158738546679698312357277977426049821498080666477363663674011121526087273579264359070435347272627989
439059448001113857969664559428074674597379879218553281807387924841989504743763659517951497337131979525860945218515929870
073095598332922388504150248051052265450631851024839701211746564417540632582635278680944499331693015181073611529405497350
498298084714157728442861970051932040162340577334537293227476271328439998419624618023368418002401860210948355699522325293
580299654432584054534667171920867256667363610238865397256860756852674086935436296251616572521630772974247646128116020031
001]
Jul 23, 2013 8:42:47 AM com.ibm.zurich.idmx.showproof.Verifier verify
INFO: on smartcard: hashing twice...
Jul 23, 2013 8:42:47 AM com.ibm.zurich.idmx.showproof.Verifier verify
INFO: c == cHat!
idemix verifies: true
DELEGATION VERIFIER.verifies:::: true
 - OK!
Jul 23, 2013 8:42:47 AM eu.abc4trust.pilot.soderhamn.servlets.AbcHandler verifyToken
INFO: verifyToken -> PresentationToken successfully verified!
```

Figure 22: Söderhamn IdM ABC System - Remote Desktop Log Windows

2.4.6.2.2.2 SSH Server

In order to enable remote administration of the Windows server, an SSH server had to be installed.

2.4.6.2.2.3 Apache Tomcat

An Apache-Tomcat 7 container was installed on the Windows server to be able to host the ‘IdM ABC System’ which is a Tomcat application. In configuration file of Tomcat ‘server.xml’, the connector’s settings for port 8080 were enhanced to support UTF-8 URI encoding with the following setting:

```
URIEncoding="UTF-8"
```

HTTPS was not used by the ‘IdM ABC System’, as the web service methods of this system were not accessible via the Internet.

Apache-Tomcat is started via the startup.bat script. This script generates a Tomcat command prompt window which outputs System.out.print logs (see Figure 22).

2.4.6.2.2.4 Windows Server Firewall

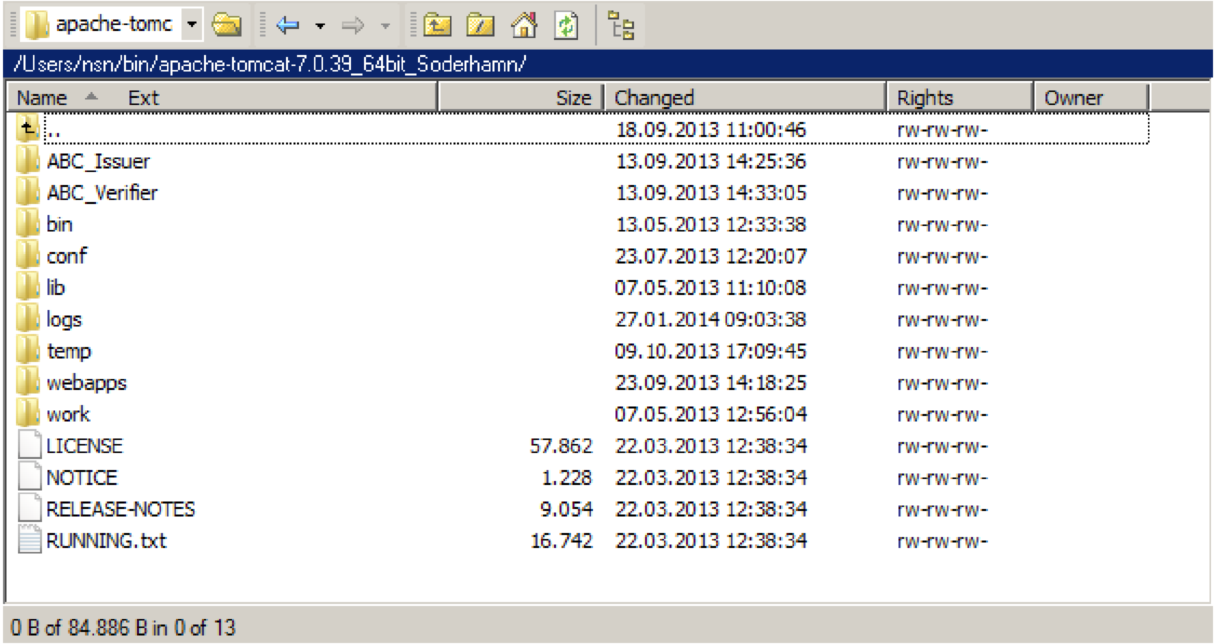
Port 8080 had to be added to the inbound rules of the Windows server firewall in order to allow the web clients (IdM Portal, IdM Application, etc) to access the Tomcat application ‘IdM ABC System’.

The port the SSH server was listening on had to be added to the inbound rules too, allowing administrative access from specific trusted IP addresses.

Finally, for accessing the Remote Desktop (Windows RDC), the corresponding port 3389 must be opened to enable remote administration.

2.4.6.2.2.5 Public and Private Storage Directories

During the first startup of the ‘IdM ABC System’, the application sets up the public and private storage directories and generates and distributes artifacts into these directories. Figure 23 depicts the first level of the storage directories separated by the ABC roles (ABC_Issuer and ABC_Verifier). The storage directories are located on the hard disc of the Windows server and are decoupled from the application directory ‘webapps’. Therefore, exchanging the ‘IdM ABC System’ application does not lead to a loss of the information gathered so far in the storage directories.



/Users/nsn/bin/apache-tomcat-7.0.39_64bit_Soderhamn/

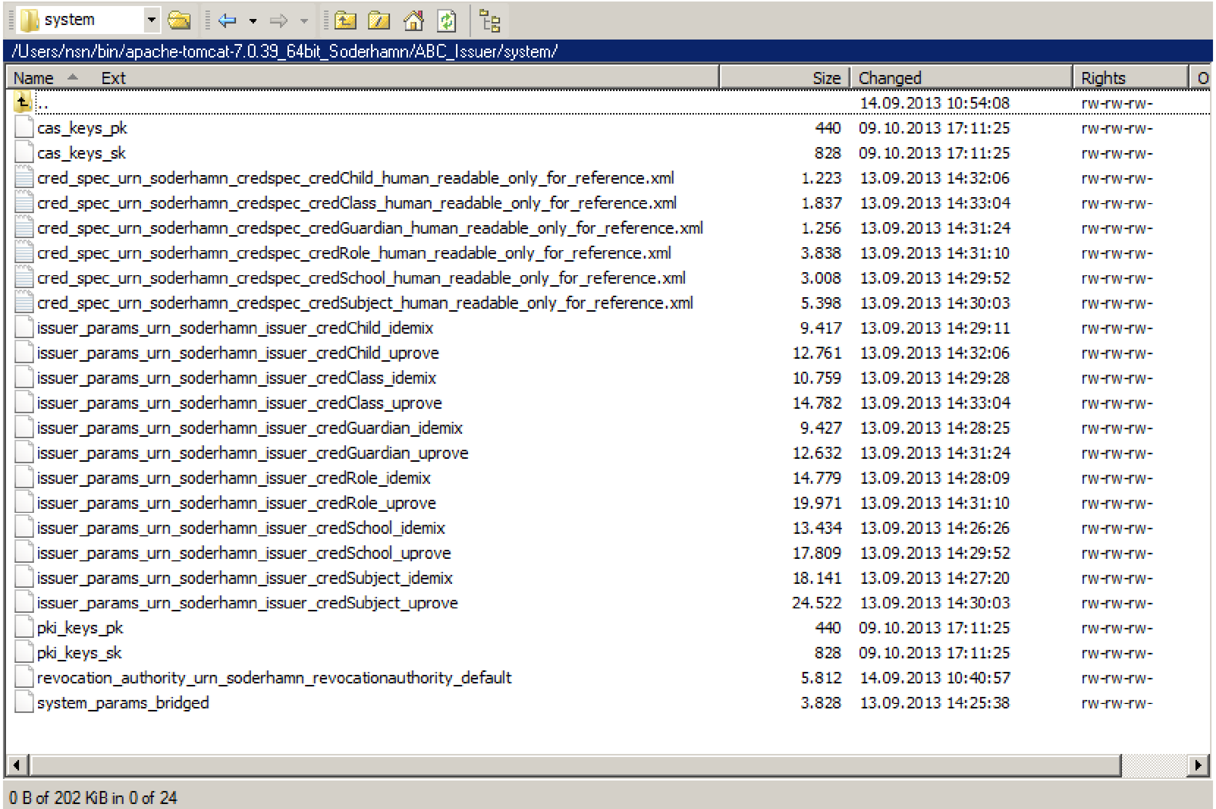
Name	Ext	Size	Changed	Rights	Owner
..			18.09.2013 11:00:46	rw-rw-rw-	
ABC_Issuer			13.09.2013 14:25:36	rw-rw-rw-	
ABC_Verifier			13.09.2013 14:33:05	rw-rw-rw-	
bin			13.05.2013 12:33:38	rw-rw-rw-	
conf			23.07.2013 12:20:07	rw-rw-rw-	
lib			07.05.2013 11:10:08	rw-rw-rw-	
logs			27.01.2014 09:03:38	rw-rw-rw-	
temp			09.10.2013 17:09:45	rw-rw-rw-	
webapps			23.09.2013 14:18:25	rw-rw-rw-	
work			07.05.2013 12:56:04	rw-rw-rw-	
LICENSE		57.862	22.03.2013 12:38:34	rw-rw-rw-	
NOTICE		1.228	22.03.2013 12:38:34	rw-rw-rw-	
RELEASE-NOTES		9.054	22.03.2013 12:38:34	rw-rw-rw-	
RUNNING.txt		16.742	22.03.2013 12:38:34	rw-rw-rw-	

0 B of 84.886 B in 0 of 13

Figure 23: Söderhamn IdM ABC System - Issuer and Verifier Storage Directories

Figure 24 depicts the public storage area of the Issuer. The files in this area are distributed to the Verifiers and to the Users. The private storage area of the Issuer is shown in Figure 25. Both the public and the private storage areas contain Issuer parameter files of both CE types for each credential.

So in effect there are 12 private Issuer parameters in the files folder and 12 public Issuer parameters in the system folder.

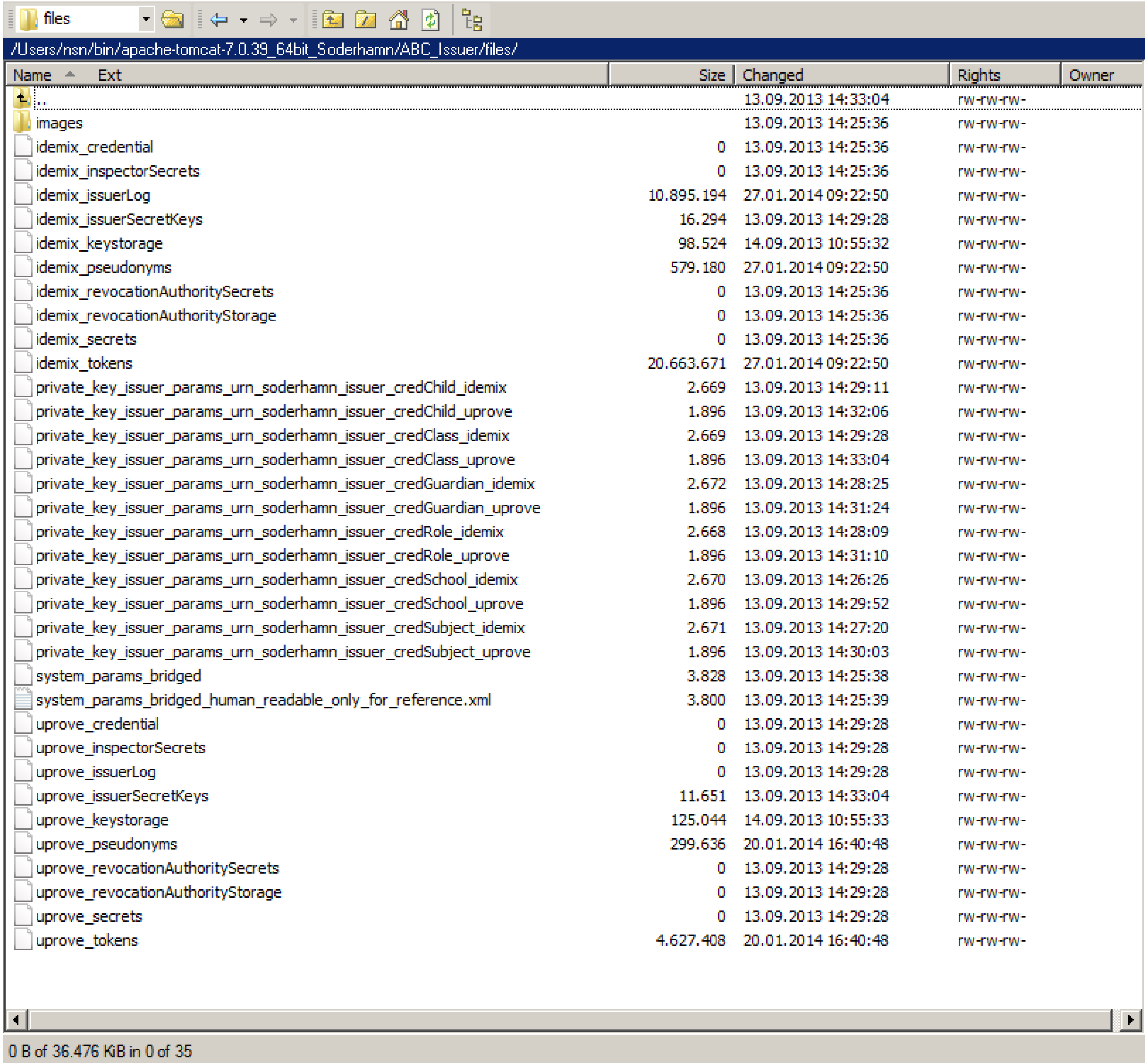


/Users/nsn/bin/apache-tomcat-7.0.39_64bit_Soderhamn/ABC_Issuer/system/

Name Ext	Size	Changed	Rights
..		14.09.2013 10:54:08	rw-rw-rw-
cas_keys_pk	440	09.10.2013 17:11:25	rw-rw-rw-
cas_keys_sk	828	09.10.2013 17:11:25	rw-rw-rw-
cred_spec_urn_soderhamn_credspec_credChild_human_readable_only_for_reference.xml	1.223	13.09.2013 14:32:06	rw-rw-rw-
cred_spec_urn_soderhamn_credspec_credClass_human_readable_only_for_reference.xml	1.837	13.09.2013 14:33:04	rw-rw-rw-
cred_spec_urn_soderhamn_credspec_credGuardian_human_readable_only_for_reference.xml	1.256	13.09.2013 14:31:24	rw-rw-rw-
cred_spec_urn_soderhamn_credspec_credRole_human_readable_only_for_reference.xml	3.838	13.09.2013 14:31:10	rw-rw-rw-
cred_spec_urn_soderhamn_credspec_credSchool_human_readable_only_for_reference.xml	3.008	13.09.2013 14:29:52	rw-rw-rw-
cred_spec_urn_soderhamn_credspec_credSubject_human_readable_only_for_reference.xml	5.398	13.09.2013 14:30:03	rw-rw-rw-
issuer_params_urn_soderhamn_issuer_credChild_idemix	9.417	13.09.2013 14:29:11	rw-rw-rw-
issuer_params_urn_soderhamn_issuer_credChild_uprove	12.761	13.09.2013 14:32:06	rw-rw-rw-
issuer_params_urn_soderhamn_issuer_credClass_idemix	10.759	13.09.2013 14:29:28	rw-rw-rw-
issuer_params_urn_soderhamn_issuer_credClass_uprove	14.782	13.09.2013 14:33:04	rw-rw-rw-
issuer_params_urn_soderhamn_issuer_credGuardian_idemix	9.427	13.09.2013 14:28:25	rw-rw-rw-
issuer_params_urn_soderhamn_issuer_credGuardian_uprove	12.632	13.09.2013 14:31:24	rw-rw-rw-
issuer_params_urn_soderhamn_issuer_credRole_idemix	14.779	13.09.2013 14:28:09	rw-rw-rw-
issuer_params_urn_soderhamn_issuer_credRole_uprove	19.971	13.09.2013 14:31:10	rw-rw-rw-
issuer_params_urn_soderhamn_issuer_credSchool_idemix	13.434	13.09.2013 14:26:26	rw-rw-rw-
issuer_params_urn_soderhamn_issuer_credSchool_uprove	17.809	13.09.2013 14:29:52	rw-rw-rw-
issuer_params_urn_soderhamn_issuer_credSubject_idemix	18.141	13.09.2013 14:27:20	rw-rw-rw-
issuer_params_urn_soderhamn_issuer_credSubject_uprove	24.522	13.09.2013 14:30:03	rw-rw-rw-
pki_keys_pk	440	09.10.2013 17:11:25	rw-rw-rw-
pki_keys_sk	828	09.10.2013 17:11:25	rw-rw-rw-
revocation_authority_urn_soderhamn_revocationauthority_default	5.812	14.09.2013 10:40:57	rw-rw-rw-
system_params_bridged	3.828	13.09.2013 14:25:38	rw-rw-rw-

0 B of 202 KiB in 0 of 24

Figure 24: Söderhamn IdM ABC System - Public Storage Area of the Issuer



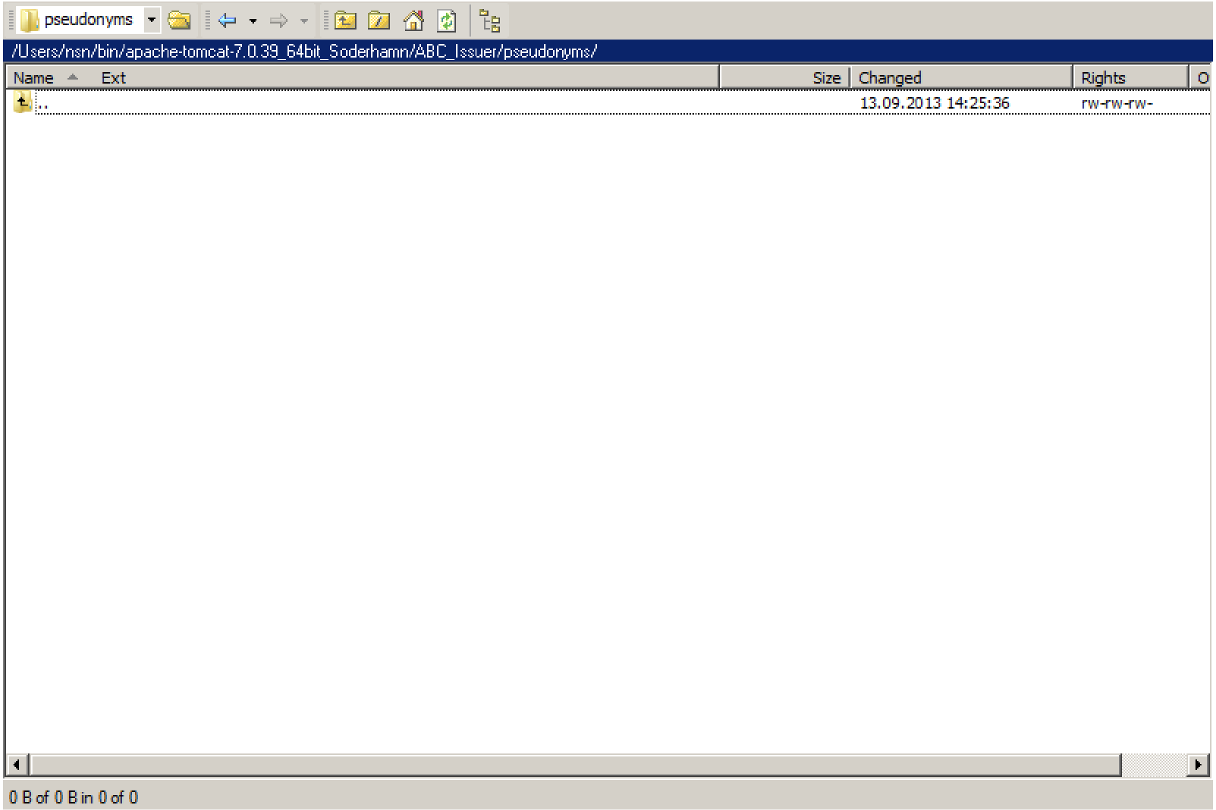
/Users/nsn/bin/apache-tomcat-7.0.39_64bit_Soderhamn/ABC_Issuer/files/

Name	Ext	Size	Changed	Rights	Owner
..			13.09.2013 14:33:04	rw-rw-rw-	
images			13.09.2013 14:25:36	rw-rw-rw-	
idemix_credential		0	13.09.2013 14:25:36	rw-rw-rw-	
idemix_inspectorSecrets		0	13.09.2013 14:25:36	rw-rw-rw-	
idemix_issuerLog		10.895.194	27.01.2014 09:22:50	rw-rw-rw-	
idemix_issuerSecretKeys		16.294	13.09.2013 14:29:28	rw-rw-rw-	
idemix_keystorage		98.524	14.09.2013 10:55:32	rw-rw-rw-	
idemix_pseudonyms		579.180	27.01.2014 09:22:50	rw-rw-rw-	
idemix_revocationAuthoritySecrets		0	13.09.2013 14:25:36	rw-rw-rw-	
idemix_revocationAuthorityStorage		0	13.09.2013 14:25:36	rw-rw-rw-	
idemix_secrets		0	13.09.2013 14:25:36	rw-rw-rw-	
idemix_tokens		20.663.671	27.01.2014 09:22:50	rw-rw-rw-	
private_key_issuer_params_urn_soderhamn_issuer_credChild_idemix		2.669	13.09.2013 14:29:11	rw-rw-rw-	
private_key_issuer_params_urn_soderhamn_issuer_credChild_uprove		1.896	13.09.2013 14:32:06	rw-rw-rw-	
private_key_issuer_params_urn_soderhamn_issuer_credClass_idemix		2.669	13.09.2013 14:29:28	rw-rw-rw-	
private_key_issuer_params_urn_soderhamn_issuer_credClass_uprove		1.896	13.09.2013 14:33:04	rw-rw-rw-	
private_key_issuer_params_urn_soderhamn_issuer_credGuardian_idemix		2.672	13.09.2013 14:28:25	rw-rw-rw-	
private_key_issuer_params_urn_soderhamn_issuer_credGuardian_uprove		1.896	13.09.2013 14:31:24	rw-rw-rw-	
private_key_issuer_params_urn_soderhamn_issuer_credRole_idemix		2.668	13.09.2013 14:28:09	rw-rw-rw-	
private_key_issuer_params_urn_soderhamn_issuer_credRole_uprove		1.896	13.09.2013 14:31:10	rw-rw-rw-	
private_key_issuer_params_urn_soderhamn_issuer_credSchool_idemix		2.670	13.09.2013 14:26:26	rw-rw-rw-	
private_key_issuer_params_urn_soderhamn_issuer_credSchool_uprove		1.896	13.09.2013 14:29:52	rw-rw-rw-	
private_key_issuer_params_urn_soderhamn_issuer_credSubject_idemix		2.671	13.09.2013 14:27:20	rw-rw-rw-	
private_key_issuer_params_urn_soderhamn_issuer_credSubject_uprove		1.896	13.09.2013 14:30:03	rw-rw-rw-	
system_params_bridged		3.828	13.09.2013 14:25:38	rw-rw-rw-	
system_params_bridged_human_readable_only_for_reference.xml		3.800	13.09.2013 14:25:39	rw-rw-rw-	
uprove_credential		0	13.09.2013 14:29:28	rw-rw-rw-	
uprove_inspectorSecrets		0	13.09.2013 14:29:28	rw-rw-rw-	
uprove_issuerLog		0	13.09.2013 14:29:28	rw-rw-rw-	
uprove_issuerSecretKeys		11.651	13.09.2013 14:33:04	rw-rw-rw-	
uprove_keystorage		125.044	14.09.2013 10:55:33	rw-rw-rw-	
uprove_pseudonyms		299.636	20.01.2014 16:40:48	rw-rw-rw-	
uprove_revocationAuthoritySecrets		0	13.09.2013 14:29:28	rw-rw-rw-	
uprove_revocationAuthorityStorage		0	13.09.2013 14:29:28	rw-rw-rw-	
uprove_secrets		0	13.09.2013 14:29:28	rw-rw-rw-	
uprove_tokens		4.627.408	20.01.2014 16:40:48	rw-rw-rw-	

0 B of 36.476 KiB in 0 of 35

Figure 25: Söderhamn IdM ABC System - Private Storage Area of the Issuer

The pseudonyms folder of the Issuer is not used by the ‘IdM ABC System’ and remains empty (see Figure 26). Instead, trusted pseudonyms are filtered out by the web client applications connecting to the ‘IdM ABC System’ (i.e.: the IdM Portal, the IdM Application, etc).



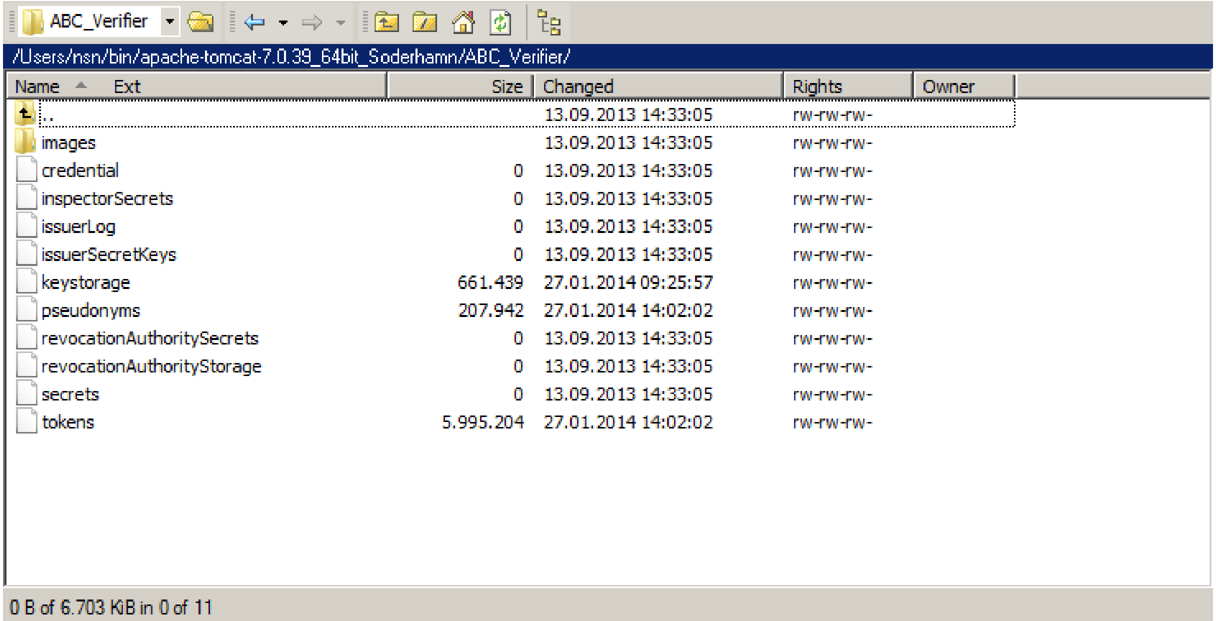
/Users/nsn/bin/apache-tomcat-7.0.39_64bit_Soderhamn/ABC_Issuer/pseudonyms/

Name	Ext	Size	Changed	Rights
..			13.09.2013 14:25:36	rw-rw-rw-

0 B of 0 B in 0 of 0

Figure 26: Söderhamn IdM ABC System - Pseudonyms Folder of the Issuer

The directory structure of the Verifier is much simpler when compared to the Issuer. The Verifier basically used the Issuer public storage area for initialization and generates its private files in its own private storage area (see Figure 27). Most of the files remain empty. So in effect, the Verifier only uses the keystorage, pseudonyms and token files.



/Users/nsn/bin/apache-tomcat-7.0.39_64bit_Soderhamn/ABC_Verifier/

Name	Ext	Size	Changed	Rights	Owner
..			13.09.2013 14:33:05	rw-rw-rw-	
images			13.09.2013 14:33:05	rw-rw-rw-	
credential		0	13.09.2013 14:33:05	rw-rw-rw-	
inspectorSecrets		0	13.09.2013 14:33:05	rw-rw-rw-	
issuerLog		0	13.09.2013 14:33:05	rw-rw-rw-	
issuerSecretKeys		0	13.09.2013 14:33:05	rw-rw-rw-	
keystorage		661.439	27.01.2014 09:25:57	rw-rw-rw-	
pseudonyms		207.942	27.01.2014 14:02:02	rw-rw-rw-	
revocationAuthoritySecrets		0	13.09.2013 14:33:05	rw-rw-rw-	
revocationAuthorityStorage		0	13.09.2013 14:33:05	rw-rw-rw-	
secrets		0	13.09.2013 14:33:05	rw-rw-rw-	
tokens		5.995.204	27.01.2014 14:02:02	rw-rw-rw-	

0 B of 6.703 KiB in 0 of 11

Figure 27: Söderhamn IdM ABC System - Private Storage Area of the Verifier

2.4.6.2.3 Initializing the Issuer and the Verifier

As stated before in this document, the Issuer and Verifier are initialized via the 'Init Servlet'. First of all, the Issuer is initialized which leads to a set of files in private and public storage areas of the Issuer. And then the Verifier is initialized, which accesses the public storage area of the Issuer and generates private Verifier files. It must be noted, that during the first initialization, the revocation parameters are not available.

In order to obtain the Revocation Authority parameters, the RevAuth application has to be setup with the Issuer parameters, credential specifications and system parameters generated by or provided for the first initialization of the Issuer. As soon as the RevAuth is powered up and the revocation/updaterevocationinformation RESTful interface is accessed, the RevAuth parameters are created and stored in a file located in /WEB-INF/classes/revocation_storage/. After this step, the 'IdM ABC System' must be stopped and the file containing the revocation parameters, i.e. 'revocation_authority_urn_soderhamn_revocationauthority_default' (see Figure 24), must be copied into the public storage area of the Issuer, i.e. the 'system' directory. Finally, the 'IdM ABC System' must be restarted. Now, when the initialization method of the Issuer is being executed, the revocation parameters will be loaded.

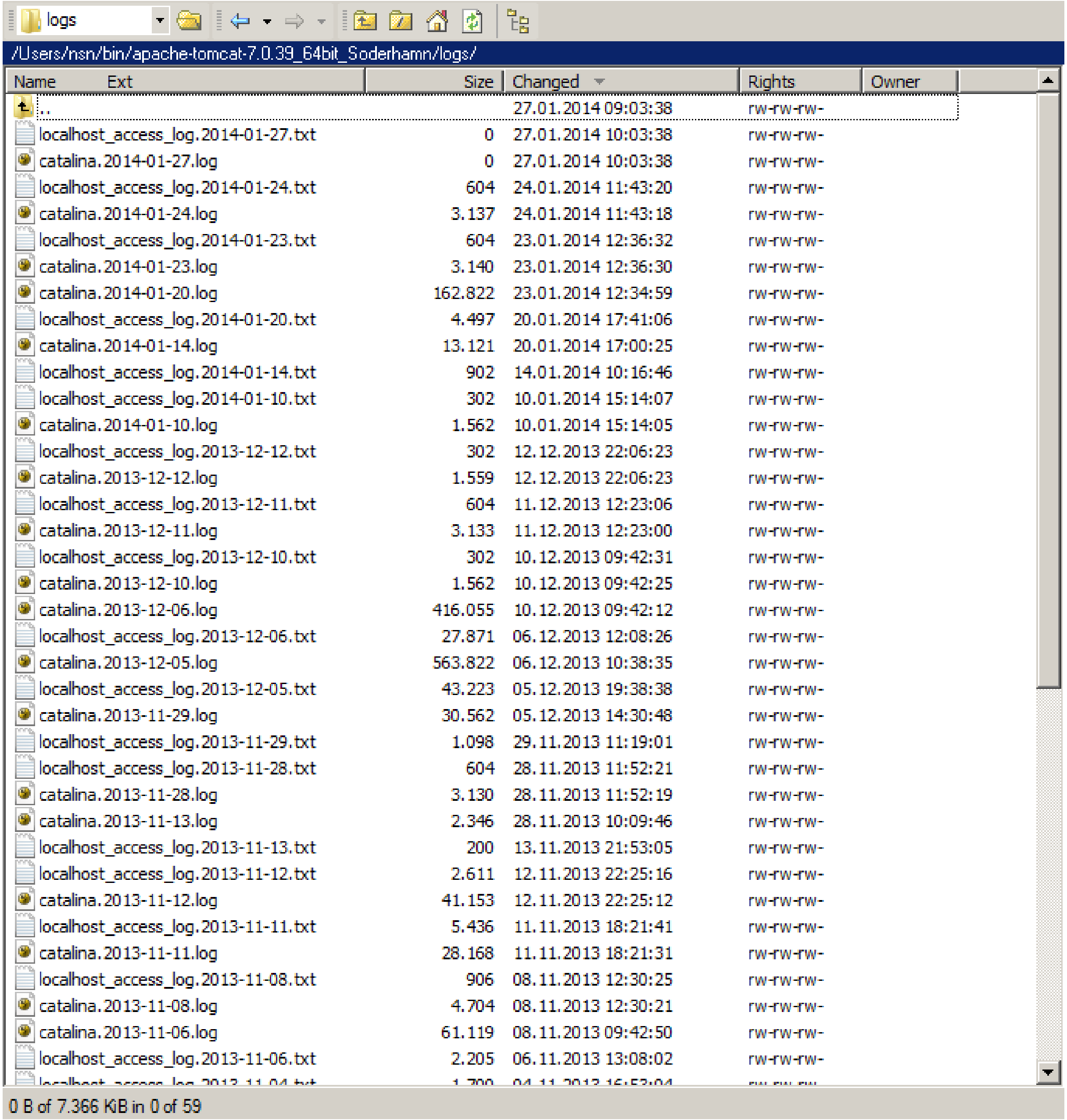
The above described procedure is complicated, time-consuming and not easy to execute. The Issuer is dependent on files produced by the Revocation Authority. And the Revocation Authority is dependent on files produced by the Issuer. So, in effect, 2 startups are required.

With a Revocation Authority and Inspectors active in the Söderhamn pilot, adding or modifying credential specifications have a wide impact. Not only must all steps be repeated. Since this impacts the public parameters of both the Issuer and the Revocation Authority, all Users (via the Installer), all Inspectors and all Verifiers are impacted too.

2.4.6.2.4 Debugging the IdM ABC System

The ‘IdM ABC System’ generates a set of log files. Next to the command prompt window output of the Tomcat and the ABC4Trust-UProve executable, the U-Prove log and timing files must be mentioned (see Subsection 2.4.6.2.2.1).

The Apache-Tomcat container generates in-depth logging in the ‘catalina log files’ triggered by the ‘IdM ABC System’ (see Subsection 2.4.6.2.2.3, Figure 28 and Figure 29). At this point, it must be mentioned, that the logs generated by the ABCE and its CEs are not included in the catalina log files so that the command prompt window of Apache-Tomcat must be viewed in parallel when debugging. The reason for this split is that the ABCE and the CEs are using the System.out.print methods for logging. Unfortunately, the buffer of the command prompt window is restricted, so that the logs of the ABCE and its CEs show only the latest snapshot of debugging information. The rest is lost. A sequential ordering of command prompt window content and catalina log content is difficult and partially impossible. This makes debugging sporadic errors significantly difficult.

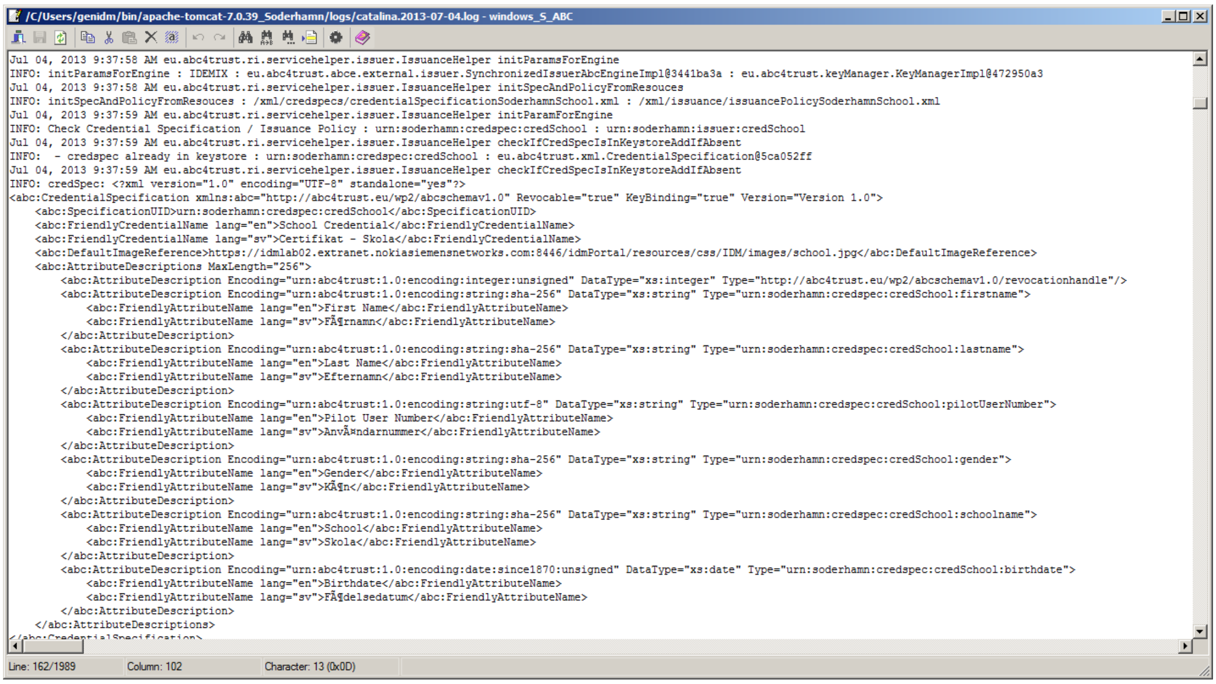


/Users/nsn/bin/apache-tomcat-7.0.39_64bit_Soderhamn/logs/

Name Ext	Size	Changed	Rights	Owner
..		27.01.2014 09:03:38	rw-rw-rw-	
localhost_access_log.2014-01-27.txt	0	27.01.2014 10:03:38	rw-rw-rw-	
catalina.2014-01-27.log	0	27.01.2014 10:03:38	rw-rw-rw-	
localhost_access_log.2014-01-24.txt	604	24.01.2014 11:43:20	rw-rw-rw-	
catalina.2014-01-24.log	3.137	24.01.2014 11:43:18	rw-rw-rw-	
localhost_access_log.2014-01-23.txt	604	23.01.2014 12:36:32	rw-rw-rw-	
catalina.2014-01-23.log	3.140	23.01.2014 12:36:30	rw-rw-rw-	
catalina.2014-01-20.log	162.822	23.01.2014 12:34:59	rw-rw-rw-	
localhost_access_log.2014-01-20.txt	4.497	20.01.2014 17:41:06	rw-rw-rw-	
catalina.2014-01-14.log	13.121	20.01.2014 17:00:25	rw-rw-rw-	
localhost_access_log.2014-01-14.txt	902	14.01.2014 10:16:46	rw-rw-rw-	
localhost_access_log.2014-01-10.txt	302	10.01.2014 15:14:07	rw-rw-rw-	
catalina.2014-01-10.log	1.562	10.01.2014 15:14:05	rw-rw-rw-	
localhost_access_log.2013-12-12.txt	302	12.12.2013 22:06:23	rw-rw-rw-	
catalina.2013-12-12.log	1.559	12.12.2013 22:06:23	rw-rw-rw-	
localhost_access_log.2013-12-11.txt	604	11.12.2013 12:23:06	rw-rw-rw-	
catalina.2013-12-11.log	3.133	11.12.2013 12:23:00	rw-rw-rw-	
localhost_access_log.2013-12-10.txt	302	10.12.2013 09:42:31	rw-rw-rw-	
catalina.2013-12-10.log	1.562	10.12.2013 09:42:25	rw-rw-rw-	
catalina.2013-12-06.log	416.055	10.12.2013 09:42:12	rw-rw-rw-	
localhost_access_log.2013-12-06.txt	27.871	06.12.2013 12:08:26	rw-rw-rw-	
catalina.2013-12-05.log	563.822	06.12.2013 10:38:35	rw-rw-rw-	
localhost_access_log.2013-12-05.txt	43.223	05.12.2013 19:38:38	rw-rw-rw-	
catalina.2013-11-29.log	30.562	05.12.2013 14:30:48	rw-rw-rw-	
localhost_access_log.2013-11-29.txt	1.098	29.11.2013 11:19:01	rw-rw-rw-	
localhost_access_log.2013-11-28.txt	604	28.11.2013 11:52:21	rw-rw-rw-	
catalina.2013-11-28.log	3.130	28.11.2013 11:52:19	rw-rw-rw-	
catalina.2013-11-13.log	2.346	28.11.2013 10:09:46	rw-rw-rw-	
localhost_access_log.2013-11-13.txt	200	13.11.2013 21:53:05	rw-rw-rw-	
localhost_access_log.2013-11-12.txt	2.611	12.11.2013 22:25:16	rw-rw-rw-	
catalina.2013-11-12.log	41.153	12.11.2013 22:25:12	rw-rw-rw-	
localhost_access_log.2013-11-11.txt	5.436	11.11.2013 18:21:41	rw-rw-rw-	
catalina.2013-11-11.log	28.168	11.11.2013 18:21:31	rw-rw-rw-	
localhost_access_log.2013-11-08.txt	906	08.11.2013 12:30:25	rw-rw-rw-	
catalina.2013-11-08.log	4.704	08.11.2013 12:30:21	rw-rw-rw-	
catalina.2013-11-06.log	61.119	08.11.2013 09:42:50	rw-rw-rw-	
localhost_access_log.2013-11-06.txt	2.205	06.11.2013 13:08:02	rw-rw-rw-	
localhost_access_log.2013-11-04.txt	1.700	04.11.2013 16:53:04	rw-rw-rw-	

0 B of 7.366 KiB in 0 of 59

Figure 28: Söderhamn IdM ABC System - Apache-Tomcat Log Files


/C/Users/genidm/bin/apache-tomcat-7.0.39_Soderhamn/logs/catalina.2013-07-04.log - windows_S_ABC
Line: 162/1989
Column: 102
Character: 13 (0x0D)

```
Jul 04, 2013 9:37:58 AM eu.abc4trust.ri.servicehelper.issuer.IssuanceHelper initParamsForEngine
INFO: initParamsForEngine : IDEMIX : eu.abc4trust.abce.external.issuer.SynchronizedIssuerAbcEngineImpl@3441ba3a : eu.abc4trust.keyManager.KeyManagerImpl@472950a3
Jul 04, 2013 9:37:58 AM eu.abc4trust.ri.servicehelper.issuer.IssuanceHelper initSpecAndPolicyFromResouces
INFO: initSpecAndPolicyFromResouces : /xml/credspecs/credentialSpecificationSoderhamnSchool.xml : /xml/issuance/issuancePolicySoderhamnSchool.xml
Jul 04, 2013 9:37:59 AM eu.abc4trust.ri.servicehelper.issuer.IssuanceHelper initParamForEngine
INFO: Check Credential Specification / Issuance Policy : urn:soderhamn:credspec:credSchool : urn:soderhamn:issuer:credSchool
Jul 04, 2013 9:37:59 AM eu.abc4trust.ri.servicehelper.issuer.IssuanceHelper checkIfCredSpecIsInKeystoreAddIfAbsent
INFO:  - credspec already in keystore : urn:soderhamn:credspec:credSchool : eu.abc4trust.xml.CredentialSpecification@5ca052ff
Jul 04, 2013 9:37:59 AM eu.abc4trust.ri.servicehelper.issuer.IssuanceHelper checkIfCredSpecIsInKeystoreAddIfAbsent
INFO: credSpec: <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<abc:CredentialSpecification xmlns:abc="http://abc4trust.eu/wp2/abcschemav1.0" Revocable="true" KeyBinding="true" Version="Version 1.0">
    <abc:SpecificationUID>urn:soderhamn:credspec:credSchool</abc:SpecificationUID>
    <abc:FriendlyCredentialName lang="en">School Credential</abc:FriendlyCredentialName>
    <abc:FriendlyCredentialName lang="sv">Certifikat - Skola</abc:FriendlyCredentialName>
    <abc:DefaultImageReference>https://idmlab02.extranet.nokiasiemensnetworks.com:8446/idmPortal/resources/css/IDM/images/school.jpg</abc:DefaultImageReference>
    <abc:AttributeDescriptions MaxLength="256">
        <abc:AttributeDescription Encoding="urn:abc4trust:1.0:encoding:integer:unsigned" DataType="xs:integer" Type="http://abc4trust.eu/wp2/abcschemav1.0/revocationhandle"/>
        <abc:AttributeDescription Encoding="urn:abc4trust:1.0:encoding:string:sha-256" DataType="xs:string" Type="urn:soderhamn:credspec:credSchool:firstname">
            <abc:FriendlyAttributeName lang="en">First Name</abc:FriendlyAttributeName>
            <abc:FriendlyAttributeName lang="sv">FÃ¶rnamn</abc:FriendlyAttributeName>
        </abc:AttributeDescription>
        <abc:AttributeDescription Encoding="urn:abc4trust:1.0:encoding:string:sha-256" DataType="xs:string" Type="urn:soderhamn:credspec:credSchool:lastname">
            <abc:FriendlyAttributeName lang="en">Last Name</abc:FriendlyAttributeName>
            <abc:FriendlyAttributeName lang="sv">Efternamn</abc:FriendlyAttributeName>
        </abc:AttributeDescription>
        <abc:AttributeDescription Encoding="urn:abc4trust:1.0:encoding:string:utf-8" DataType="xs:string" Type="urn:soderhamn:credspec:credSchool:pilotUserNumber">
            <abc:FriendlyAttributeName lang="en">Pilot User Number</abc:FriendlyAttributeName>
            <abc:FriendlyAttributeName lang="sv">AnvÃ¤ndarnummer</abc:FriendlyAttributeName>
        </abc:AttributeDescription>
        <abc:AttributeDescription Encoding="urn:abc4trust:1.0:encoding:string:sha-256" DataType="xs:string" Type="urn:soderhamn:credspec:credSchool:gender">
            <abc:FriendlyAttributeName lang="en">Gender</abc:FriendlyAttributeName>
            <abc:FriendlyAttributeName lang="sv">KÃ¶n</abc:FriendlyAttributeName>
        </abc:AttributeDescription>
        <abc:AttributeDescription Encoding="urn:abc4trust:1.0:encoding:string:sha-256" DataType="xs:string" Type="urn:soderhamn:credspec:credSchool:schoolname">
            <abc:FriendlyAttributeName lang="en">School</abc:FriendlyAttributeName>
            <abc:FriendlyAttributeName lang="sv">Skola</abc:FriendlyAttributeName>
        </abc:AttributeDescription>
        <abc:AttributeDescription Encoding="urn:abc4trust:1.0:encoding:date:since1870:unsigned" DataType="xs:date" Type="urn:soderhamn:credspec:credSchool:birthdate">
            <abc:FriendlyAttributeName lang="en">Birthdate</abc:FriendlyAttributeName>
            <abc:FriendlyAttributeName lang="sv">FÃ¶delsedatum</abc:FriendlyAttributeName>
        </abc:AttributeDescription>
    </abc:AttributeDescriptions>
</abc:CredentialSpecification>
```

Figure 29: Söderhamn IdM ABC System - An Example Content of the Catalina Log Files

2.4.6.3 Deployment

After completing all integration tests with the near-final version of the ABCE and its CEs, first steps were performed in order to be able to install the 'IdM ABC System' on the target site. EDOC provided a Windows server hosting an SSH server and configured their firewall to grant access for NSN's developers to the RDP and SSH ports of this system. In order to guarantee that the source IP address does not change over time, NSN's developers accessed the 'IdM ABC System' of the target site via their integration testing environment and not via their company laptops. This way, EDOC's firewall needed only to be configured with the static and public IP address of NSN's integration testing environment.

As soon as EDOC provided accounts for the SSH and the RDP access, NSN checked the connectivity and began the installation of other required software. Next to Java and .NET, Apache-Tomcat had to be installed. Before the target site was ready to pick up the 'IdM ABC System', the application had to be 'customized'. The latter was no minor task. Shifting the School Registration System applications from the integration testing environment impacts the credential specifications. Please note, that the credential specifications contain a 'DefaultImageReference' tag which points to the location where the User can download the credential image which will be displayed in her UI. Since the location of this image must shift from the integration testing environment to the target site too, all credential specifications must be modified. Another impact is the Issuer parameters. Since the Revocation Authority shifted its location, the Issuer parameters of revocable credentials have to change. On top of that, the Revocation Authority itself will generate different Revocation Authority Parameters. In essence, most of the files in the public and private storage areas of the 'IdM ABC System' must be re-generated.

The 'IdM ABC System' is dependent on the web service clients (i.e. the IdM Portal, the IdM Application, etc.) connecting to it. As soon as the entire School Registration System was installed at EDOC's target site, NSN performed some system tests on the 'IdM ABC System' in order to verify that the customization succeeded. After that, test User accounts were made available for WP4 and for EDOC so that all entities of the Söderhamn pilot (including the Restricted Area System) can be tested.

EDOC and NSN executed the system tests. Before the operation phase of the pilot, NSN invited to a formal handover of the School Registration System and the Revocation Authority. EDOC thereafter configured their firewall to block traffic coming from NSN. After 'cutting the lines', the pilot entered the operation state.

2.4.6.4 Operation

The IdM ABC System was hosted on a separate virtual machine at URL http://abce.abc4trust.se that was accessible only from the server hosting the IdM Application. As soon as the IdM ABC System was initialized by NSN without any exceptions it was ready to receive and respond on issuance and verification requests. During the operation phase no issues regarding the IdM ABC System came up.

During the 1st round the Users received smart cards from EDOC that had only been initialized and therefore did not contain any credentials. The administrators registered all smart cards using the Smart Card Registrar. And Users registered their smart cards and downloaded their credentials using the IdM Portal. Users with registered and valid smart cards in their possession were successfully involved in the verification process (i.e. logging-in the IdM Application using Privacy-ABC technologies) and in the issuance process (obtaining credSchool, credRole and other credentials).

In the 2nd round the Users did not need to access the School Registration System because EDOC downloaded all the credentials to the smart cards. During the issuance process, when EDOC was downloading the credentials, no exceptions appeared on the IdM ABC System Issuer and Verifier side.

2.5 Revocation Authority

The Revocation Authority (RevAuth) of the Söderhamn pilot is a customized web application developed by WP4. The RevAuth application is installed on the target site by NSN. This application is dependent on Issuer parameters generated by the 'IdM ABC System' of the School Registration System (see Subsection 2.4.6.2.3). And the Issuer of the 'IdM ABC System' is dependent on the parameters generated by the RevAuth application. In order to speed up the installation it therefore makes sense to map the tasks of Issuer parameter generation and RevAuth parameter generation to the same team.

2.5.1 Revocation Application

2.5.1.1 Implementation

The reference implementation produced in WP4 consists of an ABC engine (ABCE) providing functionality for issuing credentials, constructing proofs of ownership of a credential and verification of said proofs. In addition to these basic functionalities, the ABCE also contains functionality for revocation of credentials as well as inspection of credential attributes in a proof.

A revocation authority functions by keeping a whitelist of unrevoked revocation handles. These revocation handles are stored in a publicly available accumulator[4]. When a revocation handle is to be revoked, the revocation authority removes the handle from the accumulator (dependant on whether white or black listing is used). The new accumulator is then signed by the revocation authority and made publicly available as part of the (updated) RevocationInformation.

Since all the cryptographic functionality was already implemented, the main task in relation to the Söderhamn pilot was therefore simply to integrate the functionality of the revocation authority with the components of the ABC system (Issuers, Users and Verifiers).

2.5.1.2 Integration

The ABC engine contains the required functionality for constructing a revocation authority; however the ABCE only exposed an 'ABCE Java API'. A custom revocation authority application (RevAuth) for the Söderhamn pilot was therefore implemented.

The RevAuth is a small Java application containing a lightweight Jetty web server. When the RevAuth is started, it will start by scanning for existing trusted storage (named CredentialManager) containing private key artifacts and existing public storage (name KeyManager), containing system parameters and other public artifacts. If these are found, the RevAuth will load these objects into the ABCE and expose a REST web interface to the other parts of the ABC system (Issuers, Users and Verifiers).

If existing storage files were not found, the RevAuth will scan for necessary resources such as system parameters, Issuer parameters and other configuration files. These files allow the RevAuth to know which URL it is reachable and thereby the RevAuth is able to generate a new set of revocation authority parameters as well as a new empty trusted storage. All of these are written to disk allowing the administrator to access the revocation authority parameters and will allow the revocation authority service to be restarted without loss of data (revoked credentials).

Once the RevAuth is running and the revocation authority parameters are generated, the revocation authority parameters must be distributed to the Users, Issuers and Verifier. During normal operations an Issuer will ask the RevAuth for a new revocation handle when issuing a credential. When a User is performing a proof, she will contact the RevAuth to obtain the latest revocation information. The Verifier will also contact the RevAuth to obtain the latest revocation information when verifying a presentation proof.

In order to revoke a credential, a revocation handle must be posted to the 'ABCE REST API'. Due to the lightweight nature of the RevAuth, authentication and authorization was not implemented.

2.5.1.3 Deployment

The preparation of the installation requires the availability of one or more revocable credentials. Credentials are revocable only if their credential specifications contain a revocation handle as attribute and if the 'Revocable' parameter is set to true.

The installation is complex and begins by copying the war file of the RevAuth application into an Apache-Tomcat container (../webapps directory). In the Söderhamn pilot, the RevAuth application is hosted on a server dedicated only to revocation. No other services are handled by this application.

[4] Upon initial startup, the 'empty' accumulator will contain all possible revocation handles, i.e. integers belonging to some mathematical group. During issuance, the accumulator is not updated, since the RevAuth just randomly chooses a new revocation handle. During revocation, the accumulator needs to be adjusted. If revocation handle X is revoked, the accumulator will finally contain all possible revocation handles except X.

The server.xml connector of port 8443 is configured with a GoDaddy X.509 certificate in order to provide and enforce HTTPS. This certificate is accepted by all browsers and by Java so that the server certificate need not be installed in the User Application. After copying the war file into the container and after configuring the server.xml file, Tomcat must be started to trigger the generation of a 'service-rev-auth' directory which basically contains the unzipped contents of the war file. Then Tomcat must be stopped and the following tasks must be executed:

1. The Issuer parameters must be copied to the
   .. /webapps/service-rev-auth/WEB-INF/classes/revocation_storage folder
2. After this, the credSpecs must be copied into the
   .. /webapps/service-rev-auth/WEB-INF/classes/xml folder
3. Then the system parameters of the Issuer must be copied into the
   ../webapps/service-rev-auth/WEB-INF/classes/revocation_storage folder
4. The urlConfig file in
   .. /webapps/service-rev-auth/WEB-INF/classes/revocation_storage
   must be edited to match the external address of the RevAuth service
5. Now Tomcat must be started again
6. Use a browser to open `https://revauth.abc4trust.se:8443/service-rev-auth/service-rev-auth/revocation/generatenonrevocationevidence/dd/`
   - This will respond with an HTTP error 404, but will make the revocation authority generate the revocation parameters
7. Copy the RevAuth parameters
   revocation_authority_urn_soderhamn_revocationauthority_default
   into the system folder of the Issuer and restart the Tomcat of the Issuer

Please note that if a credSpec changes or if the location of the RevAuth changes, all above listed steps must be repeated. On top of that, new RevAuth parameters impact the User, the Verifier and the Issuer as they must be distributed to these entities before the operation state of the project.

EDOC and NSN executed the system tests of the RevAuth application. Testing revocation included *all* ABCE engines of the Söderhamn pilot *but* the Inspector ABCE. Before the operation phase of the pilot, NSN invited to a formal handover of the School Registration System and the Revocation Authority. EDOC thereafter configured their firewall to block traffic coming from NSN. After 'cutting the lines', the pilot entered the operation state.

2.5.1.4 Operation

During the operation of the pilot the Revocation Authority was up and running without any noticeable problems. The Revocation Authority is an important component that is used not only when credentials are issued and saved to the cards, it is also used and has to be up and running during verification i.e. when Users visit the Restricted Area System.

Based on the experiences gained from the testing phase and from the 1st round of the pilot it was known that downtime of the Revocation Authority means that the Restricted Area Application will not be able to respond on requests to verify the validity of credentials. As this is a crucial and important component, EDOC was continuously checking the availability of the Revocation Authority to make sure that the Users participating in the pilot were able to access it.

After the 1st round of the pilot it was decided to improve the performance of the entire system. Among many other improvements made to speed up the login process when using the Restricted Area Application it was decided to minimize the number of controls (checks) towards the Revocation Authority in the following way.

Revocation (the revocation handle) was removed from all credentials except for the main credential, credSchool. To make sure that revocation control still can take place, a default access policy was added to all Restricted Areas requiring the school name to be equal to Norrtullskolan. As this access policy was added to all Restricted Areas it guaranteed that a revocation check was performed whenever a User was trying to login at any Restricted Area.

2.6 Restricted Area System

The Restricted Area System was developed to add user functionality to the pilot and to demonstrate a real life application of the Privacy-ABC technologies. The Restricted Area System consists of following blocks:

- Restricted Area Application
- School Portal
- RA ABC System – Verifier
- Restricted Area Admin

The RA ABC System is integrated with the server-side of the Restricted Area Application. The Restricted Area Admin is connected to the Restricted Area Application database with limited functionality of operations allowing the administrator to manage data. The School Portal (see Figure 31) is the website containing general information about the pilot, links to required software, help and support materials and links to applications used by pilot Users such as a link to the URL of the Restricted Area Application (https://ra.abc4trust.se/Home/Areas) and a link to the URL of the IdM Portal (https://idm.abc4trust.se:8443/idmPortal) (see Figure 32).

2.6.1 Restricted Area Application

The Söderhamn school pilot uses Privacy-ABC technologies to enable secure, and by minimal data disclosure, privacy-preserving identification in communications between staff, pupils and guardians. The pilot application at Norrtullskolan involves privacy-preserving community access and school internal social networking for pupils via a specifically dedicated online platform called the Restricted Area Application. This pilot addresses the specific challenges posed by the fact that Internet users are getting younger and often might even be minors (Pupils 12-16 years old).

The communication services provided on the online platform entail the following possibilities for the participants:

- Chat rooms to be used by pupils and/or staff and guardians
- Online forums for discussing lessons and other school related matters as well as political discussions. These may be set up as openly accessible forums or as personal restricted areas where only a predefined group of participants can enter (e. g. children of a certain age or class).
- Online counselling sessions in restricted areas with health personnel (counsellors, social workers, nurses, coaches), where staff can provide counselling in a safe environment while pupils are not necessarily required to reveal their identity.
- Document areas where staff can share documents (e.g. grades and development plans) with pupils and their guardians.

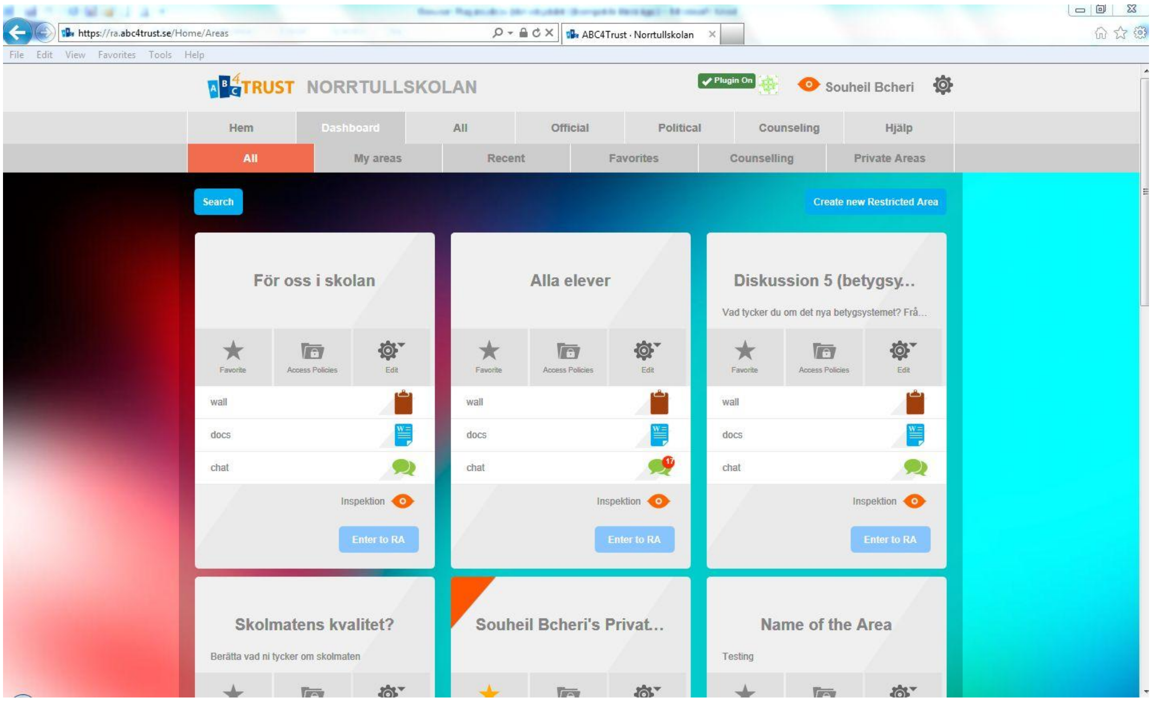


Figure 30: The Restricted Area Application - A List of Restricted Areas

As depicted in Figure 30 the Restricted Area Application consists of many different Restricted Areas. Restricted Areas (RAs) created by the school are marked as official. Other users (pupils and guardians) are also able to create new RAs for different purposes. Each RA is protected by one or several access policies[5]. An access policy defines who is allowed to enter the RA and use its functionality (chat, wall, document sharing and political discussions) and see its content. A Restricted Area for girls will have the access policy “Girls”.

Users (pupils, guardians/parents and school personnel) are able to sign in (login) to different RAs in a very secure and privacy friendly way taking advantage of the Privacy-ABC technologies. Using smart card the User can prove that she is a girl to enter a chat room restricted for “Girls”. The User can prove that she belongs to Class 9A to enter a chat room restricted for “Class 9A”. The User can prove that she is a girl between 14-15 years old to enter a chat room for “Girls 14-15 years old”. In some cases, when anonymity is desired, the User can anonymously/pseudonymously sign in and participate in political discussion groups without revealing any personal data. Another case when anonymous/pseudonymous login might be desirable is during counselling sessions.

The Restricted Area Application consists of JavaScript client and C# backend, of which both are integrated with the RA ABC System. The access to the RA Application itself and the different Restricted Areas is controlled with presentation policy alternatives (an XML translation of the access policies), which are verified against the credentials which the User has on her smart card.

Some of the functionalities provided by the RA Application are the dashboard, search and browse functions for lists of Restricted Areas. And inside a Restricted Areas Users can chat, upload files and leave messages on the wall.

[5] With ‘access policies’, the XML style ‘presentation policy alternatives’ are NOT meant. The access policies are intermediate policies which need to be translated into PPAs via the ‘XML Generator’ before sending them to the User.

Additional functionalities on top of the Restricted Area concept are the counseling, the political discussions and the alias to alias chat (private one-to-one chat).

2.6.1.1 Implementation

The RA Application backend communicates with the RA ABC System, which runs on the same server. The RA Application JavaScript client is integrated with the User ABCE via a browser plugin and the two crypto engines for Idemix and U-Prove. U-Prove runs as a separate Windows service whereas Idemix runs inside the User Application.

The Restricted Area Application hosts the client JavaScript that is executed locally in the User's browser to prevent transferring data from the User computer and to avoid linkability between the different aliases of the User. The JavaScript is of course a part of the RA Application but in this chapter and in Section 2.6.1.2 the JavaScript is referred to as the 'RA Application JS Client'. The reason is to make it clear what is processed on the server and what is processed locally on the User's side.

The RA Application provides the pilot functions, most notably the different kinds of Restricted Areas associated/protected by access policies.

The RA Application is built as traditional 3-tier architecture.

Tier 1	Presentation / JS Client	.NET / HTML, JavaScript
Tier 2	Application / business logic	.NET
Tier 3	Data	MS SQL Server

While the RA Application is installed on a single server, the application is built to be able to connect to a database on another server. The presentation and application tiers, on the other hand, are logically separated but built together, thus always to be deployed on the same server.

A firewall can naturally be placed in front of the application, allowing only HTTP / HTTPS traffic. If, at a future point in time, the database is installed on a server of its own, there can also be an extra security measure of another firewall with access restrictions allowing only certain ports or networks to access the database.

All user interaction is done via a web browser. Please note that the User interacts with the User Application, in addition to the interaction with the Restricted Area System.

The data flows for the application are described in more details in [D51].

The following data is stored in the RA Application database:

- RA metadata
    - type of RA (general chat, counselling etc.)
    - presentation policy alternatives
- Chat messages (each message connected to a specific RA)
- Files uploaded (each file connected to a specific RA)
- Alias list (aliases already used, thus reserved for the original User). However, the server does not store the identity of the original User, only lets her use an alias already stored on her smart card and already stored in the alias list.

The Restricted Area Application is interacting with two different instances of ABCEs, one which resides on the server and is named ‘RA ABC System’, and another which resides inside the User Application.

The User’s ABCE interacts indirectly with the RA ABC System’s ABCE via the Restricted Area Application in order to retrieve presentation policy alternaltives and to verify presentation tokens.

The Restricted Area (RA) Application is built using .NET Framework 4.0 and the C# programming language and is deployed on a Windows 2008 Server. It uses a Microsoft Internet Information Server (IIS) 7.5 as the web server to run the web application and for communication with the client side. The server side ABC System, also known as the Verifier, runs within a Java environment, also on Windows 2008 Server.

The the RA Application backend has dependencies to Entity Framework for ORM (object relational mapping) support, ASP.NET WebForms (for the Restricted Area admin interface), and ASP.NET MVC.

Three major components within the RA Application are

- the business logic entities
- the ‘XML Generator’ and its related components
- and an API to provide connections for the RA Application JS client.

The business logic layer and the API are directly related because business entities provide the API with data to be sent to and handled by the client. All RA Application data, i.e. the definitions and contents of the Restricted Areas, is stored in a Microsoft SQL database. As the Entity Framework has been used with a “code first” approach, the structure of the pilot database was created by ORM during deployment.

2.6.1.2 Integration

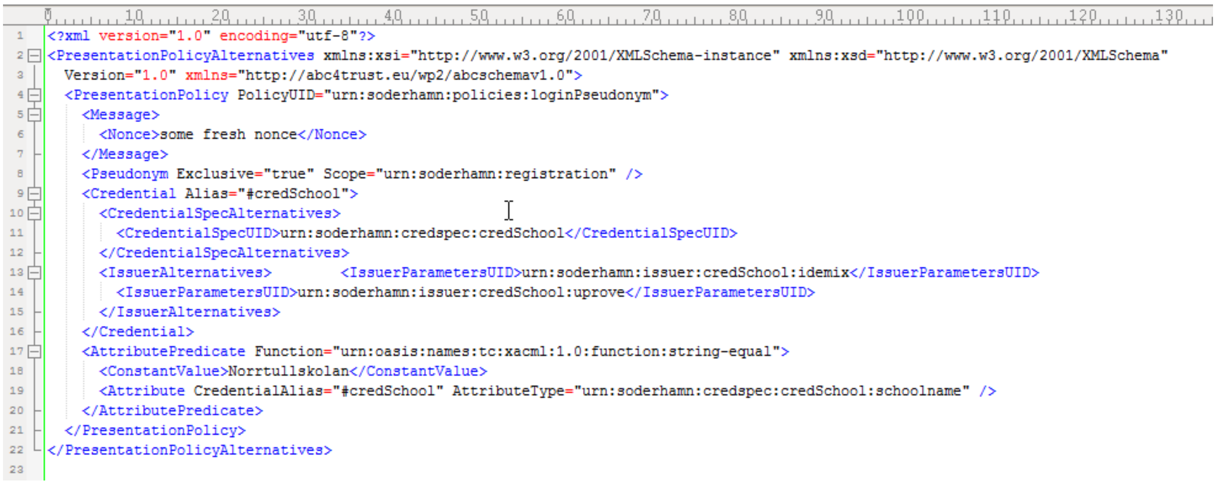
The RA Application is communicating with the RA ABC System (i.e. the Verifier), which is running on the same server, using a RESTful interface. The RA Application JavaScript client is integrated with the User ABCE via browser plugin and User service chain.

The User does not have direct access to the Verifier. The RA Application is acting as a proxy and forwards tokens from the User to the RA ABC System.

Below is a description of scenarios that require integration with the chain of messages.

To log in to the RA Application, the User has to prove that she has a valid credSchool. This does not mean that the User will get access to any specific Restricted Area or any content (messages, documents etc.). A successful login means that the User will be able to see the lists of Restricted Areas. In order to enter a specific Restricted Area the User still has to prove that her credentials satisfy the corresponding presentation policy alternatives of that Restricted Area.

- The RA Application prepares a presentation policy alternatives (XML-format) that requests the User to prove ownership of a valid school credential (credSchool). It sends the policy to the User via JavaScript call that is delegated by browser plugin:

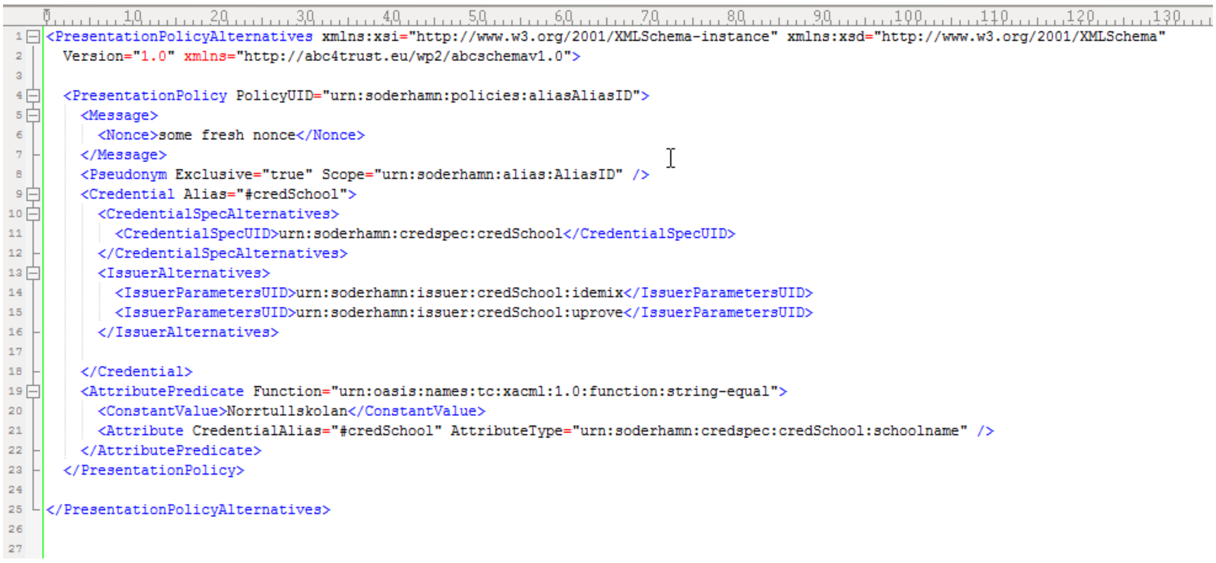


```
<?xml version="1.0" encoding="utf-8"?>
<PresentationPolicyAlternatives xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  Version="1.0" xmlns="http://abc4trust.eu/wp2/abcschemav1.0">
  <PresentationPolicy PolicyUID="urn:soderhamn:policies:loginPseudonym">
    <Message>
      <Nonce>some fresh nonce</Nonce>
    </Message>
    <Pseudonym Exclusive="true" Scope="urn:soderhamn:registration" />
    <Credential Alias="#credSchool">
      <CredentialSpecAlternatives>
        <CredentialSpecUID>urn:soderhamn:credspec:credSchool</CredentialSpecUID>
      </CredentialSpecAlternatives>
      <IssuerAlternatives>       <IssuerParametersUID>urn:soderhamn:issuer:credSchool:idemix</IssuerParametersUID>
        <IssuerParametersUID>urn:soderhamn:issuer:credSchool:uprove</IssuerParametersUID>
      </IssuerAlternatives>
    </Credential>
    <AttributePredicate Function="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <ConstantValue>Norrtullskolan</ConstantValue>
      <Attribute CredentialAlias="#credSchool" AttributeType="urn:soderhamn:credspec:credSchool:schoolname" />
    </AttributePredicate>
  </PresentationPolicy>
</PresentationPolicyAlternatives>
```

- The User Application processes the request and returns the presentation token to the RA Application JS client
- The RA Application JS client sends the received token to the RA Application backend using an asynchronous request. The backend makes local request on the server to the RA ABC System using the RESTful interface. The RA ABC System verifies the token by calling the method `VerifierABCE.verifyTokenAgainst-Policy()` of the ABCE. If verification fails, a failed page is displayed and access is refused.

The User can select one of the **aliases** that she has created previously using the Restricted Areas user interface. Those aliases are saved in a secure and privacy friendly way on the User's smart card. If the User creates a new alias, the RA Application will make sure that the chosen alias is unique system-wide using an asynchronous request to the RA Application. The new alias is also added to the smart card for later use.

- If the User chooses to establish a new alias, then
  - The User is asked to enter a name *aliasname* for the new alias.
  - The RA Application JS client checks with request to the RA Application backend whether the alias *aliasname* is already taken and there is a record in the database. If so, it asks the User to choose a different alias.
  - The RA Application backend prepares the presentation policy alternatives by feeding the access policies to the XML Generator which request the User (via RA Application JS client and User Application) to present a scope-exclusive pseudonym for scope string `urn:soderhamn:alias:alias`*`AliasID`*
    (for example, `urn:soderhamn:alias:alias154`) and a valid school credential bound to the same key. It sends the policy to the User:



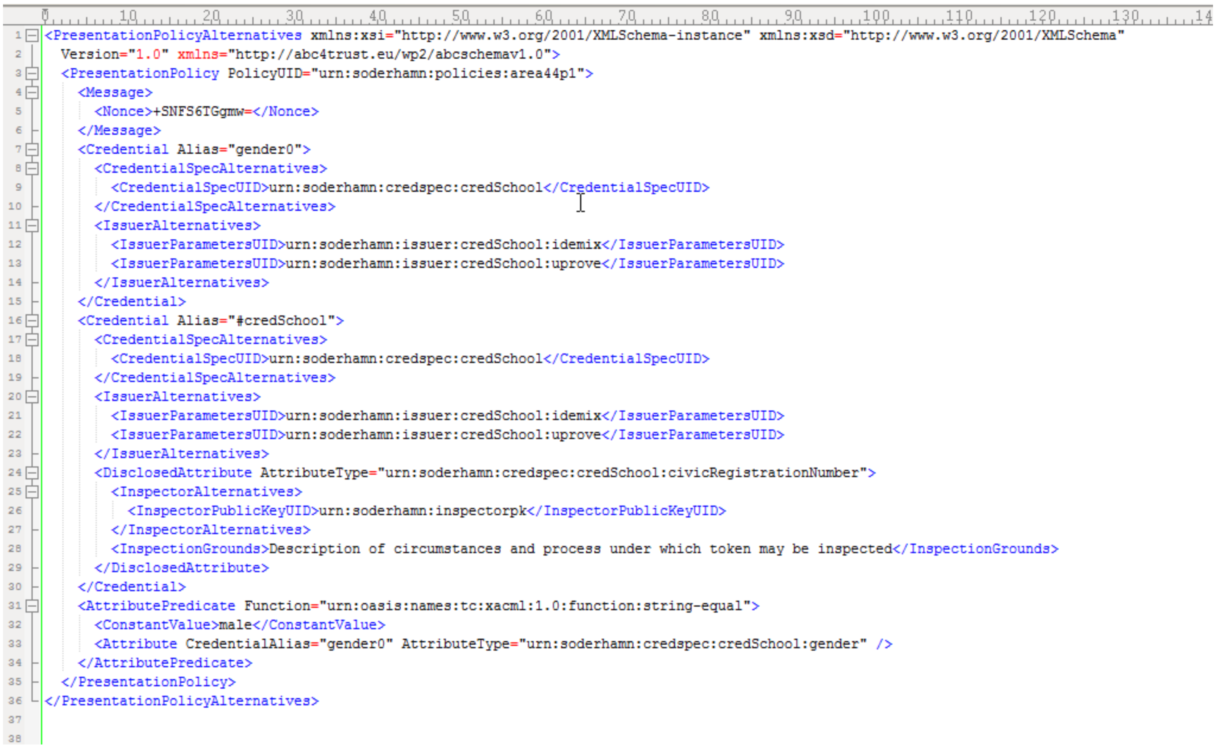
```
<PresentationPolicyAlternatives xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  Version="1.0" xmlns="http://abc4trust.eu/wp2/abcschemav1.0">

  <PresentationPolicy PolicyUID="urn:soderhamn:policies:aliasAliasID">
    <Message>
      <Nonce>some fresh nonce</Nonce>
    </Message>
    <Pseudonym Exclusive="true" Scope="urn:soderhamn:alias:AliasID" />
    <Credential Alias="#credSchool">
      <CredentialSpecAlternatives>
        <CredentialSpecUID>urn:soderhamn:credspec:credSchool</CredentialSpecUID>
      </CredentialSpecAlternatives>
      <IssuerAlternatives>
        <IssuerParametersUID>urn:soderhamn:issuer:credSchool:idemix</IssuerParametersUID>
        <IssuerParametersUID>urn:soderhamn:issuer:credSchool:uprove</IssuerParametersUID>
      </IssuerAlternatives>

    </Credential>
    <AttributePredicate Function="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <ConstantValue>Norrtullskolan</ConstantValue>
      <Attribute CredentialAlias="#credSchool" AttributeType="urn:soderhamn:credspec:credSchool:schoolname" />
    </AttributePredicate>
  </PresentationPolicy>

</PresentationPolicyAlternatives>
```

- The User Application processes the request and returns the presentation token to the RA Application JS client
- The RA Application JS client sends the received token to the RA Application backend using an asynchronous request. The backend makes local request on the server to the RA ABC System using the RESTful interface. The RA ABC System verifies the token by calling the method `VerifierABCE.verifyTokenAgainst-Policy()` of the ABCE. If verification fails, a failed page is displayed and access is refused.
- Upon a successful presentation, the Restricted Area Application stores the cryptographic pseudonym (i.e., the PseudonymValue) and associates it to the *aliasname* chosen by the User in the Restricted Area Application database.
- Otherwise, the default main screen of the RA Application is displayed.

- If the User chooses to log in under an established alias *aliasname*, then
    - The User Application sends *alias ID* to the RA Application.
    - The presentation policy alternatives will look like in the example above.
    - The User Application processes the request and returns the presentation token to the RA Application JS client
    - The RA Application JS client sends the received token to the RA Application backend using an asynchronous request. The backend makes local request on the server to the RA ABC System using the RESTful interface. The RA ABC System verifies the token by calling the method `VerifierABCE.verifyTokenAgainst-Policy()` of the ABCE. Moreover, it checks that the presented cryptographic pseudonym (i.e., the PseudonymValue) in the token is equal to the pseudonym that was associated to this alias in the RA Application records.
    - If any of these checks fail, a failed login window is displayed and access is refused.

Users are able to create **new Restricted Areas**. When a User is logged in at the RA Application (not as anonymous alias), she can create a Restricted Area, i.e., a discussion board and define one or several access policies.

- The RA Application creates the restricted area, converts the access policies entered in the GUI into an XML PresentationPolicyAlternatives element and associates it to the restricted area. Since we want inspection to be possible, when a new pseudonym is established, the PUN number is encrypted with the inspector's public key for possible use by the inspector. For example, the restricted area for pupils whose gender is equal to “male” would be something like the following:



```
<PresentationPolicyAlternatives xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  Version="1.0" xmlns="http://abc4trust.eu/wp2/abcschemav1.0">
  <PresentationPolicy PolicyUID="urn:soderhamn:policies:area44p1">
    <Message>
      <Nonce>+SNFS6TGgmw=</Nonce>
    </Message>
    <Credential Alias="gender0">
      <CredentialSpecAlternatives>
        <CredentialSpecUID>urn:soderhamn:credspec:credSchool</CredentialSpecUID>
      </CredentialSpecAlternatives>
      <IssuerAlternatives>
        <IssuerParametersUID>urn:soderhamn:issuer:credSchool:idemix</IssuerParametersUID>
        <IssuerParametersUID>urn:soderhamn:issuer:credSchool:uprove</IssuerParametersUID>
      </IssuerAlternatives>
    </Credential>
    <Credential Alias="#credSchool">
      <CredentialSpecAlternatives>
        <CredentialSpecUID>urn:soderhamn:credspec:credSchool</CredentialSpecUID>
      </CredentialSpecAlternatives>
      <IssuerAlternatives>
        <IssuerParametersUID>urn:soderhamn:issuer:credSchool:idemix</IssuerParametersUID>
        <IssuerParametersUID>urn:soderhamn:issuer:credSchool:uprove</IssuerParametersUID>
      </IssuerAlternatives>
      <DisclosedAttribute AttributeType="urn:soderhamn:credspec:credSchool:civicRegistrationNumber">
        <InspectorAlternatives>
          <InspectorPublicKeyUID>urn:soderhamn:inspectorpk</InspectorPublicKeyUID>
        </InspectorAlternatives>
        <InspectionGrounds>Description of circumstances and process under which token may be inspected</InspectionGrounds>
      </DisclosedAttribute>
    </Credential>
    <AttributePredicate Function="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <ConstantValue>male</ConstantValue>
      <Attribute CredentialAlias="gender0" AttributeType="urn:soderhamn:credspec:credSchool:gender" />
    </AttributePredicate>
  </PresentationPolicy>
</PresentationPolicyAlternatives>
```

Note that the policy above does not yet specify the full scope string for the scope-exclusive pseudonym. The actual scope string depends on the alias of the User who wants to access the RA, so this part of the policy must be generated on-the-fly.

In order to **enter a Restricted Area** the User still has to prove that her credentials satisfy the presentation policy alternatives.

- The policy is sent from the Restricted Area Application to the User Application.
- The identity selector pops up informing the User about the information that will be revealed and allows him to choose between multiple options of credentials, policies, etc. The User indicates his choice in the identity selector.
- The User Application prepares the presentation token and sends it to the RA Application.
- The RA Application verifies the token and checks that the PseudonymValue associated to this token is the same as what has been registered for this alias. If all checks succeed, the application updates the timestamp of the last successful presentation.
- If the Restricted Area is set as ‘inspectable’ during creation, the RA Application stores the unique token identifier returned via RESTful interface from the RA ABC System and associates it to all the messages written during this session, so that it later can fetch the token back from the database in case inspection is demanded. (The token itself is stored by default by the RA Application in its database.)

The build of the RA Application was done from source code located on the EDOC SVN. The build is set to be launched manually but can be set up to be launched automatically by a scheduler or after each commit.

2.6.1.3 Deployment

The Restricted Area Application was deployed on one of the virtual machines running on servers located within the school DMZ, this way it was deployed on the same sub network as the School Registration System and the Revocation Authority.

The installation of IIS, MSSQL and .NET Framework was done on Windows 2008 R2 Server operating system.

2.6.1.4 Operation

During testing and the operation phase of the 1st round it was hard to debug problems as it was difficult to isolate the problems in order to know if the problem was caused by the Restricted Area Application itself or by any of the other applications. Before the 2nd round of the pilot the logging functionality of the User Application and the Restricted Area Application were significantly improved and debugging became more efficient.

During operation of the 2nd round the Restricted Area Application served the Users well and no known issues or problems were discovered by EDOC or reported by the school or by the pilot Users.

Users were able to use the Restricted Area Application in the way it was intended to be used. Teachers could create Restricted Areas and define access policies. Pupils could enter different Restricted Areas and post and receive messages and documents etc.

2.6.2 RA ABC System – Verifier

2.6.2.1 Implementation

The Verifier was taken from a demo application that WP4 had made. It was generic enough that there was no need for further implementations to make it work. The one exception was a way of storing the revocation information, but this is discussed in Section 2.6.2.2.

2.6.2.2 Integration

During integration it became clear that we needed to save time somewhere in order for the system to operate faster. One way of doing this was minimizing communication time with the revocation authority as well as the computation time saved by re-using the same revocation information several times. The idea was to only update the revocation information when a certain time has passed. This makes efficiency better, but it also enables a revoked User's credential to still be valid at the Verifier until the revocation information is updated. This trade-off was decided to be acceptable.

The Verifier ran on a separate server, which was contacted only by the Restricted Area Application when needed. The RA Application contacts the REST interface of the Verifier in two situations

- When the RA needs to add revocation information to the presentation policy alternatives for the User
- When the RA needs to validate a presentation token

The XML needed for the proofs was merely redirected from the User through the RA to the Verifier. The Verifier could then answer the RA back if it should display an error page or accept the User into the area.

As it is just a Verifier, there is nothing special needed to run it. Several Verifiers for a system are highly likely, and so it is just a matter of providing the correct resources (Issuer parameters, system parameters etc.) to the ABCE. Feeding the Verifier ABCE these parameters was done using the helper method provided by WP4.

2.6.2.3 Deployment

As it was written in 2.6.2.2, The RA ABC System is integrated with the RA Application via the RESTful interface. The Verifier RESTful interface is locally accessible for the RA Application at the location with URL http://localhost:9500. The port of the RESTful interface does not accept any external connections.

The Verifier was deployed manually and configured to launch automatically on system start at the production server where all other Restricted Area System components reside.

EDOC replaced the old version of the RA ABC System that was emulating the Privacy-ABC technologies with a new version of the application that can verify Privacy-ABC presentation tokens.

The deployment of the Verifier in the RA ABC System in the Söderhamn pilot is different from the deployment in the Patras pilot in the way of how the presentation policy alternatives are generated. The presentation policy alternatives, which in the Söderhamn pilot are based on the access policies that are associated with (and protect) each Restricted Area, have to be generated dynamically (on the fly). The reason is that each Restricted Area is associated with one or more access policies, e.g. a RA for girls in class 9A will have two different policies combined with a logical AND. And as those access policies can’t be known in advance the RA Application has to deal with this matter in a different way from what is the case in the Patras pilot.

For this reason the Restricted Area Application has to be deployed so that it generates different presentation policy alternatives dynamically after retrieving the access policies that are applied for the Restricted Area.

2.6.2.4 Operation

During operation and when a User is trying to enter a certain RA the RA Application retrieves the access policies for that specific RA from the database. Based on those access policies and other policies associated with the RA, such as whether inspection is required or not, the RA Application generates a valid XML file containing the presentation policy alternatives and sends it to the User. The RA Application sends the XML file containing the presentation token AND the presentation policy alternatives to the Verifier via REST interface and depending on the decoded XML answer the RA Application will allow the User to enter the RA or will inform the User that the verification of his presentation token did not succeed. There are of course many reasons why a User is not allowed to enter a RA. One is of course if the User’s credentials cannot meet the presentation policy alternatives. Another reason is a non-valid (revoked) credSchool. But there are also other reasons such as if the card is removed from the card reader or if the reader is not properly connected to the PC etc.

2.6.3 Restricted Area Admin

The Restricted Area Admin (RA Admin) is an application created mainly for two purposes, for Restricted Area database administration and for early stage testing (emulating Privacy-ABC technologies). Initially, before the ABCE was fully developed by WP4 and ready to be integrated into the RA ABC System by EDOC, the RA Admin was used to create different test Restricted Areas and to create test Users and attributes emulating Privacy-ABC technologies.

At this early stage the RA Admin made it possible to test the different functionalities of the Restricted Area Application before the ABCE and other components such as the browser plugins, the Identity Selector and the User smart cards were fully developed and ready for integration, testing and usage.

During the two rounds of the pilot the main purpose of the RA Admin was to make it easier for administrators to edit (add, change and delete) data in the different database tables needed by the RA Application. The RA Admin was used to upload and edit default aliases, it was also used to edit the list of emails matching the default aliases of counselors used for notification purposes.

The Restricted Area Admin doesn’t use ABC technologies for any purposes; it has a separate login via admin username and admin password and utilizes only data from the database.

2.6.3.1 Implementation

The Restricted Area Admin was developed using ASP.NET WebForms as CRUD (create, read, update, delete) user interface performing typical operations on data saved in different tables in the database.

Login to the system requires login and password via a web form. The interface is implemented as tabs. Every tab corresponds (is mapped) to its own table in the database. URL’s are not parsed (and therefore cannot be changed) by input from command line so the application gives no ability to edit other tables than the allowed ones.

After the testing phase was over and the 1st round of the pilot started, all old tables used for testing and simulation of verification were removed from the Restricted Area Admin possibility.

2.6.3.2 Integration

The RA Admin was not integrated with any of the other components. The only interface that the RA Admin shares with other systems is the access to the Restricted Area database directly via database connection. There was no need for integration of the RA Admin with any of the other components such as the Restricted Area Application, the User Application, the IdM Portal or the RA ABC System.

2.6.3.3 Deployment

The Restricted Area Admin is a web application. Its deployment is similar to the deployment of the School Portal, with its own separate URL http://ra.abc4trust.se/Admin/

The Restricted Area Admin was deployed to a server with an IIS 7.5 web server and ASP.NET installed.

During the testing phase all Privacy-ABC technologies were simulated and all test data (attributes) about the Users were saved in the database. EDOC manually added some test Users to the database (First name, last name, gender, age, class, role etc.). As no Privacy-ABC technologies or smart cards with credentials was used during the test phase all data was retrieved from the database. All test data was of course deleted before the 1st round of the pilot started.

During the pilots all attributes about the Users were stored on the User's smart card as Privacy-ABC credentials. At this stage the RA Admin was used to enter default aliases (the real name of the User) and to enter the list of aliases and other inappropriate words that should not be possible to select as alias.

The functionality of the RA Admin was continuously improved and tested during the testing phases, before the 1st round of the pilot and finally before the 2nd round of the pilot.

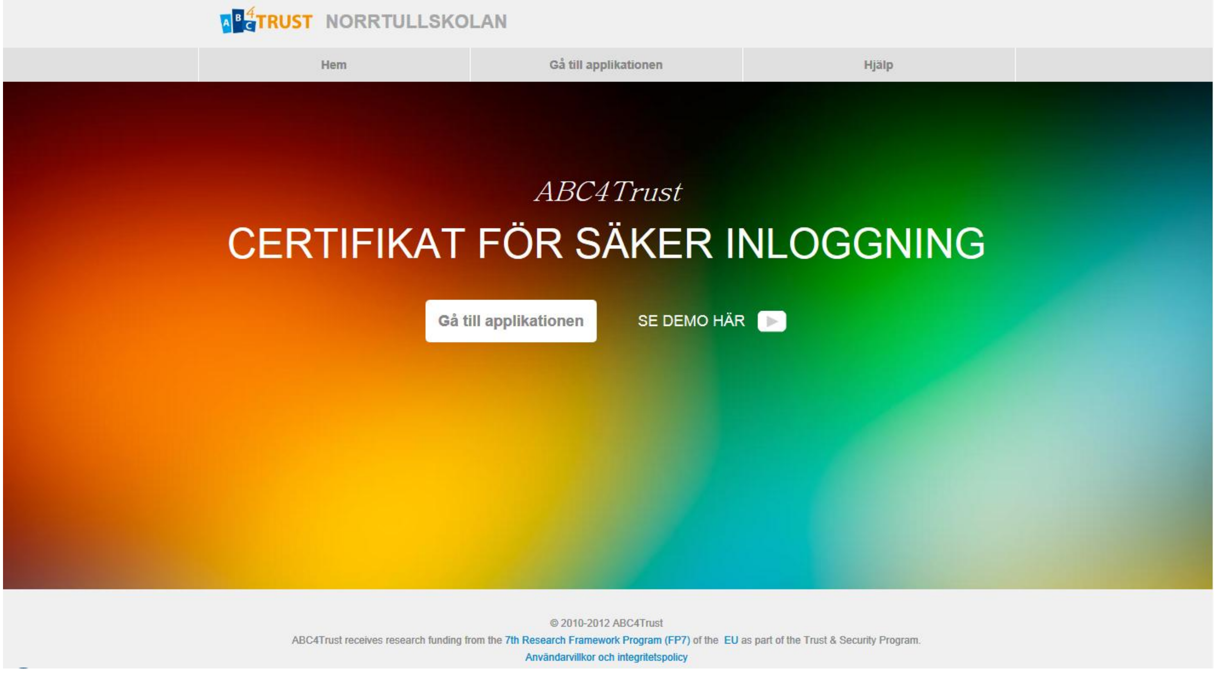
2.6.3.4 Operation

In the beginning of 2nd round of the pilot, The Restricted Area Admin was used to pre-upload default aliases to prevent Users from taking other Users' names as aliases and fake their identity. The uploaded data contained only the aliases, after the Users gained access to the Restricted Area Application a token was saved along with the default alias of the User and proved the ownership of the correct alias by the User. Teachers who were supposed to be counselors had their email address linked to their default alias for notification purposes.

2.6.4 School Portal

The School Portal is the website containing general information about the pilot, links to required software, support materials and links to applications used by pilot Users.

The School Portal user interface has the same design concept as the Restricted Area Application.



© 2010-2012 ABC4Trust

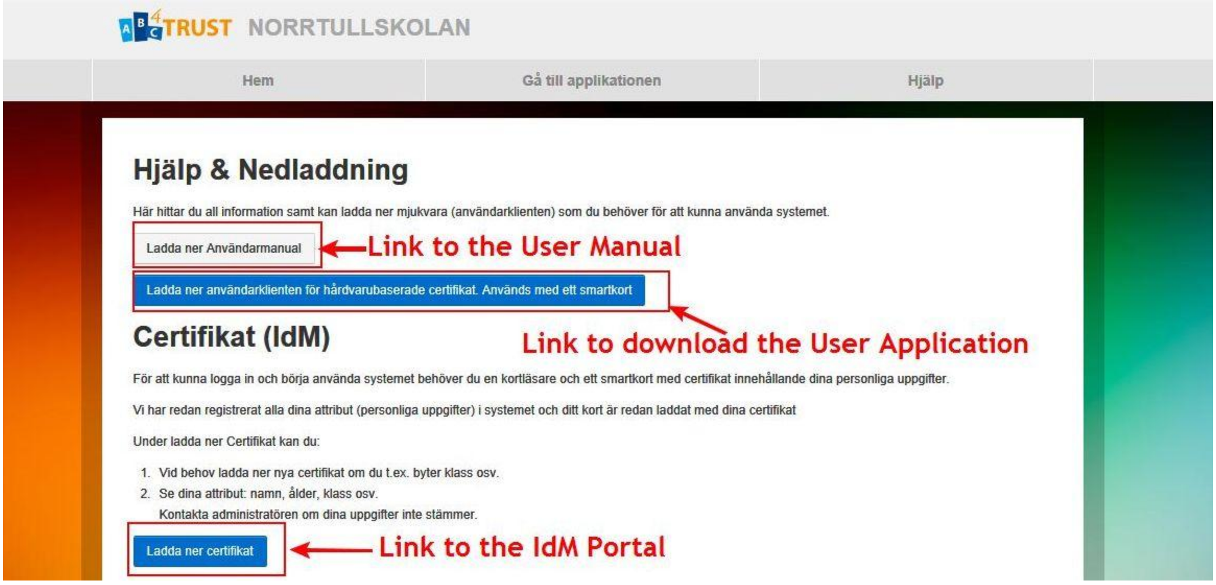
ABC4Trust receives research funding from the 7th Research Framework Program (FP7) of the EU as part of the Trust & Security Program.

Användarvillkor och integritetspolicy

Figure 31: School Portal - Start page

Figure 31 is a screenshot of the School Portal starting page which consists of a menu (links) to the following different web pages. (Swedish translation within the parenthesis):

- Home (Hem): Link to the starting page.
- Enter the application (Gå till applikationen): Link to the Restricted Area Application which requires login using Privacy-ABC technologies (smart cards with Privacy-ABC credentials)
- Help (Hjälp): Contains FAQ, User Manual, link to download the User Application Installer, link to the IdM Portal and link to download the smart card reader drivers. (see Figure 32)
- See demo here (Se demo här): A link to instructional video available at a YouTube channel.



Hjälp & Nedladdning

Här hittar du all information samt kan ladda ner mjukvara (användarklienten) som du behöver för att kunna använda systemet.

Link to the User Manual

Link to download the User Application

Certifikat (IdM)

För att kunna logga in och börja använda systemet behöver du en kortläsare och ett smartkort med certifikat innehållande dina personliga uppgifter.

Vi har redan registrerat alla dina attribut (personliga uppgifter) i systemet och ditt kort är redan laddat med dina certifikat

Under ladda ner Certifikat kan du:

1. Vid behov ladda ner nya certifikat om du t.ex. byter klass osv.
2. Se dina attribut: namn, ålder, klass osv.
   Kontakta administratören om dina uppgifter inte stämmer.

Link to the IdM Portal

Figure 32: School Portal - Help page

The School Portal is not making use of any Privacy-ABC technologies.

2.6.4.1 Implementation

For the 1st round of the pilot the School Portal was created using open source Orchard content management system written on ASP.NET, but since this application required separate deployment and was more resource consuming for the 2nd round it was decided to substitute the old version of the School Portal with a new version. The new version consists of flat HTML pages rendered using ASP.NET templates. On top of that, the design was significantly improved in order to have the same design as the new Restricted Area application style.

2.6.4.2 Integration

The School Portal doesn’t use the RA ABC System and is loosely coupled with the Restricted Area Application’s GUI Even though the School Portal and the RA Application are contained in the same build, they can be considered as independent entities.

In the 1st round of the pilot the School Portal the production environment was located at URL http://portal.abc4trust.se. For the 2nd round it was located at URL www.abc4trust.se which is a more user-friendly URL that redirects the User to the new School Portal located at URL https://ra.abc4trust.se/Start.

All the links from the School Portal to the Restricted Area Application are leading to https://ra.abc4trust.se which is already a separate application on its own subdomain.

The School Portal contains links to the IdM Portal (http://idm.abc4trust.se) under the section “Help”.

2.6.4.3 Deployment

The School Portal was filled with links to needed software such as the User Application Installer and the smart card reader installer. The School Portal was also updated with different support material needed by the Users such as User Manuals, FAQ section, inspection grounds document and links to the ABC4Trust official webpage and other support materials.

Support materials include the following:

- Description of the pilot and an overview about the Restricted Area application
- User Manual in Swedish, PDF format
- Inspection grounds
- Embedded YouTube videos with basic training concerning the usage of the RA Application
- The required software during the pilot i.e. the User Application Installer which automatically will install the browser plugin, the Idemix and U-Prove services and will set up its own dependencies.
- Another required component is the smart card reader driver which normally should be included to the default Windows driver database but might as well be installed separately in specific cases.

2.6.4.4 Operation

The School Portal was used in two different ways during the two rounds of the pilot.

In the 1st round the Users received their smart cards that was initialized and prepared but did not contain the credentials. Each User had to visit the School Portal and navigate to the IdM Portal in order to download her own credentials.

In the 2nd round this was changed. EDOC initialized and downloaded all needed credentials to the smart cards before they were handed over to the Users. So there was no need for the Users to visit the IdM Portal.

But the link to the IdM was not removed from the portal in case a User needed to download new credentials. This happened in a couple of cases when the name of one User was misspelled. The name was changed in the IdM Database via the IdM Admin Tool which automatically triggered the revocation of the old credential. A new credential was issued and downloaded by the User herself.

Not only the design of the School Portal was changed between the two rounds of the pilot but also the content such as the FAQ section, the user manuals and the links to the User Application Installer was updated to reflect the latest changes of development.

While the two rounds of the pilot were running there were no reports about downtime or unavailability of the School Portal. The School Portal can serve more intensive and much larger number of simultaneous requests than what the pilot required.

2.7 Legal Topics for the Söderhamn pilot

As already explained in Section 1.2, this document solely aims at providing feedback from the pilots from a more technical perspective. So besides the feedback meant to be fed into WP4, this deliverable shall contain information which can be helpful for potential developers as well as implementing decision-makers with regard to the Privacy-ABC technologies used in the pilots.

Whereas legal requirements are always a difficult topic for developers and implementing entities, it must be considered that this issue is not only of concern when taking Privacy-ABC technologies into account. Rather, whenever the processing of personal data is intended, a valid legal ground must be given, and the correlating legal requirements must be fulfilled. However, in the context of Privacy-ABC technologies, the complexity is even less since the inbuilt privacy-enhancing characteristics of these eliminate a great number of risks to the personal data of its users. The legal European data protection framework on the generic European level (directive 95/46EC) as well as on the level of the national law of the EU member states requires that all processing of personal data must be purpose-bound. This means that the processing is in principle only lawful if it is conducted for the initial purpose for which the processing operation was set up. From this purpose-boundary derive the principles of data avoidance and data minimization. Both allow only the processing of personal data which is absolutely necessary to perform the given task for which the data was collected initially. As a consequence, deployment of Privacy-ABC technologies is strongly advisable to comply with these legal requirements.

To provide a valid legal ground for the processing of the pilot participant's personal data within the pilot architecture, valid informed consent must be given by the participants before the start of the trial. However, since this deliverable mainly covers the perspective of the technical staff in the project, the necessary content of such consent forms will not be described in this document. Rather, all primarily user-centric documents like the consent forms, the information sheet and the user manual, will be explained in the deliverables focusing on the specific detail 'feedback and evaluation of the project pilots', namely [D63] (Evaluation of the school pilot) and [D73] (Evaluation of the Student pilot).

Regarding the legal matters related to the pilots, this deliverable will not concentrate on the user perspective, but explain which aspects must be considered by entities who want to further develop, or integrate existing Privacy-ABC technologies. This encompasses the lessons learned from the two specific pilots of this project as well as general advices on how to approach the usage of Privacy-ABCs in different settings. In this chapter, the specific frame conditions and requirements of the Söderhamn school pilot are explained and the lessons learned are explained.

2.7.1 Role Allocation

One of the challenges to be addressed when setting up the school pilot was the Norrtullskolan in Sweden not having either the technical means or the expertise to completely set up and maintain a Privacy-ABC architecture on their own. Therefore, the assistance of other project partners was necessary.

The infrastructure (i.e. the hard- and software) to use Privacy-ABCs, with the exclusion of the school PC's from which the system would also be accessible, was mainly provided by Eurodocs, a Swedish IT company (in the following: EDOC). This included the setup and operation of a Restricted Area System for providing communication network functionalities to the school for direct, anonymous or pseudonymous interaction among pupils, teachers, and parents. Furthermore, they were involved in the set up and administration of the credentials (e. g. issuance, revocation), in debugging tasks, and entrusted with the oversight of the performance of the pilot architecture overall.

Moreover, Nokia Solutions and Networks Management International GmbH (NSN) provided the School Registration System and supported EDOC with the setup, administration, debugging, maintenance, and performance control of it. On top of that, NSN deployed the Revocation Authority and supported EDOC in operating it.

This situation made it necessary to allocate different legal roles to the persons and legal entities involved before the start of the pilot. Each role correlates with a certain set of rights and obligations, which in some cases have to be fulfilled before the initiation of the data processing or even the development of new applications. While both pilots of the project are governed under the umbrella of the European Data Protection Directive 95/46 EC, a certain complexity was added since two different national privacy laws (Swedish[6] and German[7]) had to be considered. However, a detailed explanation of the legal as well as of the organisational roles has already been published in [D51] (Scenario definition for both pilots). A first mapping of legal roles was conducted in Section 3.6.2 of [D51]. However, since the drafting of this specific deliverable, the technical as well as the organisational frame conditions of the Söderhamn pilot changed in parts, therefore this mapping is outdated. Rather, the initial allocation of legal roles had to be altered since the factual circumstances became clearer after the pilot went into the implementation phase. This is due to the fact that the school, as a public entity under Swedish law, holds the definite legal responsibility for its pupils. Moreover, in preparation of the pilot (see 8 of Figure 2), the school reserved itself the right to determine the means, the purpose, and the frame conditions of the personal data processing which was intended to be done with the Privacy-ABC technologies used. While EDOC provided guidance and assistance, the school maintained the decision power over diverse organisational and conceptual matters related to the execution of the pilot (see 9 of Figure 2). Examples would be which functionalities of the Privacy-ABC architecture the school wanted to be ready for the pilot. According to the European data Protection Directive 95/46 EC, a data controller determines the purposes and means of the processing of personal data, while a data processor only processes the data according to the instructions of the controller. Therefore, the data processor only applies the technical means set out by the data controller on his behalf and processes the personal data instead of him. Nevertheless, the data controller remains fully responsible for the data processing, while the processor is bound to the instructions of the controller. Another entity, which acts on behalf of the processor, has to be considered being a sub-processor.

Correlating to this, the school was the decisive partner in the project with whom the work packages developed the central communication services provided within the pilot architecture. By request of the school, these services entailed e. g.:

- Chat rooms
- Online forums
- Online counselling sessions in restricted areas
- Document sharing functionalities
- Online polls

[6] English translations of the Swedish "Personal Data Act (1991:204)" and the Personal Data Ordinance (1998:1191)" are available online: http://www.datainspektionen.se/in-english/legislation/the-personal-data-act/.

[7] Bundesdatenschutzgesetz (BDSG), an English translation can be found online at: http://www.bfdi.bund.de/EN/DataProtectionActs/Artikel/BDSG_idFv01092009.pdf?__blob=publicationFile.

Moreover the school was the key partner with whom it was decided how many Restricted Areas should be set up, as well as which and how many of them shall be inspectable. Furthermore, the distribution of crucial information, consent forms, and smart cards was conducted with the help of teachers on behalf and under the supervision of the school administration.

All these aspects led to the conclusion that the Norrtullskolan, in consideration of Art. 3 of the Swedish Personal Data Act, fully determines the means and the purpose of the personal data processing. Consequently, a joint controllership of the Norrtullskolan and EDOC is not given; rather it must be assumed that the school is the sole controller with the final legal responsibility for what happens with the participant's personal data during the pilot. EDOC then was categorised as a personal data assistant in the sense of the Swedish Personal Data Act (which would correlate to the role of a processor according to the European Data Protection Directive 95/46 EC). NSN acting on behalf of EDOC was categorised as a sub-processing entity.

2.7.2 Processing Contracts between Norrtullskolan, EDOC and NSN

For the processing of the personal data of pupils, parents, and teachers within the Söderhamn pilot, contractual relationship between the above mentioned entities were necessary. In the prior section, it was already explained that EDOC and NSN only processed data on behalf of Norrtullskolan and were therefore only data processors instead of being data controllers.

For the Söderhamn pilot, three entities were involved in the execution of the trial. EDOC was strictly bound to the instructions of the school, while in turn NSN is furthermore bound to the instructions of EDOC. To ensure the fulfilment of all legal requirements for the personal data processing as well as to safeguard that all measures for the protection of said data are taken, a chain of processing contracts needed to be set up. Therefore, two types of contracts have been drafted before the start of the Söderhamn pilot:

- **Processing contract** between Norrtullskolan as controller and EDOC as processor
- **Sub-processing contract** between EDOC as processor (according to Swedish data protection law named Personal Data Assistant) and NSN as sub-processor

Ultimately, these two contracts between the involved entities needed to convey the scope of the data processing as well as the rights and obligations of the parties, as stipulated by the law. While in Section 4.4, we will give an overview of the general set of rules to be stated in processing contracts as defined on European level by the Directive 95/46/EC, the Söderhamn pilot required the national data protection law of Sweden to be taken into account. According to Article 4 p.1a of Directive 95/46EC, the location of the responsible entity's establishment (in this case the Söderhamn-based Norrtullskolan) determines the applicable national law (which was here the Swedish Data Protection Act).

To determine the content of the processing contracts, we have to rely on Section 30 of the Swedish Personal Data Act, which conveys the requirements of processing contracts correlating to the European umbrella Directive. It states as follows:

Section 30

A personal data assistant and a person or those persons who work under the assistant's or the controller of personal data's direction may only process personal data in accordance with instructions from the controller of personal data.

There shall be a written contract on the processing by the personal data assistant of personal data on behalf of the controller of personal data. It shall be specifically stipulated in the contract that the personal data assistant may only process personal data in accordance with instructions from the controller of personal data and that the personal data assistant is liable to take those measures referred to in Section 31, first paragraph.

If there are special provisions in a statute or other enactment concerning processing of personal data in public operations as regards matters referred to in the first paragraph, these shall apply instead of that stated in the first paragraph

By the setup of the processing contract between Norrtullskolan and EDOC, which explicitly manifests the instructions of the school and the frame conditions of the data processing within the pilot architecture, the basic requirements were fulfilled. Likewise, this also applies for the sub-processing contract between EDOC and NSN. According to the Swedish Data Protection Act, any other special provisions were not applicable in this specific pilot setting. Hence, the contractual fixation according to Section 30 was sufficient. Thereby, Section 30 explicitly refers to the following Section 31, which stipulates the minimum security measures to be taken:

Section 31

The controller of personal data shall implement appropriate technical and organisational measures to protect the personal data that is processed. The measures shall provide a level of security that is appropriate having regard to

a) the technical possibilities available,
b) what it would cost to implement the measures,
c) the special risks that exist with processing of personal data, and
d) how sensitive the personal data processed really is.

If the controller of personal data engages a personal data assistant, the controller of personal data shall ensure for him/herself that the personal data assistant can implement the security measures that must be taken and ensure that the personal data assistant actually takes the measures.

The supervisory authority may decide on security measures.

In accordance with Section 31 Swedish Data Protection Act, technical and organisational measures guaranteeing the protection of the pilot participant's personal data were stipulated in both contracts. In particular, measures were taken to adhere to the principles of data avoidance and data minimization. Therefore, NSN would by default not be involved with the processing of any personal data. Prior to the start of the Söderhamn pilot, NSN assisted EDOC in setting up the pilot architecture to be tested. Once these preparations were finished, EDOC deactivated NSN's direct administrative access to this system, so NSN would never have direct insight into the personal data of the pilot participants. To enable NSN to perform their assistance and debugging tasks, a strict communication procedure was set up for cases EDOC needed the help of NSN for troubleshooting technical problems. This entailed the anonymisation of log files or screenshots by EDOC before forwarding them to NSN. Only in exceptional cases when such an anonymisation would not be possible or sufficient to perform the debugging or other mandatory contractual obligations, NSN would be authorized to receive and process log files or screenshots of the system still containing personal data. For such specific cases, the sub-processing contract foresees that NSN is bound to the same precautionary measures of data protection as EDOC.

Furthermore, the realisation of the minimum requirements on European level by the Swedish national law still allowed for an even higher level of protection for the personal data of the pilot participants. Since NSN as German partner based in Munich is in itself bound to German law, the even more strict requirements for processing contracts derived from the German data protection law (BDSG) were also integrated into the sub-processing contract between EDOC and NSN. § 11 (2) BDSG contains a list of ten minimum requirements with regard to the collection, processing or use of personal data on behalf of others. In specific, Section 11 (2) of the German BDSG states:

(2)
The processor shall be chosen carefully, with special attention to the suitability of the technical and organizational measures applied by the processor. The work to be carried out by the processor shall be specified in writing, including in particular the following:

1. *the subject and duration of the work to be carried out,*
2. *the extent, type and purpose of the intended collection, processing or use of data, the type of data and category of data subjects,*
3. *the technical and organizational measures to ensure the security of the processed personal data,*
4. *the rectification, erasure and blocking of data,*
5. *the processor's obligations, in particular monitoring,*
6. *any right to issue subcontracts,*
7. *the controller's rights to monitor and the processor's corresponding obligations to accept and cooperate,*
8. *violations by the processor or its employees of provisions to protect personal data or of the terms specified by the controller which are subject to the obligation to notify,*
9. *the extent of the controller's authority to issue instructions to the processor,*
10. *the return of data storage media and the erasure of data recorded by the processor after the work has been carried out.*

To enable NSN in performing their tasks and yet to limit the contact with personal data as much as possible, a step-by-step procedure regarding the fixing of technical errors was outlined in the contract. By adhering to this routine rigidly, the exposure of NSN to personal data was kept to a minimum. Additionally, certain obligations of keeping protocols and logging were integrated into the contract. Thereby, the controller was able to check the documentation for cases of misuse, unauthorized access, or actions not in compliance with the given instruction. Last but not least, the contract stated that data should be generally anonymised, wherever possible.

To comply with the generic minimum requirements as stated in the Directive 96/45EC, the contract had also to guarantee that NSN had implemented

> *"[…] appropriate technical and organizational measures to protect personal data against accidental or unlawful destruction or accidental loss, alteration, unauthorized disclosure or access, in particular where the processing involves the transmission of data over a network, and against all other unlawful forms of processing."*[8]

[8] Art. 17 (2) Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data; thereinafter EU Data Protection Directive 95/46/EC.

However, since NSN is established in Germany, these necessary measures had to be in compliance with the national data protection law of Germany (BDSG).[9] Therefore, employees of NSN had to be obliged to confidentiality[10], and their technical and organisational measures had to be in compliance with § 9 BDSG.

Notwithstanding the fact that the minimum requirements were already fulfilled, the contract entails several additional elements. So it informs that NSN is bound to the German BDSG and has, compliant to § 4f BDSG, a data protection officer appointed. Furthermore, the contract informs that a documentation according to §§ 4e and 4g ff. BDGS exists. It was decided to include these requirements for two main reasons. Primarily, the goal was to protect the data of the data subjects as much as possible, while still being able to test a prototype using Privacy-ABC technologies in a real life scenario. Secondly, including requirements stated by the German BDSG, it could be shown that the operation of the pilot architecture under the governance of one of the strictest data protection laws in Europe is possible.

2.7.3 Specific Legal Requirements of the Söderhamn Pilot

In a next step, distinctive requirements of the Swedish Personal Data Act had to be considered. In this specific case, Section 36 of the Swedish Data Protection Act requires a notification of the 'supervisory authority' regarding the automated processing of personal data prior to the start of the processing. According to Section 37, such a notice is not necessary when a 'personal data representative' has been appointed. In such a case, the supervisory authority needs only to be informed about the appointment of this representative and his or her identity. To comply with these legal requirements, the responsible Swedish data protection authority was notified about the upcoming pilot and given information about the identity of the personal data representative of the Söderhamn commun, who governs the personal data protection actions of the Norrtullskolan. Furthermore, the personal data representative was also informed about the pilot and he was given detail information about the means and purposes of the personal data processing, including the two data processing contracts, the consents forms for the participants and a rough overview over the technical details of the pilot architecture to be implemented.

2.7.4 Considerations about inspectable Areas

The pilot architecture set up for the 2nd round of the Söderhamn pilot should provide the participants with a greater number of Restricted Areas in which they were able to interact. It was agreed that in general, this system should have fully anonymous Restricted Areas by default. However, since the school had no experience with such a communication platform, it could not anticipate how the pupils would behave when using this system. Taking into account the strict legal responsibility of the Norrtullskolan for the minors, it was agreed upon that the inspection feature should be implemented for most of the RA's, so the school could maintain the ability to interfere if the physical and mental safety of a participant would be at risk. This intention clearly contradicted the goal of showing the full capacity of the pilot architecture regarding completely anonymous activity of its users. To cushion this contradiction, some RA's were set up which did not provide any inspection functionality. But the number of such un-inspectable areas was smaller, so the school was still able to monitor these RA's comparable to the activity of any forum moderator. Moreover, strict guidelines were put into place in which cases inspection would be allowed only. These guidelines also had to be aligned with the

[9] Art. 17 (2) EU Data Protection Directive 95/46/EC, (§ 11 (4) S.1 BDSG.

[10] § 5 BDSG

school’s legal responsibilities as prescribed by the Swedish law. Therefore, the cases, in which inspection should be possible, were stipulated as ‘inspection grounds’ as follows:

- Situations implying a severe threat to the life, or the physical/mental integrity of a person
- Breaches of the Norrtullskolan policy against discrimination and degrading treatment.
- An existing court order or other valid administrative request

To protect the participants against arbitrary identity revelation, a two-step process was implemented by setting up a School Inspection Board which should perform an initial assessment of reported content. This School Inspection Board did not consist of teachers of the school only, but included also parties representing the interests of the participants (parents/pupils). Only if the School inspection Board decided that a valid inspection ground was given, the case would be forwarded to the school principal, who, as the ultimate legally responsible person within the sphere of the school, would fulfill the role of the inspector and reveal the identity of the reported participant. Before the start of the pilot, each participant was informed about the inspection possibility, the inspection process itself, and the cases in which inspection could be performed. This information was also necessary to enable the participants to give a valid, informed consent for the processing of their personal data during the pilot.

3 Experiences and Feedback of the Patras Pilot

Students of the University of Patras are the only Users of this pilot. The main focus is to offer a privacy-respecting means of evaluating a course of this university without being influenced by the professors who gave the course. In the past, the evaluation was paper-based and made directly in the lecture room in the presence of the professors. In the Patras pilot, the course evaluation can now be made from the student's home. Using Privacy-ABC technologies, the students can even perform the course rating multiple times with only the last rating being counted. This has the advantage that ratings made in the presence of others (e.g. the professors) can easily be 'overwritten' when the student has moved to a trusted environment.

The Tombola System and the Inspector are new entities which were introduced in the 2nd round of the Patras pilot. The idea is to offer a 'special prize' to one student who participated in the course evaluation. With this measure, it is assumed that more students will make use of the online course rating via Privacy-ABC technologies. After a student successfully completed the course evaluation, she will receive a tombola credential which contains the matriculation number. So contrary to the Söderhamn pilot, the Patras pilot has a 2nd Issuer, the Course Evaluation System. After the evaluation period ended, the students can visit the Tombola System and provide this system with a presentation token containing the 'inspectable' matriculation number encrypted with the public key of the Inspector. In the end, one presentation token is drawn as 'winner' of the tombola. Now the Inspector comes in and reveals the identity of the winner using his private Inspector's key. Contrary to the Söderhamn pilot, the inspection grounds in this pilot are not critical. Inspection is only performed on the winner token. It must only be guaranteed, that the identity of the winner is only revealed; not the identity of all students who sent a presentation token to the Tombola System (see [D71] for more details on pilot scenarios).

3.1 Setup

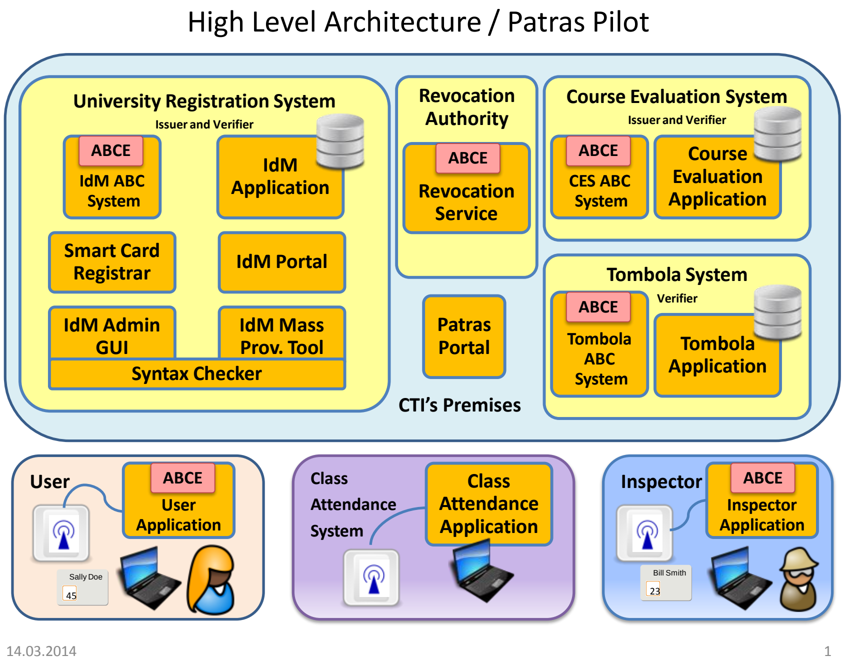


Figure 33: Final High Level Architecture of the Patras Pilot

As in the Söderhamn pilt, three teams developed the pilot applications. WP4 of ABC4Trust developed the Revocation Authority (RevAuth), the Inspector Application and the User Application. CTI developed the Patras Portal, the Course Evaluation System, the Tombola System and the Class Attendance System. All entities within the University Registration System were developed by NSN.

The above mentioned teams have provided contents in the following chapters on the development phases until the Integration Test (see 3 to 6 of Figure 2). Contributions to the deployment phase (7) and the system test phases (8) were provided by CTI and NSN. And finally, the operation phase (9) is handled by CTI alone.

Contrary to the Söderhamn pilot, every User is provided with a smart card which stores one university credential of type Idemix, one tombola credential of type Idemix and one course credential of type U-Prove. The key length of both Idemix and U-Prove is set to 1024. Even though the course credential is of type U-Prove, there is no need to support reIssuance since Users visiting the Course Evaluation System will be requested to present their scope-exclusive pseudonym (in order to disable multiple evaluations from the same User) and since this pseudonym makes the Users linkable anyhow.

The students in Patras will apply for credentials themselves and therefore receive smart cards which are not mapped to any identity. The 2nd round of the Patras pilot is making use of the new crypto architecture. Therefore, the Course Evaluation System can launch ‘advanced issuance’ and pass a hidden ‘carry-over-attribute’ (i.e. the matriculation number) into the tombola credential. Because of the new crypto architecture, the key-length of 1024 for both Idemix and U-Prove credentials can be chosen.

Contrary to the Söderhamn pilot, the Revocation Authority is generic, meaning that it can be used in other pilots. This pilot is the first to prove the interoperability between U-Prove and Idemix as the presentation policy requested by the Course Evaluation System requires the User to have a course (U-Prove) and a university (Idemix) credential. Firefox is the default browser in the Patras pilot.

3.2 User

As mentioned in earlier documents, the Users of this pilot are University of Patras students that attend courses at the Computer Engineering and Informatics Department. The User block presented in the architecture Figure 33, consists of the User herself operating her personal computer. On her personal computer the ABC4Trust software which includes a User Application and a Firefox browser plugin is installed. Attached to her PC, there is a USB smart card reader and her ABC4Trust smart card is inserted in it.

3.2.1 User Application

See Section 2.2.1 for an in-depth explanation of the User Application. The only difference from the Söderhamn pilot lies in the browser plugin, and that the 2nd round of the Patras pilot was run on the new crypto architecture.

3.2.1.1 Implementation

See Section 2.2.1.1 for the details of the implementation steps.

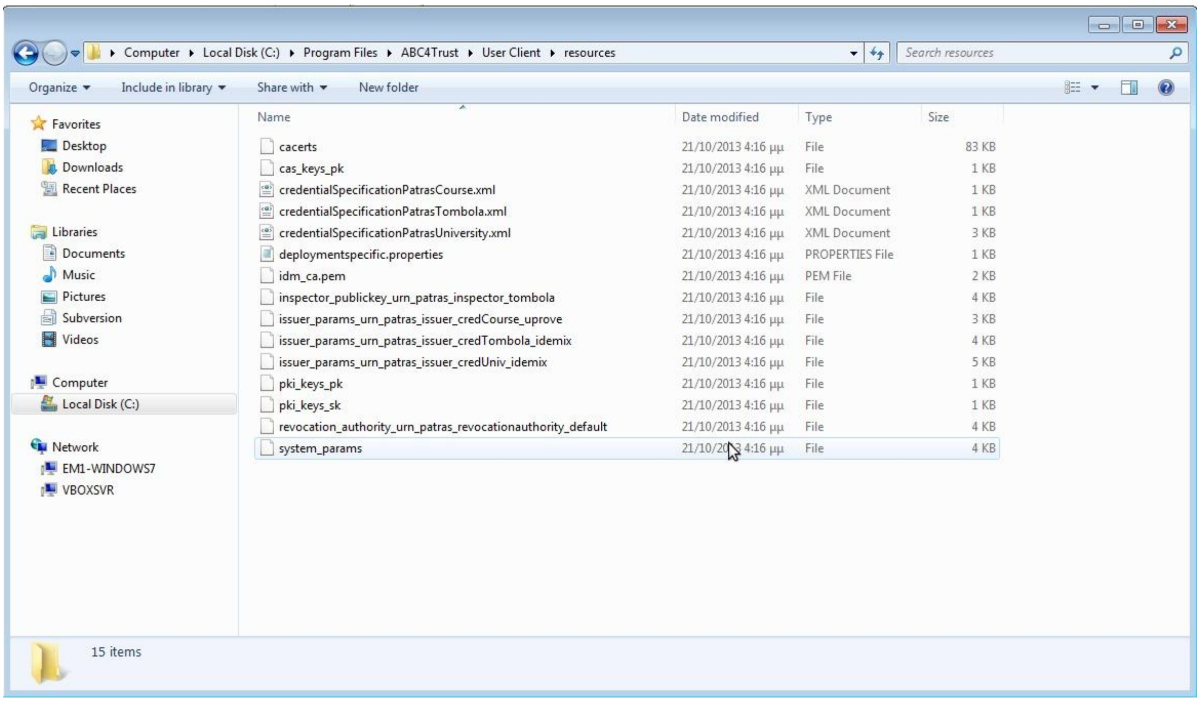
The Patras pilot had the additional requirement for the browser plugin that it needed to be able to show the attendance data for the student. This was a simple addition, but useful in the pilot.

3.2.1.2 Integration

See Section 2.2.1.2 for the details of the integration steps.

3.2.1.3 Deployment

Building the User Application was done by compiling the corresponding sample project provided by WP4. The pilot administrators had to make sure that the appropriate resources (system parameters, Issuer resources, PKI-keys, credential specifications etc.) were located inside this project. The result of the compilation was the creation of some bundles (containing all the necessary resources) that were uploaded to the SFTP server, where the installer was hosted. As soon as a User would execute the ABC4Trust installer, the resources would be downloaded from the corresponding bundle (which was built for hardware smart cards) of the SFTP server and would be extracted on his PC.



Name	Date modified	Type	Size
cacerts	21/10/2013 4:16 μμ	File	83 KB
cas_keys_pk	21/10/2013 4:16 μμ	File	1 KB
credentialSpecificationPatrasCourse.xml	21/10/2013 4:16 μμ	XML Document	1 KB
credentialSpecificationPatrasTombola.xml	21/10/2013 4:16 μμ	XML Document	1 KB
credentialSpecificationPatrasUniversity.xml	21/10/2013 4:16 μμ	XML Document	3 KB
deploymentspecific.properties	21/10/2013 4:16 μμ	PROPERTIES File	1 KB
idm_ca.pem	21/10/2013 4:16 μμ	PEM File	2 KB
inspector_publickey_urn_patras_inspector_tombola	21/10/2013 4:16 μμ	File	4 KB
issuer_params_urn_patras_issuer_credCourse_uprove	21/10/2013 4:16 μμ	File	3 KB
issuer_params_urn_patras_issuer_credTombola_idemix	21/10/2013 4:16 μμ	File	4 KB
issuer_params_urn_patras_issuer_credUniv_idemix	21/10/2013 4:16 μμ	File	5 KB
pki_keys_pk	21/10/2013 4:16 μμ	File	1 KB
pki_keys_sk	21/10/2013 4:16 μμ	File	1 KB
revocation_authority_urn_patras_revocationauthority_default	21/10/2013 4:16 μμ	File	4 KB
system_params	21/10/2013 4:16 μμ	File	4 KB

Figure 34: User Application Resources

As can be seen on Figure 34, the installer extracted into the installation folders of the User Application the necessary resources for interacting with the rest of the pilot systems. Among these resources were the system parameters, the credential specifications, the Issuer parameters, the revocation authority parameters and the Inspector’s public key.

In order to verify the User Application, a series of tests were performed using hardware and software smart cards. Firstly, the pilot developers/testers wanted to test how the User Application behaves when a User interacts with the pilot systems i.e. when the User is involved in Privacy-ABC issuance and verification protocols. Using software smart cards they interacted with the University Registration System, in order to check if the functionalities were working as they should. They tested the smart card registration functionality (presentation of a scope exclusive pseudonym for scope “urn:patras:registration”) as well as the University and Course credentials issuance procedure. After that, communicating with the Course Evaluation System was required in order to test the presentation of both the University and Course credentials as well as the issuance of the Tombola credential. Finally, by interacting with the Tombola System they checked if presenting the Tombola credential was working as expected.

As soon as hardware smart cards were supported, all of the above tests were repeated using properly initialized smart cards. Additionally, the pilot developers tested the functionalities provided by the Firefox browser plug-in i.e. changing the smart card PIN value, unlocking the smart card using the PUK value, backing up the smart card contents and restoring it on a new smart card, finding out the number of attendances stored on the card and finding out the credentials stored on the card.

Any problems or bugs detected during these tests were reported immediately to WP4, by sending the appropriate log files. They were responsible for fixing the possible bugs or advise us on how to solve the issues that came up.

3.2.1.4 Operation

The administrators provided the pilot Users with an ABC4Trust installer (installer.exe) in order to simplify the setup of the User Application at their PCs. When executed, the installer would check the environment for the necessary programs (e.g. Java) and then would download the pilot resources from the SFTP server. Finally, it would install and start the ABC4Trust User Service as well as the Firefox plugin.

Throughout the pilot operation the participants reported to the administrators some issues that they had to deal with. Below these issues are listed:

Some students had in their possession a smart card whose PUK value was less than 8 digits long (this case could happen during smart card initialization). If the student's smart card was locked (after inserting a wrong PIN for three times in a row) and she tried to unlock it using the User Application, it would not be possible as it would require the PUK value to be strictly 8 digits long. Thus, the Users could not unlock their smart card and they had to contact a pilot administrator to assist them with this issue (a special script was developed for this issue).

Moreover, some students noticed that when their smart card was locked and they removed it out of the smart card reader and re-insert it, the User Application would still ask them to enter the PIN for any operation whereas it should ask for the PUK value, since the card was actually locked.

Sometimes, when a student was trying to enter the Course Evaluation System by presenting her credUniv along with her credCourse the User Application along with the smart card could not complete the proof successfully. The message that was reported at the log files of the User Application was "incompatible secret binding". However, this issue would go away if the User restarted the User Application and thus it was not so important. Moreover, this issue was rather related to the smart card instead of to the User Application. Finally, an issue that was discovered was that the User Application along with the smart card could not complete successfully a proof towards the Course Evaluation System when the User who had obtained the required credentials on her smart card (credUniv, credCourse) had attempted to make such a proof when her attendance counter had not reached the pre-defined threshold (we here note that the attendance threshold is defined by the pilot administrator upon smart card initialization – see Section 3.2.2.3). During the proof that would fail due to insufficient counter, the 'state' of the stored credUniv on the card would change from 'presentable' to 'presentation committed' and would not change back upon the proof failure. Thus, when the counter had reached the threshold and the User was engaged in a new proof the User Application would throw an exception due to the credential not being in a state for presentation (the User ABCE was confused since the credential's state was not 'presentable' and would try to do issuance). This issue would be resolved if the User would re-obtain her credUniv from the University Registration System, since re-issuing the credential would change its state again to 'presentable'.

3.2.2 User Smart Cards

Section 2.2.2 explains the problematic issues which came up with the smart cards. The difference from the Söderhamn pilot is that a new CE architecture was used in the 2nd round of Patras. This means that one could have remedied one of the major issues we had with the smart card, namely that the hash function for U-Prove was not the same that Idemix and the smart card used. However, to keep compatibility with the real U-Prove implementation, it was chosen not to change this. This means that the smart card retained the two different implementation calls for the hash function.

Also, one of the main differences in the cards were the parameters of the cards themselves, e.g. the amount of “Issuers” allowed on the cards. This essentially means that there is another maximum allowed number of credentials than in Söderhamn.

The difference in particular for how the smart cards were used in the Patras pilot compared to the Söderhamn pilot, was to rely on the counter mechanism for enabling the generation of a proof in the evaluation of the course.

For the backup and restore feature there were some discussions as to the security aspects of it. If e.g. only the counter was backed up, this feature could be misused to distribute the backup to other students who could then restore and suddenly have a working card for evaluation even though they really didn’t attend enough lectures. In order to solve the issue that students might try to restore a foreign counter value, restore was only enabled if the device ID of the old and new card is identical.

And in order to solve the cloning issue, backup was limited to the counter value and deliberately excluded the device secret. As consequence, the new card will contain a new device secret.

Unfortunately, this introduces a new risk: an adversary can evaluate the course, claim his card stolen, and evaluate again.

The solution selected by this pilot in order to prevent multiple evaluations by the same User is to only allow new cards before the evaluation period has started. This measure only works because

- the old scope-exclusive pseudonym will be deleted from the IdM Database preventing the old smart card from logging in to the IdM Portal and from obtaining new credentials
- the User’s credentials will be revoked

once a card is reported lost.

The Patras pilot thus needs all the features of the ABC4Trust framework in order to make it work as intended.

3.2.2.1 Implementation

See Section 2.2.2.1 for further details on this subject.

3.2.2.2 Integration

See Section 2.2.2.2 for further details on this subject.

3.2.2.3 Deployment

The smart card customization was done by the CTI pilot administrators. After obtaining some blank MULTOS smart cards the administrators did the following steps:

- They printed and placed stickers containing the ABC4Trust logo and a unique identifier on each smart card in order to be able to distinguish between them, as can be seen on Figure 35
- Using the MUtil tool, they could load on each smart card the appropriate precompiled application (ALU) for the Patras pilot. In order to do so, they should also load the "Application Loading Certificate" (ALC) that was provided by CRX.
- Finally, using a script written in Java, the administrator could initialize each smart card with a unique identifier, the smart card PIN/PUK values, the necessary pilot PKI keys and Issuer resources, the card's scope exclusive pseudonym for scope "urn:patras:registration" as well as the card's attendance threshold.

Figure 35: An ABC4Trust Smart Card

The smart card is involved in every pilot operation, assisting its owner in some computations and storing data in its memory. In order to test the functionality of the smart cards and the ABC4Trust application loaded on them, the pilot developers had to perform all the pilot steps using the User Application along with a hardware smart card.

As described also in Section 3.2.1.3, they interacted with the pilot systems in order to verify that the User Application is working smoothly with the smart card when it is involved in Privacy-ABC issuance or verification protocols. During the issuance of University and Course credentials at the University Registration System they tested if the credentials were stored successfully on the smart card. When trying to do a presentation of both University and Course credentials at the Course Evaluation System they verified that the smart card (along with the ABCE layer) is able to create an appropriate presentation token. At this point, they also verified that a User can only present the Course credential if her attendance counter has reached the threshold that was set during the smart card initialization procedure. Finally, the issuance of the Tombola credential at the Course Evaluation System was tested as well as its successful presentation to the Tombola System.

Moreover, the pilot developers tested how the smart card responds to the operations provided by the Firefox plugin. They tested whether the smart card PIN can change without any problems and that a locked smart card (after inserting 3 times in a row the wrong PIN) can be unlocked using the PUK value. They verified that the "Manage Credentials" functionality is working as it should i.e. displaying on a Firefox window the credentials stored on the card and that the User can delete credentials from her smart card. Also, they tested the backup/restore functionality of the smart card content and checked that the User can successfully read the value of her attendance counter.

Finally, they used some smart cards along with the Class Attendance System in order to test the functionality of increasing the attendance counter stored on the card. This way they verified that the communication protocol between the smart card and the Class Attendance System can be completed without any problems and checked that the counter on the card is increased by one after its execution (see also Section 3.8.1.4).

3.2.2.4 Operation

In the beginning of the semester the pilot administrators provided the participants with a properly initialized smart card and its PIN/PUK values, as well as a smart card reader. During the semester, when attending a lecture, the Users would collect attendance units which would be stored on their cards. Finally, the smart card was required for obtaining their credentials and doing proofs with them when interacting with the pilot systems.

Some issues that were reported by the students during the pilot are the following:

Some students faced some difficulties with pilot operations (e.g. smart card registration at the IdM Portal) because the User Application could not communicate properly with the smart card through the smart card reader. This issue existed because the smart card reader (Omnikey 3021 USB) drivers were not installed successfully by the Windows OS. The administrators advised the Users to install manually the drivers from the Omnikey website and this issue was resolved.

Another issue that came up during the pilot, was that some students who had their smart cards in locked mode (see also Section 3.2.1.4), could not obtain attendance units during the lecture. This was expected taking into account the ABC4Trust-Lite Smart Card Application[11]. In order to deal with this issue the Users had to unlock their smart card first (using their PUK) at home, before coming to the lecture.

It was also observed by some students that the smart card could not store more than three credentials. More specifically, some students obtained next to their credUniv credential their credCourse credential twice from the University Registration System. Then, when they tried to obtain the Tombola credential from the Course Evaluation System the issuance protocol would fail due to insufficient storage on the smart card. This issue could be easily resolved since the students could delete credentials from their smart card using the Credential Manager. Thus, in cases that this issue occurred they could delete the extra credentials and continue normally with the pilot operations.

As mentioned also in Subsection 3.2.1.4, the User Application along with the User's smart card could not complete a proof towards the Course Evaluation System although the card had the required credentials and a sufficient attendance counter. The exception reported at the log files of the User Application was "Incompatible Secret Binding" and it would only go away if the User restarted the User Application. After this procedure, the User could log-in successfully at the Course Evaluation System.

Finally, the most crucial issue that was discovered during the pilot operation was the one mentioned also in Subsection 3.2.1.4. This issue was showing up when a student would try to log-in to the Course Evaluation System using her smart card which did not have a sufficient attendance counter. The result of this proof that would fail (due to insufficient counter) was that the state of the credUniv would change to 'presentation committed' and not change back to 'presentable'. This issue came up due to a bug in the ABC4Trust-Lite Smart Card Application. So, when trying to do some proofs in the future (even with sufficient counter) the credUniv could not be used since its' state was not 'presentable'. This issue could be resolved if the User would re-obtain her credUniv from the University Registration System.

[11] The ABC4Trust-Lite Smart Card Application is the precompiled application (ALU) that is loaded on the smart cards by an administrator. It was developed by CRX.

3.3 Inspector

The role of the Inspector for the University of Patras pilot was assigned to a student representative. This student representative was responsible for revealing the matriculation number from the presentation token that was selected as the winner of the raffle, from the Tombola System. The pilot administrators provided the Inspector with a smart card which contains the Inspector's secret key. Using this smart card along with a specialized tool (Inspect Tool) running on her PC, the Inspector could decrypt the matriculation number from the presentation token that won the raffle and announce the student who should collect the prize.

3.3.1 Inspector Application and Inspector Smart Cards

3.3.1.1 Implementation

The tools created for the Patras pilot are almost identical to the tools used for the Söderhamn pilot described in 2.3.1.1. Minor changes have been made to command line arguments plus an extra tool had been added specifically to run and test Inspection using a private key stored in a file.

Contrary to the Söderhamn pilot, no 'Inspector Wrapper' was implemented.

3.3.1.2 Integration

Building the Inspector Application was done by compiling the sample projects provided by WP4. We had to include in the corresponding resources folder the system parameters as well as the Issuer parameters and the credential specification of the tombola credential. When compiling the project, the Inspector Setup Tool and the Inspect Tool were created. The Inspector Setup Tool was used for generating the Inspector key pair and loading it on to a smart card and the Inspect Tool was used for performing the actual inspection of a presentation token.

In order to test the Inspector Application, the pilot developers used the Inspector Setup Tool in order to generate the Inspector's key pair (private/public key). As soon as the Inspector's key pair was generated they loaded on the Inspector smart card the corresponding secret key using the Inspector Setup Tool. After that, they picked up some presentation tokens from the database of the Tombola System. These presentation tokens contained the Users' matriculation numbers encrypted with the public key of the Inspector. Using the Inspect Tool along with the Inspector smart card they tried to reveal the matriculation number from these presentation tokens. This way, it was verified that the Inspect Tool was working as expected and that the smart card was loaded with the correct secret key.

The pilot developers also did some tests using a different smart card (with another Inspector secret key loaded in it) than the Inspector's along with the Inspect Tool. They tried to reveal the matriculation number from the presentation tokens but it failed. This way they verified that only the Inspector can reveal the private information from the Users' presentation tokens.

3.3.1.3 Deployment

Following the same procedure as described in Section 3.2.2.2, the pilot administrators initialized a smart card that would be used for the Inspector. Using the Inspector Setup Tool, they loaded on this smart card the Inspector's secret key which they had generated earlier.

In order to test the Inspector's smart card functionality they had to follow the same procedure as described in Section 0. This way they verified that the Inspector' smart card was working smoothly along with the Inspect Tool.

3.3.1.4 Operation

When the pilot officially started, the CTI administrators used the tools that were created from the Inspector project compilation. They used the Inspector Setup Tool in order to generate the Inspector key pair and loaded it on a smart card. This smart card was provided along with its PIN/PUK values to a student representative. The representative was elected by the students in one of the meetings with them in the lecture room.

The Inspector smart card was used along with the Inspect Tool at the end of the tombola. The matriculation number was revealed out of the presentation token that was selected as the winner of the lottery from the Tombola System. No technical issues came up when using the Inspector smart card.

When the lottery was over, the pilot administrators came in contact with the Inspector and was invited at the CTI premises. The administrators provided to the Inspector the presentation token that the Tombola System selected as the winner of the raffle. Then the Inspector, using the Inspect Tool along with her smart card found out the matriculation number of the winner and announced it to the rest of the students. No technical issues came up when using the Inspector Setup Tool or the Inspect Tool during the pilot.

3.4 University Registration System

The University Registration System is very similar to the School Registration System (see Section 2.4 and Figure 3).

The main differences are only that

- all GUIs are in English
- the attribute values stored in the IdM Database are in English
- reIssuance is not required, but still supported
- support of the IE is not required, but still supported
- other credential specifications, other issuance and presentation policies are being used
- a different set of images for the credentials is stored in the IdM Portal

As the Users of the Patras pilot will not receive customized smart cards, they need to visit the University Registration System in order register their smart card and to obtain their university and course credentials.

3.4.1 IdM Application

3.4.1.1 Implementation

Please refer to Subsection 2.4.1.1 for details on the implementation. Please note, that the Patras pilot's scope for generating the scope-exclusive pseudonym is different, i.e. 'urn:patras:registration'.

3.4.1.2 Integration

The integration of the IdM Application is the same as described in 2.4.1.2.

3.4.1.3 Deployment

CTI is not using a 'hosted' Ubuntu firewall. Instead, a pair of dedicated firewalls (Cisco Pix-535) located at the edge of the University Registration System protects all applications (and all servers) located within this zone.

Shifting the IdM Application from the integration testing environment to the target site involves identical tasks as described in 2.4.1.3. This includes adjusting the WSDL entries in the pom.xml and in the abcHander.wsdl files and in customizing the URLs on the web page of the IdM Application.

Contrary to the Söderhamn pilot, the Apache-Tomcat containers hosting the University Registration System applications enforce HTTPS using self-signed X.509 certificates which contain the FQDNs of the server hosting the application as 'common name'.

CTI and NSN executed the system tests. Before the operation phase of the pilot, NSN invited to a formal handover of the University Registration System and the Revocation Authority. CTI thereafter configured their firewalls to block traffic coming from NSN. After 'cutting the lines', the pilot entered the operation state.

3.4.1.4 Operation

During the pilot operation, the administrators had to make sure that the Tomcat container hosting the IdM Application was up and running at the server hosting the IdM system (see [D72], figure 6). As soon as an administrator would initialize the IdM Application on the Tomcat server, he should make sure that it was launched without any exceptions.

Throughout the pilot, the students faced some usability difficulties when they interacted with the IdM Application. Some issues that were reported are mentioned below:

Some of the students had trouble logging in the IdM Application with their matriculation number and the One Time Password that was provided by the pilot administrators. By checking the log files it was discovered that these students' browsers were presenting old invalid cookies. In order to solve this issue, they were advised by the administrators to clear their browser's cache/cookies and re-try to log-in. After cleaning the cache this issue would go away and they could log-in successfully to the IdM Application.

Some students stated that they could not log-in to the University Registration System via the IdM Application (either using their OTP or Privacy-ABC technologies) because they tried to do so by contacting the IdM Application directly. As a result, the session identifier variable (which would be set by the IdM Portal) did not exist and the students were not able to log-in. But this was an intended behaviour as logging in will only be allowed via a SAML client.

Finally, some students configured by mistake their Firefox browser to "remember" the One Time Password that was provided to them for logging-in the University Registration System. Thus, as soon as they had registered their smart card and the One Time Passwords was disabled, every time they visited the University Registration System the browser was trying to log them in with the password. As the password was disabled, authentication would fail and after a few times their account would be locked. After that, they could not log-in to the University Registration System even using Privacy-ABC technologies and they had to contact a pilot administrator in order to re-enable their accounts.

3.4.2 IdM Portal

3.4.2.1 Implementation

Please refer to 2.4.2.1 for details on the implementation.

Since the IdM Portal issues only two different types of credentials, only 8 servlets are implemented which are dedicated to handling Privacy-ABC technologies.

3.4.2.2 Integration

The integration of the IdM Portal is the same as described in 2.4.2.2.

Since the course credential is automatically mapped to U-Prove and since the university credential is automatically mapped to Idemix, there is no need to query the IdM Database any more prior to launching issuance, to determine which crypto engine is being used. In the IdM Portal of the Patras pilot, the crypto engine types are hardcoded to the credential types.

As in the Söderhamn integration, the IdM Portal automatically launches revocation in case the student applies for a university credential she already obtained.

3.4.2.3 Deployment

Shifting the IdM Portal from the integration testing environment to the target site involves identical tasks as described in 2.4.2.3.

After completing the system tests, NSN and CTI performed a formal handover. After CTI disabled the firewall rules allowing NSN developers to access the University Registration System, the operation phase began.

3.4.2.4 Operation

Before the pilot, the administrators had to clean the LDAP database from all the test Users' attributes and import the attributes of the students that would participate in the pilot. During the pilot the administrator's task was to make sure that the Tomcat server hosting the IdM Portal was up and running at the server hosting the IdM system.

Throughout the pilot some minor problems were reported when interacting with the IdM Portal.

Some students reported that they were unable to register their smart cards through the IdM Portal. Having a look at IdM Portal log files, the pilot administrators could see that the User Application was not responding with the token for the smart card registration after receiving the presentation policy. Thus, they advised the students to make sure that the User Application was able to communicate with their smart card through the smart card reader by installing the Omnikey drivers. As soon as the drivers were installed successfully, the students could finally register their smart card and obtain their credentials from the University Registration System.

Another issue regarding the smart card registration was discovered. Some students contacted the pilot administrators claiming that upon trying to register their smart card they received a message from the IdM Portal that registration was not successful. By checking the log files the administrators could see that the smart card registration actually had worked successfully. The reason that the User did not get the appropriate message in her browser could be that she had visited a web page that had caused JavaScript errors or that some of her active browser plugins contained errors.

3.4.3 IdM Mass Provisioning Tool

3.4.3.1 Implementation

Please refer to 2.4.3.1 for details on the implementation.

Contrary to the implementation provided for the Söderhamn pilot, the version of the Patras pilot does not allow the administrator to enter the scope-exclusive pseudonym, the smart card ID and the Crypto Engine Type. In effect, the administrators must always perform steps 1 to 5 described in 2.4.3.1.

3.4.3.2 Integration

The integration of the IdM Mass Provisioning Tool is the same as described in 2.4.3.2.

3.4.3.3 Deployment

Shifting the IdM Mass Provisioning Tool from the integration testing environment to the target site involves identical tasks as described in 2.4.3.3.

The two firewalls on CTI’s target site must be configured to allow LDAP requests originating from the Windows server to the server hosting the IdM Database.

3.4.3.4 Operation

The IdM Mass Provisioning Tool was used by the pilot administrators in order to load the participants’ attributes into the LDAP database. This tool was hosted on the Windows server and provided to the administrators an interface for handling the LDAP database. The administrators had to create appropriately formatted .csv files containing the Users’ attributes in order to feed the IdM Mass Provisioning Tool.

The only issue that the administrators had to deal with regarding the IdM Mass Provisioning Tool was that some special characters (e.g. #,$,^,&) contained in the One Time Passwords generated by an administrator script were not accepted as valid characters and could not be stored on the LDAP database. Thus, the administrators had to re-generate some Users’ One Time Passwords with valid characters only and then they would be stored on the database without any problems. Moreover, this issue could have been resolved by configuring the rules for selecting the passwords via REGEX.

3.4.4 IdM Admin GUI

3.4.4.1 Implementation

Please refer to 2.4.4.1 for details on the implementation. Please note, that the Patras pilot’s scope for generating the scope-exclusive pseudonym is different, i.e. ‘urn:patras:registration’.

The IdM Admin GUI offers to Users with their specialRole attribute equal to ‘university administrator’ special administration rights.

3.4.4.2 Integration

The integration of the IdM Application is the same as described in 2.4.4.2.

3.4.4.3 Deployment

Shifting the IdM Application from the integration testing environment to the target site involves identical tasks as described in 2.4.4.3.

After completion of the system tests and a formal handover between NSN and CTI, the operation phase began.

3.4.4.4 Operation

In order for the IdM Admin GUI to be available online the pilot administrators had to ensure that the Tomcat web server hosting the IdM Admin application was up and running. Moreover, they had to set a firewall rule so that the port on which the application was running would be accessible only from the administrator IP addresses.

During the pilot operation, no issues regarding the IdM Admin GUI came up. The pilot administrators using their smart cards could log-in to the IdM Admin GUI and if necessary revoke the participants’ credentials or reset their One Time Password.

3.4.5 Smart Card Registrar

3.4.5.1 Implementation

Please refer to 2.4.5.1 for details on the implementation. Please note, that the Patras pilot’s scope for generating the scope-exclusive pseudonym is different, i.e. ‘urn:patras:registration’.

Contrary to the Söderhamn implementation, the Patras instance does not request the administrators to enter the crypto engine types.

3.4.5.2 Integration

The integration of the Smart Card Registrar is the same as described in 2.4.5.2.

3.4.5.3 Deployment

Shifting the Smart Card Registrar from the integration testing environment to the target site involves identical tasks as described in 2.4.5.3.

NSN installed the Smart Card Registrar on the target site of CTI. After system tests and a formal handover, the operation phase began.

3.4.5.4 Operation

The Smart Card Registrar application was hosted on a Tomcat web server so the pilot administrators had to make sure that the server was running during the pilot. Additionally, since only administrators should be able to access the Smart Card Registrar they had set a firewall rule which restricted access to certain IP addresses.

During the pilot operation, no problems regarding the Smart Card Registrar were discovered. The pilot administrators using the Smart Card Registrar could register to the LDAP database the pseudonyms of the valid smart cards that would be distributed to the pilot participants.

3.4.6 IdM ABC System – Issuer and Verifier

The ABC System of the University Registration System and the ABC System of the School Registration System (see Section 2.4.6) are based on the same application code. The main differences between both are

- the ABCE library (Patras is using version '1.1.13' and Söderhamn is using the older version 'soderhamn-1.0.16')
- each credential is mapped to a specific crypto engine type
- the XMLs (credSpecs, issuance policies and presentation policies)
- all parameters
- the key length (Söderhamn uses 2048; Patras uses 1024)

3.4.6.1 Implementation

Please refer to 2.4.1.1 for details on the implementation. Please note, that the Patras pilot's scope for generating the scope-exclusive pseudonym is different, i.e. 'urn:patras:registration'.

Even though reIssuance is not required by this pilot, the functionality is still available. So in effect, since the University Registration System offers issuance of only 2 credentials, the 'IdM ABC System' handles 9 web service methods, whereas the 3 reIssuance methods are not used.

3.4.6.2 Integration

As the Patras pilot is using an ABCE with the new crypto architecture, there are marginal differences in the integration of the 'IdM ABC System' when compared to the Söderhamn integration described in 2.4.6.2.

3.4.6.2.1 Deployed ABCE Methods

The next subchapters will focus on the differences in the ABCE methods deployed in the Patras pilot only, as not to repeat the content of 2.4.6.2.

3.4.6.2.1.1 Init Servlet (automatically executed during startup)

- eu.abc4trust.ri.servicehelper.issuer.SpecAndPolicy

  SpecAndPolicy university = new SpecAndPolicy(CommonDefinitions.SPEC_AND_POLICY_UNIVERSITY, CryptoTechnology.IDEMIX, null, 6, 0, "/xml/credspecs/credentialSpecificationPatrasUniversity.xml", "/xml/issuance/issuancePolicyPatrasUniversity.xml", "urn:patras:revocationauthority:default", "en", "Patras");

  Next to defining the crypto engine type, the first of the yellow marked parameters fixes the maximalNumberOfAttributes and the second of the yellow marked parameters defines the numberOfTokens. The latter is relevant for U-Prove credentials only.

  Naturally, the Revocation Authority URN is different to the RevAuth of the Söderhamn pilot.

- eu.abc4trust.ri.servicehelper.issuer.IssuanceHelper

  IssuanceHelper.**initInstance**(1024, systemAndIssuerParamsPrefix, fileStoragePrefix, new SpecAndPolicy[]{university, course}, revocationAuthorityParameters_resources);

  Contrary to the Söderhamn pilot, only one keylength value can be input which is valid for both crypto engine types. The number of U-Prove tokens per issuance has also been removed from this method.

- eu.abc4trust.ri.servicehelper.verifier.VerificationHelper

  VerificationHelper.**initInstance**(systemParamsResource, issuerParamsResourceList, credSpecResourceList, inspectorPublicKeyResourceList, revocationAuthorityParameters_resources, fileStoragePrefix, presentationPolicyResourceList);

  The crypto engine type 'BRIDGED' has been removed from the initInstance method.

- eu.abc4trust.ri.servicehelper.smartcard.PKIKeyTool

  PKIKeyTool.**generateSignatureKeys**(ISSUER_SYSTEM_FOLDER, "cas_keys");

  The cas keys are not required by the Issuer. These keys are necessary for setting up the Class Attendance System (see Section 3.8.1.1). They are generated for the convenience of the administrators of the pilot.

3.4.6.2.1.2 Issuance (executed on request during operation)

- eu.abc4trust.ri.servicehelper.issuer.IssuanceHelper
  eu.abc4trust.xml.IssuanceMessage
  eu.abc4trust.xml.IssuanceMessageAndBoolean

  IssuanceMessage im_with_policy = IssuanceHelper.getInstance().**initIssuance**(specAndPolicyId, attributeValueMap);

  IssuanceMessageAndBoolean IssuanceHelper.getInstance().**issueStep**(issuanceMessage);

  The crypto engine parameter has been removed from these methods.

3.4.6.2.1.3 Revocation

Since the 2nd round of the Partas pilot is the first to make use of the ‘generic Revocation Service’ and since the interface to the RevAuth changed significantly when introducing the new crypto architecture, launching revocation from the IdM Admin GUI and the IdM Portal is significantly different.

- eu.abc4trust.xml.RevocationMessage

  The RevocationMessage class is not used any more.

- eu.abc4trust.xml.RevocationInformation

  The generic Revocation Service requires a different handling in order to extract the revocation information.

  Client client = Client.create();

  RevocationInformation riFromRevocation = null;

  String revocationURL = "unknown";

  WebResource revokeResource;

  revokeResource = client.resource(revocationURL);

  riFromRevocation = revokeResource.**post**(RevocationInformation.class, payload);

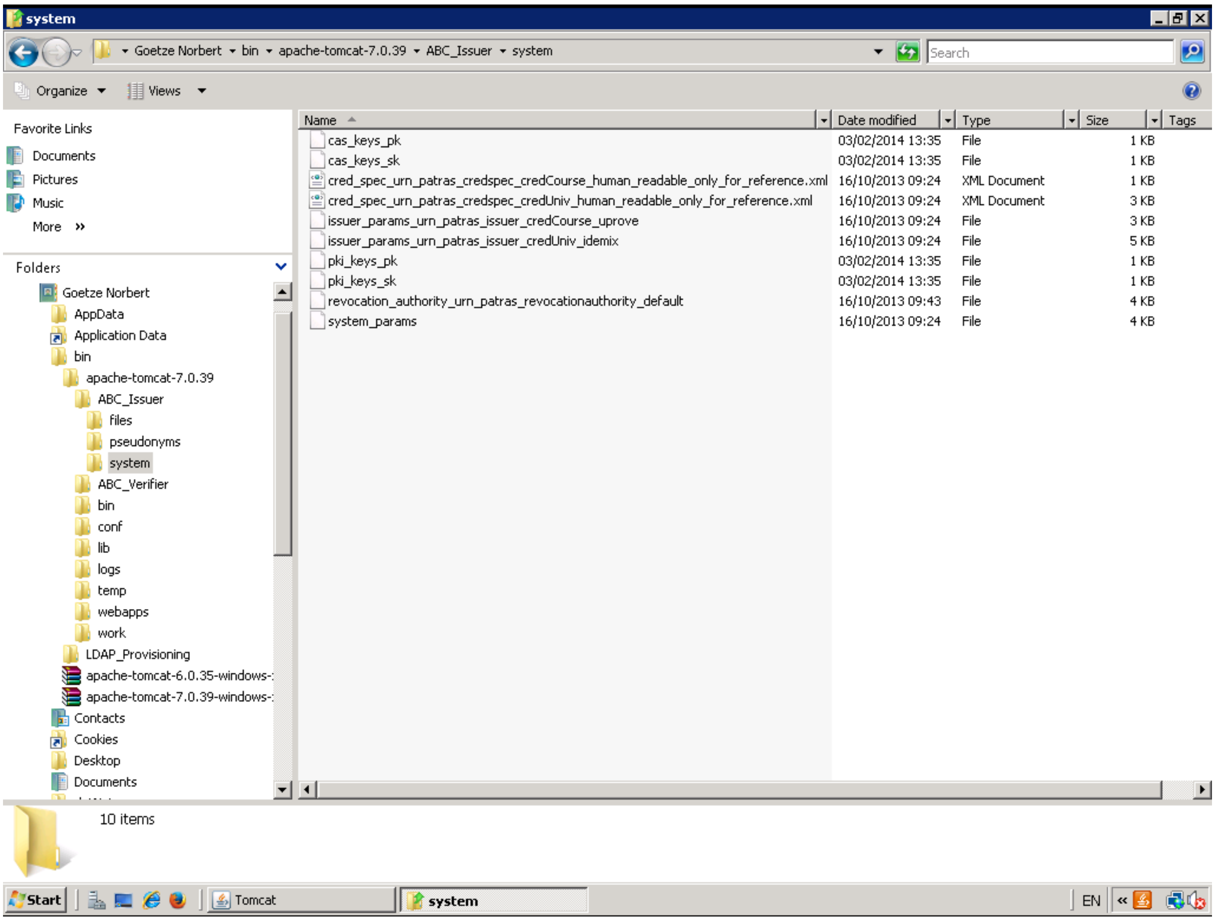
  String result = riFromRevocation.**getRevocationInformationUID**().toString();

  If the revocation information UID can be extracted from the revocation information, the revocation itself was successful.

3.4.6.2.2 *Setting up the IdM ABC System*

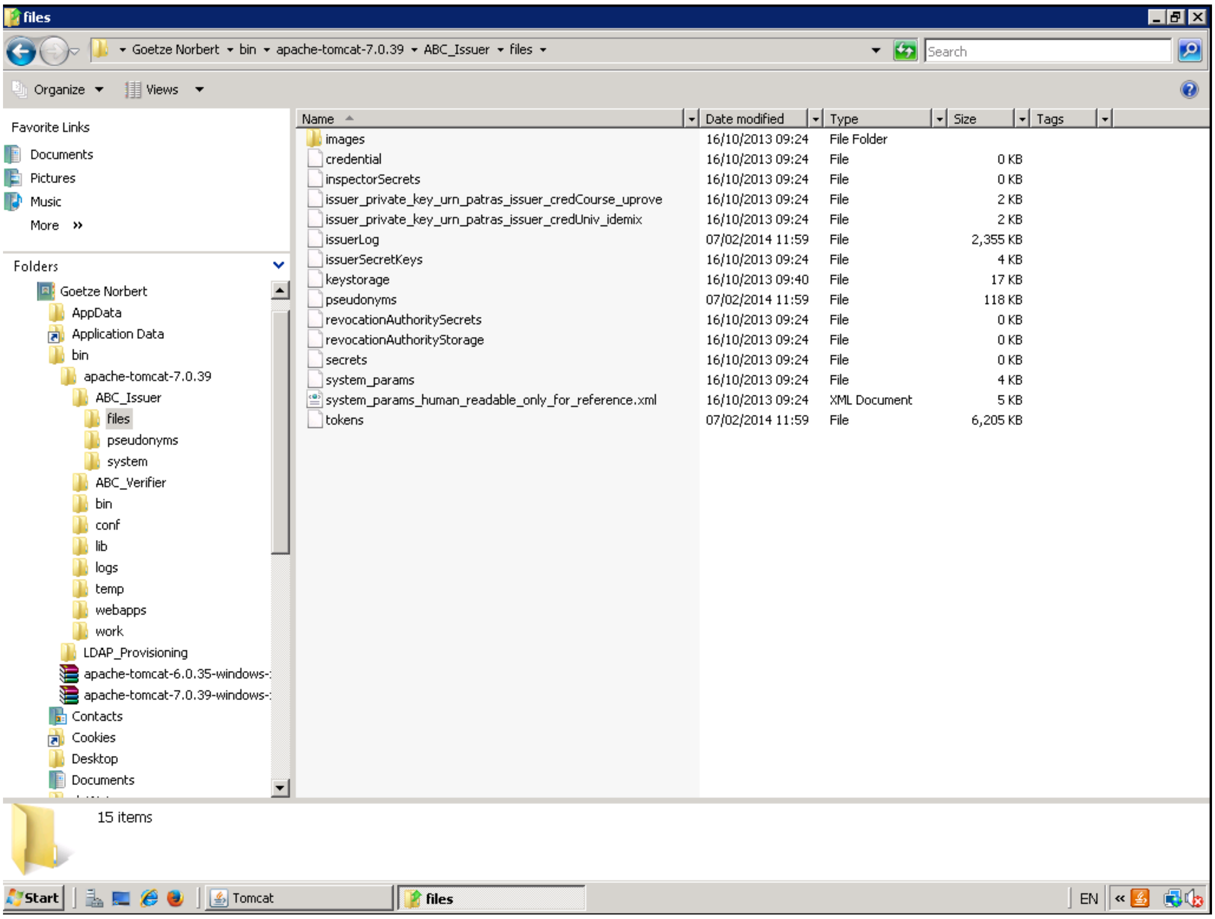
Setting up the 'IdM ABC System' of the University Registration System is identical to the steps required to set up the 'IdM ABC System' of the School Registration System (see Subsection 2.4.6.2.2). The only differences are related to the new crypto architecture:

- .NET is no longer required
- the generated files: per credential, only one version of Issuer parameters and Issuer private keys will be generated
- several U-Prove and Idemix specific files from the Söderhamn pilot are combined into a single file, e.g. the revocation authority secrets, revocation authority storage files, Inspector secrets, tokens, etc.
- the system_params_bridged generated by the old crypto architecture is now only called system_params.



Name	Date modified	Type	Size	Tags
cas_keys_pk	03/02/2014 13:35	File	1 KB	
cas_keys_sk	03/02/2014 13:35	File	1 KB	
cred_spec_urn_patras_credspec_credCourse_human_readable_only_for_reference.xml	16/10/2013 09:24	XML Document	1 KB	
cred_spec_urn_patras_credspec_credUniv_human_readable_only_for_reference.xml	16/10/2013 09:24	XML Document	3 KB	
issuer_params_urn_patras_issuer_credCourse_uprove	16/10/2013 09:24	File	3 KB	
issuer_params_urn_patras_issuer_credUniv_idemix	16/10/2013 09:24	File	5 KB	
pki_keys_pk	03/02/2014 13:35	File	1 KB	
pki_keys_sk	03/02/2014 13:35	File	1 KB	
revocation_authority_urn_patras_revocationauthority_default	16/10/2013 09:43	File	4 KB	
system_params	16/10/2013 09:24	File	4 KB	

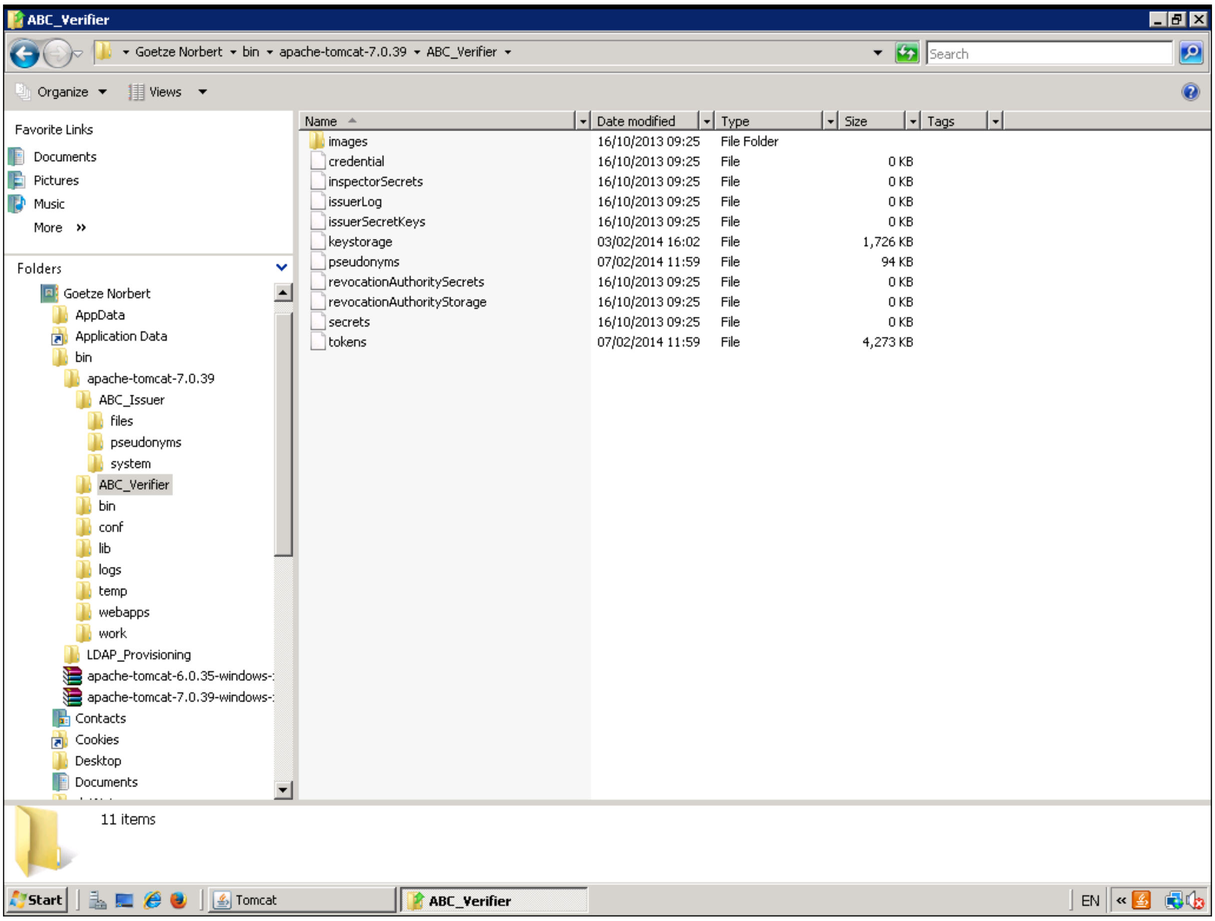
Figure 36: Patras IdM ABC System - Public Storage Area of the Issuer



Name	Date modified	Type	Size	Tags
images	16/10/2013 09:24	File Folder		
credential	16/10/2013 09:24	File	0 KB	
inspectorSecrets	16/10/2013 09:24	File	0 KB	
issuer_private_key_urn_patras_issuer_credCourse_uprove	16/10/2013 09:24	File	2 KB	
issuer_private_key_urn_patras_issuer_credUniv_idemix	16/10/2013 09:24	File	2 KB	
issuerLog	07/02/2014 11:59	File	2,355 KB	
issuerSecretKeys	16/10/2013 09:24	File	4 KB	
keystorage	16/10/2013 09:40	File	17 KB	
pseudonyms	07/02/2014 11:59	File	118 KB	
revocationAuthoritySecrets	16/10/2013 09:24	File	0 KB	
revocationAuthorityStorage	16/10/2013 09:24	File	0 KB	
secrets	16/10/2013 09:24	File	0 KB	
system_params	16/10/2013 09:24	File	4 KB	
system_params_human_readable_only_for_reference.xml	16/10/2013 09:24	XML Document	5 KB	
tokens	07/02/2014 11:59	File	6,205 KB	

Figure 37: Patras IdM ABC System - Private Storage Area of the Issuer

No changes can be noticed when comparing the private storage area of the Verifier of both pilots.



Name	Date modified	Type	Size	Tags
images	16/10/2013 09:25	File Folder		
credential	16/10/2013 09:25	File	0 KB	
inspectorSecrets	16/10/2013 09:25	File	0 KB	
issuerLog	16/10/2013 09:25	File	0 KB	
issuerSecretKeys	16/10/2013 09:25	File	0 KB	
keystorage	03/02/2014 16:02	File	1,726 KB	
pseudonyms	07/02/2014 11:59	File	94 KB	
revocationAuthoritySecrets	16/10/2013 09:25	File	0 KB	
revocationAuthorityStorage	16/10/2013 09:25	File	0 KB	
secrets	16/10/2013 09:25	File	0 KB	
tokens	07/02/2014 11:59	File	4,273 KB	

Figure 38: Patras IdM ABC System - Private Storage Area of the Verifier

3.4.6.3 Deployment

Shifting the ‘IdM ABC System’ from the integration testing environment to the target site involves identical tasks as described in 2.4.6.3.

3.4.6.4 Operation

The ABC System of the University Registration System was hosted on a Tomcat web server on a Windows virtual machine. At the beginning of the pilot, the administrators had to clear the log files and restart the Tomcat hosting the ABC System. As soon as the application was initialized without any exceptions the system was ready for service requests.

During the pilot operation no issues regarding the ABC System of the University Registration System came up. The Users with valid pilot smart cards in their possession were involved successfully in verification protocols (i.e. logging-in the University Registration System using Privacy-ABC technologies) and issuance protocols (obtaining credUniv and credCourse). Moreover, by checking the log files no exceptions were discovered during issuance/verification protocols.

3.5 Revocation Authority

The 2nd round of the Patras pilot is using the revocation service provided in the publicly available ABC4Trust code (see https://abc4trust.eu/download/source/ABCEngine-source.zip). Contrary to the Söderhamn Revocation Authority, the RevAuth used in the Patras pilot is generic, so it can be deployed in other scenarios and other use-cases without modification. Initialization of the RevAuth is performed by sending HTTP messages to the RESTful interface of the application instead of copying data into its directories.

3.5.1 Revocation Application

3.5.1.1 Implementation

The ABC engine (ABCE) used in the Patras pilot is close to identical to the one used in the Söderhamn pilot, described in Section 2.5.1.1 A minor change did occur though, namely the removal of a wrapping XML element in the messages being sent to and from the revocation authority.

While the ABCE did not incur major changes, the reference implementation was expanded to contain a series of generic web services. One of these abce-services is the RevocationService which was enhanced with an 'ABCE REST API'. The 'ABCE REST API' contains a complete set of methods for invoking all revocation services as an alternative to the legacy 'ABCE Java API'.

3.5.1.2 Integration

With the generic web services implemented as part of the reference implementation, there was no need to build a custom revocation authority application. The generic revocation authority service has to be initialized the same way as the custom revocation authority service; however it does not scan for the various resource files. Instead an administrator has to use the HTTP protocol (using curl), to make a series of method calls on the ABCE in order to configure it. These method calls include a method to pass the system parameters to the ABCE and a call that causes a set of revocation authority parameters to be generated.

Once these initial setup calls have been performed, the behavior is identical to that of the revocation authority service described in Section 2.5.1.2. As in that case, the generic web service does not perform any kind of User authentication or authorization, the methods provided for administrators only must be protected by a firewall.

3.5.1.3 Deployment

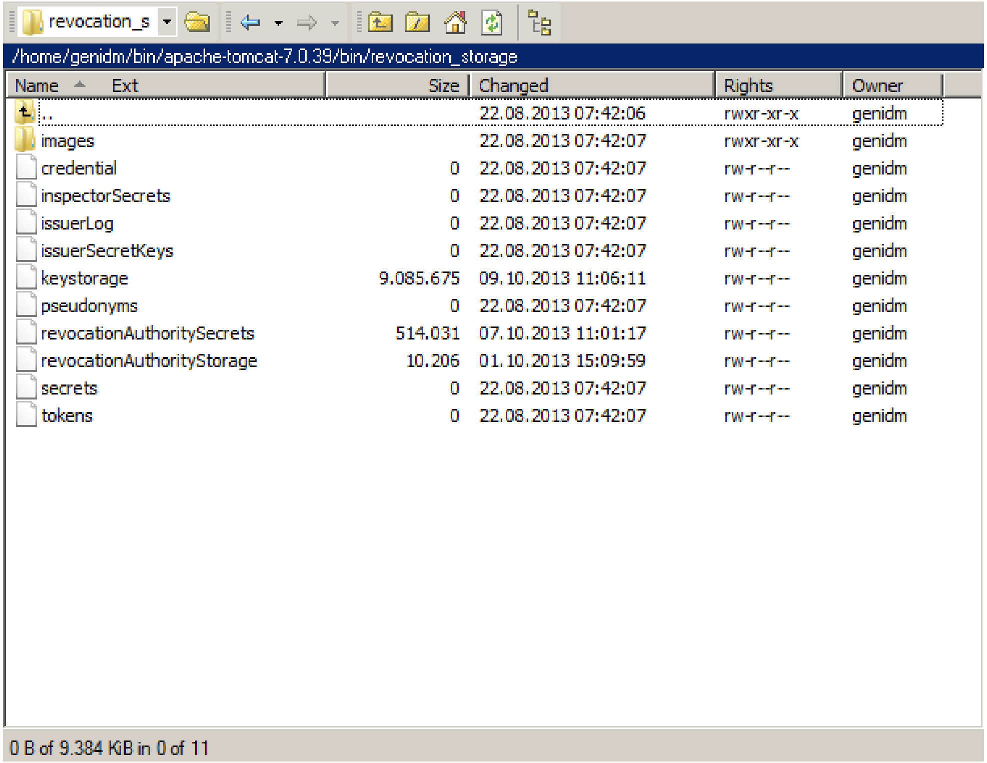
As stated in the previous chapter, the generic Revocation Authority is part of the reference implementation provided by WP4 and publicly available in https://abc4trust.eu/download/source/ABCEngine-source.zip

As preparatory measure, a self-signed X.509 certificate must be generated for the server hosting the RevAuth application. This X.509 certificate must contain a 'common name' matching the FQDN of the server. This server is only dedicated to hosting the RevAuth. No other applications other than the Apache-Tomcat container will be installed.

The RevAuth application is a war file that results in launching the following command in the abce-services directory of the ABCE code:

mvn -P revocation-service install –DskipTests

In the very first deployment, the directory ../bin/revocation_storage will not be available (see Figure 39). This directory will be generated and filled as soon as the application starts up.



/home/genidm/bin/apache-tomcat-7.0.39/bin/revocation_storage

Name	Ext	Size	Changed	Rights	Owner
..			22.08.2013 07:42:06	rwxr-xr-x	genidm
images			22.08.2013 07:42:07	rwxr-xr-x	genidm
credential		0	22.08.2013 07:42:07	rw-r--r--	genidm
inspectorSecrets		0	22.08.2013 07:42:07	rw-r--r--	genidm
issuerLog		0	22.08.2013 07:42:07	rw-r--r--	genidm
issuerSecretKeys		0	22.08.2013 07:42:07	rw-r--r--	genidm
keystorage		9.085.675	09.10.2013 11:06:11	rw-r--r--	genidm
pseudonyms		0	22.08.2013 07:42:07	rw-r--r--	genidm
revocationAuthoritySecrets		514.031	07.10.2013 11:01:17	rw-r--r--	genidm
revocationAuthorityStorage		10.206	01.10.2013 15:09:59	rw-r--r--	genidm
secrets		0	22.08.2013 07:42:07	rw-r--r--	genidm
tokens		0	22.08.2013 07:42:07	rw-r--r--	genidm

0 B of 9.384 KiB in 0 of 11

Figure 39: Generic Revocation Authority Storage Area

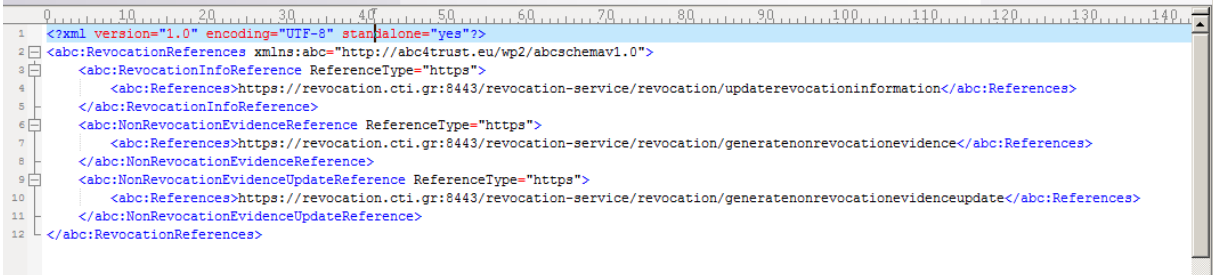
Should this area already be available, the admin must decide if it is necessary to delete it. This is typically the case if the Issuer parameters have changed.

Now the admin must perform the following steps:

1. Start (or restart) the Tomcat container.
2. Copy the ‘human readable’ system parameters generated by the Issuer to the server hosting the RevAuth
3. launch in a command window of the server hosting the RevAuth the following command:
   curl -X POST -d @system_params_human_readable_only_for_reference.xml -H 'Content-Type: application/xml' http://localhost:8080/revocation-service/revocation/storeSystemParameters/
   (This command assumes that, next to port 8443, the container is also listening locally on port 8080)
4. Generate a RevocationReferences.xml file fitting to the location where the RevAuth is installed (see Figure 40)
5. Then launch the next command which will generate not only the RevocationAuthorityParameters.xml file but also the directory <tomcat_location>/bin/revocation_storage:
   curl -X POST -d @RevocationReferences.xml -H 'Content-Type: application/xml' "http://localhost:8081/revocation-

service/revocation/setupRevocationAuthorityParameters?keyLength=1024&uid=urn:patras:revocationauthority:default" -o RevocationAuthorityParameters.xml

6. Copy the revocation authority parameters (in this case revocation_authority_urn_patras_revocationauthority_default) located in <tomcat_location>/bin/revocation_storage to the system folder of the Issuer



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<abc:RevocationReferences xmlns:abc="http://abc4trust.eu/wp2/abcschemav1.0">
    <abc:RevocationInfoReference ReferenceType="https">
        <abc:References>https://revocation.cti.gr:8443/revocation-service/revocation/updaterevocationinformation</abc:References>
    </abc:RevocationInfoReference>
    <abc:NonRevocationEvidenceReference ReferenceType="https">
        <abc:References>https://revocation.cti.gr:8443/revocation-service/revocation/generatenonrevocationevidence</abc:References>
    </abc:NonRevocationEvidenceReference>
    <abc:NonRevocationEvidenceUpdateReference ReferenceType="https">
        <abc:References>https://revocation.cti.gr:8443/revocation-service/revocation/generatenonrevocationevidenceupdate</abc:References>
    </abc:NonRevocationEvidenceUpdateReference>
</abc:RevocationReferences>
```

Figure 40: RevocationReferences.xml

As soon as the revocation authority application was installed on the corresponding server at the Patras site, the pilot administrators started a series of tests in order to verify its correct operation. Using the IdM Admin Tool they revoked a test User's University credential and then they attempted to use it. To verify that the credential was revoked, they browsed the User's credential manager in order to see if her credUniv revocation box contained a tick. Moreover, for extra assurance they tried to log-in at the Course Evaluation System and verify that they were not allowed to do so due to a revoked credUniv.

During the tests performed at CTI's site, one issue was discovered. When a pilot administrator revoked a User's credUniv, her credential was actually revoked successfully but a witness value was not updated correctly and the credential managers of the rest of the system Users were throwing an exception ("Witness Update Failed"), i.e. they could not browse and use their credentials properly. This issue was immediately reported to WP4 which provided a fix for it at the next version available.

3.5.1.4 Operation

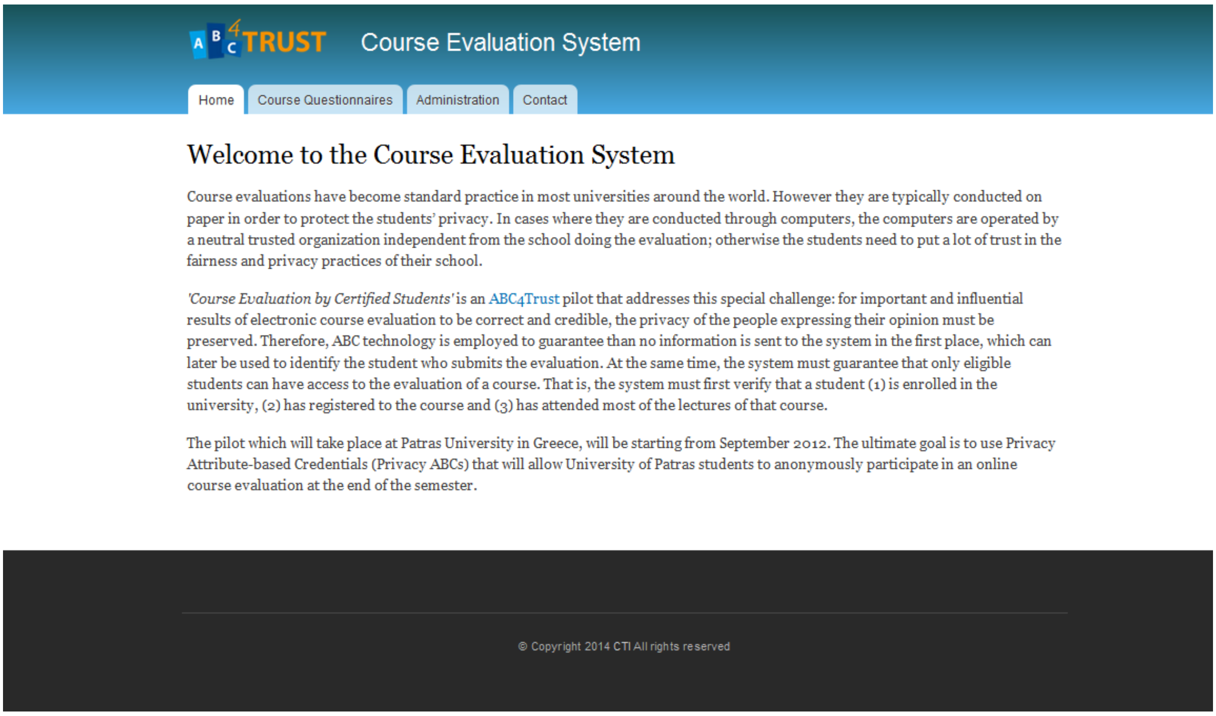
Throughout the Patras pilot the administrators had to ensure that the Revocation Authority was up and running. This task was important since the rest of the systems (University Registration System, Course Evaluation System) needed to be in communication with the Revocation Authority during issuance/verification protocols. Thus, the pilot administrators had to make sure that the Tomcat container hosting the Revocation Authority on the revocation server was running normally during the pilot operation.

No issues regarding the Revocation Authority were discovered, during the Patras pilot operation.

3.6 Course Evaluation System

The Course Evaluation System is responsible for the realization of the course evaluation process. It is the system that University students interact with in order to anonymously evaluate courses that they have registered for and attended to. Moreover, professors can interact with this system in order to upload questionnaires for the courses that they teach and members of the HQAA can access it in order to obtain the evaluation results.

As can be seen in Figure 33, the Course Evaluation System consists of the Course Evaluation Application and the Course Evaluation ABC System. In the following subsections we describe these components in more detail.



Welcome to the Course Evaluation System

Course evaluations have become standard practice in most universities around the world. However they are typically conducted on paper in order to protect the students' privacy. In cases where they are conducted through computers, the computers are operated by a neutral trusted organization independent from the school doing the evaluation; otherwise the students need to put a lot of trust in the fairness and privacy practices of their school.

'Course Evaluation by Certified Students' is an ABC4Trust pilot that addresses this special challenge: for important and influential results of electronic course evaluation to be correct and credible, the privacy of the people expressing their opinion must be preserved. Therefore, ABC technology is employed to guarantee than no information is sent to the system in the first place, which can later be used to identify the student who submits the evaluation. At the same time, the system must guarantee that only eligible students can have access to the evaluation of a course. That is, the system must first verify that a student (1) is enrolled in the university, (2) has registered to the course and (3) has attended most of the lectures of that course.

The pilot which will take place at Patras University in Greece, will be starting from September 2012. The ultimate goal is to use Privacy Attribute-based Credentials (Privacy ABCs) that will allow University of Patras students to anonymously participate in an online course evaluation at the end of the semester.

© Copyright 2014 CTI All rights reserved

Figure 41: Course Evaluation System GUI

3.6.1 Course Evaluation Application

The Course Evaluation Application is a web application (see Figure 41) that provides the user interface for the course evaluation process, i.e. an interface for professors that can upload questionnaires for the courses that they teach, an interface for students that can submit their evaluations for the courses that they attend and HQAA members that can browse the evaluation results. The Course Evaluation Application (using the ABC System) performs access control to the course questionnaires so that only certified students can access them. Finally, the Course Evaluation Application is responsible for storing the student submissions as well as the evaluation results on a database.

3.6.1.1 Implementation

The Course Evaluation Application has been developed with Drupal version 7. Drupal is a free and open source Content Management System (CMS) written in PHP and distributed under the GNU General Public License. Drupal can run on any computing platform that supports both a web server capable of running PHP (e.g. Apache) and a database (such as MySQL or PostgreSQL) that can store content and settings. Drupal 7 requires PHP 5.2.5 or higher. Our development decisions for this pilot were that the Course Evaluation Application would run on an Apache web server and that the content would be stored on a MySQL database.

At the early stages of the ABC4Trust project, the pilot developers implemented the Course Evaluation Application as a stand-alone application, i.e. without Privacy-ABC technologies. At that point, they implemented the main functionalities that were required from the Course Evaluation System i.e. the user interfaces, the evaluation submission procedure, the process of uploading a course questionnaire, the calculation of the evaluation results etc. Then, some testing was performed in order to verify that the Course Evaluation Application was working according to the requirements.

As soon as the first version of the ABC4Trust Reference Implementation was available, they started with the integration of Privacy ABC technologies into the Course Evaluation Application (see Section 3.6.2.2).

3.6.1.2 Integration

The integration of Privacy ABC technologies to the Course Evaluation Application was relatively easy. The pilot developers had to introduce the Course Evaluation ABC System (CES ABC System for short) hosting the necessary Privacy-ABC issuance/verifier services (will be described later on Section 3.6.2) and then they had to enhance the Issuer/Verifier web pages of the Course Evaluation Application with some JavaScript code. This piece of code was triggered upon some page event (e.g. a User's button click) and started a communication flow between the Course Evaluation ABC System and the User Application through the Firefox browser plugin. Depending on the type of the web page (Issuer or Verifier) the corresponding ABCE services were called. At the end of this communication flow and whether the issuance or verification protocol was completed successfully or not, the User was re-directed to the appropriate web page e.g. if verification was successful she was re-directed to the course questionnaire web page where she could state her opinion about the course.

3.6.1.3 Deployment

In the beginning of the ABC4Trust project, the pilot developers implemented the Course Evaluation Application as a stand-alone application i.e. without any Privacy-ABC technologies. This application was hosted on the developer's PC in order to easily make changes if required. At that point, they performed some testing in order to verify that the application was working as expected. First of all, they tested that the web pages and the user interface were displayed correctly. Moreover, they created some "test" Users that could log-in to the system via username/password and submit their evaluation for a course. As a result, they could check that the evaluation procedure could run smoothly and that the student evaluations were stored successfully at the database. Additionally, some "professor" accounts were created in order to test if a professor could upload and modify without problems a questionnaire for his course. Finally, the application developers created some "HQAA" accounts that could access the evaluation results and this way it was possible to verify that the results were produced without any errors. Having performed these initial tests successfully, the developers were ready to start with the integration of Privacy-ABC technologies in the Course Evaluation System.

3.6.1.4 Operation

Before the students could access the Course Evaluation Application, the pilot administrators should make sure that the Apache server which hosted the application was up and running. Moreover, they should set the status of the evaluation questionnaire to “active” so that certified students can access it and make sure that the database was clean from previous evaluations (possibly from the testing phase).

During the pilot operation the administrators did not face any problems with the Course Evaluation Application, since it had been extensively tested. The fact that some students could not log in to the Course Evaluation System was rather related to a problem of another entity of the overall system (e.g. User’s smart card) instead of the Course Evaluation Application itself.

3.6.2 CES ABC System – Issuer and Verifier

For the 2nd round of the Patras pilot, the Course Evaluation System was playing the ABC roles of Verifier and Issuer. As a Verifier, it allowed access to the course evaluation questionnaires to certified students only i.e. students who possessed the necessary credentials and had attended at the majority of the course lectures. As an Issuer, the Course Evaluation System allowed the students who had submitted their evaluation for the course, to obtain a credential (called the Tombola Credential) that would enable them to participate in an anonymous online lottery.

3.6.2.1 Implementation

The Course Evaluation ABC System has been implemented as Java REST web services that are exposed to the application layer through HTTPS. The implementation of these web services was based on the example projects that were provided by the ABC4Trust reference implementation. These example projects were using some helpers that enabled a developer to deploy the necessary services of a Verifier and an Issuer. The pilot developers modified these examples according to their application needs and enhanced them with some checks that would increase their application's security e.g. they did not allow a student who had not submitted her course evaluation questionnaire to access the Tombola Issuer web services. The compiled application (.war file) containing the Issuer/Verifier services, was hosted on Jetty web server version 7.0.1. Jetty is a pure Java based HTTP server and has been developed as a free and open source project.

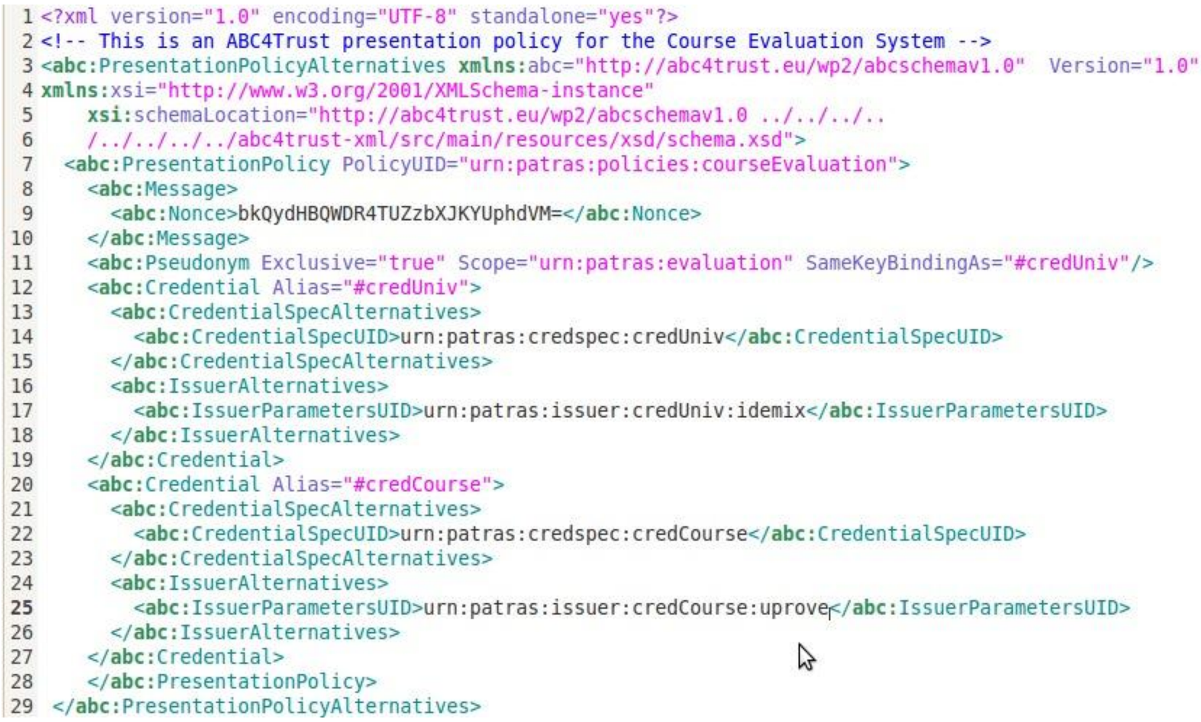
The Course Evaluation ABC System offers the following services:

- getPolicy
- verifyToken
- start*Tombola*
- issueStep

The first two services (getPolicy and verifyToken) were called when a student wanted to log-in to the Course Evaluation System and submit her evaluation for a course. The next two services were called when a student had submitted her evaluation for a course and desired to obtain the Tombola Credential that would allow her to participate in the online lottery. The start*Tombola* service takes as a query parameter the scope exclusive pseudonym of the user that is logged-in at the Course Evaluation System for that session and embeds it in the issuance policy.

3.6.2.2 Integration

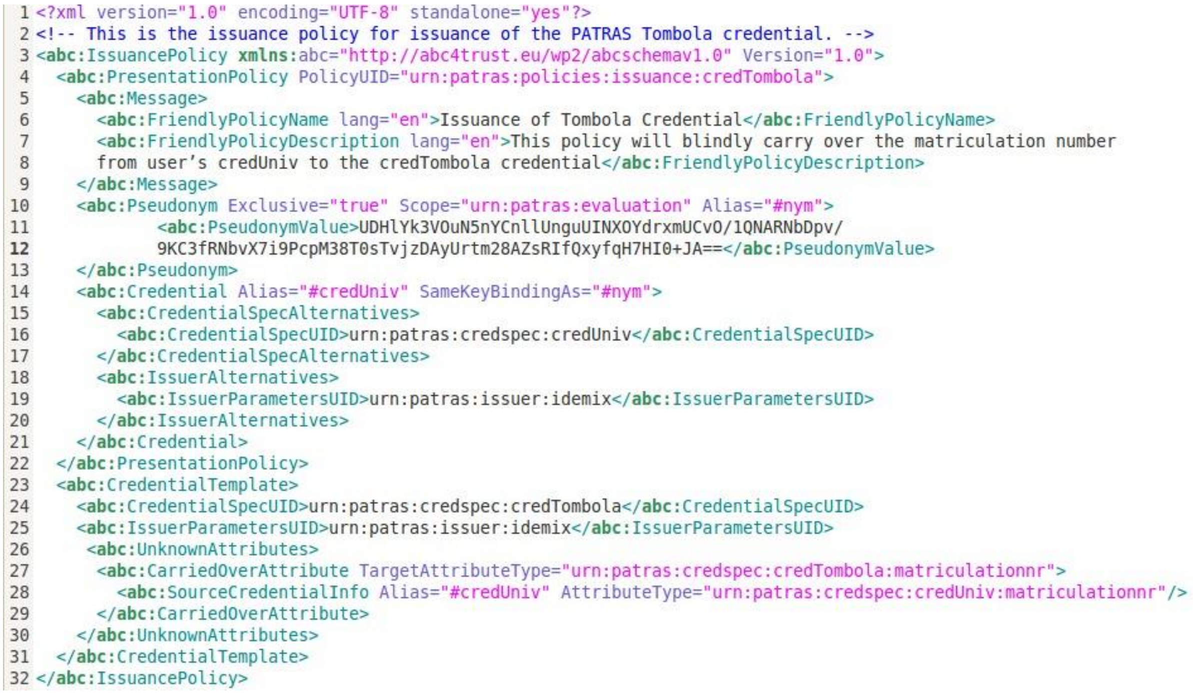
The Course Evaluation ABC System (CES ABC System for short), when acting as a Verifier, starts communicating to the Users who tried to access it by sending a presentation policy. This policy stated that they should have in their possession a valid (i.e. non-revoked) University Credential and a Course Credential (see Figure 42). This policy also requires the User to present a scope exclusive pseudonym based on the scope "urn:patras:evaluation". When the CES ABC System received a presentation token for the User side, it was trying to verify it against its presentation policy. When a token was verified successfully the CES ABC System was extracting the scope exclusive pseudonym from the token and through HTTPS POST requests it was informing the Course Evaluation Application which would log-in the User with that pseudonym and re-direct her to the evaluation web page.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- This is an ABC4Trust presentation policy for the Course Evaluation System -->
<abc:PresentationPolicyAlternatives xmlns:abc="http://abc4trust.eu/wp2/abcschemav1.0"  Version="1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://abc4trust.eu/wp2/abcschemav1.0 ../../../..
    /../../../../abc4trust-xml/src/main/resources/xsd/schema.xsd">
  <abc:PresentationPolicy PolicyUID="urn:patras:policies:courseEvaluation">
    <abc:Message>
      <abc:Nonce>bkQydHBQWDR4TUZzbXJKYUphdVM=</abc:Nonce>
    </abc:Message>
    <abc:Pseudonym Exclusive="true" Scope="urn:patras:evaluation" SameKeyBindingAs="#credUniv"/>
    <abc:Credential Alias="#credUniv">
      <abc:CredentialSpecAlternatives>
        <abc:CredentialSpecUID>urn:patras:credspec:credUniv</abc:CredentialSpecUID>
      </abc:CredentialSpecAlternatives>
      <abc:IssuerAlternatives>
        <abc:IssuerParametersUID>urn:patras:issuer:credUniv:idemix</abc:IssuerParametersUID>
      </abc:IssuerAlternatives>
    </abc:Credential>
    <abc:Credential Alias="#credCourse">
      <abc:CredentialSpecAlternatives>
        <abc:CredentialSpecUID>urn:patras:credspec:credCourse</abc:CredentialSpecUID>
      </abc:CredentialSpecAlternatives>
      <abc:IssuerAlternatives>
        <abc:IssuerParametersUID>urn:patras:issuer:credCourse:uprove</abc:IssuerParametersUID>
      </abc:IssuerAlternatives>
    </abc:Credential>
    </abc:PresentationPolicy>
 </abc:PresentationPolicyAlternatives>
```

Figure 42: The Presentation Policy of the Course Evaluation ABC System (Verifier)

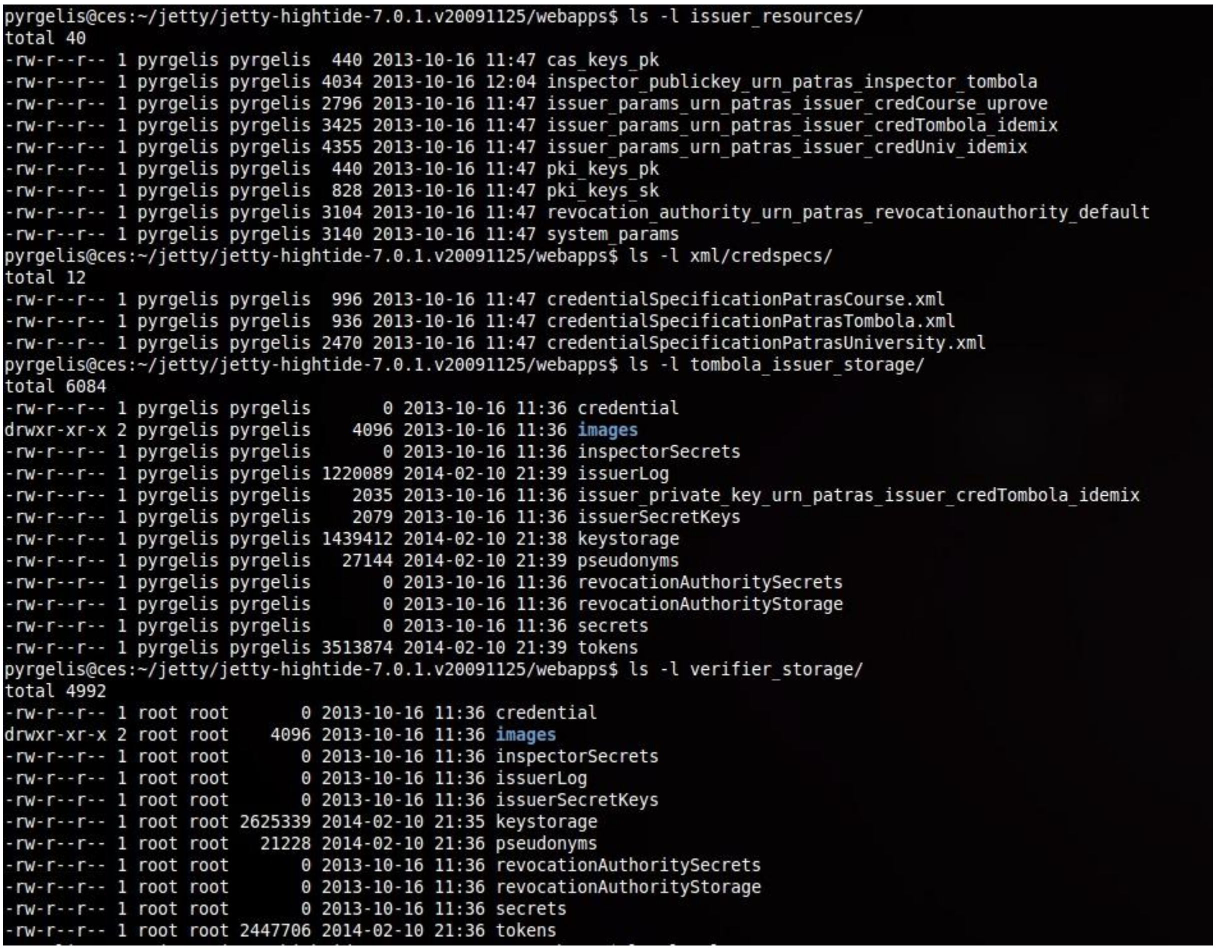
The Course Evaluation ABC System when acting as an Issuer, starts communicating to the Users who will be granted access to the Tombola web page (i.e. after submitting the evaluation for the course) by sending an issuance policy stating that the User should prove possession of the scope exclusive pseudonym that she was logged-in with to the Course Evaluation Application and that she should possess a University Credential from which the matriculation number would be carried over blindly to the newly issued Tombola Credential (see Figure 43). When the service that would start the issuance protocol was called, the CES ABC System would make a HTTPS POST request to the Course Evaluation Application in order to verify that the user that is about to be issued the Tombola Credential has submitted the course evaluation. If this check was successful the issuance protocol would continue normally. It was possible for a student to obtain a Tombola credential multiple times but he could not gain anything out of it (e.g. registering multiple times for the lottery) since the Tombola System performs consumption control based on a scope exclusive pseudonym for the scope “urn:patras:tombola” (see Section 3.7).



```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- This is the issuance policy for issuance of the PATRAS Tombola credential. -->
<abc:IssuancePolicy xmlns:abc="http://abc4trust.eu/wp2/abcschemav1.0" Version="1.0">
  <abc:PresentationPolicy PolicyUID="urn:patras:policies:issuance:credTombola">
    <abc:Message>
      <abc:FriendlyPolicyName lang="en">Issuance of Tombola Credential</abc:FriendlyPolicyName>
      <abc:FriendlyPolicyDescription lang="en">This policy will blindly carry over the matriculation number
      from user’s credUniv to the credTombola credential</abc:FriendlyPolicyDescription>
    </abc:Message>
    <abc:Pseudonym Exclusive="true" Scope="urn:patras:evaluation" Alias="#nym">
            <abc:PseudonymValue>UDHlYk3VOuN5nYCnllUnguUINXOYdrxmUCvO/1QNARNbDpv/
            9KC3fRNbvX7i9PcpM38T0sTvjzDAyUrtm28AZsRIfQxyfqH7HI0+JA==</abc:PseudonymValue>
    </abc:Pseudonym>
    <abc:Credential Alias="#credUniv" SameKeyBindingAs="#nym">
      <abc:CredentialSpecAlternatives>
        <abc:CredentialSpecUID>urn:patras:credspec:credUniv</abc:CredentialSpecUID>
      </abc:CredentialSpecAlternatives>
      <abc:IssuerAlternatives>
        <abc:IssuerParametersUID>urn:patras:issuer:idemix</abc:IssuerParametersUID>
      </abc:IssuerAlternatives>
    </abc:Credential>
  </abc:PresentationPolicy>
  <abc:CredentialTemplate>
    <abc:CredentialSpecUID>urn:patras:credspec:credTombola</abc:CredentialSpecUID>
    <abc:IssuerParametersUID>urn:patras:issuer:idemix</abc:IssuerParametersUID>
     <abc:UnknownAttributes>
      <abc:CarriedOverAttribute TargetAttributeType="urn:patras:credspec:credTombola:matriculationnr">
        <abc:SourceCredentialInfo Alias="#credUniv" AttributeType="urn:patras:credspec:credUniv:matriculationnr"/>
      </abc:CarriedOverAttribute>
    </abc:UnknownAttributes>
  </abc:CredentialTemplate>
</abc:IssuancePolicy>
```

Figure 43: Issuance Policy for the Tombola Credential

In order for an issuance/verification protocol to complete without exceptions on the ABCE layer, the pilot administrators had to make sure that both communication ends (User and Issuer/Verifier) were using the same resources i.e. system parameters, Issuer parameters, credential specifications, revocation authority parameters and inspector public key. Thus, a crucial administrator’s task was to create the User installer bundles containing the correct resource files and moreover upload the same files to the Course Evaluation ABCE storage space. The public system, issuer and revocation authority parameters were provided by the administrators of the University Registration System (NSN) to the administrators of the pilot (CTI). The issuer parameters of the Tombola Credential were generated by the administrators of the Course Evaluation System (using the public system parameters).



```
pyrgelis@ces:~/jetty/jetty-hightide-7.0.1.v20091125/webapps$ ls -l issuer_resources/
total 40
-rw-r--r-- 1 pyrgelis pyrgelis  440 2013-10-16 11:47 cas_keys_pk
-rw-r--r-- 1 pyrgelis pyrgelis 4034 2013-10-16 12:04 inspector_publickey_urn_patras_inspector_tombola
-rw-r--r-- 1 pyrgelis pyrgelis 2796 2013-10-16 11:47 issuer_params_urn_patras_issuer_credCourse_uprove
-rw-r--r-- 1 pyrgelis pyrgelis 3425 2013-10-16 11:47 issuer_params_urn_patras_issuer_credTombola_idemix
-rw-r--r-- 1 pyrgelis pyrgelis 4355 2013-10-16 11:47 issuer_params_urn_patras_issuer_credUniv_idemix
-rw-r--r-- 1 pyrgelis pyrgelis  440 2013-10-16 11:47 pki_keys_pk
-rw-r--r-- 1 pyrgelis pyrgelis  828 2013-10-16 11:47 pki_keys_sk
-rw-r--r-- 1 pyrgelis pyrgelis 3104 2013-10-16 11:47 revocation_authority_urn_patras_revocationauthority_default
-rw-r--r-- 1 pyrgelis pyrgelis 3140 2013-10-16 11:47 system_params
pyrgelis@ces:~/jetty/jetty-hightide-7.0.1.v20091125/webapps$ ls -l xml/credspecs/
total 12
-rw-r--r-- 1 pyrgelis pyrgelis  996 2013-10-16 11:47 credentialSpecificationPatrasCourse.xml
-rw-r--r-- 1 pyrgelis pyrgelis  936 2013-10-16 11:47 credentialSpecificationPatrasTombola.xml
-rw-r--r-- 1 pyrgelis pyrgelis 2470 2013-10-16 11:47 credentialSpecificationPatrasUniversity.xml
pyrgelis@ces:~/jetty/jetty-hightide-7.0.1.v20091125/webapps$ ls -l tombola_issuer_storage/
total 6084
-rw-r--r-- 1 pyrgelis pyrgelis       0 2013-10-16 11:36 credential
drwxr-xr-x 2 pyrgelis pyrgelis    4096 2013-10-16 11:36 images
-rw-r--r-- 1 pyrgelis pyrgelis       0 2013-10-16 11:36 inspectorSecrets
-rw-r--r-- 1 pyrgelis pyrgelis 1220089 2014-02-10 21:39 issuerLog
-rw-r--r-- 1 pyrgelis pyrgelis    2035 2013-10-16 11:36 issuer_private_key_urn_patras_issuer_credTombola_idemix
-rw-r--r-- 1 pyrgelis pyrgelis    2079 2013-10-16 11:36 issuerSecretKeys
-rw-r--r-- 1 pyrgelis pyrgelis 1439412 2014-02-10 21:38 keystorage
-rw-r--r-- 1 pyrgelis pyrgelis   27144 2014-02-10 21:39 pseudonyms
-rw-r--r-- 1 pyrgelis pyrgelis       0 2013-10-16 11:36 revocationAuthoritySecrets
-rw-r--r-- 1 pyrgelis pyrgelis       0 2013-10-16 11:36 revocationAuthorityStorage
-rw-r--r-- 1 pyrgelis pyrgelis       0 2013-10-16 11:36 secrets
-rw-r--r-- 1 pyrgelis pyrgelis 3513874 2014-02-10 21:39 tokens
pyrgelis@ces:~/jetty/jetty-hightide-7.0.1.v20091125/webapps$ ls -l verifier_storage/
total 4992
-rw-r--r-- 1 root root       0 2013-10-16 11:36 credential
drwxr-xr-x 2 root root    4096 2013-10-16 11:36 images
-rw-r--r-- 1 root root       0 2013-10-16 11:36 inspectorSecrets
-rw-r--r-- 1 root root       0 2013-10-16 11:36 issuerLog
-rw-r--r-- 1 root root       0 2013-10-16 11:36 issuerSecretKeys
-rw-r--r-- 1 root root 2625339 2014-02-10 21:35 keystorage
-rw-r--r-- 1 root root   21228 2014-02-10 21:36 pseudonyms
-rw-r--r-- 1 root root       0 2013-10-16 11:36 revocationAuthoritySecrets
-rw-r--r-- 1 root root       0 2013-10-16 11:36 revocationAuthorityStorage
-rw-r--r-- 1 root root       0 2013-10-16 11:36 secrets
-rw-r--r-- 1 root root 2447706 2014-02-10 21:36 tokens
```

Figure 44: Course Evaluation ABC System Resources

As can be seen in Figure 44, the resources required for the Verifier of the Course Evaluation ABC System are the system parameters, the issuer resources, the revocation authority parameters and the credential specifications which were provided by the administrators of the University Registration System (NSN). In the “issuer_storage” folder there are the necessary resources for the Course Evaluation ABC System Issuer (Issuer private key) for issuing the Tombola credential. These resources were generated by the Course Evaluation System administrators (CTI).

3.6.2.2.1 Deployed ABCE Methods

In this subsection we describe with small code samples the ABCE methods that were used for the deployment of the Course Evaluation ABC System.

Please also refer to Section 2.4.6.2.1 for the reasoning behind listing these methods.

- eu.abc4trust.ri.servicehelper.verifier.VerificationHelper

  VerificationHelper. **initInstance**(String systemParamsResource, String[] issuerParamsResourceList, String[] credSpecResourceList, String[] inspectorPublicKeyResourceList, String[]

revocationAuthorityParametersResourceList, String fileStoragePrefix, String presentationPolicyResourceList)

This method initializes the instance of the Verifier with the appropriate resources and presentation policies.

byte[] nonce = VerificationHelper.getInstance().**generateNonce**();

This method is used for generating a nonce that will be embedded in the presentation policy.

PresentationPolicyAlternatives ppa =VerificationHelper.getInstance().**createPresentationPolicy**(policyId, nonce, applicationData, null);

Generating the presentation policy before communicating it back to the User side.

boolean ok = VerificationHelper.getInstance().**verifyToken**(policyId, nonce, applicationData, presentationToken);

This method is used for verifying a presentation token against the presentation policy and the nonce value.

- eu.abc4trust.xml.PresentationToken;

byte[] pse_bytes = presentationToken.getPresentationTokenDescription().getPseudonym().get(0).**getPseudonymValue**();

This method is used for extracting the pseudonym value out of a received presentation token.

- eu.abc4trust.ri.servicehelper.issuer.SpecAndPolicy

```
public static final SpecAndPolicy tombola = new SpecAndPolicy(
    CREDSPEC_TOMBOLA, // logical key
    CryptoTechnology.IDEMIX, // Crypto Technology
    null, // IssuerParameter UID : null will default to issuerparam uid from
            issuancepolicy->credentialtemplate and append cryptotechnology...
    2, // max number of attributes
    0, // number of U-Prove tokens...
    "src/main/resources/xml/credspecs/credentialSpecificationPatrasTombola.xml",
    "src/main/resources/xml/issuance/issuancePolicyPatrasTombola.xml",
    null, // Revocation Authority UID
    "en", "Description of Tombola Credential");
```

This way the credential specification and the issuance policy of the Tombola Credentia are defined. The Tombola Credential is of type Idemix.

- eu.abc4trust.ri.servicehelper.issuer.IssuanceHelper
  eu.abc4trust.xml.IssuanceMessage
  eu.abc4trust.xml.IssuanceMessageAndBoolean
  eu.abc4trust.xml.ObjectFactory;

IssuanceHelper.**initInstanceWithExistingSystemParameters**(String systemParametersResource, String[] foreignCredSpecResourceList, String[] foreignIssuerParamResourceList, String publicResources, String fileStoragePrefix, SpecAndPolicy[] specAndPolicyList, String[] revocationAuthorityParametersResourcesList)

With this method the Tombola Issuer Parameters are generated, using the existing system parameters

IssuanceMessage im_with_policy =

IssuanceHelper.getInstance().**initIssuance**(newInstanceOfSpecAndPolicy, attributeValueMap);

The issuance policy template is loaded and modified.

ObjectFactory of = new ObjectFactory();

JAXBElement<IssuanceMessage> imJAXB = of.**createIssuanceMessage**(im_with_policy);

Using this method an issuance message is create and sent to the User side.

IssuanceMessageAndBoolean response= IssuanceHelper.getInstance().**issueStep**(issuanceMessage);

Using this method, it is determined if the last issuance message should be communicated back to the User side otherwise the issuance protocol continues.

3.6.2.3 Deployment

Deploying the Course Evaluation ABCE System was done by loading the compiled application (.war file) to the Jetty server and restarting it. During the integration stage the Issuer/Verifier services were hosted locally at the development PC in order to check their operation and make easily any necessary changes. Later on, these services were hosted on the actual Course Evaluation server and the pilot testers started testing the Course Evaluation System in whole (i.e. Course Evaluation Application and CES ABC System).

In order to check the functionalities of the Course Evaluation ABCE System, the pilot administrators had to perform some tests with Users having software or hardware smart cards when interacting with the Course Evaluation System.

In order to be able to log-in to the Course Evaluation System, a User should have sufficient attendance units stored on her smart card. Moreover, the presentation policy of the Course Evaluation System required from the Users to have in their possession a valid (i.e. non-revoked) University Credential as well as a Course Credential. Using some test Users who had both credentials was therefore necessary in order to verify that access to the questionnaire was allowed for these Users. Additionally, some tests were performed with Users that did not have any credentials or had only one out of the two required credentials. This way, the pilot testers verified that the Course Evaluation ABCE System did not allow access to Users who could not fulfill the requirements of the presentation policy. Finally, they did some tests with Users whose University Credential had been revoked by a pilot administrator, in order to be sure that revocation was working as expected and that the CES ABC System was picking up the latest revocation information when attempting to verify a presentation token.

Testing the Issuer functionality of the Course Evaluation System was done with Users that had stored in their smart cards the necessary credentials and had submitted their evaluation for the course. After submitting their evaluation for the course they were re-directed to a web page enabling them to apply for the Tombola Credential. This way, it was verified that the issuance process was completed without any problems. Finally, the special case where a User who has submitted the evaluation for the course changes the smart card on her smart card reader with a different one when she is re-directed at the Tombola web page was tested. At this case, the Course Evaluation ABCE System should not allow the issuance protocol to take place as it requires from the User to present the scope exclusive pseudonym that the she is logged in to the Course Evaluation System with. This fact is stated in the issuance policy of the Tombola Credential. Thus, it was verified that cheating Users would not be able to obtain a Tombola Credential and participate in the lottery without having submitted the evaluation for the course.

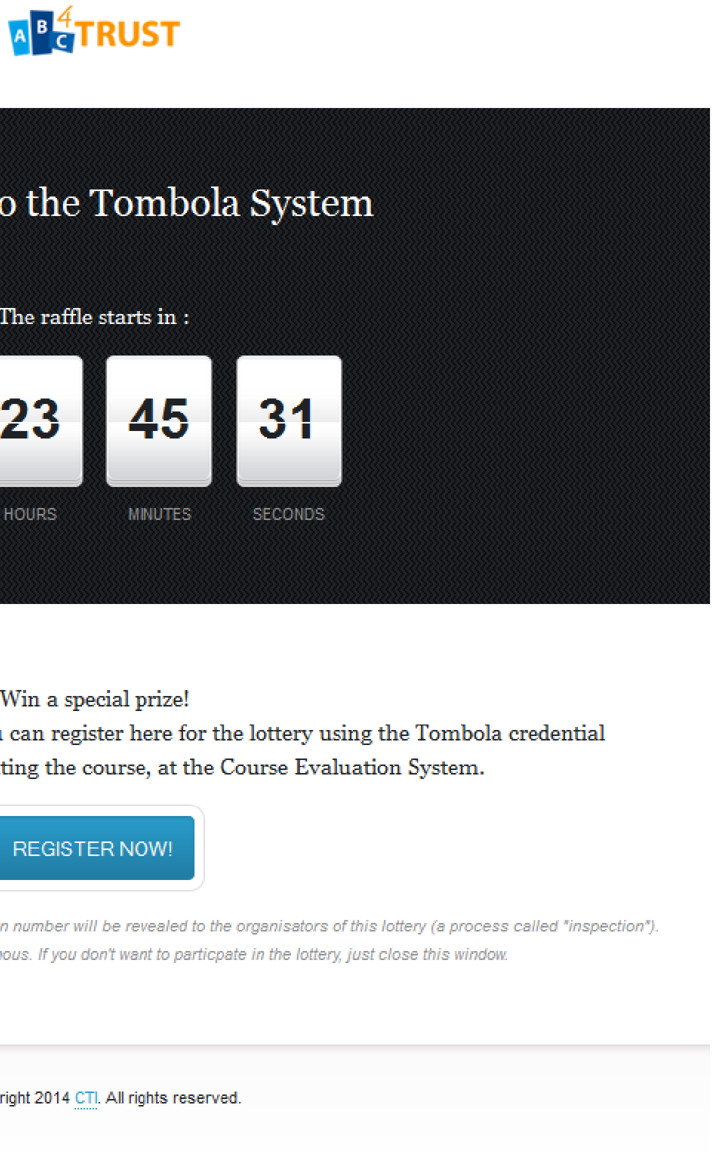
3.6.2.4 Operation

During the pilot operation, the administrators had to make sure that the Jetty server hosting the Course Evaluation ABC system was up and running. Moreover, they should ensure that the appropriate resources (system parameters, Issuer resources etc.) were in place so that the Privacy-ABC protocols would run smoothly.

Some students faced difficulties logging-in the Course Evaluation System but the reasons for this were related to the User smart card rather than to the Course Evaluation System (see Section 3.2.2.4). Moreover, some students had trouble obtaining the Tombola credential from the Course Evaluation System but that was due to insufficient storage on the card (see Section 3.2.2.4). Other than these, no issues came up regarding the operation of the Course Evaluation ABC System, during the pilot.

3.7 Tombola System

The Tombola System is responsible for conducting an online raffle for the students that participate in the Patras pilot. The prize for the winner of the raffle is a free registration for the ABC4Trust Summer School that will take place in Patras, on September of 2014. The students can interact with this system and register for the lottery, after they submit their evaluation for the pilot course and obtain the Tombola Credential from the Course Evaluation System. As seen on Figure 33, the Tombola System consists of the Tombola Application and the Tombola ABC System.



Welcome to the Tombola System

The raffle starts in :

Win a special prize!
If you are a student of course 23330E, you can register here for the lottery using the Tombola credential that you obtained after evaluating the course, at the Course Evaluation System.

After the winner of the Tombola is drawn, his or her matriculation number will be revealed to the organisators of this lottery (a process called "inspection"). All other participants will remain pseudonymous. If you don't want to particpate in the lottery, just close this window.

© Copyright 2014 CTI. All rights reserved.

Figure 45: Tombola System GUI

3.7.1 Tombola Application

The Tombola System was introduced in the 2nd round of the Patras pilot so that the students can participate anonymously in an online raffle. The Tombola Application is a web application (see Figure 45) that the students can interact with and register for the lottery using Privacy-ABC technologies.

3.7.1.1 Implementation

The Tombola Application was developed using the server side scripting language PHP. Moreover, a MySQL database was designed and installed according to the application needs. These were, to store the lottery details (lottery status, lottery expiration time, winner matriculation number) as well as the presentation tokens of the Users who had registered for the lottery. Additionally, a MySQL script was developed in order to set an event (lottery expiration date) that would pick randomly the winning token out of the database. The web server that was chosen to host the Tombola Application was the Apache HTTP server.

Initially, the Tombola Application was developed as a standalone web application, without any Privacy-ABC technologies. At that point, the application database was designed according to the requirements defined and the web pages. Along with that, the User interfaces were created. A first version of the registration functionality was implemented– Users could enter the matriculation number in a field and that was registered in the database. Finally, the “lottery” functionality where the application chooses randomly a matriculation number which is announced as the winner of the raffle was developed.

As soon as the ABC4Trust Reference Implementation was available including the newly created cryptographic architecture, the integration of Privacy-ABC technologies to the Tombola System started (see Section 3.7.2.2).

3.7.1.2 Integration

The procedure of integrating of Privacy ABC technologies to the Tombola System was similar to the one which was followed for the Course Evaluation System. The pilot developers introduced the Tombola ABC System and enhanced the Verifier web page of the Tombola Application with some JavaScript code. This piece of code was triggered by the User clicking on the registration button and started a communication flow between the Tombola ABC System and the User Application through the Firefox browser plugin. At the end of this communication flow (independent of whether the verification protocol was completed successfully or not), the User was informed whether he was registered for the raffle or not.

When the first version of the Tombola Application was ready, it was deployed locally in order to check its operation and make any changes required. After that, some tests were performed in order to verify that the application was working properly. Using some dummy Users, who registered their matriculation numbers using the interface of the Tombola main web page it was verified that they were stored correctly to the Tombola database. Moreover, the “lottery” functionality was tested, in order to check that the application was picking a random matriculation number among the registered ones as winner of the raffle, at the time of the lottery expiration.

After the above tests were completed successfully, the developers were in position to start integrating Privacy-ABC technologies into the Tombola System.

3.7.1.3 Deployment

Before allowing the pilot participants to register for the lottery, the administrators should perform some tasks for the operation of the Tombola System. First of all, they should make sure that the Apache web server that hosted the Tombola Application was up and running at the tombola server. Moreover, they should clean up the database from any registered presentation tokens that were stored during the testing phase. Finally, they should configure the expiration date of the lottery and deploy the mySQL script that would pick up the winner of the lottery at the appropriate date.

3.7.1.4 Operation

During the pilot operation, there were no issues reported by the participants regarding the Tombola Application. All the students who had successfully obtained their Tombola Credential from the Course Evaluation System could register for the online lottery.

3.7.2 Tombola ABC System – Verifier

For the 2nd round of the Patras pilot, the Tombola System was playing the ABC role of a Verifier. More specifically, it allowed only certified students to register for the online lottery i.e. students who had in their possession a Tombola Credential issued by the Course Evaluation System.

3.7.2.1 Implementation

Similarly to the Course Evaluation ABC System, the Tombola ABC System has been implemented as Java REST web services that are exposed to the application layer through HTTPS. The implementation of these web services was based on the example projects that were provided by the ABC4Trust reference implementation. These example projects contained some helpers that utilized the ABC4Trust API for the deployment of the necessary services of a Verifier. The pilot developers modified these examples according to their application needs. The compiled application (.war file) containing the Verifier services, was hosted on Jetty web server version 7.0.1. Jetty is a pure Java based HTTP server and has been developed as a free and open source project.

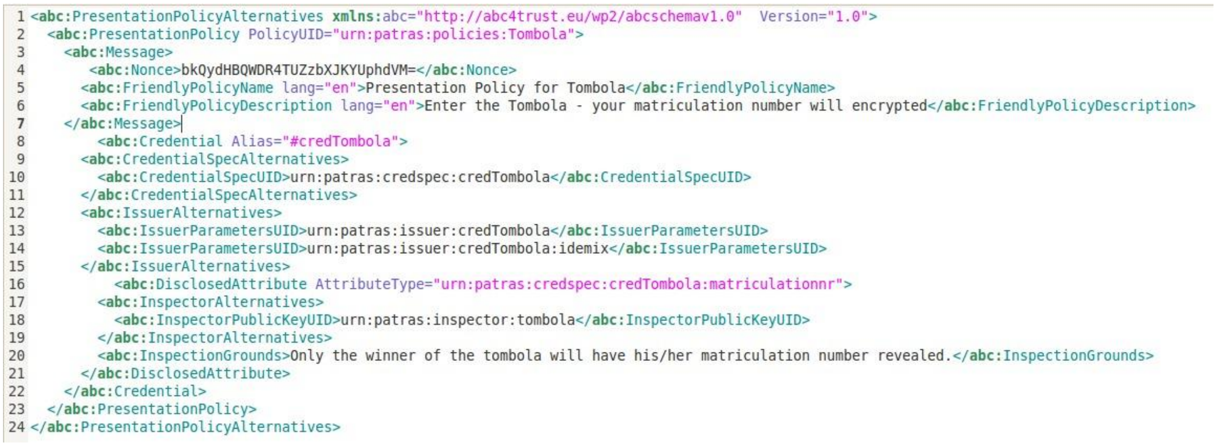
The Tombola ABC System offers the following services:

- getPolicy
- verifyToken

These services (getPolicy and verifyToken) were called when a student wanted to register a presentation token to the Tombola System and participate in the lottery.

3.7.2.2 Integration

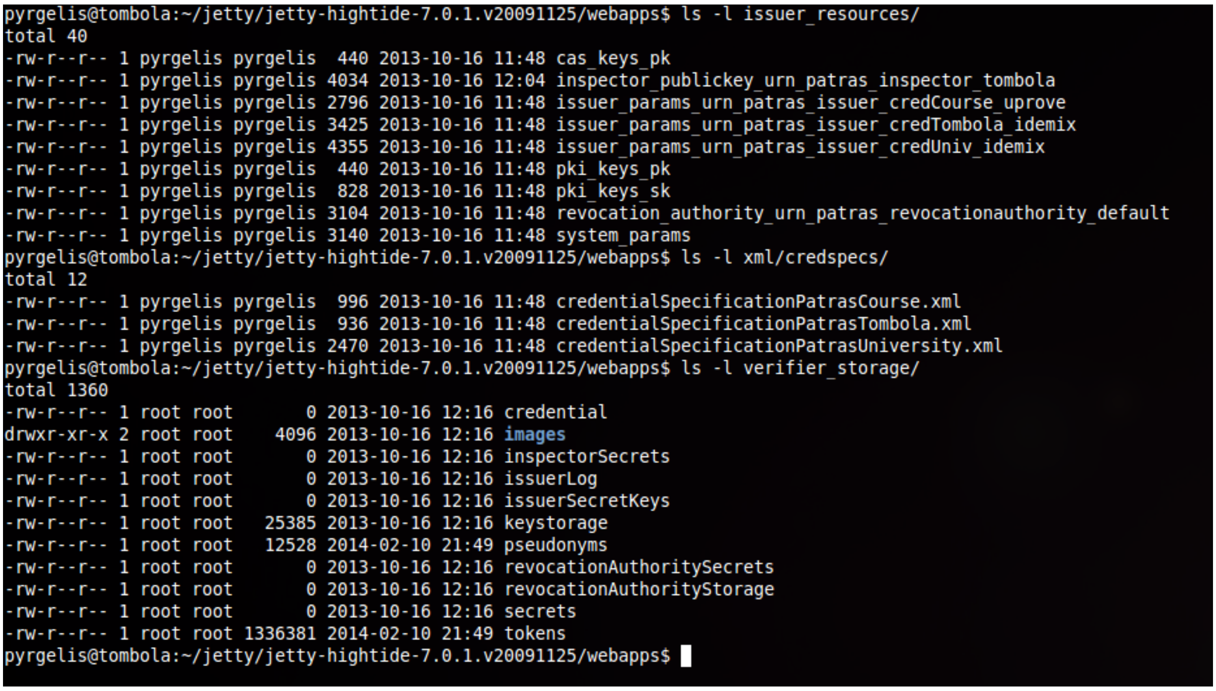
When a student was trying to register for the online lottery, the Tombola ABC System sent a presentation policy to the User side. This policy stated that the User should present a scope exclusive pseudonym for scope “urn:patras:tombola” and have in her possession a Tombola Credential issued by the Course Evaluation System. Moreover, the User should present the matriculation number from the Tombola Credential as an inspectable attribute i.e. encrypted with the Inspector’s public key (see Figure 46). When the Course Evaluation ABC System was receiving a token from the User side and it verified it, it was making an HTTPS POST request to the Tombola Application in order to store the student pseudonym and the corresponding presentation token to the database table with the received tokens. When a student tried to re-register for the lottery, the Tombola ABC System would check if the pseudonym was already in the database and would not allow the User to register more than once for the raffle.



```xml
<abc:PresentationPolicyAlternatives xmlns:abc="http://abc4trust.eu/wp2/abcschemav1.0"  Version="1.0">
  <abc:PresentationPolicy PolicyUID="urn:patras:policies:Tombola">
    <abc:Message>
       <abc:Nonce>bkQydHBQWDR4TUZzbXJKYUphdVM=</abc:Nonce>
      <abc:FriendlyPolicyName lang="en">Presentation Policy for Tombola</abc:FriendlyPolicyName>
      <abc:FriendlyPolicyDescription lang="en">Enter the Tombola - your matriculation number will encrypted</abc:FriendlyPolicyDescription>
    </abc:Message>
        <abc:Credential Alias="#credTombola">
      <abc:CredentialSpecAlternatives>
        <abc:CredentialSpecUID>urn:patras:credspec:credTombola</abc:CredentialSpecUID>
      </abc:CredentialSpecAlternatives>
      <abc:IssuerAlternatives>
        <abc:IssuerParametersUID>urn:patras:issuer:credTombola</abc:IssuerParametersUID>
        <abc:IssuerParametersUID>urn:patras:issuer:credTombola:idemix</abc:IssuerParametersUID>
      </abc:IssuerAlternatives>
          <abc:DisclosedAttribute AttributeType="urn:patras:credspec:credTombola:matriculationnr">
        <abc:InspectorAlternatives>
          <abc:InspectorPublicKeyUID>urn:patras:inspector:tombola</abc:InspectorPublicKeyUID>
        </abc:InspectorAlternatives>
        <abc:InspectionGrounds>Only the winner of the tombola will have his/her matriculation number revealed.</abc:InspectionGrounds>
      </abc:DisclosedAttribute>
    </abc:Credential>
  </abc:PresentationPolicy>
</abc:PresentationPolicyAlternatives>
```

Figure 46: The Presentation Policy of the Tombola ABC System

In order for a verification protocol to complete without exceptions on the ABCE layer, the pilot administrators had to make sure that both communication ends (User and Verifier) were using the same resources i.e. system parameters, Issuer parameters, credential specifications, revocation authority parameters and inspector public key. The system parameters, Issuer parameters, credential specifications and revocation information were provided by the administrators of the University Registration System. The credential specification for the Tombola Credential and the Inspector public key was generated by the administrators of the Tombola System. Thus, a crucial administrator's task was to create the User installer bundles containing the correct resource files and moreover upload the same files to the Tombola ABC storage space.



```
pyrgelis@tombola:~/jetty/jetty-hightide-7.0.1.v20091125/webapps$ ls -l issuer_resources/
total 40
-rw-r--r-- 1 pyrgelis pyrgelis  440 2013-10-16 11:48 cas_keys_pk
-rw-r--r-- 1 pyrgelis pyrgelis 4034 2013-10-16 12:04 inspector_publickey_urn_patras_inspector_tombola
-rw-r--r-- 1 pyrgelis pyrgelis 2796 2013-10-16 11:48 issuer_params_urn_patras_issuer_credCourse_uprove
-rw-r--r-- 1 pyrgelis pyrgelis 3425 2013-10-16 11:48 issuer_params_urn_patras_issuer_credTombola_idemix
-rw-r--r-- 1 pyrgelis pyrgelis 4355 2013-10-16 11:48 issuer_params_urn_patras_issuer_credUniv_idemix
-rw-r--r-- 1 pyrgelis pyrgelis  440 2013-10-16 11:48 pki_keys_pk
-rw-r--r-- 1 pyrgelis pyrgelis  828 2013-10-16 11:48 pki_keys_sk
-rw-r--r-- 1 pyrgelis pyrgelis 3104 2013-10-16 11:48 revocation_authority_urn_patras_revocationauthority_default
-rw-r--r-- 1 pyrgelis pyrgelis 3140 2013-10-16 11:48 system_params
pyrgelis@tombola:~/jetty/jetty-hightide-7.0.1.v20091125/webapps$ ls -l xml/credspecs/
total 12
-rw-r--r-- 1 pyrgelis pyrgelis  996 2013-10-16 11:48 credentialSpecificationPatrasCourse.xml
-rw-r--r-- 1 pyrgelis pyrgelis  936 2013-10-16 11:48 credentialSpecificationPatrasTombola.xml
-rw-r--r-- 1 pyrgelis pyrgelis 2470 2013-10-16 11:48 credentialSpecificationPatrasUniversity.xml
pyrgelis@tombola:~/jetty/jetty-hightide-7.0.1.v20091125/webapps$ ls -l verifier_storage/
total 1360
-rw-r--r-- 1 root root       0 2013-10-16 12:16 credential
drwxr-xr-x 2 root root    4096 2013-10-16 12:16 images
-rw-r--r-- 1 root root       0 2013-10-16 12:16 inspectorSecrets
-rw-r--r-- 1 root root       0 2013-10-16 12:16 issuerLog
-rw-r--r-- 1 root root       0 2013-10-16 12:16 issuerSecretKeys
-rw-r--r-- 1 root root   25385 2013-10-16 12:16 keystorage
-rw-r--r-- 1 root root   12528 2014-02-10 21:49 pseudonyms
-rw-r--r-- 1 root root       0 2013-10-16 12:16 revocationAuthoritySecrets
-rw-r--r-- 1 root root       0 2013-10-16 12:16 revocationAuthorityStorage
-rw-r--r-- 1 root root       0 2013-10-16 12:16 secrets
-rw-r--r-- 1 root root 1336381 2014-02-10 21:49 tokens
pyrgelis@tombola:~/jetty/jetty-hightide-7.0.1.v20091125/webapps$ 
```

Figure 47: Tombola ABC System Resources

As can be seen on Figure 47, the administrators need to upload on the Tombola ABC System storage space the system parameters, the Issuer resources, the inspector public key as well as the credential specifications. With these resources in place the verification protocols would work without any exceptions at the ABCE layer.

In order to deploy the Tombola ABC System, the developers had to compile the corresponding project that created the verifier services and upload the .war file to the Jetty web server. During the integration phase, they executed these services locally along with the Tombola Application in order to check their operation and fix minor bugs.

3.7.2.2.1 Deployed ABCE Methods

The methods that were used for the deployment of the Tombola ABC System were similar to those used for the Verifier of the Course Evaluation System (see also Subsection 3.6.2.2.1).

3.7.2.3 Deployment

After correcting some minor bugs, the administrators deployed the Tombola ABC System on the Tombola server and they performed some testing for the Tombola System in the whole.

In order to test the operation of the Tombola ABC System test Users possessing hardware or software smart cards when interacting with the Tombola System had to be introduced.

As mentioned before, the Tombola ABC System deploys a Verifier service. The presentation policy requires from the Users to present a scope exclusive pseudonym for the scope “urn:patras:tombola” as well as to prove their possession of a Tombola Credential issued by the Course Evaluation System and encrypt the matriculation number from this credential with the Inspector’s public key. Firstly, the system developers did some tests with Users having the Tombola credential to make sure that they could get registered without any problems. Then, they did some tests with test Users who did not have the Tombola credential in order to ensure that they could not get registered for the online lottery. Finally, they tried to register some Users multiple times for the raffle in order to make sure that they would be registered only once and not multiple times, preventing them to cheat and increase their chances to win the lottery.

When enough test Users were registered to the Tombola System, the lottery procedure was tested. More precisely, it was checked that when the lottery timer had expired the system picked a random winning presentation token among the registered ones and announced it as the winner of the raffle. Then, using the Inspect Tool (see Section 3.3.1) along with the Inspector's smart card the administrators tried to reveal the matriculation number from the presentation token. Thus, it was verified that this number was one of the matriculation numbers of the test Users we had registered and the whole raffle process was working as expected.

3.7.2.4 Operation

During the pilot operation the administrators should make sure that the Jetty server hosting the Tombola ABC System was up and running. Moreover, they should ensure that the necessary resources (system parameters, Issuer resources, inspector public key etc.) were placed on the Tombola ABC System storage space. No issues were reported by the pilot participants regarding the operation of the Tombola ABC System. The students who had obtained successfully a Tombola credential could register for the online lottery without any problems.

3.8 Class Attendance System

The Class Attendance System is the system that the pilot students interact with every time they attend to a course lecture. This system is placed inside the lecture room and is setup by an administrator from CTI. It consists of a laptop running the Class Attendance Application and an NFC smart card reader attached to its' USB port. The NFC reader is responsible for the communication between the smart card and the application. When the Class Attendance Application is initiated and a student waves her smart card in front of the NFC smart card reader a communication protocol takes place between the smart card and the Class Attendance Application. At the end of this protocol, the course attendance counter stored on the smart card is incremented by one.

3.8.1 Class Attendance Application

We here note that no Privacy-ABC technologies are used for the Class Attendance System. The Class Attendance System has a key pair similar to PKI scenarios. The Class Attendance key pair was generated by the administrators of the University Registration System and it was provided to the application developers. The secret key is known only to the Class Attendance System itself whereas its public key is loaded on to the smart cards during the initialization process. Moreover, each smart card is initialized with a counter blob which contains a counter identifier, an index (indicating the counter value - initially set to zero) and a cursor (indicating the last attended lecture identifier).

The main functionality of the Class Attendance Application is to find out when a smart card is present in the communication field of the NFC reader. As soon as a smart card is detected, a protocol between the smart card and the application is executed.

This protocol includes the following actions:

- The application requests from the smart card to generate a fresh random challenge
- The challenge is communicated to the class attendance application
- The application (using its secret key) generates a signature on a message including the counter identifier, the current lecture identifier and the challenge from the card
- The signature is sent to the smart card which tries to verify it. If verification is successful and the lecture identifier is larger than the currently stored identifier, the attendance counter is increased and the last lecture identifier (cursor) is updated.

We here note that when a User interacts with the Class Attendance Application there is no need to enter her PIN.

The smart card commands from the ABC4Trust-Lite API that are used for this protocol are the following:

- The command “GET CHALLENGE(16)” which when called the smart card returns a 16 byte fresh challenge
- The command “INCREMENT COUNTER(counterID, sig)” which when called the smart card tries to verify the signature “sig” increase by one the counter with identifier “counterID” stored on it

The communication protocol is described in more detail on the next subsection.

3.8.1.1 Implementation

The Class Attendance Application is setup offline, prior to each lecture by an administrator from CTI. The administrator initializes the application with a fresh lecture identifier which must be strictly increasing for every new lecture (the lecture identifier is assigned to the variable “newcursor” below).

When initialized, the Class Attendance Application continuously polls the NFC reader to check if a smart card is present. When a student waves his smart card near the NFC reader, the following protocol is executed between the smart card and the Class Attendance Application:

- The smart card generates a random nonce “challenge”. It sends this nonce to the Class Attendance Application and also stores it locally.

- Upon receiving the random challenge, the Class Attendance Application using the Class Attendance System secret key (sk_cas) produces the following signature “sig” (based on a signature algorithm supported also by the smart card) :

  sig = Sign(sk_cas, counterId || newcursor || challenge).

- The smart card attempts to increase by one the counter identified by counterId, verifying the signature sig with the public key of the class attendance system:
  - Verify(pk_cas, sig, m) = true for m = counterID || newcursor || challenge, i.e. it verifies the signature against the stored public key and for the message including the nonce “challenge” that was generated by the smart card in previous step.
  - cursor < newcursor, i.e. it sees a fresh lecture identifier

If one of the checks fails, or no counter blob was stored, it indicates failure towards the Class Attendance Application. Otherwise, the smart card increments the counter value by one, sets cursor = newcursor and indicates successful counter update towards the Class Attendance Application.

Figure 48: The Omnikey 5321 Dual Interface PC-linked Reader

The laptop hosting the Class Attendance Application is a HP laptop (HP Compaq 6720s) with an Intel Core 2 Duo Processor running at 1,6Ghz and has 1 GB of RAM. The laptop has Windows XP service pack 3 installed. Attached to its' USB port is the Omnikey 5321 (Figure 48) smart card reader which offers a dual interface and can communicate with both contact and contactless smart cards. The Class Attendance Application was developed in Java using the package javax.smartcardio (http://docs.oracle.com/javase/6/docs/jre/api/security/smartcardio/spec/javax/smartcardio/package-summary.html) which provides an API for communicating with smart cards.

3.8.1.2 Integration

Deploying the Class Attendance System was done by configuring the Class Attendance Application with a lecture identifier and by compiling it. The resulting .jar file was deployed on a laptop which had the Omnikey 5321 reader attached to its USB port. When starting the .jar file with the appropriate command (java –jar class-attendance.jar), the application would enable its contactless interface and one could play around with the operation of the Class Attendance Application.

In order to test the Class Attendance System the pilot developers initialized some smart cards with a zero counter blob. They did setup the Class Attendance Application with a fresh lecture identifier and deployed it on a PC with the Omnikey 5321smart card reader attached to its USB port. Then, they waved each of the smart cards in front of the NFC reader, waiting for an indication that the communication protocol between the smart card and the application completed successfully (beep sound). Using another script that takes as input the smart card's PIN and prints out the smart card counter information, it was verified that the counter was indeed increased by one and that the last lecture identifier was updated correctly. Moreover, they verified that waving a smart card multiple times in front of the NFC reader does not increase the counter more than once for the same lecture. Setting up the Class Attendance Application with a new fresh lecture identifier (larger than the previous one) the above steps were repeated in order to verify that the application was working as it should.

Another test that was performed was to deploy the Class Attendance Application using a different secret key than the original application's key (the one whose public key was loaded on the smart cards during initialization). With this test it could be verified that when the smart card communicates with a “fake” Class Attendance System, the signature verification fails and thus the counter cannot be increased.

Finally, the developers did some tests regarding the execution time required for the communication protocol between the smart card and the Class Attendance Application. With these tests it was found out that the protocol requires approximately 1.5 seconds in order to complete. Thus, the Class Attendance Application was configured to wait for approximately 2 seconds for the communication protocol to complete, every time a smart card is present in the communication field of the NFC reader. The application was also configured to throw an exception if the smart card present in the field of the NFC reader is removed before the protocol is completed successfully.

3.8.1.3 Deployment

Prior to each lecture, the Class Attendance Application was properly initialized by a pilot administrator with a fresh lecture identifier. The application was compiled into a .jar file which was securely transferred to the laptop hosting the Class Attendance System. The pilot administrator then just had to place the laptop inside the the lecture room, connect the Omnikey 5321 reader to the USB port of the laptop and start the application with the appropriate command (java –jar class-attendance.jar).

3.8.1.4 Operation

When a student waved his smart card in front of the reader, the communication protocol took place increasing the attendance counter on her smart card. Some students, eager to obtain their attendance certification were removing the smart card from the communication field of the NFC reader before the communication protocol was completed. Thus, they got a notification that the protocol was not successful and that they should repeat the process. As soon as the students were instructed by the administrator that they should keep their smart card close to the NFC reader until they heard the confirmation sound (which meant that the protocol was successful) they were able to collect their attendance without any problems.

Another issue that came up during the lectures was that some students could not obtain their attendance unit because their smart card was in “locked” mode. To resolve this issue the students first had to unlock their smart card using the PUK value and then contact the pilot administrators in order to obtain the attendance that they had missed.

3.9 Patras Portal

The Patras Portal is the students' entry for the University of Patras pilot. It is the first link that a student should visit and where she can find information about the pilot.

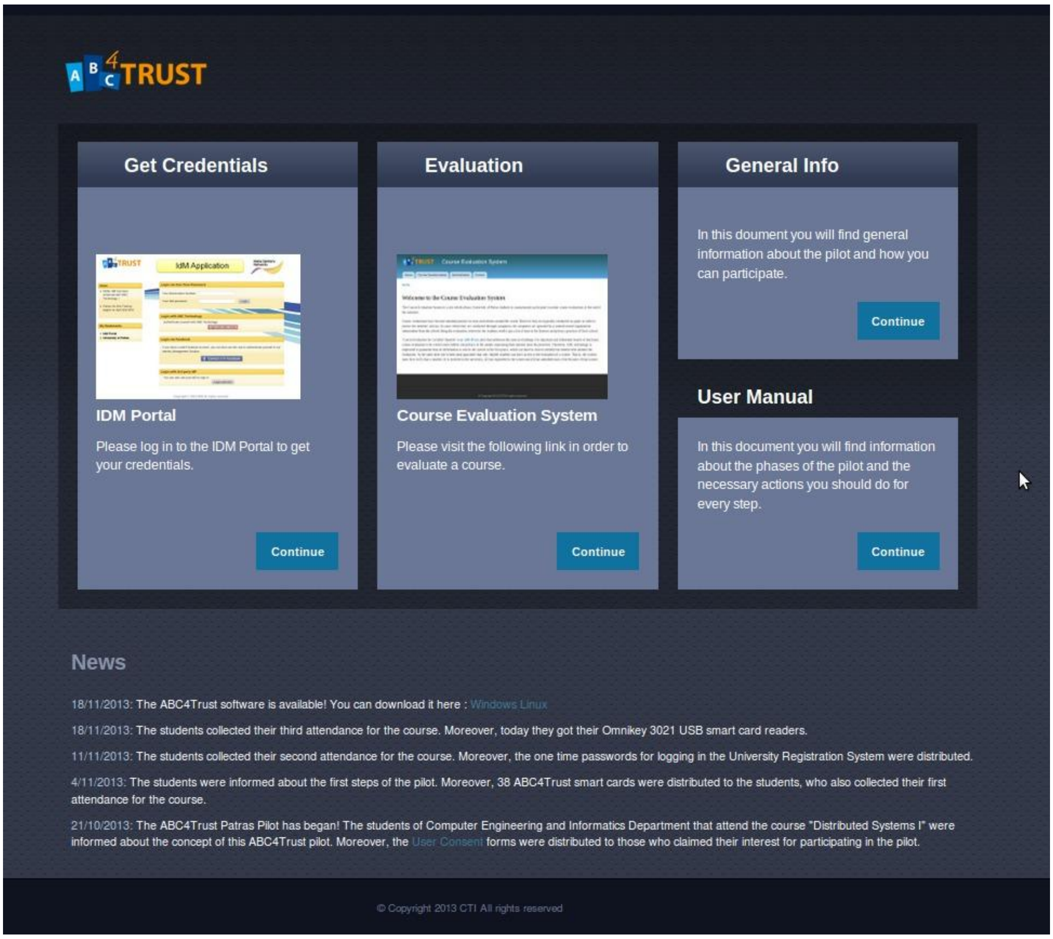
3.9.1 University Portal GUI

The Patras Portal can be seen on Figure 49. It is a simple web page that contains all the information that a User would need for her participation in the Patras pilot. This page is hosted on the same server as the Course Evaluation System and it can be found on the URL https://ces.cti.gr/Portal/Portal.html.

Uploaded on the Patras Portal are documents that are helpful for the participants. As an example, the Information document is where the students can find details about the concept of the pilot and how they can participate. Another document is the User Manual where the students can be informed about all the pilot steps they will go throughout the semester and find some tips in cases they face some difficulties when interacting with the pilot systems. Finally, the User Consent Form that the Users need to sign before participating in the pilot is available on the Patras Portal.

Moreover, the Patras Portal has some blocks that point to the URLs of the pilot systems, i.e. the University Registration System, the Course Evaluation System and the Tombola System. So, the students need only to remember/bookmark the URL of the Patras Portal.

Finally, at the bottom of the page there is a "News" block which is updated by the pilot administrators every time that happens something important for the pilot operation.



Get Credentials

IDM Portal

Please log in to the IDM Portal to get your credentials.

Evaluation

Course Evaluation System

Please visit the following link in order to evaluate a course.

General Info

In this doument you will find general information about the pilot and how you can participate.

User Manual

In this document you will find information about the phases of the pilot and the necessary actions you should do for every step.

News

18/11/2013: The ABC4Trust software is available! You can download it here : Windows Linux

18/11/2013: The students collected their third attendance for the course. Moreover, today they got their Omnikey 3021 USB smart card readers.

11/11/2013: The students collected their second attendance for the course. Moreover, the one time passwords for logging in the University Registration System were distributed.

4/11/2013: The students were informed about the first steps of the pilot. Moreover, 38 ABC4Trust smart cards were distributed to the students, who also collected their first attendance for the course.

21/10/2013: The ABC4Trust Patras Pilot has began! The students of Computer Engineering and Informatics Department that attend the course "Distributed Systems I" were informed about the concept of this ABC4Trust pilot. Moreover, the User Consent forms were distributed to those who claimed their interest for participating in the pilot.

© Copyright 2013 CTI All rights reserved

Figure 49: The Patras Portal

3.9.1.1 Implementation

As described above, the Patras Portal is nothing more than a web page (html/css) which provides basic information about the pilot and the necessary links for the pilot systems. Thus, no Privacy-ABC technologies were used for the implementation of the Patras Portal.

3.9.1.2 Integration

The Patras Portal was initially designed and hosted locally on the developer's PC. Some minor bugs regarding the interface were corrected before it was uploaded on the production server.

3.9.1.3 Deployment

The only testing that was performed regarding the Patras Portal, was to verify that all the links provided in the web page work as they should. Thus, it was made sure that the page content is reachable by the students.

3.9.1.4 Operation

The pilot administrators had to make sure that the Apache web server that hosted the Patras Portal was up and running.

No problems were detected throughout the pilot regarding the operation of the Patras Portal. The “News” section of the page was updated by the pilot administrators every time there was an update for the status of the pilot.

3.10 Legal Topics for the Patras pilot

Referring to Section 1.2, this chapter elaborates on the necessary legal considerations when deploying Privacy-ABC technologies in settings like the Patras university pilot. As already mentioned, these considerations will only cover the mere technical perspective, leaving the user focused aspects of the pilot preparations (e. g. the matters of drafting the consent forms, information sheets, and the user manual) to the pilot specific deliverables [D63] (for the school pilot), and [D73] (for the university pilot). So the following sections focus solely on the specific frame conditions and requirements in the context of course evaluation by university students and the lessons learned during preparation and execution of this pilot.

3.10.1 Role Allocation

Similarly to the Swedish school pilot, the difficulty of the Patras pilot was that the Computer Technology Institute "Diophantus" (CTI) was not able to completely set up and operate the pilot architecture on their own. Therefore, the help of other project partners was enlisted here as well. In this case, only the assistance of NSN was needed for the provision of the application and the administration of the University Registration System. This system was run on 2 machines residing at the premises of the CTI. Furthermore, NSN deployed the Revocation Authority and supported CTI with the debugging, the maintenance, and the performance control of the running IdM system.

Just as in the Söderhamn pilot, it was again necessary to allocate different legal roles to the persons and legal entities involved. Each role correlates with a certain set of rights and obligations, which in some cases have to be fulfilled prior to the data processing, or even the development of new applications. As in the Swedish school pilot, the laws of two different states had to be considered. In the Patras pilot, the Greek[12] and the German[13] data protection laws were applicable. A detailed definition each legal role for the pilot at the University of Patras can be found in Section 4.6. of [D51].

A first mapping of legal roles was conducted in chapter 4.6 of D5.1 (Scenario definition for both Pilots). In contrast to the Söderhamn school pilot, the organisational frame conditions remained generally the same, therefore an adaption of this mapping was not necessary. According to this adequate mapping of legal roles, CTI fulfilled the role of a data controller as responsible party, whereas NSN is data processor acting on behalf of CTI.

[12] Namely Law 2472/1997, which is the enactment of the Data Protection Directive, and Law 3471/2006 being the enactment of the E-Privacy Directive). Both laws are available in an English translation: http://www.dpa.gr/portal/page?_pageid=33,43560&_dad=portal&_schema=PORTAL.

[13] Bundesdatenschutzgesetz (BDSG), an English translation can be found online at: http://www.bfdi.bund.de/EN/DataProtectionActs/Artikel/BDSG_idFv01092009.pdf?__blob=publicationFile.

3.10.2 Processing Contract between CTI and NSN

As explained before, the data processor only applies the technical means set out by the data controller on his behalf and processes the personal data according to his instructions. Nevertheless, CTI as the data controller remains fully responsible for the data processing. To stipulate the rights and obligations of both parties, a contractual fixation was necessary. According to Art. 10 (4) Greek Law 2472/1997, the contract had to be in written form and needed to transfer the legal obligations regarding confidentiality and security of processing from the controller onto the processor. Similar to the Söderhamn pilot, to ensure a higher level of personal data protection, the contract governing the pilots also integrated the stricter requirements for processing contracts from § 11 (2) of the German BDSG, which were already presented in Section 2.8.2 (Processing Contracts between Norrtullskolan, EDOC and NSN).

Just as in the Söderhamn pilot, NSN helped with the technical preparations of the pilot, thereby being closely involved in the setup of the Privacy-ABC architecture for the Patras University. Prior to the start of the pilot, CTI also deactivated direct administrative access of NSN. So also in the Patras pilot, NSN was by default not involved with the processing of any personal data of pilot participants and would be authorized to access such data only by the contractually foreseen procedures and only in the exceptional cases as described above in the section about the ‘Processing Contracts between Norrtullskolan, EDOC and NSN’.

3.10.3 Specific Legal Requirements of the Patras Pilot

Prior to the pilot, the responsible Greek Data Protection Authority had to be notified in detail about the upcoming processing of personal data. According to Art. 6 Law 2472/1997, the controller CTI had to declare:

- *his/her name, trade name or distinctive title, as well as his/her address.*
- *the address where the file or the main hardware supporting the data processing are established.*
- *the description of the purpose of the processing of personal data included or about to be included in the file.*
- *the category of personal data that are being processed or about to be processed or included or about to be included in the file.*
- *the time period during which s/he intends to carry out data processing or preserve the file.*
- *the recipients or the categories of recipients to whom such personal data are or may be communicated.*
- *any transfer and the purpose of such transfer of personal data to third countries.*
- *the basic characteristics of the system and the safety measures taken for the protection of the file or data processing.*

All this information was conveyed in the data processing contract to comply with the requirements of Art. 6 Law 2472/1997.

3.10.4 Considerations about inspectable Areas

The architecture set up for the 2nd round of the Patras pilot did not require any inspection feature for the course evaluation which the participating students were asked to conduct with the help of this system. The characteristics of the architecture were sufficient to hinder any unauthorized multiple evaluations. The alteration of the evaluation ratings was only possible until the final submission, so misuse cases of course evaluation were precluded by the system itself.

However, to motivate the students to participate in the pilot and in the completion of the evaluation task, a tombola was offered to all students who had submitted their course evaluation through the Privacy-ABC architecture. This tombola was conducted within the pilot architecture through an advanced issuance feature enabled by the new crypto architecture. This made it possible that students who submitted a course evaluation could afterwards obtain a credential to use it for participating in the tombola if so desired. The system then drew a winner by an automated raffle and the inspection feature was used in this particular case only to reveal the matriculation number of the winner to distribute the tombola prize. Therefore, the inspection functionality was assigned to be performed only once throughout the whole pilot. The students were informed beforehand, and the participation in the tombola was not mandatory for the general pilot participation and course evaluation at all. Consequently, legal requirements were far less complex and strict than in the Söderhamn pilot, only demanding prior information of the participants and setting up a clear process for the execution of the winning credential inspection.

4 Lessons Learned

Sections 2 and 3 gave an overview on how Privacy-ABC technologies were used in the pilots and what had to be taken into account in the various phases of the project. When mapping the knowledge attained so far to potential future projects, the following questions come up:

- What did the developers, integrators and administrators learn out of these pilots?
- Which areas can be improved and which problems still need to be solved?

The next session will cover these topics and, on top of this, give some hints and tips for project leaders and developers interested in adopting these technologies.

Please note, that only the technical ‘Lessons Learned’ topics will be analyzed. Non-technical topics mapped to the Users, especially the ‘usability’ topics, are content of [D63] and [D73].

4.1 Lessons Learned of Söderhamn Pilot

4.1.1 1st Round

The 1st round of the Söderhamn pilot started on May 13th 2013 and ended on June 10th 2013. Most of the issues detected in the 1st round have been fixed before the start of the 2nd round. One issue that could not be solved was the limited size of memory space on the smart cards - for which a work-around was made. As memory space on the card is limited the number and the structure of the credentials have to be carefully planned. i. e. the number of possible subject attributes in the credential called credSubject had to be limited so that it did not occupy too much memory space on the smart card. Another issue related to the limited size of memory was that the scenarios also had to be analyzed and planned in a way so that they use as little memory space as possible. One way to achieve this was to add functionality to the User Application allowing the User to be able to free memory space on the smart card by removing revoked credentials from the card’s memory.

The school network firewall settings caused problems during the preparation of the 1st round of the pilot. Some of the services and components needed by the pilot were communicating via port 8443 and 8444 that were not opened (allowed for communication) at the school network. One solution could of course have been to change and reconfigure the pilot applications to use other ports but as this would require more time a decision was made to solve the problem much faster by asking the school administrator to change the firewall settings and the domain security policy. After those changes were done the pilot was successfully conducted.

4.1.2 2nd Round

The 2nd round of the Söderhamn pilot concluded successfully on February 2014 and was based on the old crypto architecture. Before the start of the 2nd round of Söderhamn pilot many changes were made to improve the performance and enhance the user experience.

Entering the PIN code on every interaction involving the smart card was reported by Users as annoying. The solution was to cache the PIN code of the smart card after the first entry and to provide the PIN from the cache during a session. The PIN is deleted from the cache on session end (e.g. system shutdown), restart of services, closed browser or ejection of smart card.

Another action taken to enhance the overall performance and to optimize the smart card memory usage was to remove revocation from all credentials except the credSchool credential – for more details please see Section 2.5.1.4. The project also made efforts to improve performance by decreasing the key length in the crypto engines. Unfortunately, decreasing the key length for U-Prove to 1024 failed in the old crypto architecture.

One issue during the pilot was that the systems of the pilot depend heavily on the parameters defined in the credentials specification. Even a minor change of one attribute requires not only an update of the credential specification itself but also an update of the ABC System binaries. Re-deployment of all systems that contain the ABCE would be required.

For the 2nd round of the pilot, a new version of the Smart Card Initialization Tool was implemented (see Section 2.2.1.2). The tool was used by EDOC to initialize and prepare approximately 400 smart cards for the 2nd round of the pilot. The new tool eliminated the need of using the Smart Card Registrar.

The Smart Card Initialization Tool used to prepare the smart cards had to be manually supplied with different parameters such as the crypto engine type (Idemix or U-Prove) and the smart card ID. This process was time consuming and the risk of making errors increased when a large number of smart cards had to be prepared for the 2nd round. To solve this issue EDOC enhanced the smart cards initialization process by creating a script that automatically fed the tool with the crypto engine type and the smart card ID. The script had other features such as a voice notification about the initialization results and a functionality of concatenating the output of the Smart Card Initialization Tool into one single text file instead of separate file for each initialized smart card. The concatenated file containing the smart card ID and the PIN/PUK were imported to Microsoft Excel and used while distributing the smart cards to the Users.

The conclusion is that the 2nd round of the Söderhamn pilot was overall successful. Issuance, verification, inspection and revocation in combination with the Restricted Area Application functionality served the pilot scenarios well. One important issue that needs to be improved for the future is performance. The time it takes to sign in into a Restricted Area should come down to the same level as a login using a username and a password.

4.2 Lessons Learned of Patras Pilot

4.2.1 1st Round

The 1st round of the Patras pilot was completed on February of 2013 and two major issues came up during its operation. The first one was related with the smart card's available memory and the second one had to do with the concurrent access to the services provided by the ABCE.

Regarding the first issue, some students reported to the pilot administrators that when they attempted to obtain multiple times the University and Course credentials from the University Registration System, their smart card was no longer working. As it was discovered by the log files provided by the students, the smart card in that case had run out of RAM. As a result, it could not function properly and the students could no longer interact with the pilot systems. Thus, the students had to contact a pilot administrator who would obtain the broken smart card and re-initialize it. At that point, he would provide the User with the newly initialized smart card along with the new PIN and PUK values and advise them to re-obtain only once their credentials so that they could interact with the pilot systems without any further problems. As will be described in Section 4.2.2, this issue was resolved in the 2nd round of the Patras pilot.

Regarding the second issue, some students reported to the pilot administrators that at some points in time they could not log-in to the Course Evaluation System whereas at some other times they could do so, without any problems. Having a look at the server log-files the administrators found out some exceptions at the ABCE layer which were provoked by Users trying to enter Course Evaluation System simultaneously. More specifically, it was noticed that when the Course Evaluation System ABCE layer had to process two or more concurrent presentation tokens an exception was thrown. As a result, only one of the concurrent presentation tokens could be verified successfully and thus only one user could log-in at the Course Evaluation System. We here note, that the ABCE layer was not designed to be thread-safe at that point in time and such an issue was expected. This issue was reported to the ABC4Trust consortium and it was resolved in the next versions of the ABC4Trust reference implementation.

4.2.2 2nd Round

The 2nd round of the Patras pilot concluded successfully on February 2014. As mentioned earlier in the document, it was based on the new crypto architecture that was developed by WP4. As a result the interoperability of the Idemix and U-Prove technologies were demonstrated, i.e. there was no distinction between U-Prove and Idemix smart cards any longer. Moreover, some new features were deployed (e.g. issuance with carry-over attribute, revocation, inspection etc.) that would make the overall scenario demonstrate more Privacy-ABC functionalities as well as make it look much more attractive for a User.

Regarding the functionality of the ABCE layer, no major issues came up during the pilot operation. The new crypto architecture was working as expected and no exceptions during issuance/verification protocols were discovered throughout the pilot. Some issues that were discovered during the testing phase e.g. see Section 3.5.1.3, were reported to the corresponding developers and were resolved prior to the pilot beginning. Moreover, the issues that were reported to WP4 after the completion of the 1st round of the Patras pilot were now solved i.e. the ABCE was now designed to be thread safe and the smart card would not break down if you stored more credentials that it could hold in its memory.

One issue that came up during the pilot was related to the Smart Card Application. When a student would try to access the Course Evaluation System without her smart card having reached the attendance threshold, the state of her university credential would change from'presentable' to 'presentation committed' and stay in that state when the proof failed. As a result, this credential could not be used in future proofs due to its inconsistent state (see also Section 3.2.2.4). However, this issue did not affect the pilot Users as they were instructed by the administrators to access the Course Evaluation System only at the end of the semester. At that point in time the students had collected enough lecture attendances and this problem did not affect their interaction with the pilot systems.

Having the experience from the 1st round of the Patras pilot, the administrators were much better prepared about the issues that they had to deal with when a User would complain about something not working. As a typical example, some students could not communicate with their smart card through their smart card readers. This issue was related to the smart card reader drivers not being installed properly. Most of the problems that the students had to deal with during the 2nd round of the Patras pilot were usability issues, e.g. the One Time Password for the IdmPortal not working properly or unlocking the smart card when a PUK is not 8-digits long would not work as expected. Despite this, the administrators were now experienced enough to support the Users and help them to resolve their problems.

Finally, one can conclude that the 2nd round of the Patras pilot was successful. Its overall goal which was to deploy the new crypto architecture was achieved. Moreover, the pilot scenario was enhanced so that it would deploy new features of the Privacy-ABC technologies (i.e. inspection, revocation, issuance with carry over attribute). The issues that the Users had to deal with were minor usability issues that could have been prevented with a much more thorough testing phase or by providing a more detailed user manual.

4.3 Technical Recommendations for future Pilots

4.3.1 Strategy for adopting Privacy-ABC Technologies

The ABC4Trust project has proven, that splitting a project adopting these new technologies into several rounds can be highly recommended. The experiences from one round can be taken into account in the next round. It can also be recommended to start with less complexity and with a small number of participants i.e. Users. In the ABC4Trust project, the 1st round on the Patras pilot was the first pilot to adopt Privacy-ABC technologies. This pilot had around 50 Users. Even though the number of participants in the 2nd round of the Patras pilot was nearly the same, the architecture of that pilot was very much more complex: next to introducing an Inspector, a new issuing entity was added and the new crypto architecture was deployed. Contrary to the Patras pilot, the Söderhamn pilot had nearly the same complexity in both rounds if you only look at the architecture. But the number of participants increased from 30 to over 380. The 1st round of Söderhamn was the first to make use of the reIssuance feature and the 2nd round of Patras was the first to make use of the advanced issuance feature (carry-over).

Since adding revocation to a system adds complexity especially to the integration and deployment phases of a project, one might consider including this Privacy-ABC feature at some later point in time. Please note, that Privacy ABC credentials can also contain attributes limiting the validity e.g. an 'expiration date' attribute. Both pilots did not make use of such an alternative to revocation, but the Söderhamn pilot with the birthdate attribute encoded in urn:abc4trust:1.0:encoding:date:since1870:unsigned proves that this is doable.

State-of-the-art development requires a testing environment which is decoupled from the target site. In the case of the Registration Systems of both pilots, such publicly accessible (integration) testing environments were provided. So when problems come up during the operation state of a project, the developers and testers of the applications can try to reproduce these problems in this environment, identify their root cause and test fixes without disturbing the live system.

Also, future pilots should ensure that the requirements in the pilot changes minimally. It takes up a lot of resources and time to discuss and implement new features in the middle of the time plan.

The Issuer and RevAuth parameter handling prior to the operation of the pilot restricts the flexibility and scalability of the entire system. There is no possibility of adding new credSpecs, new Inspectors, new Issuers or new Revocation Authorities during the operation state. This is why the Söderhamn pilot introduced auxiliary attributes in the 'role' credential.

Modifying the applications of the School Registration System to support Swedish required significant changes to all applications. Not only had the text and the JavaScript in the GUIs to be modified, but also all XML files (because of the 'friendly' names). More difficult to conceive was the required code addition to the servlets of the School Registration System applications to which Users have direct access to: They had to be configured to send responses in UTF-8 format as the following code snippet shows.

➢ HttpServletResponse response;
response.setContentType("text/xml;charset=UTF-8");

The ABCE provides so-called 'Helper' methods to the developers. In the case that the adopters implement front-ends that do not directly handle the ABC messages but forward them to another entity, they might need to extract XML content from these messages in order to map this content e.g. to a session or to a User. There are several such cases in the Registration System where this mapping is necessary. The IdM Portal for instance forwards the issuance message it received from the 'IdM ABC System' to the User. In this case it was necessary to extract the revocation handle from this message. After successful completion of issuance, the IdM Portal stores the revocation handle in the IdM database. Two aspects must be taken here into account. Firstly, there is no Helper method available for extracting the revocation handle from an issuance message, so some code had to be written in order to parse the XML message coming from the Issuer in the 'IdM ABC System'. And secondly, both the location of the elements and the syntax of the elements can change over time. The revocation handle is just one example. The IdM Portal also extracts the scope-exclusive pseudonym (sent from the User in base64 format) and the nonce (sent by the Issuer and the Verifier / returned by the User).

So if the Helper methods could be expanded to extract information from the Privacy-ABC messages which is important for entities proxying the messages (e.g. the servlets, GUIs, application front-ends), there would not be any dependency on the version of the ABCE and there would be no risk in inserting new errors when trying to parse them.

4.3.2 Integration of the Privacy-ABC Technologies

Improve and standardize the architecture of systems that have to be integrated:

- Add WSDL description to ABC System REST interfaces
- Validation algorithms and messages, error codes

Asynchronous requests inside the Restricted Area Application between client and server and JavaScript functions which make calls to the REST interface of the ABC System have to be better controlled. This can be done via improving functionality made with KnockoutJS and BackboneJS, which are used more on front-end, with additional server side validations, same changes could be achieved with switch to AngularJS. Actual choice of technology is the subject of separate investigation, but general achievement would be same beneficial – decrease dependency of different systems on each other and improve handling of irregular behavior.

4.3.3 Debugging and Forensics

A commercial system using Privacy-ABC technologies may have several instances of ABCEs installed in several places e.g. on the Issuer, on the Verifier or on the User side. As all of those ABCEs have to be compatible a commercial application using Privacy-ABC technologies should be provided with a functionality that displays (or a method that returns) the build or version number of the ABCE and the embedded crypto engines. This improvement would enhance the process of integration and debugging.

During the operation phase of the pilots, another issue was detected: The RevAuth applications do not produce logs with timestamps. Since the RevAuth application is located on a server dedicated to revocation only, it is therefore nearly impossible to correlate the logs produced by the RevAuth with the logs produced by other servers (e.g. the IdM Portal or the 'IdM ABC System' of the Registration System).

Unfortunately, most of the logs of the ABCE are of type 'System.out.println()'. Therefore, the logs are output in the command window which started the Apache-Tomcat container instead of outputting the logs into the catalina files. Forensics is therefore extremely difficult as 2 issues come up concurrently:

a) the command window has only a limited buffer and loses information over time
b) the logs are distributed over 2 outputs and cannot easily be mapped time wise to each other

In both pilots, due to legal restrictions, the developers of the applications had no free access to the target sites during operation. Therefore projects in the same situation should consider during the implementation phase to generate a sufficient amount of self-explaining logs about the sanity of the systems they are developing. Goal would be to enable debugging and fixing by the local administrators without requiring the support of the application developers. In case that development support is unavoidable, parts of the logs (e.g. the exceptions or an anonymized excerpt of the log file) can be shared. But here, the corresponding agreed-upon measures in the legal framework must be taken into account. The legal framework must also be followed when the last and most effective debugging possibility is chosen: Allowing the developers to access the system.

4.3.4 Security

With the new crypto architecture, generic RESTful 'abc-services' were introduced concurrently in the ABCE (see also Section 3.5.1.1). The 2nd round of the Patras pilot incorporated one of these services (unchanged) as independent server directly into the architecture: namely the 'generic' Revocation Authority. Unfortunately, after the operation phase of the pilot began, it became clear that the generic Revocation Authority opens up a security risk. The reason for this lies especially in the fact that there is no authentication on this interface and that this interface provides not only a public service for obtaining the revocation information (required for generating and verifying proofs), but also the private services for setting up the RevAuth itself and revoking credentials. From a network point of view it is expensive to protect a subset of RESTful methods taking into account that all methods are listening on the same port.

So if an adopter intends to make use of the generic RESTful Issuer, Revocation or Verification services provided by this project, it must be clear that either high-end firewalls must be deployed or that additional development efforts must be spent in order to protect the private methods of these services. The Söderhamn pilot e.g. placed an entity in front of the 'IdM ABC System' of the Restricted Area System, protecting the verification service from direct access over the Internet. In the case of the School and University Registration Systems, the ‘IdM ABC System’ was only accessible for the frontend applications (e.g. the IdM Portal) which acted as proxies when looking at the flow of the Privacy-ABC messages. On top of that, there is always the possibility of installing multiple Apache-Tomcat containers on a single server, whereas each container listens on different ports. The latter was done in both pilots in order to separate the public applications (IdM Portal and IdM Application) from the administration applications (IdM Admin GUI and Smart Card Registrar). With this measure, the firewalls can be easily configured to block Internet access to the administration applications. See [D62] and [D72] for more details on the infrastructure of the pilots.

In order to avoid sniffing and man-in-the-middle attacks, web access was protected by HTTPS. The problem that came up using this technology was that self-signed X.509 certificates not only lead to security alerts in the browsers, but were also rejected by the Java installation on the PCs. This problem became obvious when introducing the Revocation Authority in the 1st round of Söderhamn. The Revocation Authority must be contacted via its REST interface. So there is no browser alert which can be overridden by Users. In order to overcome this problem, the installer program for setting up the User PCs had to be enhanced to add the self-signed root certificate to the keystore of the Java installation. Since the Söderhamn pilot had pupils as Users, it was decided to exchange the self-signed certificate with a certificate that was accepted by all browsers without posting any security warning.

4.3.5 Performance

The largest efficiency problem (and time consuming with respect to debugging) was by far the smart card. Thus, a recommendation would be to focus on other devices, which fulfil the same purpose with the same security. This could e.g. be a smartphone with a tamperproof SIM card or similar device which could do the same computations as the smart card. This means still having to deal with the debugging part, but it would significantly shorten the development time as the hardware could be supported much better. Also, from a usability point of view, it is way easier to carry around your smartphone as you always do anyway, and just install an app instead of having a smart card and a smart card reader.

The efficiency problem could also be remedied a bit by sacrificing security. Instead of using a key-size of 2048 bit, which was the case for the Söderhamn pilot, a future pilot could decrease security to e.g. 1024 bit keys which was the case for the Patras pilot. One has to consider the use-case though, and check if would be okay if the security can be hacked within a certain timeframe. If the information hidden is important enough, one should not lower the key-size.

Yet another thing a future pilot should think about is make the policies used as simple and as few as possible. Each additional condition put into a policy increases the complexity and thus efficiency of the system. This also goes for credential specifications, which should be as simple and concise as possible to lower complexity and strengthen efficiency. Specifically one has to be careful about when to use the power of revocation as this could consume several seconds of each proof due to extra latency because of the revocation authority and added complexity because of the cryptographic layer.

In order to increase efficiency even more, the Verifier could also cache the revocation information needed for the proofs instead of always fetching the latest revocation information. This means that there is a window between the point in time where a credential is revoked and the time the Verifier rejects use of this credential. The tolerated length of this time window would be determined from an analysis of how important it is to prevent bad guys from entering the system.

In general one should cache as much information as possible to save time, but two things in particular are important, namely the PIN code of the User and the values retrieved from the smartcard. The PIN is important to cache as it is in no way a usable system if it continuously asks the User for the PIN code. This means a reduction to security as an adversary could then act as the User if the User leaves the PC without removing her smartcard. Any real system would accept this trade-off though. As for the values from the smartcard, these are important to cache as the communication time with the smart card alone takes up several seconds. Caching data from the smart card makes sense, as some of the cryptographic evidence might be the same depending on the input to the card.

4.3.6 Introducing new Credentials

In order to introduce a new credential for a pilot, some basic steps should be followed. First of all, the pilot administrators need to define the credential specification i.e. the XML structure that describes its features (revocable, key-bound) as well as the attributes that it will contain. In order to generate the Issuer parameters for the new credential, the administrator should have the overall system parameters in place. Then, it will be possible to generate the Issuer public/private parameters for the new credential. Next, the Issuer can determine the credential type (Idemix/U-Prove) (see Subsection 3.4.6.2.1.1). When the Issuer is setup, the credential specification as well as the Issuer public parameters, should be made available to the other entities (Users, Verifiers). In a running pilot, that would mean that the User Application should be re-built with bundles that contain the newly created parameters. Thus, all the Users of the system would have to re-install the User Application. Finally, all Verifiers would have to update their storage space with the latest resources.

Introducing in a pilot a credential that is not key-bound could be risky. That is because if such a credential could be copied to another User’s (than the original owner) storage space, it could be used by her in future proofs. For this reason it is advised that not-key-bound credentials should be stored on a smart card and not on the disk space of a PC. If the Smart Card Application provides a backup mechanism, it should be implemented such that copying a non-key-bound credential to another card is not possible. If there are cases where a not-key-bound credential is stored on the User’s PC (and can be provided easily to other Users), the Verifiers should be aware of it. This way, they could enhance their presentation policies in order to check if the presentation token is coming from the original User (e.g. by asking Users to prove that their attribute in one credential has the same value as the attribute in the non-key-bound credential. As an example, if in Patras pilot we had decided to define the tombola credential as non-key-bound, then the value of the matriculation number in the university credential, could be checked for equality against the same attribute in the Tombola credential).

4.3.7 Introducing new Attributes

If a pilot wants to change the attributes of a credential, it is not easy to do so in a running pilot. You can somewhat easily change them on the Issuer and Verifier, but the problem is all the Users. However, even assuming that the client is in the cloud or similar, the problem is that any old credentials needs to be revoked since they no longer are valid. Then every User has to obtain new credentials from the Issuer. Only then is the system reconfigured, but it can be done.

During the integration phase of the Söderhamn pilot, a problem was detected when credentials stored a large number of attributes. Errors occurred specifically in the subject credential which originally contained more than 20 attributes. So ABCE documentation should clearly state this limitation and the code itself should generate appropriate error printouts if this limit is exceeded. The error printouts should be logged when the Issuer generates the Issuer parameters for this credential and not later on, when the issuance for this credential is triggered by the Users.

4.3.8 The Revocation Authority

If a pilot is running a revocation service, it is of paramount importance that it has a very high availability. It is needed by the Users to be able to perform proofs if those proofs include a revocable credential. If the revocation service is not running, all operations will fail and the entire system will be down. The remedy for this problem is to put the revocation service into the cloud and create a robust network with replications of the application and its database. This should eliminate the problem of having a single point of failure. However, the same could be said for the Verifier, so the cost of doing this would most likely not be worth the effort.

4.3.9 HW vs. SW Smart Card

Built into the ABCE is also a possibility of using software smart cards. However, unless you completely trust that your computer is not infected or that no one else can enter the computer and have enabled encryption of your hard drive, you should not use this in production. The software smart card works by serializing the Java class down to disc. This can easily be reversed by a third party program and the adversary could then read the secret key of the User and can impersonate him. Also, doing multiple copies of the smart card is very easy to do. The software smart card is only created for the purpose of testing so, though working, one should only use hardware smart cards in production. It is possible to do a little extra work to heighten security by encrypting the serialized smart card and, when it should be used, ask for the password used in encryption to decrypt it again. However, this only increases security insignificantly, as an adversary can always launch an online attack or a combined offline and phishing attack for the password.
However, it is obvious that using a hardware smart card is more tedious and slow, making the software smart card best for usability purposes, although it is by far easier to carry the hardware smart card around if a single User uses multiple computers during the pilot.

4.3.10 ABC4Trust Source Code

The code produced in the project is available on GitHub under the URL:
https://github.com/p2abcengine/p2abcengine

What is here provided are a number of core components for authentication with privacy-preserving attribute-based credentials. These core components deal with the policy language that specifies the authentication requirements, a User interface that allows User to select which credentials they want to use to satisfy the authentication policy and then some components to generate and verify authentication tokens.

Building a full-fledged authentication and authorization solution requires a number of additional components such as credential storage or key management. The ABC4Trust code provides basic implementations of such components and an example application as well. While these components could be useful as well, adopters might have to replace them with own ones to integrate the core components into the applications.

Finally, the Github repository does not provide the cryptographic building blocks that generate the cryptographic values in the authentication token. However, the ABCE is designed to be used with Identity Mixer and U-Prove, a separate download of Idemix or U-Prove is necessary. The links for those CEs are also to be found in the Github repository.

Under the Github repository, one can also find the list of licenses:
https://github.com/p2abcengine/p2abcengine/blob/master/Code/core-abce/packages.txt
This list is exhaustive and includes only licenses that are not viral and are all compatible with each other. This also means that a third party could use the ABCE code and develop something on top of this as long as the licenses are included in code and documentation as mandated by the licenses.

4.3.11 Inspector Application Enhancements

For the Söderhamn pilot a custom Inspector Wrapper was built - but it might be useful to have the User Application extended with Inspector functionality. The User Application would need an extra method in the API's on the browser plugins to identify the URI of the Inspector and forward the presentation token which needs to be inspected. Something like :

```
inspect(URI inspectorKey, XML PresentationToken)
```

This way a full 'ABCE' based web application could be built using normal ABCE presentation for logging in the Inspector to the application and standard hooks to perform the actual inspection.
On top of that, basic functionalities like changing the PIN of the smart card or unlocking the smart card via PUK will then also be inherited.
Creating this extension to make the User Application support both normal end User and Inspector functionality is - from WP6 - considered to be a minor task.

4.3.12 Restricted Area Application Enhancements

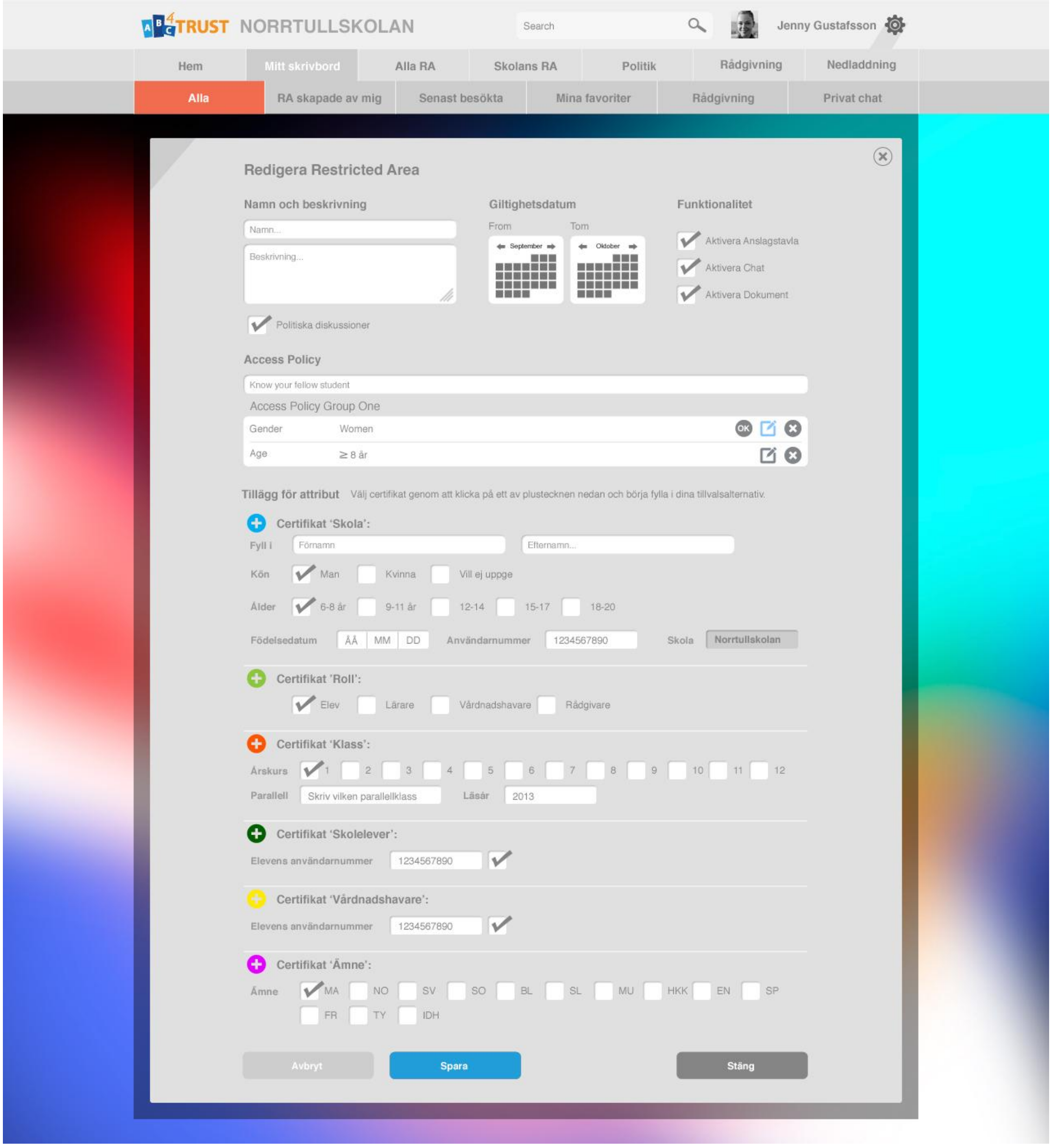
The following are some enhancement suggestions that will make the Restricted Area Application more user-friendly.

The design of the user interface can be improved so that the user is always aware when he switches aliases (context) by changing the background colour of the whole interface etc.

Add a progress indicator to inform the user when the application is busy processing in the background for every asynchronous or long operation.

The application should keep track of all attributes exchanged (used or proven) during any communication thread. Which attribute were exchanged, with who, when did this happen and in which communication thread. This information can be used later to inform the user of how much of his personal data he have revealed and to whom.

The access policy editor GUI used in the pilot to define and to add access policies to Restricted Areas was built as a wizard type of user interface. Based on the experience of both rounds of the pilot the conclusion is that a single page layout which shows all possible options on one page is a more user friendly and faster way of defining access policies. A Mockup of the suggested layout is shown in Figure 50.



Redigera Restricted Area

Namn och beskrivning

Namn...

Beskrivning...

☑ Politiska diskussioner

Giltighetsdatum

From: September

Tom: Oktober

Funktionalitet

☑ Aktivera Anslagstavla
☑ Aktivera Chat
☑ Aktivera Dokument

Access Policy

Know your fellow student

Access Policy Group One

Gender	Women
Age	≥ 8 år

Tillägg för attribut Välj certifikat genom att klicka på ett av plustecknen nedan och börja fylla i dina tillvalsalternativ.

Certifikat 'Skola':

Fyll i: Förnamn Efternamn...

Kön: ☑ Man □ Kvinna □ Vill ej uppge

Ålder: ☑ 6-8 år □ 9-11 år □ 12-14 □ 15-17 □ 18-20

Födelsedatum: ÅÅ MM DD Användarnummer: 1234567890 Skola: Norrtullskolan

Certifikat 'Roll':

☑ Elev □ Lärare □ Vårdnadshavare □ Rådgivare

Certifikat 'Klass':

Årskurs: ☑ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7 □ 8 □ 9 □ 10 □ 11 □ 12

Parallell: Skriv vilken parallellklass Läsår: 2013

Certifikat 'Skolelever':

Elevens användarnummer: 1234567890 ☑

Certifikat 'Vårdnadshavare':

Elevens användarnummer: 1234567890 ☑

Certifikat 'Ämne':

Ämne: ☑ MA □ NO □ SV □ SO □ BL □ SL □ MU □ HKK □ EN □ SP □ FR □ TY □ IDH

Figure 50: Access Policy Editor GUI - a more User friendly Version

4.4 Conclusions on legal Topics

The usage of Privacy-ABCs is a chance to realise each kind of automated personal data processing in a lawful manner while respecting the required purpose-boundary, as well as the correlating principles of data avoidance and data minimisation at the same time. This concerns especially, but not exclusively, all IT processes, where the identity of the user plays a role in distributing some sort of service or product online. These aforementioned principles, as stipulated in the European and national legal data protection framework, demand the development of IT processes which are capable of limiting the collection and use of personal data to the least amount necessary. Moreover, since the user of Privacy-

ABCs is able to only disclose the bits of information which are absolutely needed for obtaining or using a specifically desired online service, a new level of trust between user and service provider can be achieved. For a broader applicability of this chapter regarding the implementation of Privacy-ABC technologies in the future, the following remarks will solely be based on European Data Protection Law. On a European level, the European Data Protection Directive 95/46 EC outlines the legal minimum requirements which had to be adopted in each EU Member state in its national law. Still, these minimum requirements can be exceeded by the national legislation of an EU member state. A higher protection standard is not only permitted by the Directive 95/46/EC but also encouraged. Thus, it is well possible that additional obligations may arise, depending on where the Privacy-ABC architecture is set up. While this document focuses solely on the minimum requirements as demanded by the European common framework, it has to be mentioned that the differences in the national law of the EU member states must be considered diligently.

Both pilots have shown for several reasons that the involvement of legal experts at a very early stage of a pilot architecture development or integration is essential. To facilitate a lawful and effective realisation of Privacy-ABC development and implementation, a clear understanding of the desired technical system functionalities, as well as the specific setting is mandatory. This encompasses the factual, technical, and organisational circumstances of the setup, operation, and maintenance of the Privacy-ABC architecture, as well as a clear conception which parties will be involved in each phase, and especially which parties will be involved in the processing of personal data in the sense of the European Data Protection Directive 95/46EC.

If the data controller is dependent on the assistance of another entity, this entity will be data processor. Such a processor must be chosen with sufficient guarantees regarding technical and organizational measures to protect the personal data which is going to be processed. Therefore, the following questions have to be addressed by each entity considering the development and/or implementation of a Privacy-ABC architecture:

- For which purpose and under which circumstances will the Privacy-ABC technologies be used (including the parties involved and the types of data meant to be processed)??
- Which kind of service or functionality shall be enabled by the Privacy-ABC architecture and how?
- What is the specific purpose of the data processing?

In a next step, a closer look has to be taken towards the data and information that ought to be processed and collected. This concerns the whole lifecycle of the personal data, from the moment of its collection, until it will be deleted when it is no longer needed for the purpose.

- Which information will be collected and processed?
- Is this information personal data in the sense of Directive 95/46 EC and the correlating applicable national law?
- If yes: Does this personal data belong to a special category of data (meaning sensitive data, like health, sex life, ethnicity, etc.) in the sense of Directive 95/46 EC and the correlating applicable national law?
- Is it absolutely necessary to process personal data or will non-personal data suffice to serve the purpose?
- Would inspection feature be mandatory to fulfil a legal obligation to identify a User?
- Whose data will be processed? Are there special types of data subjects demanding specific attention with focus on eventual legal boundaries (like minors, elderly etc.)?
- From which source will the data be obtained?

Furthermore, the involved parties and their role in the data processing have to be examined. This entails a more detailed outline of the whole lifecycle of the processing operation and the exact route of transmission of the data across the network. This is essential to enable a determination which involved party has which legal role and what the scope of their legal rights and obligations is.

- Which is the specific task of each party involved?
- Who determines the purpose and the means of the data processing?
- Who acts only on behalf of the controlling entity and on basis of its instructions?
- In which country are the involved entities established and located?
- In which country is the collection and the processing of the data operated?
- Is cross-border transfer of personal data involved?
- Is the transfer for the purpose of executing a processing operation in another country or is it only for transit?
- Which countries are involved in the cross-border transfer?
- Are these countries EEC member states?
- If not: Are these countries states with an adequate level of data protection as acknowledged by the European Commission
- If not: Does a bilateral agreement between the EEC and the concerned country/ies exist (like the EU-U.S. Safe Harbor Agreement)?

Based on the answers derived from the above mentioned questions, the applicable national law and the legal ground of the processing must be determined:

- Which national law will be applicable?
- Does it convey additional requirements going beyond the minimum requirements of the Directive 95/46 EC?
- On which specific legal grounds will the data be processed (e.g. by law, consent etc.)?
- Is there an obligation to notify a supervisory authority about the personal data processing prior to the start of the actual collection and processing operation?
- Are there any exemptions from this obligation?
- Which information has to be disclosed to the supervisory authority?

Prior to the data collection and processing operation, a written contract between data controller and data processor must be concluded. This contract must stipulate at least:

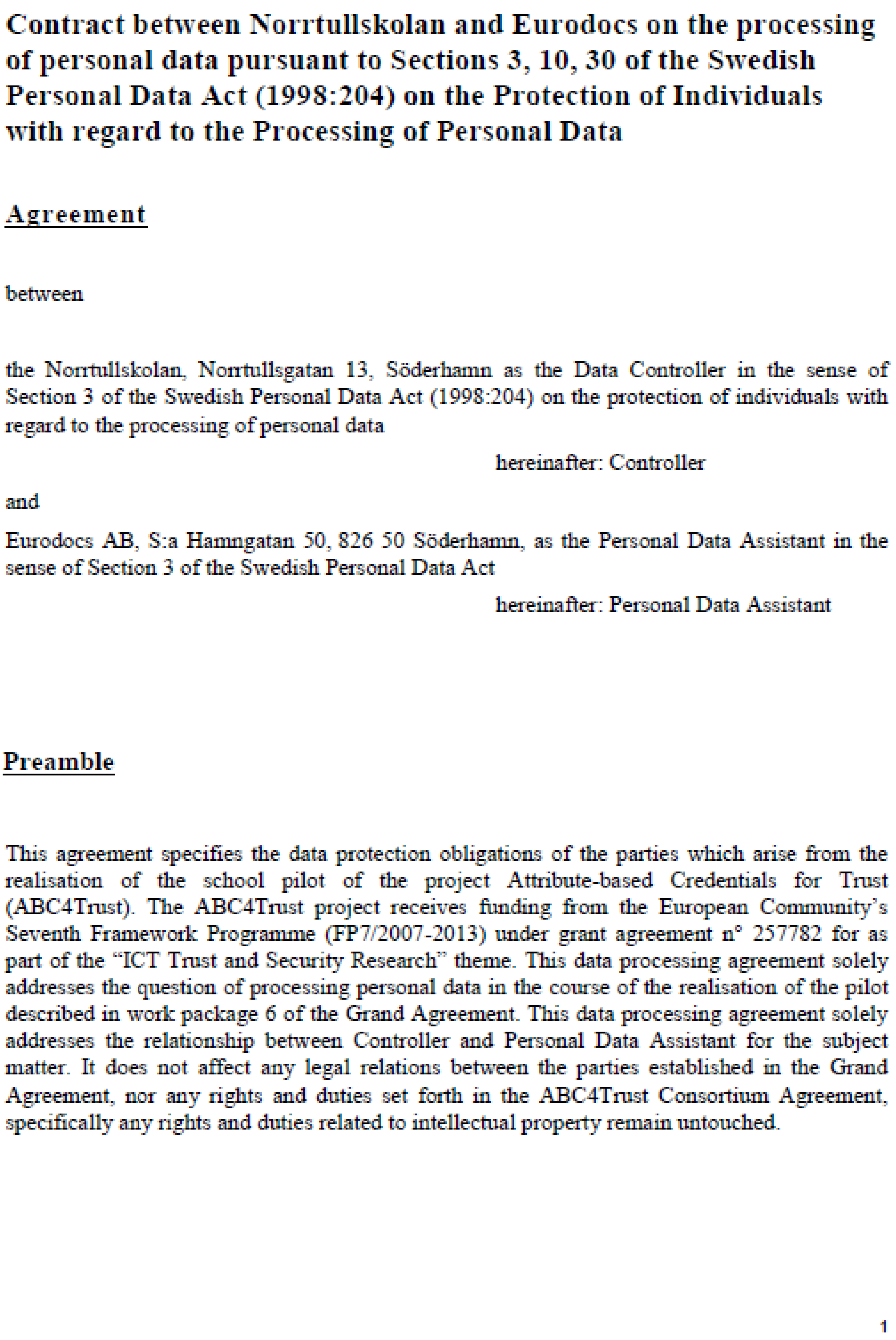
- Who is party of this contract
- The scope of the processing
- The categories of personal data
- That the processor is bound to the controller’s instructions;
- which appropriate technical and organisational measures must be taken by which party for protecting the personal data against accidental loss, alteration, unauthorised disclosure or access;
- Eventual additional content, e. g. based on legal requirements derived from the national data protection law of the state, where the processor is established.

In the ABC4Trust pilots, already the allocation of organisational and legal roles helped tremendously to gain a clearer picture of the whole setting. This was especially helpful in determining the scope, means, and purpose of the personal data processing. Furthermore, it became clear that with a more in-depth analysis of the pilot-specific frame conditions, the initial role mapping, as performed in deliverable [D51] (Scenario definition for both Pilots) needed adaption in the case of the Söderhamn school pilot. As a consequence, legal experts concerned with the preliminary examination of the case, need to pay close attention to any changes in the setting and frame conditions, as they may become relevant for a continued evaluation and eventual adaption of legal conclusions.

The precise set-up and the above questions compose the specific circumstances of the processing operation. While they form a complete picture of the setting, this information must be communicated to the concerned data subject, since it is the essential basis for a valid, informed consent. This is especially important if the processing of special categories of personal data is intended. Such sensitive data would entail all information about racial or ethnic origin, the political opinions, the religious or philosophical beliefs, a trade-union membership, or the health or sex life of a person. The processing of such sensitive data is allowed only under much stricter legal preconditions than it is normally the case, such as in certain scenarios provided by national law or on the basis of a valid, explicit and informed consent of the data subject. Regarding the regular requirements of a valid informed consent when not some kind of sensitive data is involved, [D63] and [D73] will provide an overview of necessary content. Regarding the essential content of processing and sub-processing contracts in general, this chapter already provided some first pointers. However, an update of this document is to be considered to provide for a more comprehensive guidance through the mandatory requirements as demanded by the European legal data protection framework.

Appendix A Processing Contracts

A.1 Processing Contract of Söderhamn Pilot



Contract between Norrtullskolan and Eurodocs on the processing of personal data pursuant to Sections 3, 10, 30 of the Swedish Personal Data Act (1998:204) on the Protection of Individuals with regard to the Processing of Personal Data

Agreement

between

the Norrtullskolan, Norrtullsgatan 13, Söderhamn as the Data Controller in the sense of Section 3 of the Swedish Personal Data Act (1998:204) on the protection of individuals with regard to the processing of personal data

hereinafter: Controller

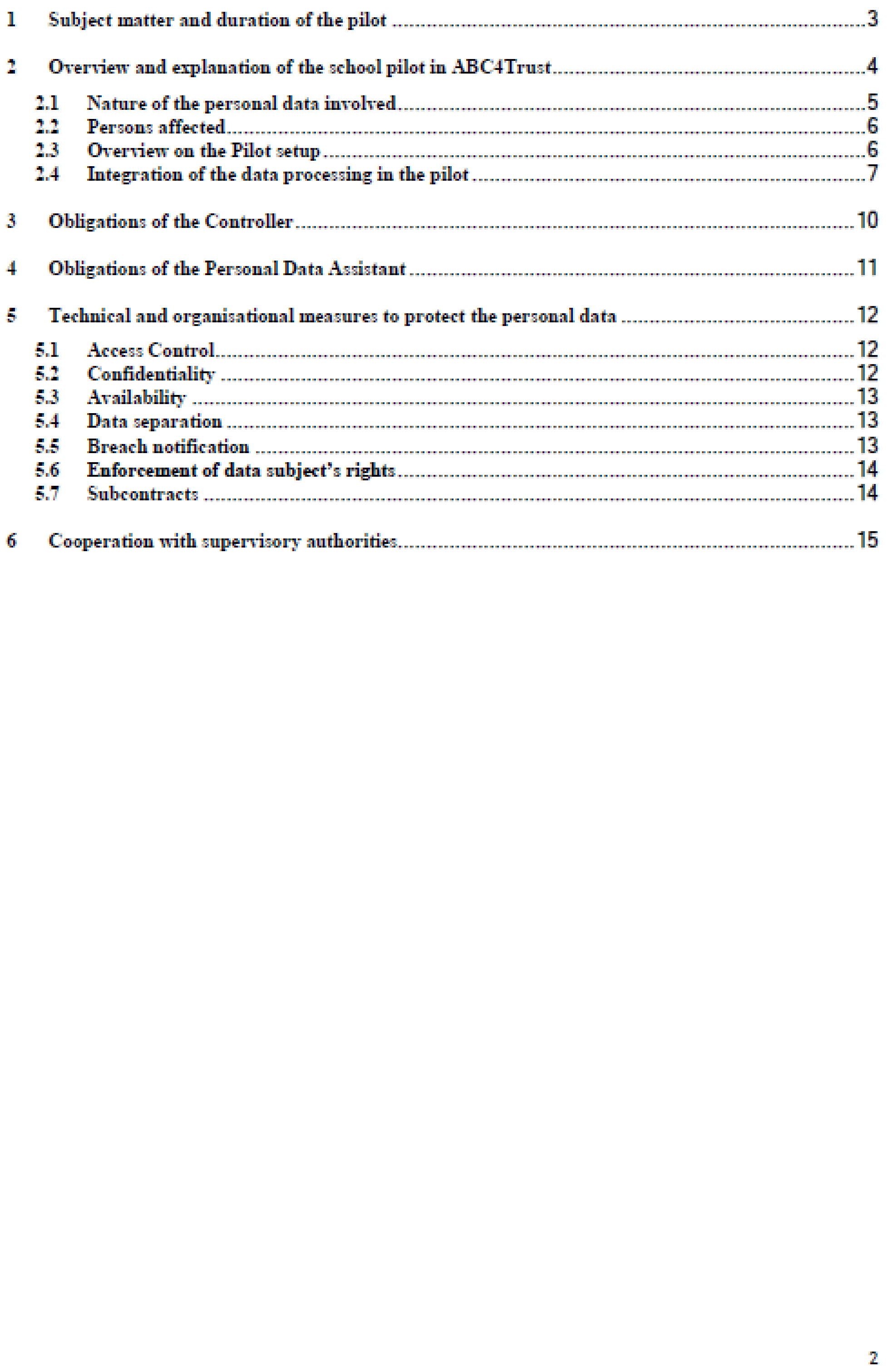
and

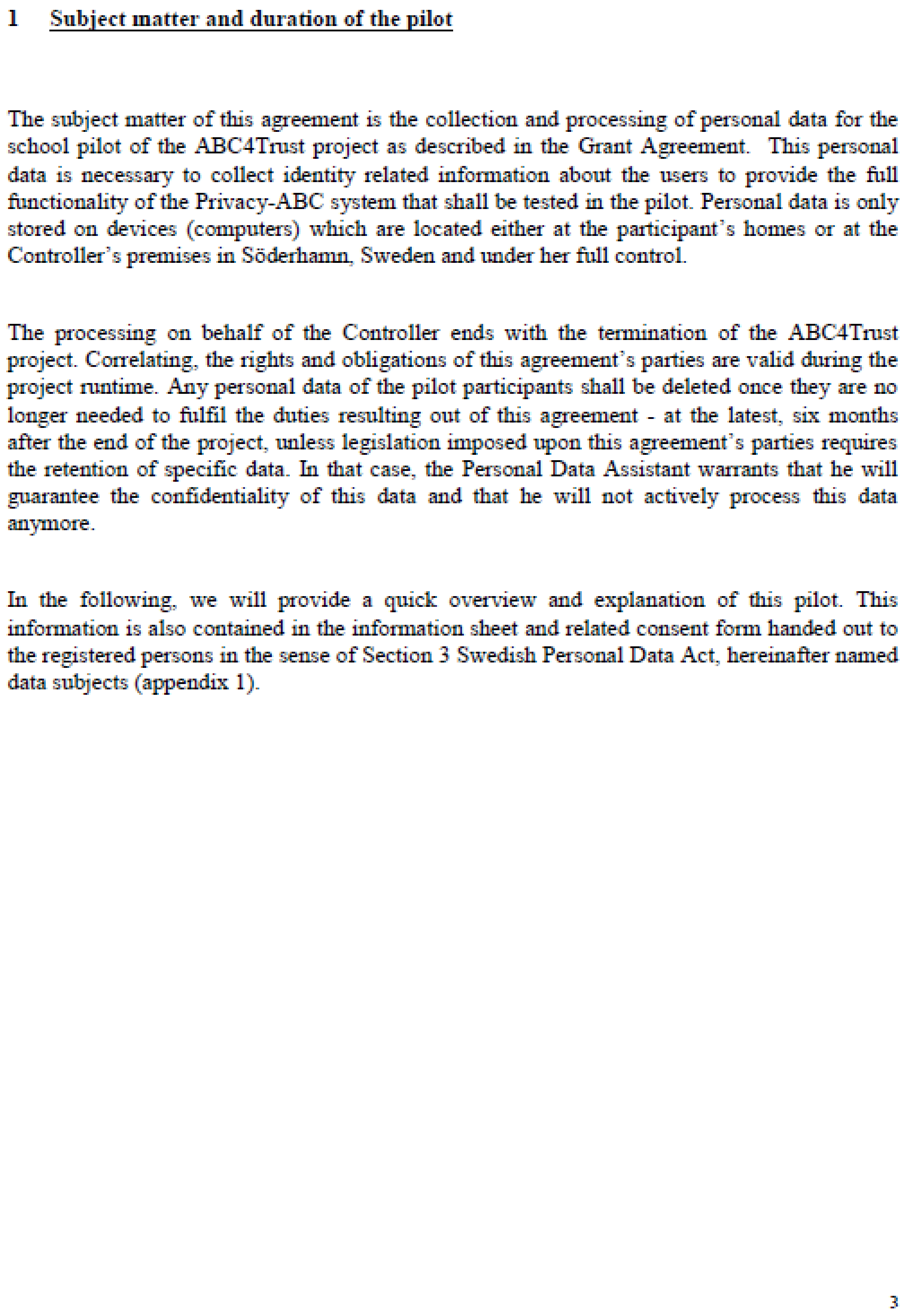
Eurodocs AB, S:a Hamngatan 50, 826 50 Söderhamn, as the Personal Data Assistant in the sense of Section 3 of the Swedish Personal Data Act

hereinafter: Personal Data Assistant

Preamble

This agreement specifies the data protection obligations of the parties which arise from the realisation of the school pilot of the project Attribute-based Credentials for Trust (ABC4Trust). The ABC4Trust project receives funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 257782 for as part of the "ICT Trust and Security Research" theme. This data processing agreement solely addresses the question of processing personal data in the course of the realisation of the pilot described in work package 6 of the Grand Agreement. This data processing agreement solely addresses the relationship between Controller and Personal Data Assistant for the subject matter. It does not affect any legal relations between the parties established in the Grand Agreement, nor any rights and duties set forth in the ABC4Trust Consortium Agreement, specifically any rights and duties related to intellectual property remain untouched.



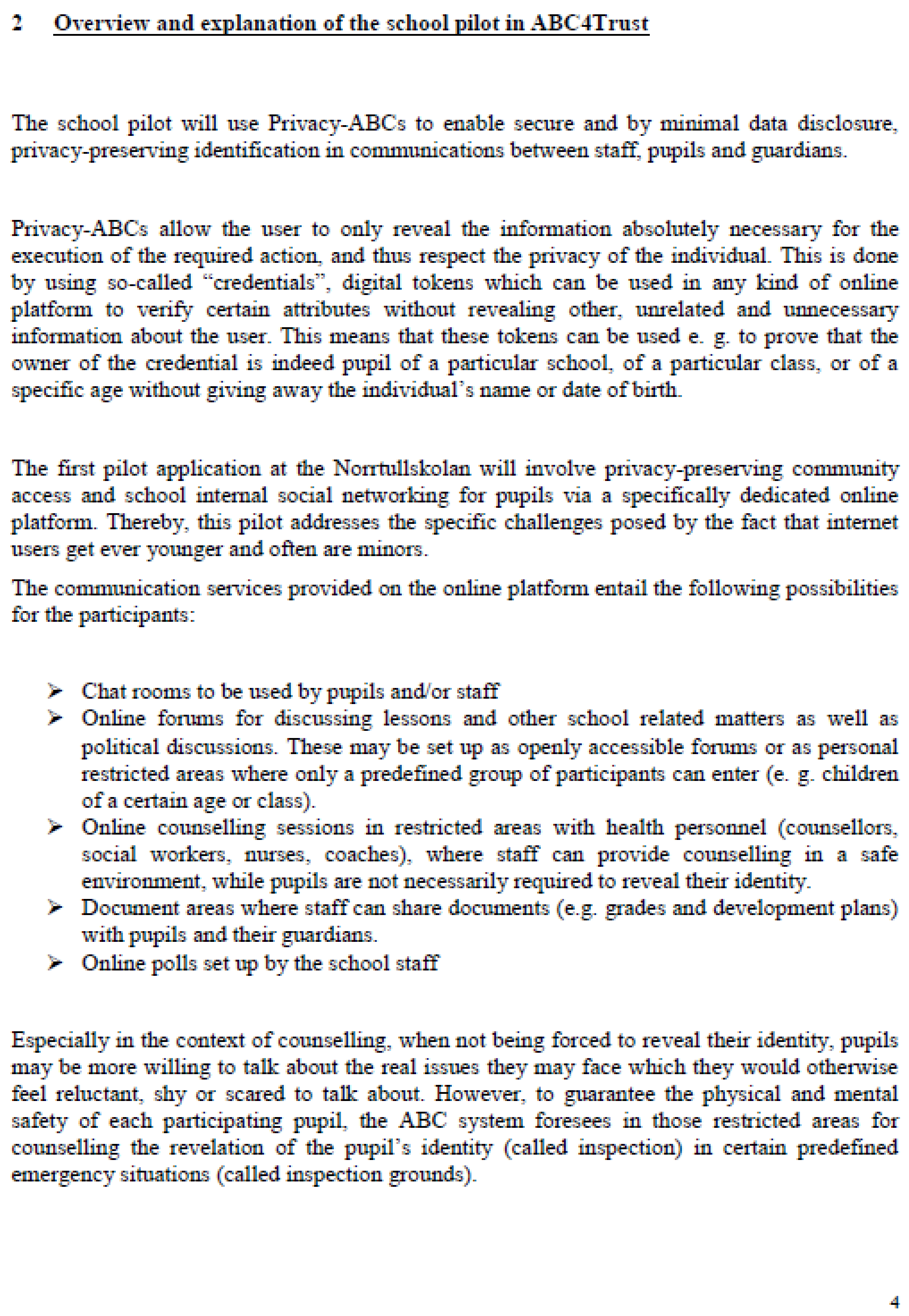


1 Subject matter and duration of the pilot

The subject matter of this agreement is the collection and processing of personal data for the school pilot of the ABC4Trust project as described in the Grant Agreement. This personal data is necessary to collect identity related information about the users to provide the full functionality of the Privacy-ABC system that shall be tested in the pilot. Personal data is only stored on devices (computers) which are located either at the participant's homes or at the Controller's premises in Söderhamn, Sweden and under her full control.

The processing on behalf of the Controller ends with the termination of the ABC4Trust project. Correlating, the rights and obligations of this agreement's parties are valid during the project runtime. Any personal data of the pilot participants shall be deleted once they are no longer needed to fulfil the duties resulting out of this agreement - at the latest, six months after the end of the project, unless legislation imposed upon this agreement's parties requires the retention of specific data. In that case, the Personal Data Assistant warrants that he will guarantee the confidentiality of this data and that he will not actively process this data anymore.

In the following, we will provide a quick overview and explanation of this pilot. This information is also contained in the information sheet and related consent form handed out to the registered persons in the sense of Section 3 Swedish Personal Data Act, hereinafter named data subjects (appendix 1).



2 Overview and explanation of the school pilot in ABC4Trust

The school pilot will use Privacy-ABCs to enable secure and by minimal data disclosure, privacy-preserving identification in communications between staff, pupils and guardians.

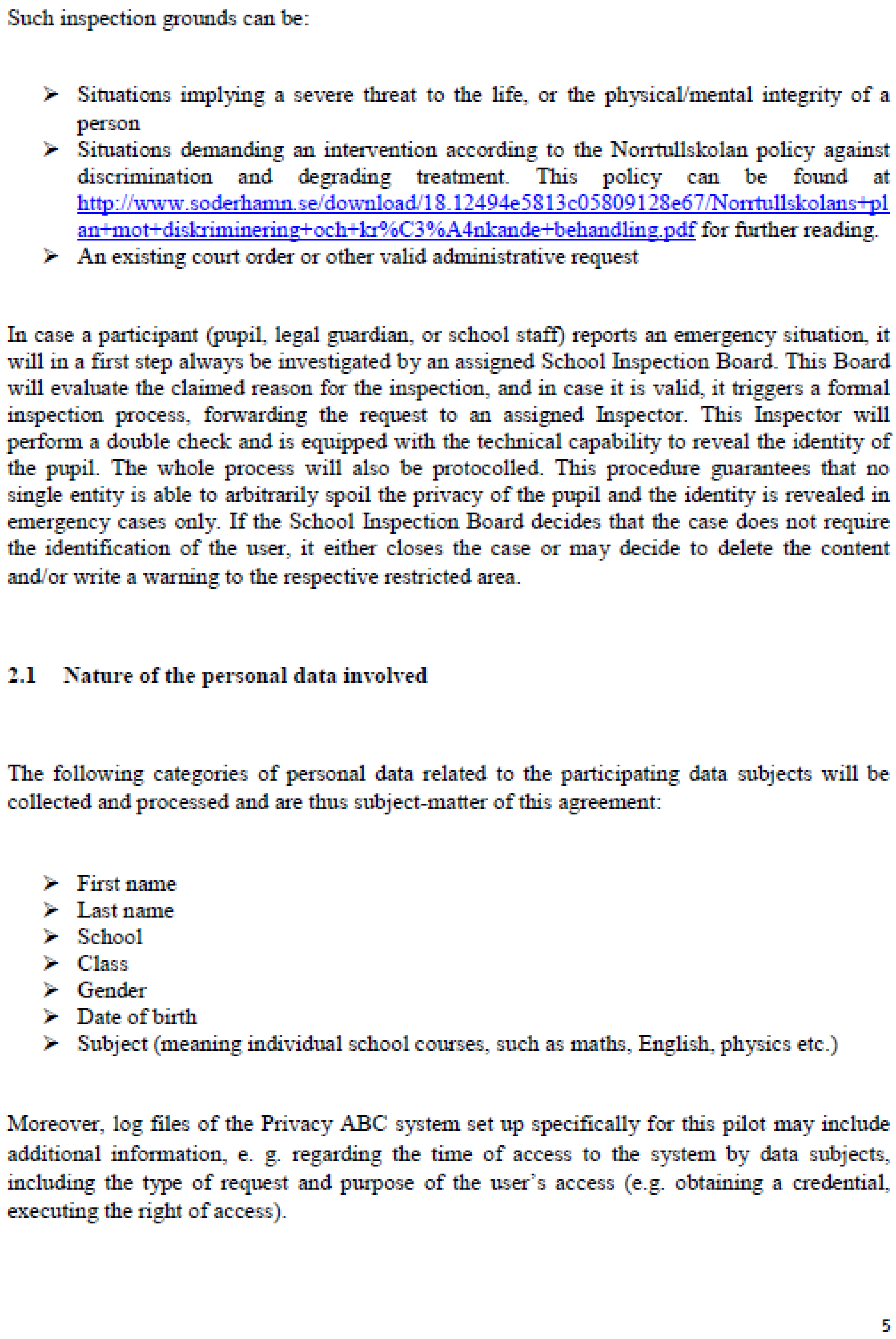
Privacy-ABCs allow the user to only reveal the information absolutely necessary for the execution of the required action, and thus respect the privacy of the individual. This is done by using so-called "credentials", digital tokens which can be used in any kind of online platform to verify certain attributes without revealing other, unrelated and unnecessary information about the user. This means that these tokens can be used e. g. to prove that the owner of the credential is indeed pupil of a particular school, of a particular class, or of a specific age without giving away the individual's name or date of birth.

The first pilot application at the Norrtullskolan will involve privacy-preserving community access and school internal social networking for pupils via a specifically dedicated online platform. Thereby, this pilot addresses the specific challenges posed by the fact that internet users get ever younger and often are minors.

The communication services provided on the online platform entail the following possibilities for the participants:

- ➢ Chat rooms to be used by pupils and/or staff
- ➢ Online forums for discussing lessons and other school related matters as well as political discussions. These may be set up as openly accessible forums or as personal restricted areas where only a predefined group of participants can enter (e. g. children of a certain age or class).
- ➢ Online counselling sessions in restricted areas with health personnel (counsellors, social workers, nurses, coaches), where staff can provide counselling in a safe environment, while pupils are not necessarily required to reveal their identity.
- ➢ Document areas where staff can share documents (e.g. grades and development plans) with pupils and their guardians.
- ➢ Online polls set up by the school staff

Especially in the context of counselling, when not being forced to reveal their identity, pupils may be more willing to talk about the real issues they may face which they would otherwise feel reluctant, shy or scared to talk about. However, to guarantee the physical and mental safety of each participating pupil, the ABC system foresees in those restricted areas for counselling the revelation of the pupil's identity (called inspection) in certain predefined emergency situations (called inspection grounds).



Such inspection grounds can be:

- Situations implying a severe threat to the life, or the physical/mental integrity of a person
- Situations demanding an intervention according to the Norrtullskolan policy against discrimination and degrading treatment. This policy can be found at http://www.soderhamn.se/download/18.12494e5813c05809128e67/Norrtullskolans+plan+mot+diskriminering+och+kr%C3%A4nkande+behandling.pdf for further reading.
- An existing court order or other valid administrative request

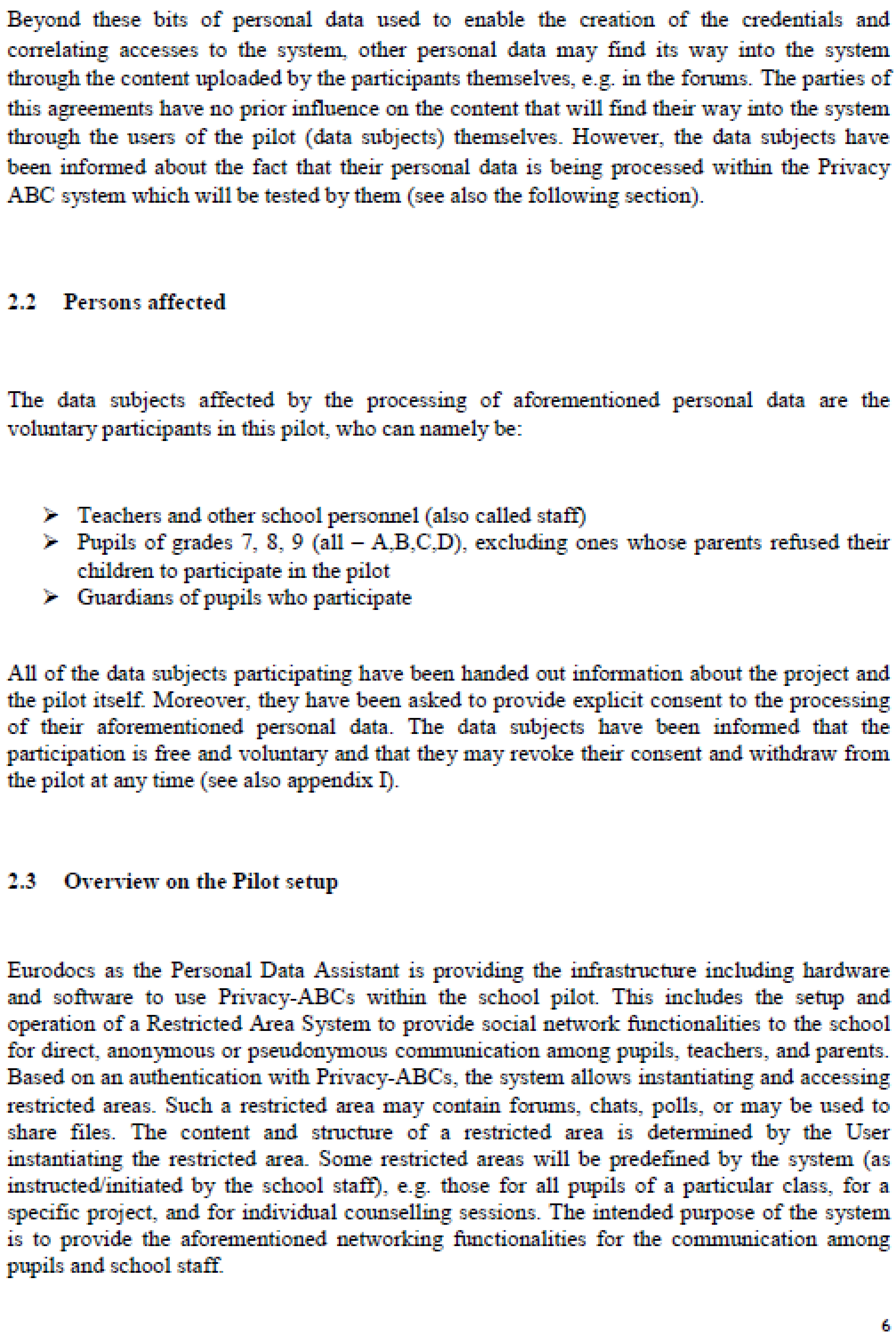
In case a participant (pupil, legal guardian, or school staff) reports an emergency situation, it will in a first step always be investigated by an assigned School Inspection Board. This Board will evaluate the claimed reason for the inspection, and in case it is valid, it triggers a formal inspection process, forwarding the request to an assigned Inspector. This Inspector will perform a double check and is equipped with the technical capability to reveal the identity of the pupil. The whole process will also be protocolled. This procedure guarantees that no single entity is able to arbitrarily spoil the privacy of the pupil and the identity is revealed in emergency cases only. If the School Inspection Board decides that the case does not require the identification of the user, it either closes the case or may decide to delete the content and/or write a warning to the respective restricted area.

2.1 Nature of the personal data involved

The following categories of personal data related to the participating data subjects will be collected and processed and are thus subject-matter of this agreement:

- First name
- Last name
- School
- Class
- Gender
- Date of birth
- Subject (meaning individual school courses, such as maths, English, physics etc.)

Moreover, log files of the Privacy ABC system set up specifically for this pilot may include additional information, e. g. regarding the time of access to the system by data subjects, including the type of request and purpose of the user's access (e.g. obtaining a credential, executing the right of access).



Beyond these bits of personal data used to enable the creation of the credentials and correlating accesses to the system, other personal data may find its way into the system through the content uploaded by the participants themselves, e.g. in the forums. The parties of this agreements have no prior influence on the content that will find their way into the system through the users of the pilot (data subjects) themselves. However, the data subjects have been informed about the fact that their personal data is being processed within the Privacy ABC system which will be tested by them (see also the following section).

2.2 Persons affected

The data subjects affected by the processing of aforementioned personal data are the voluntary participants in this pilot, who can namely be:

- Teachers and other school personnel (also called staff)
- Pupils of grades 7, 8, 9 (all – A,B,C,D), excluding ones whose parents refused their children to participate in the pilot
- Guardians of pupils who participate

All of the data subjects participating have been handed out information about the project and the pilot itself. Moreover, they have been asked to provide explicit consent to the processing of their aforementioned personal data. The data subjects have been informed that the participation is free and voluntary and that they may revoke their consent and withdraw from the pilot at any time (see also appendix I).

2.3 Overview on the Pilot setup

Eurodocs as the Personal Data Assistant is providing the infrastructure including hardware and software to use Privacy-ABCs within the school pilot. This includes the setup and operation of a Restricted Area System to provide social network functionalities to the school for direct, anonymous or pseudonymous communication among pupils, teachers, and parents. Based on an authentication with Privacy-ABCs, the system allows instantiating and accessing restricted areas. Such a restricted area may contain forums, chats, polls, or may be used to share files. The content and structure of a restricted area is determined by the User instantiating the restricted area. Some restricted areas will be predefined by the system (as instructed/initiated by the school staff), e.g. those for all pupils of a particular class, for a specific project, and for individual counselling sessions. The intended purpose of the system is to provide the aforementioned networking functionalities for the communication among pupils and school staff.

2.4 Integration of the data processing in the pilot

The school pilot itself resides on a server, which users can access through thin clients (web browsers). Thereby, users also need a local ABC client installed on their device, to be used together with smart cards and readers handed out to them. The communication channels for the transfer of personal data are shown in the visual depiction of the technical high level architecture of the Söderhamn pilot below (figure 1).

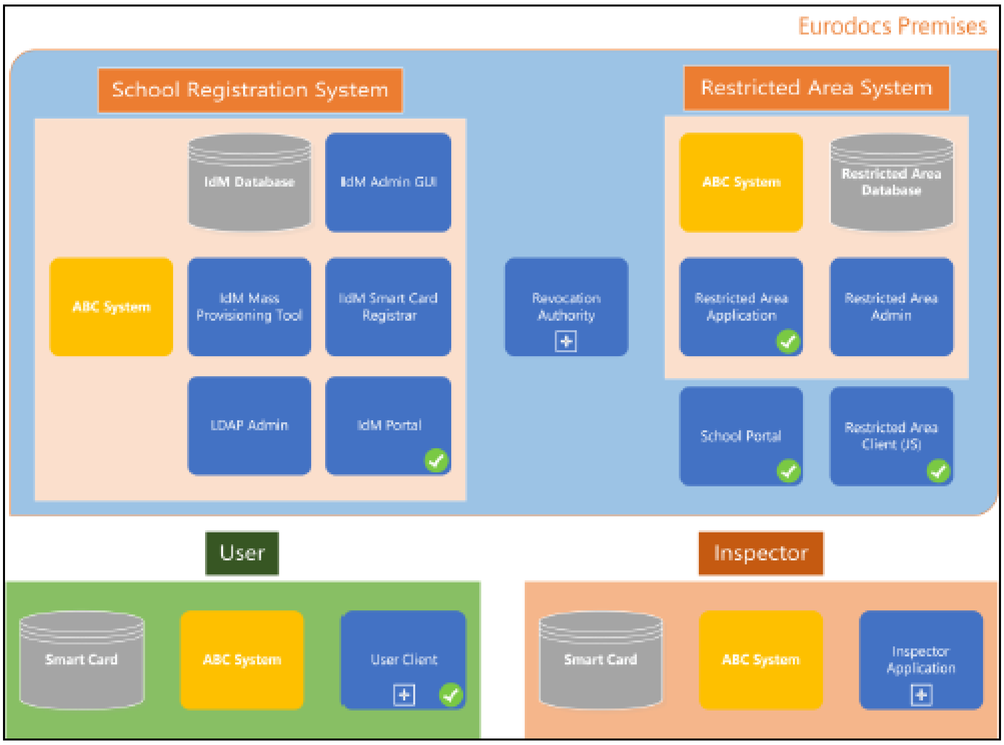
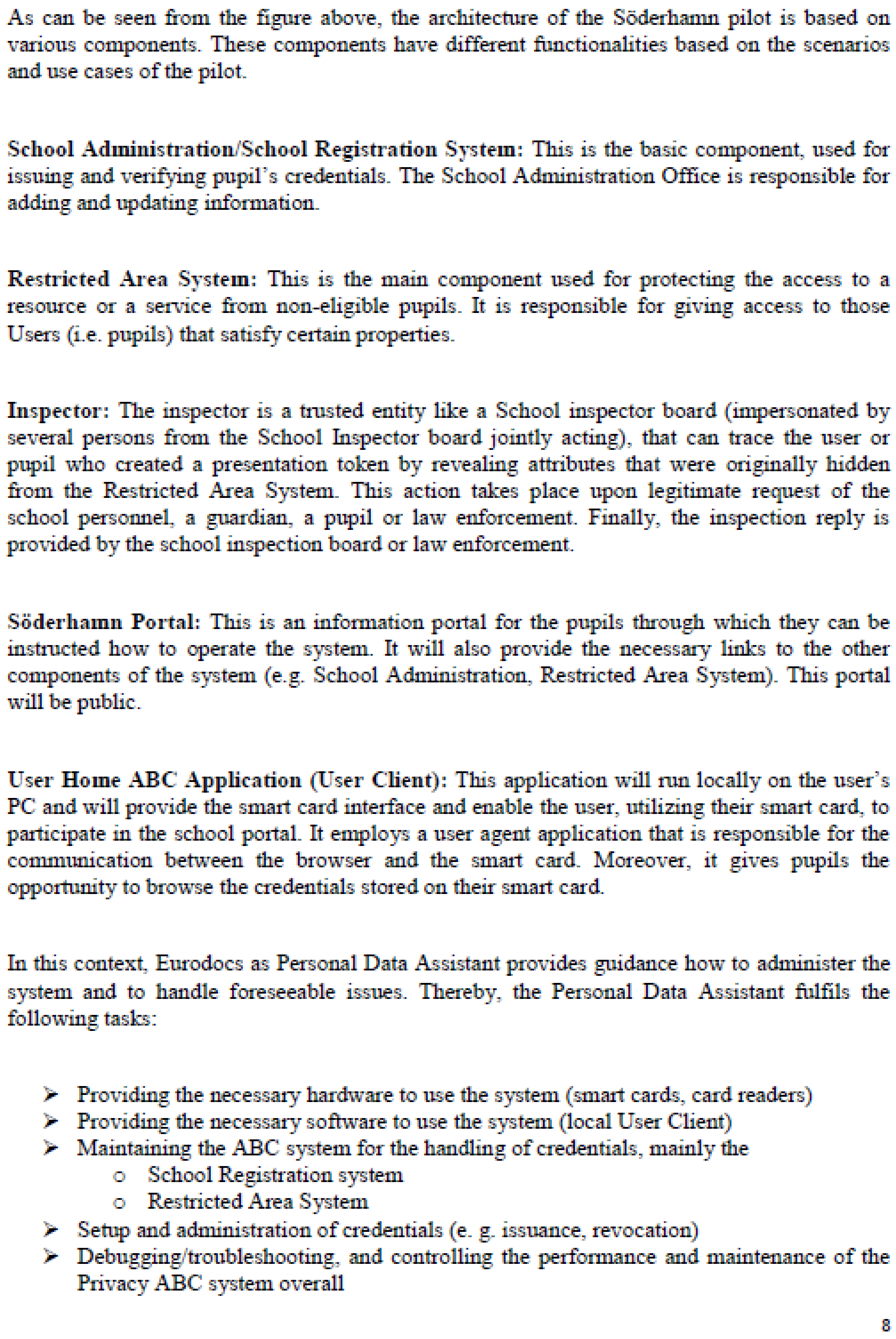


Figure 1: High level architecture of the Söderhamn pilot



As can be seen from the figure above, the architecture of the Söderhamn pilot is based on various components. These components have different functionalities based on the scenarios and use cases of the pilot.

School Administration/School Registration System: This is the basic component, used for issuing and verifying pupil's credentials. The School Administration Office is responsible for adding and updating information.

Restricted Area System: This is the main component used for protecting the access to a resource or a service from non-eligible pupils. It is responsible for giving access to those Users (i.e. pupils) that satisfy certain properties.

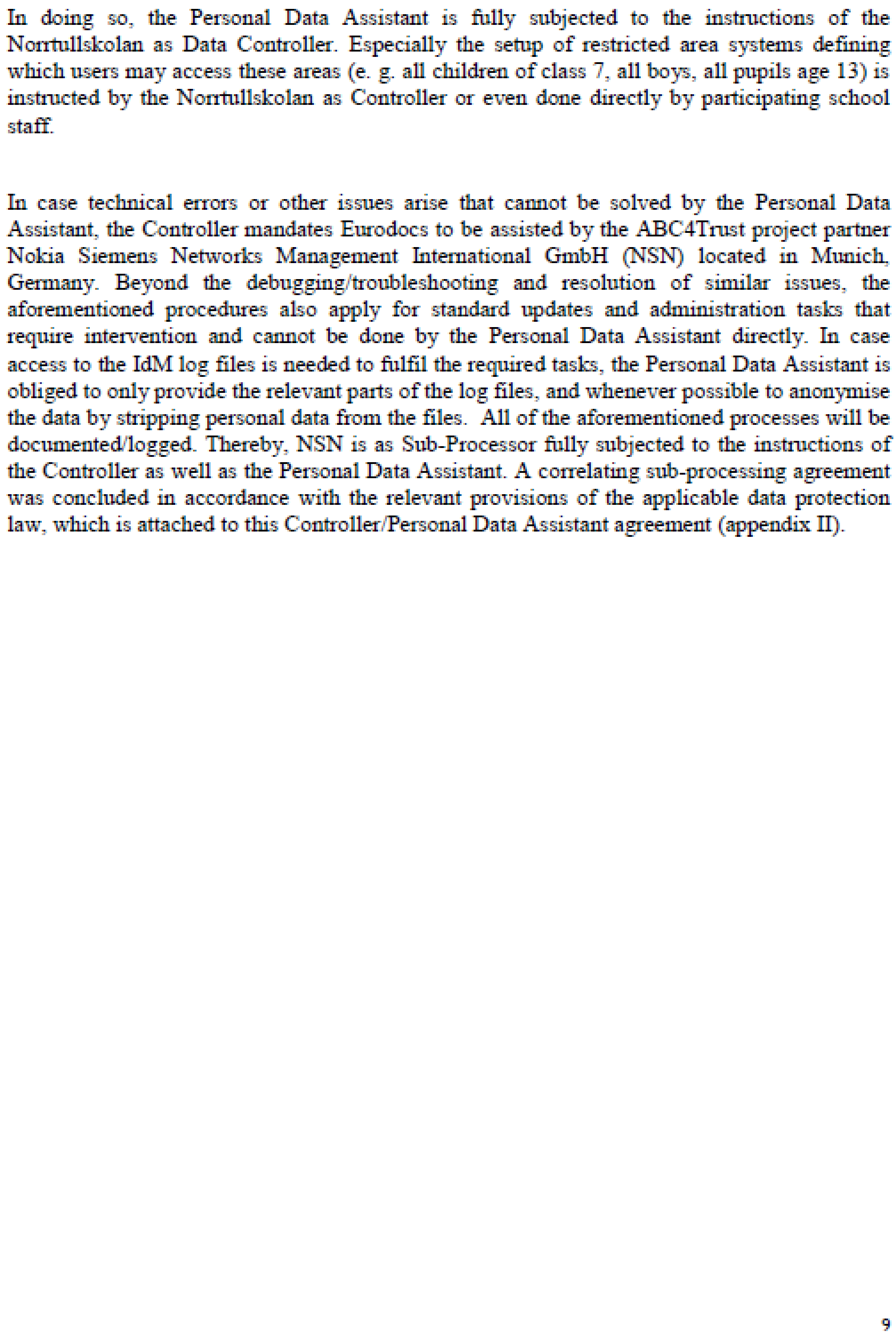
Inspector: The inspector is a trusted entity like a School inspector board (impersonated by several persons from the School Inspector board jointly acting), that can trace the user or pupil who created a presentation token by revealing attributes that were originally hidden from the Restricted Area System. This action takes place upon legitimate request of the school personnel, a guardian, a pupil or law enforcement. Finally, the inspection reply is provided by the school inspection board or law enforcement.

Söderhamn Portal: This is an information portal for the pupils through which they can be instructed how to operate the system. It will also provide the necessary links to the other components of the system (e.g. School Administration, Restricted Area System). This portal will be public.

User Home ABC Application (User Client): This application will run locally on the user's PC and will provide the smart card interface and enable the user, utilizing their smart card, to participate in the school portal. It employs a user agent application that is responsible for the communication between the browser and the smart card. Moreover, it gives pupils the opportunity to browse the credentials stored on their smart card.

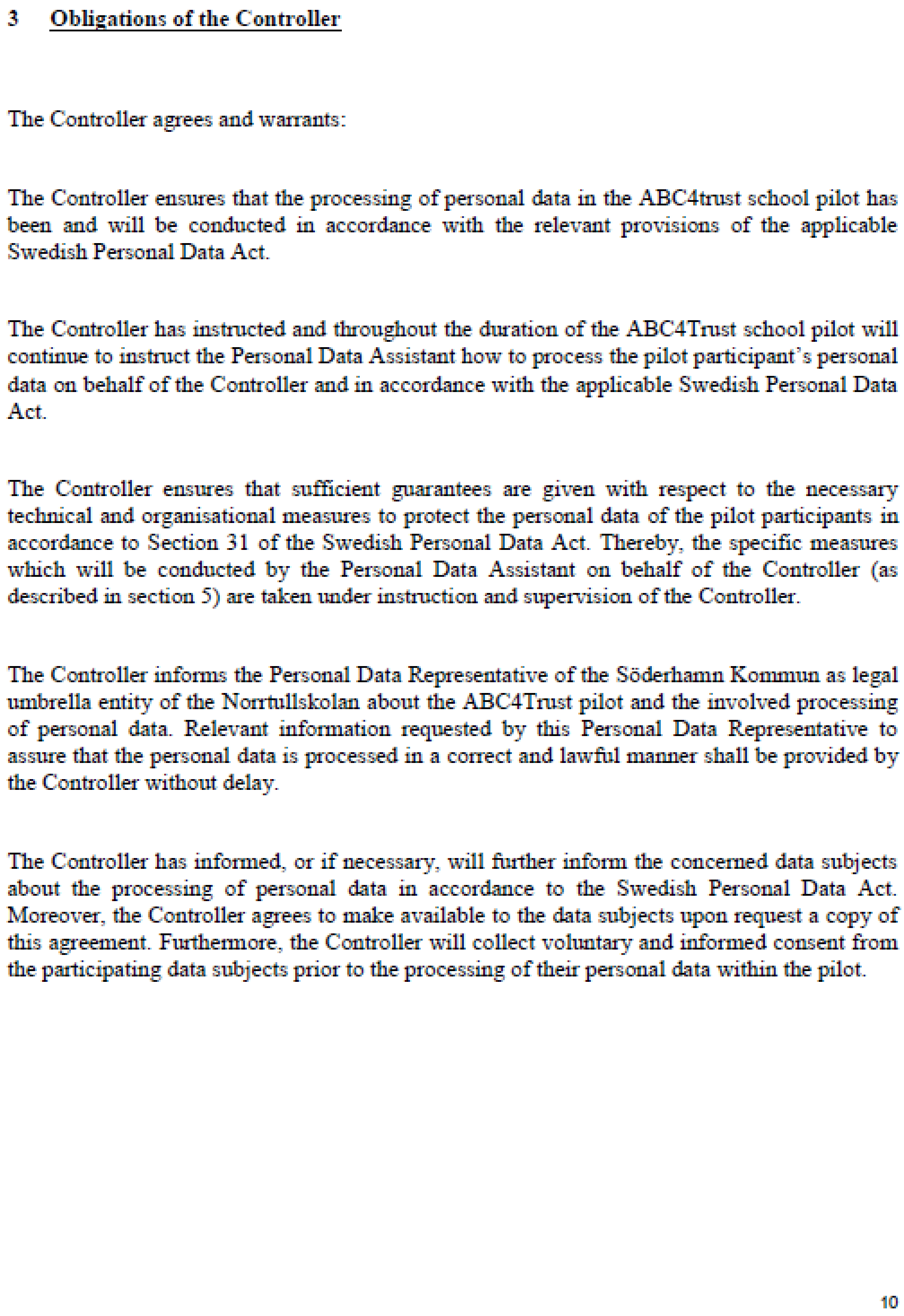
In this context, Eurodocs as Personal Data Assistant provides guidance how to administer the system and to handle foreseeable issues. Thereby, the Personal Data Assistant fulfils the following tasks:

- ➢ Providing the necessary hardware to use the system (smart cards, card readers)
- ➢ Providing the necessary software to use the system (local User Client)
- ➢ Maintaining the ABC system for the handling of credentials, mainly the
  - o School Registration system
  - o Restricted Area System
- ➢ Setup and administration of credentials (e. g. issuance, revocation)
- ➢ Debugging/troubleshooting, and controlling the performance and maintenance of the Privacy ABC system overall



In doing so, the Personal Data Assistant is fully subjected to the instructions of the Norrtullskolan as Data Controller. Especially the setup of restricted area systems defining which users may access these areas (e. g. all children of class 7, all boys, all pupils age 13) is instructed by the Norrtullskolan as Controller or even done directly by participating school staff.

In case technical errors or other issues arise that cannot be solved by the Personal Data Assistant, the Controller mandates Eurodocs to be assisted by the ABC4Trust project partner Nokia Siemens Networks Management International GmbH (NSN) located in Munich, Germany. Beyond the debugging/troubleshooting and resolution of similar issues, the aforementioned procedures also apply for standard updates and administration tasks that require intervention and cannot be done by the Personal Data Assistant directly. In case access to the IdM log files is needed to fulfil the required tasks, the Personal Data Assistant is obliged to only provide the relevant parts of the log files, and whenever possible to anonymise the data by stripping personal data from the files. All of the aforementioned processes will be documented/logged. Thereby, NSN is as Sub-Processor fully subjected to the instructions of the Controller as well as the Personal Data Assistant. A correlating sub-processing agreement was concluded in accordance with the relevant provisions of the applicable data protection law, which is attached to this Controller/Personal Data Assistant agreement (appendix II).



3 Obligations of the Controller

The Controller agrees and warrants:

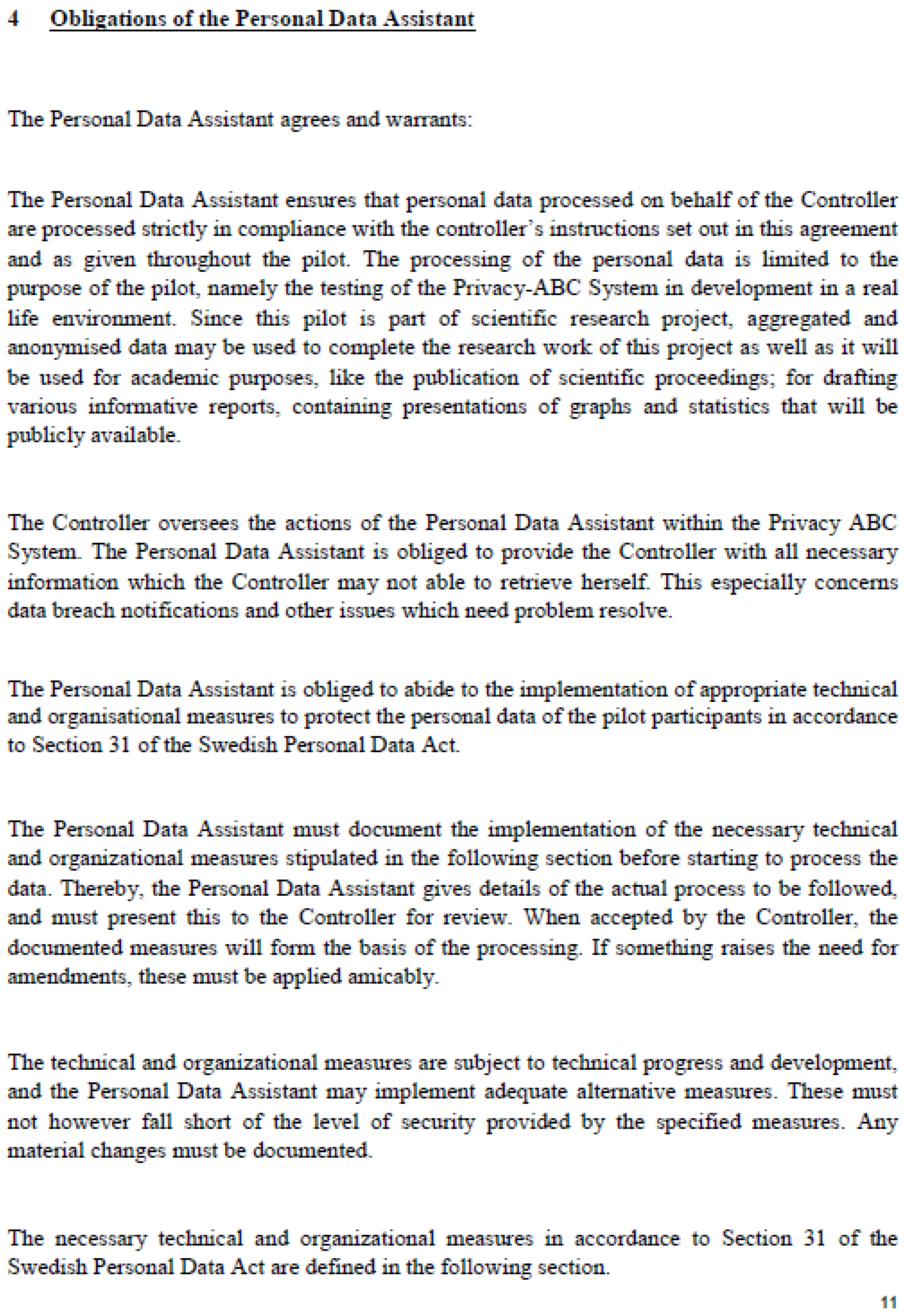
The Controller ensures that the processing of personal data in the ABC4trust school pilot has been and will be conducted in accordance with the relevant provisions of the applicable Swedish Personal Data Act.

The Controller has instructed and throughout the duration of the ABC4Trust school pilot will continue to instruct the Personal Data Assistant how to process the pilot participant's personal data on behalf of the Controller and in accordance with the applicable Swedish Personal Data Act.

The Controller ensures that sufficient guarantees are given with respect to the necessary technical and organisational measures to protect the personal data of the pilot participants in accordance to Section 31 of the Swedish Personal Data Act. Thereby, the specific measures which will be conducted by the Personal Data Assistant on behalf of the Controller (as described in section 5) are taken under instruction and supervision of the Controller.

The Controller informs the Personal Data Representative of the Söderhamn Kommun as legal umbrella entity of the Norrtullskolan about the ABC4Trust pilot and the involved processing of personal data. Relevant information requested by this Personal Data Representative to assure that the personal data is processed in a correct and lawful manner shall be provided by the Controller without delay.

The Controller has informed, or if necessary, will further inform the concerned data subjects about the processing of personal data in accordance to the Swedish Personal Data Act. Moreover, the Controller agrees to make available to the data subjects upon request a copy of this agreement. Furthermore, the Controller will collect voluntary and informed consent from the participating data subjects prior to the processing of their personal data within the pilot.



4 Obligations of the Personal Data Assistant

The Personal Data Assistant agrees and warrants:

The Personal Data Assistant ensures that personal data processed on behalf of the Controller are processed strictly in compliance with the controller's instructions set out in this agreement and as given throughout the pilot. The processing of the personal data is limited to the purpose of the pilot, namely the testing of the Privacy-ABC System in development in a real life environment. Since this pilot is part of scientific research project, aggregated and anonymised data may be used to complete the research work of this project as well as it will be used for academic purposes, like the publication of scientific proceedings; for drafting various informative reports, containing presentations of graphs and statistics that will be publicly available.

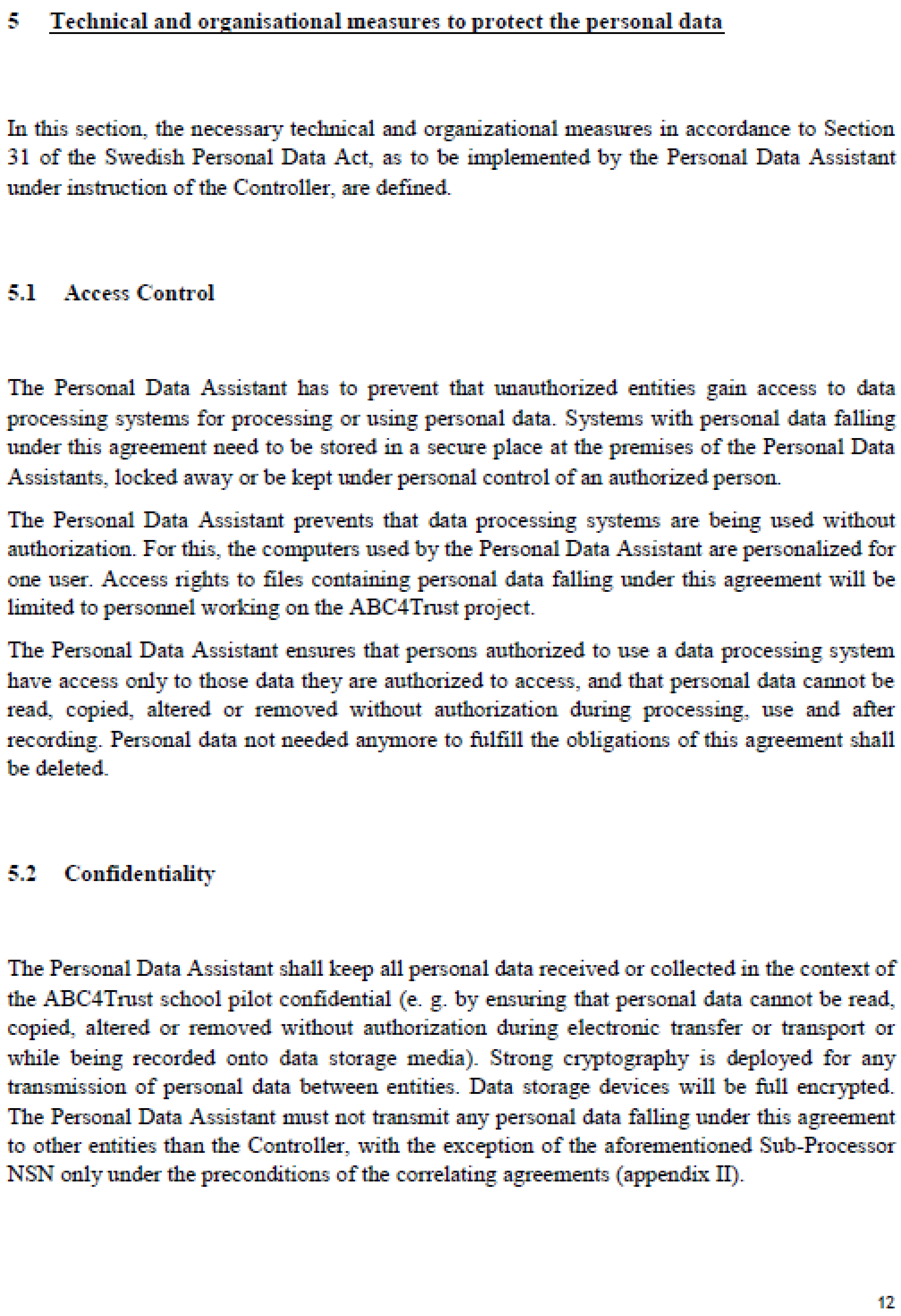
The Controller oversees the actions of the Personal Data Assistant within the Privacy ABC System. The Personal Data Assistant is obliged to provide the Controller with all necessary information which the Controller may not able to retrieve herself. This especially concerns data breach notifications and other issues which need problem resolve.

The Personal Data Assistant is obliged to abide to the implementation of appropriate technical and organisational measures to protect the personal data of the pilot participants in accordance to Section 31 of the Swedish Personal Data Act.

The Personal Data Assistant must document the implementation of the necessary technical and organizational measures stipulated in the following section before starting to process the data. Thereby, the Personal Data Assistant gives details of the actual process to be followed, and must present this to the Controller for review. When accepted by the Controller, the documented measures will form the basis of the processing. If something raises the need for amendments, these must be applied amicably.

The technical and organizational measures are subject to technical progress and development, and the Personal Data Assistant may implement adequate alternative measures. These must not however fall short of the level of security provided by the specified measures. Any material changes must be documented.

The necessary technical and organizational measures in accordance to Section 31 of the Swedish Personal Data Act are defined in the following section.



5 Technical and organisational measures to protect the personal data

In this section, the necessary technical and organizational measures in accordance to Section 31 of the Swedish Personal Data Act, as to be implemented by the Personal Data Assistant under instruction of the Controller, are defined.

5.1 Access Control

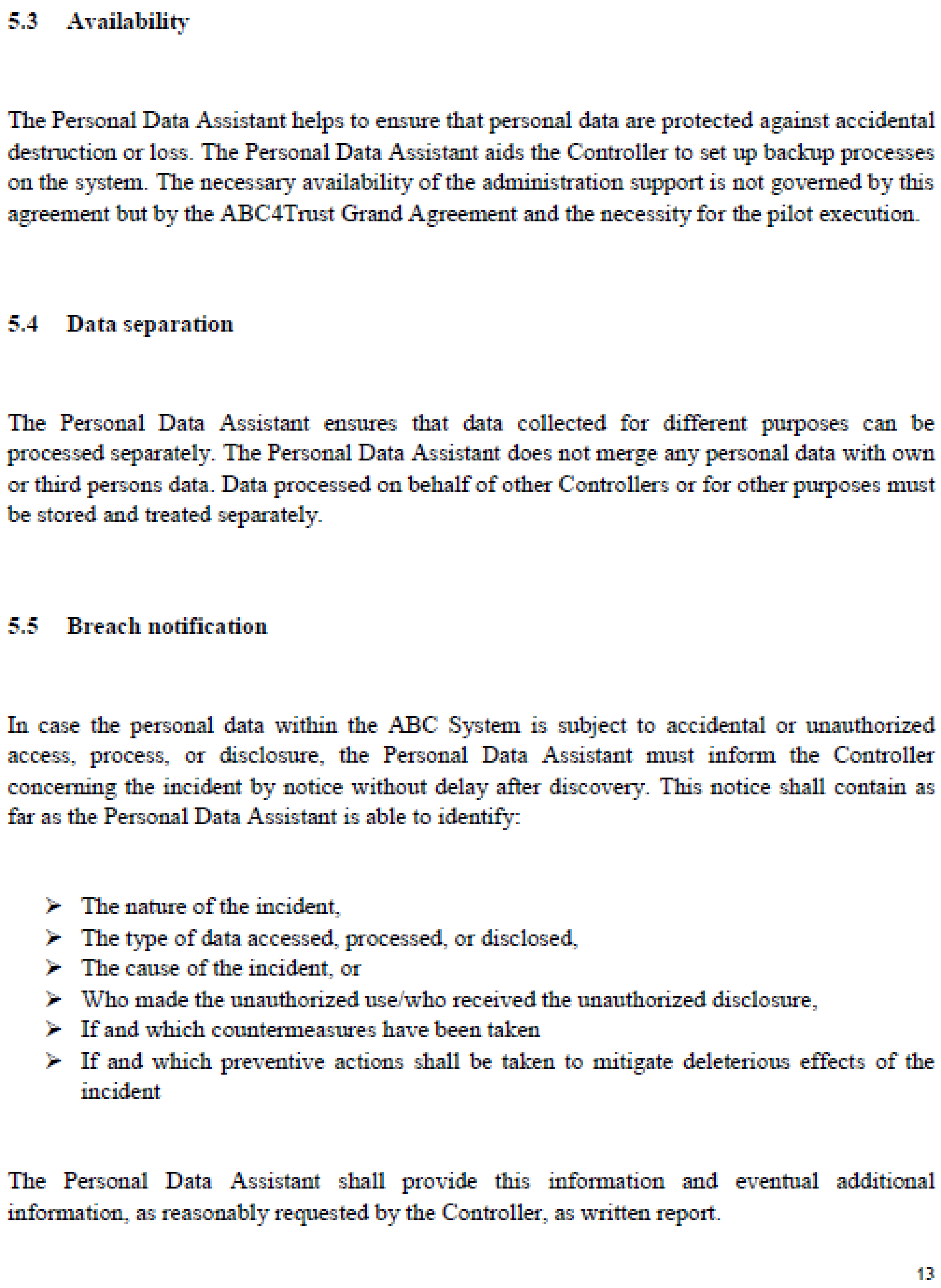
The Personal Data Assistant has to prevent that unauthorized entities gain access to data processing systems for processing or using personal data. Systems with personal data falling under this agreement need to be stored in a secure place at the premises of the Personal Data Assistants, locked away or be kept under personal control of an authorized person.

The Personal Data Assistant prevents that data processing systems are being used without authorization. For this, the computers used by the Personal Data Assistant are personalized for one user. Access rights to files containing personal data falling under this agreement will be limited to personnel working on the ABC4Trust project.

The Personal Data Assistant ensures that persons authorized to use a data processing system have access only to those data they are authorized to access, and that personal data cannot be read, copied, altered or removed without authorization during processing, use and after recording. Personal data not needed anymore to fulfill the obligations of this agreement shall be deleted.

5.2 Confidentiality

The Personal Data Assistant shall keep all personal data received or collected in the context of the ABC4Trust school pilot confidential (e. g. by ensuring that personal data cannot be read, copied, altered or removed without authorization during electronic transfer or transport or while being recorded onto data storage media). Strong cryptography is deployed for any transmission of personal data between entities. Data storage devices will be full encrypted. The Personal Data Assistant must not transmit any personal data falling under this agreement to other entities than the Controller, with the exception of the aforementioned Sub-Processor NSN only under the preconditions of the correlating agreements (appendix II).



5.3 Availability

The Personal Data Assistant helps to ensure that personal data are protected against accidental destruction or loss. The Personal Data Assistant aids the Controller to set up backup processes on the system. The necessary availability of the administration support is not governed by this agreement but by the ABC4Trust Grand Agreement and the necessity for the pilot execution.

5.4 Data separation

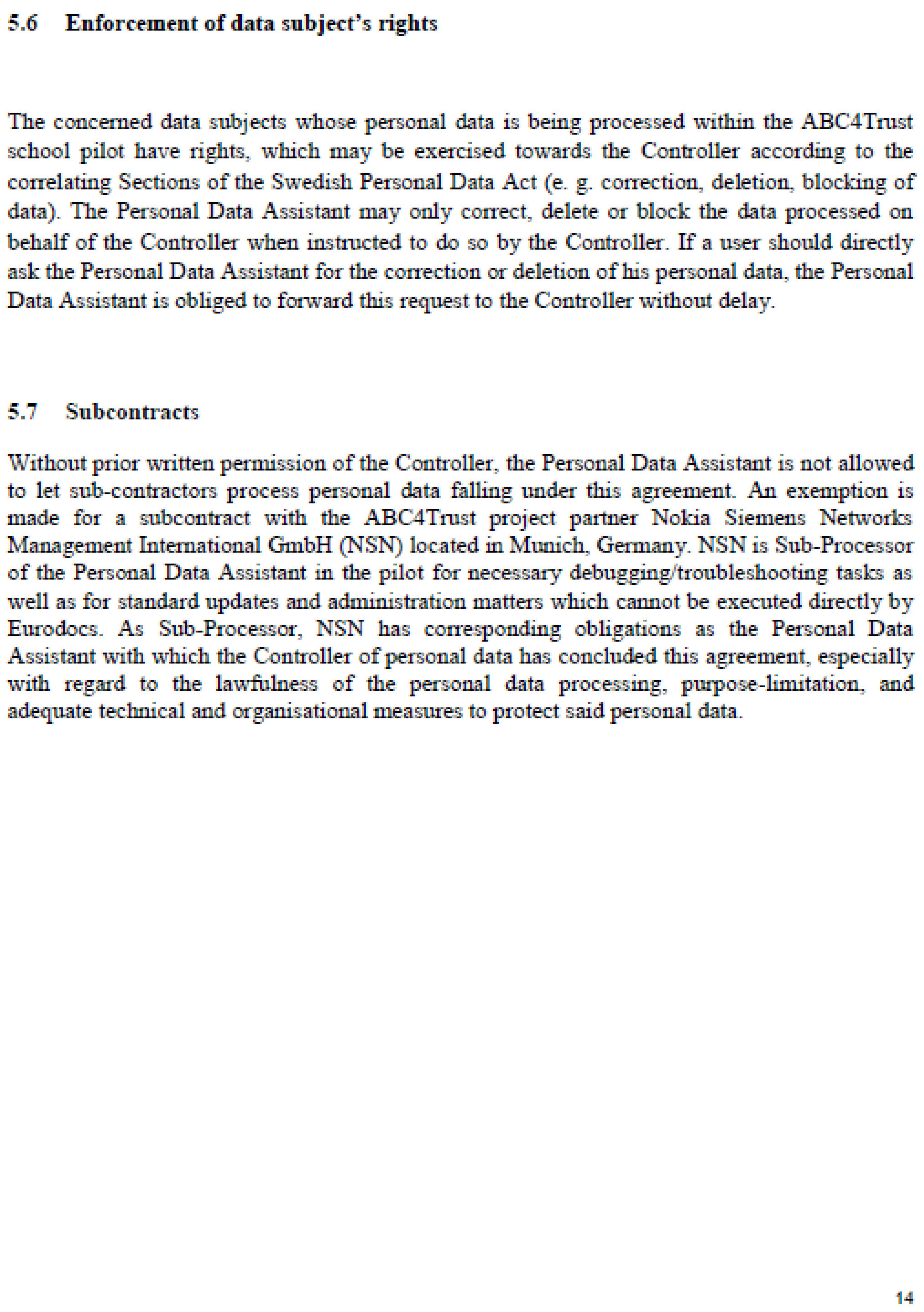
The Personal Data Assistant ensures that data collected for different purposes can be processed separately. The Personal Data Assistant does not merge any personal data with own or third persons data. Data processed on behalf of other Controllers or for other purposes must be stored and treated separately.

5.5 Breach notification

In case the personal data within the ABC System is subject to accidental or unauthorized access, process, or disclosure, the Personal Data Assistant must inform the Controller concerning the incident by notice without delay after discovery. This notice shall contain as far as the Personal Data Assistant is able to identify:

- The nature of the incident,
- The type of data accessed, processed, or disclosed,
- The cause of the incident, or
- Who made the unauthorized use/who received the unauthorized disclosure,
- If and which countermeasures have been taken
- If and which preventive actions shall be taken to mitigate deleterious effects of the incident

The Personal Data Assistant shall provide this information and eventual additional information, as reasonably requested by the Controller, as written report.

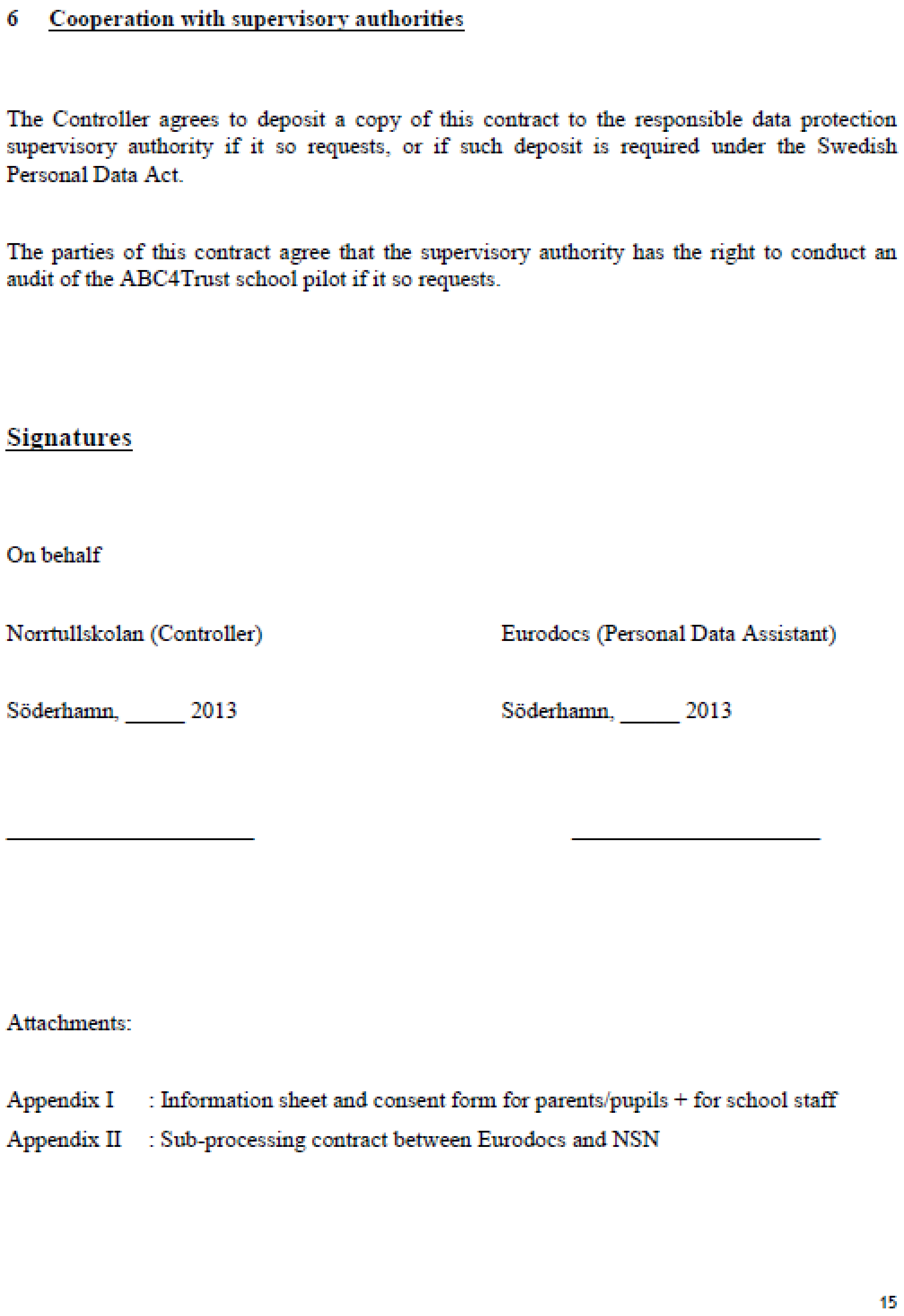


5.6 Enforcement of data subject's rights

The concerned data subjects whose personal data is being processed within the ABC4Trust school pilot have rights, which may be exercised towards the Controller according to the correlating Sections of the Swedish Personal Data Act (e. g. correction, deletion, blocking of data). The Personal Data Assistant may only correct, delete or block the data processed on behalf of the Controller when instructed to do so by the Controller. If a user should directly ask the Personal Data Assistant for the correction or deletion of his personal data, the Personal Data Assistant is obliged to forward this request to the Controller without delay.

5.7 Subcontracts

Without prior written permission of the Controller, the Personal Data Assistant is not allowed to let sub-contractors process personal data falling under this agreement. An exemption is made for a subcontract with the ABC4Trust project partner Nokia Siemens Networks Management International GmbH (NSN) located in Munich, Germany. NSN is Sub-Processor of the Personal Data Assistant in the pilot for necessary debugging/troubleshooting tasks as well as for standard updates and administration matters which cannot be executed directly by Eurodocs. As Sub-Processor, NSN has corresponding obligations as the Personal Data Assistant with which the Controller of personal data has concluded this agreement, especially with regard to the lawfulness of the personal data processing, purpose-limitation, and adequate technical and organisational measures to protect said personal data.



6 Cooperation with supervisory authorities

The Controller agrees to deposit a copy of this contract to the responsible data protection supervisory authority if it so requests, or if such deposit is required under the Swedish Personal Data Act.

The parties of this contract agree that the supervisory authority has the right to conduct an audit of the ABC4Trust school pilot if it so requests.

Signatures

On behalf

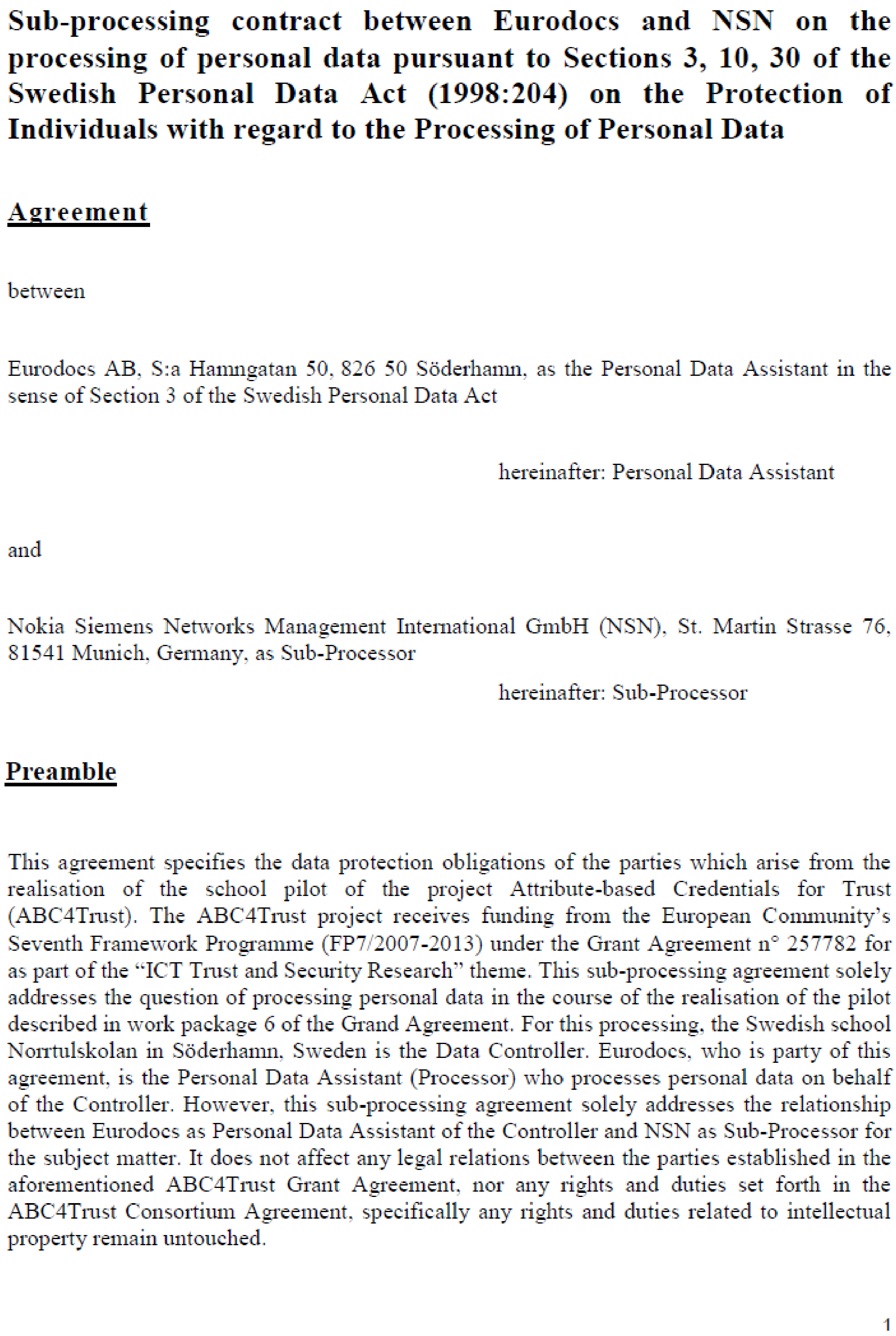
Norrtullskolan (Controller)	Eurodocs (Personal Data Assistant)
Söderhamn, _____ 2013	Söderhamn, _____ 2013
______________________	______________________

Attachments:

Appendix I : Information sheet and consent form for parents/pupils + for school staff

Appendix II : Sub-processing contract between Eurodocs and NSN

A.2 Sub-Processing Contract of Söderhamn Pilot



Sub-processing contract between Eurodocs and NSN on the processing of personal data pursuant to Sections 3, 10, 30 of the Swedish Personal Data Act (1998:204) on the Protection of Individuals with regard to the Processing of Personal Data

Agreement

between

Eurodocs AB, S:a Hamngatan 50, 826 50 Söderhamn, as the Personal Data Assistant in the sense of Section 3 of the Swedish Personal Data Act

hereinafter: Personal Data Assistant

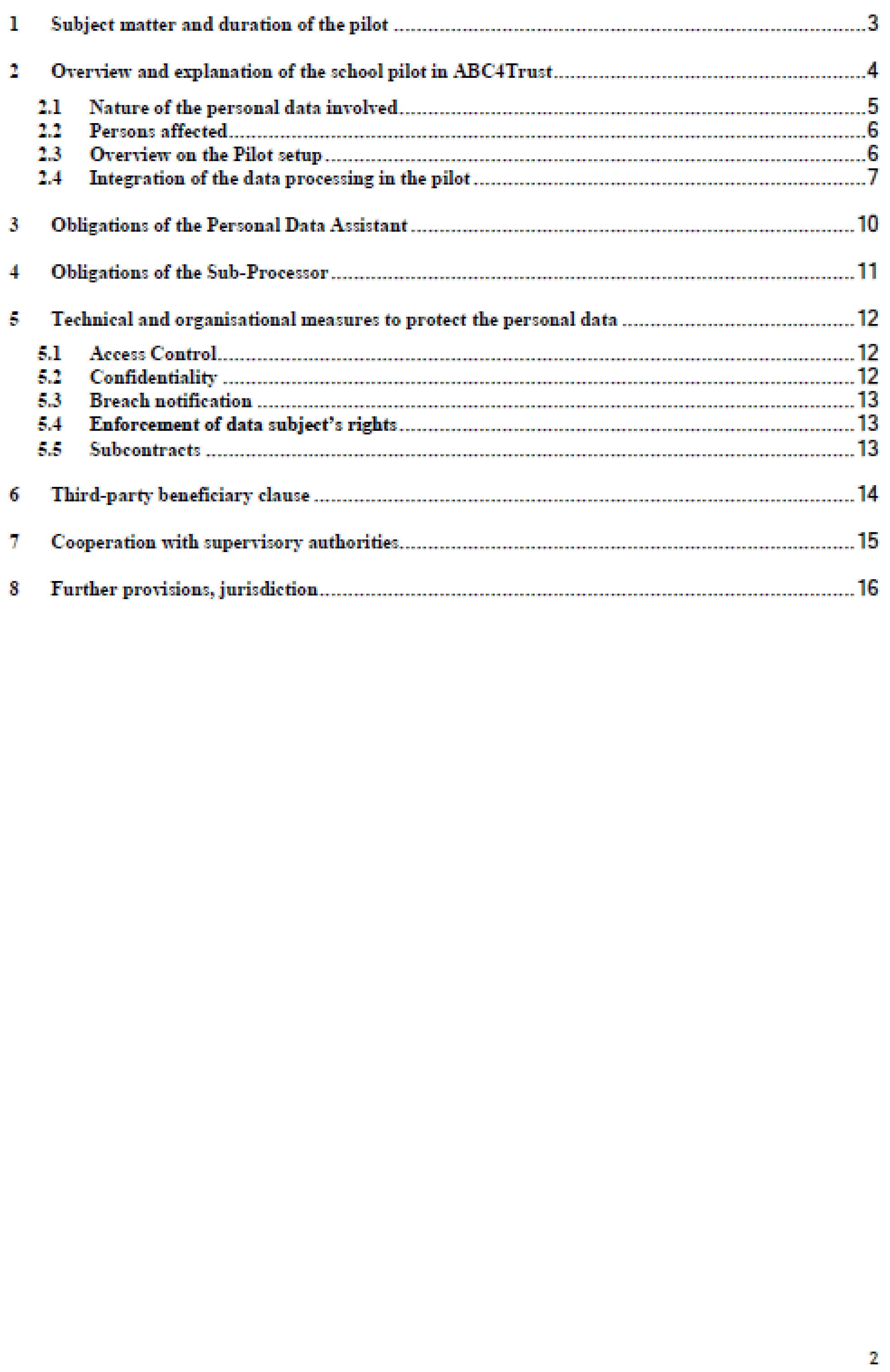
and

Nokia Siemens Networks Management International GmbH (NSN), St. Martin Strasse 76, 81541 Munich, Germany, as Sub-Processor

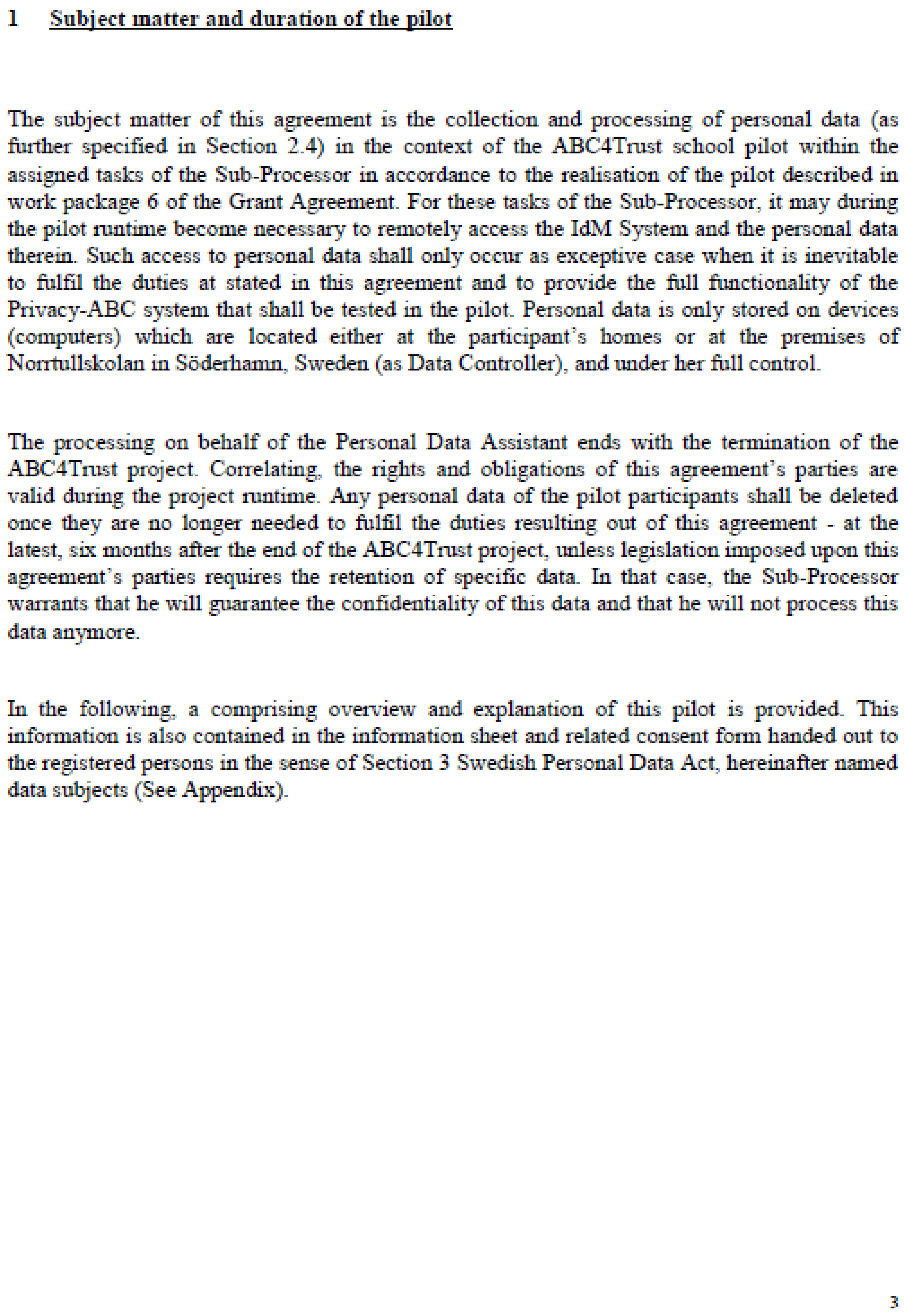
hereinafter: Sub-Processor

Preamble

This agreement specifies the data protection obligations of the parties which arise from the realisation of the school pilot of the project Attribute-based Credentials for Trust (ABC4Trust). The ABC4Trust project receives funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under the Grant Agreement n° 257782 for as part of the "ICT Trust and Security Research" theme. This sub-processing agreement solely addresses the question of processing personal data in the course of the realisation of the pilot described in work package 6 of the Grand Agreement. For this processing, the Swedish school Norrtulskolan in Söderhamn, Sweden is the Data Controller. Eurodocs, who is party of this agreement, is the Personal Data Assistant (Processor) who processes personal data on behalf of the Controller. However, this sub-processing agreement solely addresses the relationship between Eurodocs as Personal Data Assistant of the Controller and NSN as Sub-Processor for the subject matter. It does not affect any legal relations between the parties established in the aforementioned ABC4Trust Grant Agreement, nor any rights and duties set forth in the ABC4Trust Consortium Agreement, specifically any rights and duties related to intellectual property remain untouched.



2

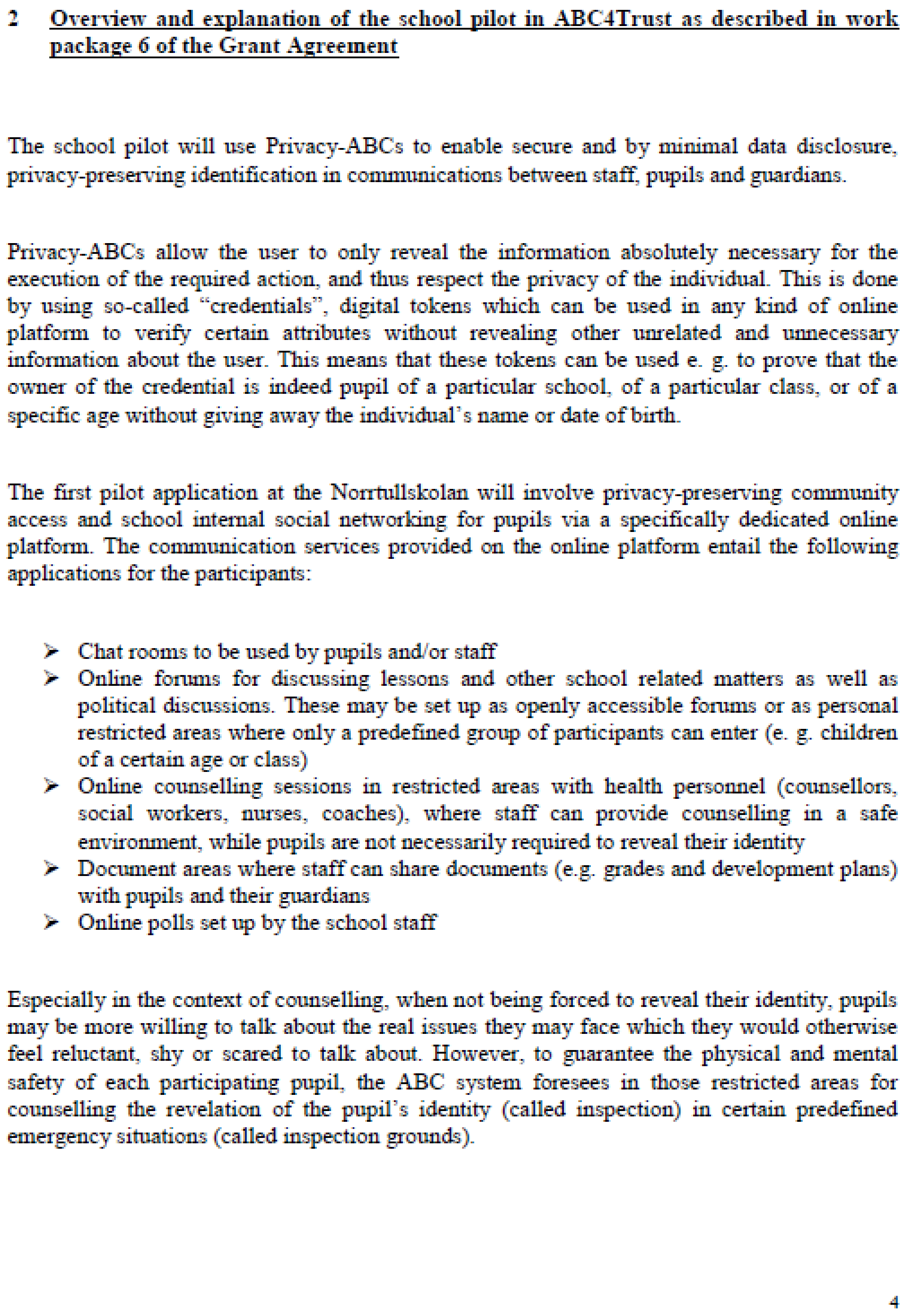


1 Subject matter and duration of the pilot

The subject matter of this agreement is the collection and processing of personal data (as further specified in Section 2.4) in the context of the ABC4Trust school pilot within the assigned tasks of the Sub-Processor in accordance to the realisation of the pilot described in work package 6 of the Grant Agreement. For these tasks of the Sub-Processor, it may during the pilot runtime become necessary to remotely access the IdM System and the personal data therein. Such access to personal data shall only occur as exceptive case when it is inevitable to fulfil the duties at stated in this agreement and to provide the full functionality of the Privacy-ABC system that shall be tested in the pilot. Personal data is only stored on devices (computers) which are located either at the participant's homes or at the premises of Norrtullskolan in Söderhamn, Sweden (as Data Controller), and under her full control.

The processing on behalf of the Personal Data Assistant ends with the termination of the ABC4Trust project. Correlating, the rights and obligations of this agreement's parties are valid during the project runtime. Any personal data of the pilot participants shall be deleted once they are no longer needed to fulfil the duties resulting out of this agreement - at the latest, six months after the end of the ABC4Trust project, unless legislation imposed upon this agreement's parties requires the retention of specific data. In that case, the Sub-Processor warrants that he will guarantee the confidentiality of this data and that he will not process this data anymore.

In the following, a comprising overview and explanation of this pilot is provided. This information is also contained in the information sheet and related consent form handed out to the registered persons in the sense of Section 3 Swedish Personal Data Act, hereinafter named data subjects (See Appendix).



2 Overview and explanation of the school pilot in ABC4Trust as described in work package 6 of the Grant Agreement

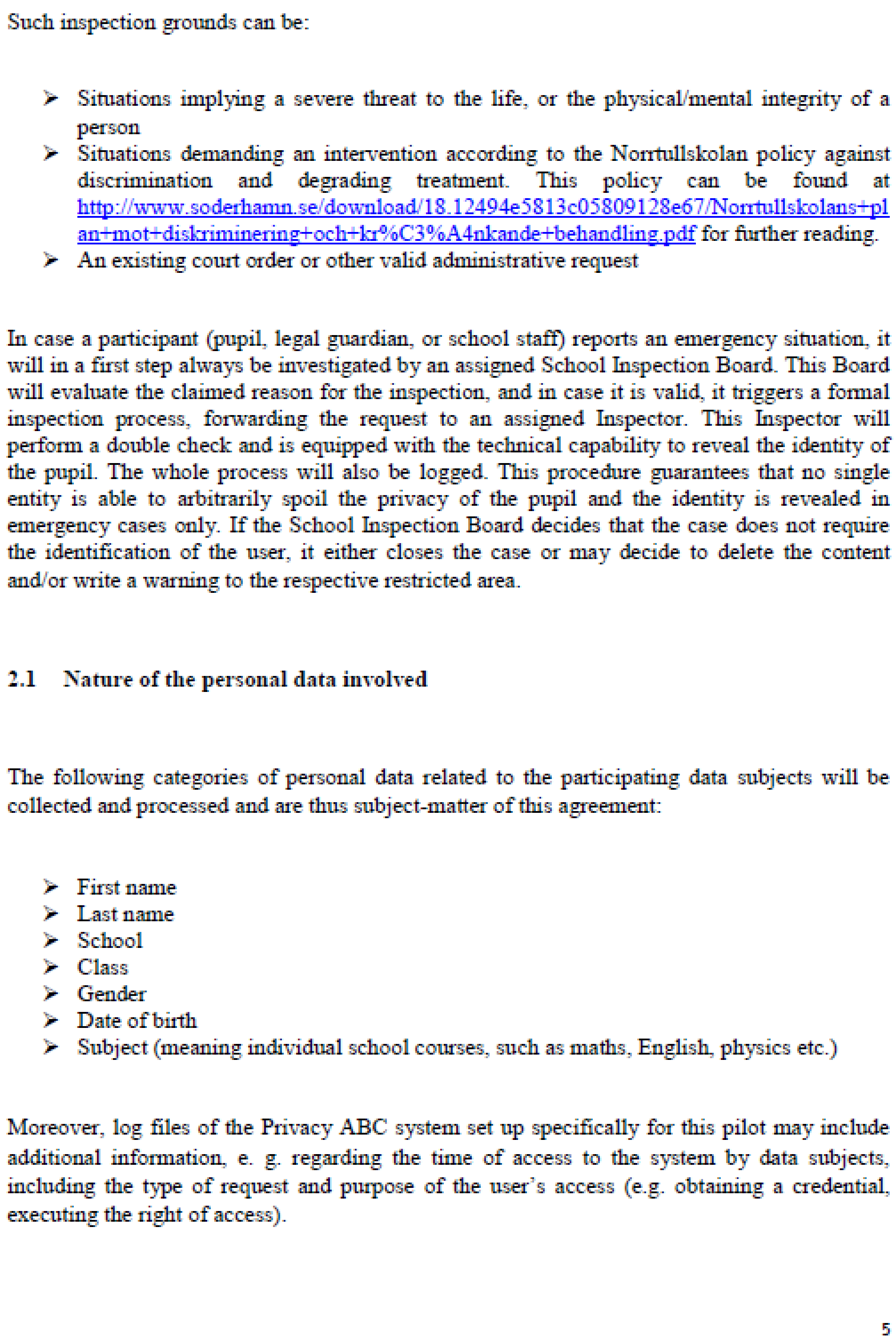
The school pilot will use Privacy-ABCs to enable secure and by minimal data disclosure, privacy-preserving identification in communications between staff, pupils and guardians.

Privacy-ABCs allow the user to only reveal the information absolutely necessary for the execution of the required action, and thus respect the privacy of the individual. This is done by using so-called “credentials”, digital tokens which can be used in any kind of online platform to verify certain attributes without revealing other unrelated and unnecessary information about the user. This means that these tokens can be used e. g. to prove that the owner of the credential is indeed pupil of a particular school, of a particular class, or of a specific age without giving away the individual’s name or date of birth.

The first pilot application at the Norrtullskolan will involve privacy-preserving community access and school internal social networking for pupils via a specifically dedicated online platform. The communication services provided on the online platform entail the following applications for the participants:

- Chat rooms to be used by pupils and/or staff
- Online forums for discussing lessons and other school related matters as well as political discussions. These may be set up as openly accessible forums or as personal restricted areas where only a predefined group of participants can enter (e. g. children of a certain age or class)
- Online counselling sessions in restricted areas with health personnel (counsellors, social workers, nurses, coaches), where staff can provide counselling in a safe environment, while pupils are not necessarily required to reveal their identity
- Document areas where staff can share documents (e.g. grades and development plans) with pupils and their guardians
- Online polls set up by the school staff

Especially in the context of counselling, when not being forced to reveal their identity, pupils may be more willing to talk about the real issues they may face which they would otherwise feel reluctant, shy or scared to talk about. However, to guarantee the physical and mental safety of each participating pupil, the ABC system foresees in those restricted areas for counselling the revelation of the pupil’s identity (called inspection) in certain predefined emergency situations (called inspection grounds).



Such inspection grounds can be:

- Situations implying a severe threat to the life, or the physical/mental integrity of a person
- Situations demanding an intervention according to the Norrtullskolan policy against discrimination and degrading treatment. This policy can be found at http://www.soderhamn.se/download/18.12494e5813c05809128e67/Norrtullskolans+plan+mot+diskriminering+och+kr%C3%A4nkande+behandling.pdf for further reading.
- An existing court order or other valid administrative request

In case a participant (pupil, legal guardian, or school staff) reports an emergency situation, it will in a first step always be investigated by an assigned School Inspection Board. This Board will evaluate the claimed reason for the inspection, and in case it is valid, it triggers a formal inspection process, forwarding the request to an assigned Inspector. This Inspector will perform a double check and is equipped with the technical capability to reveal the identity of the pupil. The whole process will also be logged. This procedure guarantees that no single entity is able to arbitrarily spoil the privacy of the pupil and the identity is revealed in emergency cases only. If the School Inspection Board decides that the case does not require the identification of the user, it either closes the case or may decide to delete the content and/or write a warning to the respective restricted area.

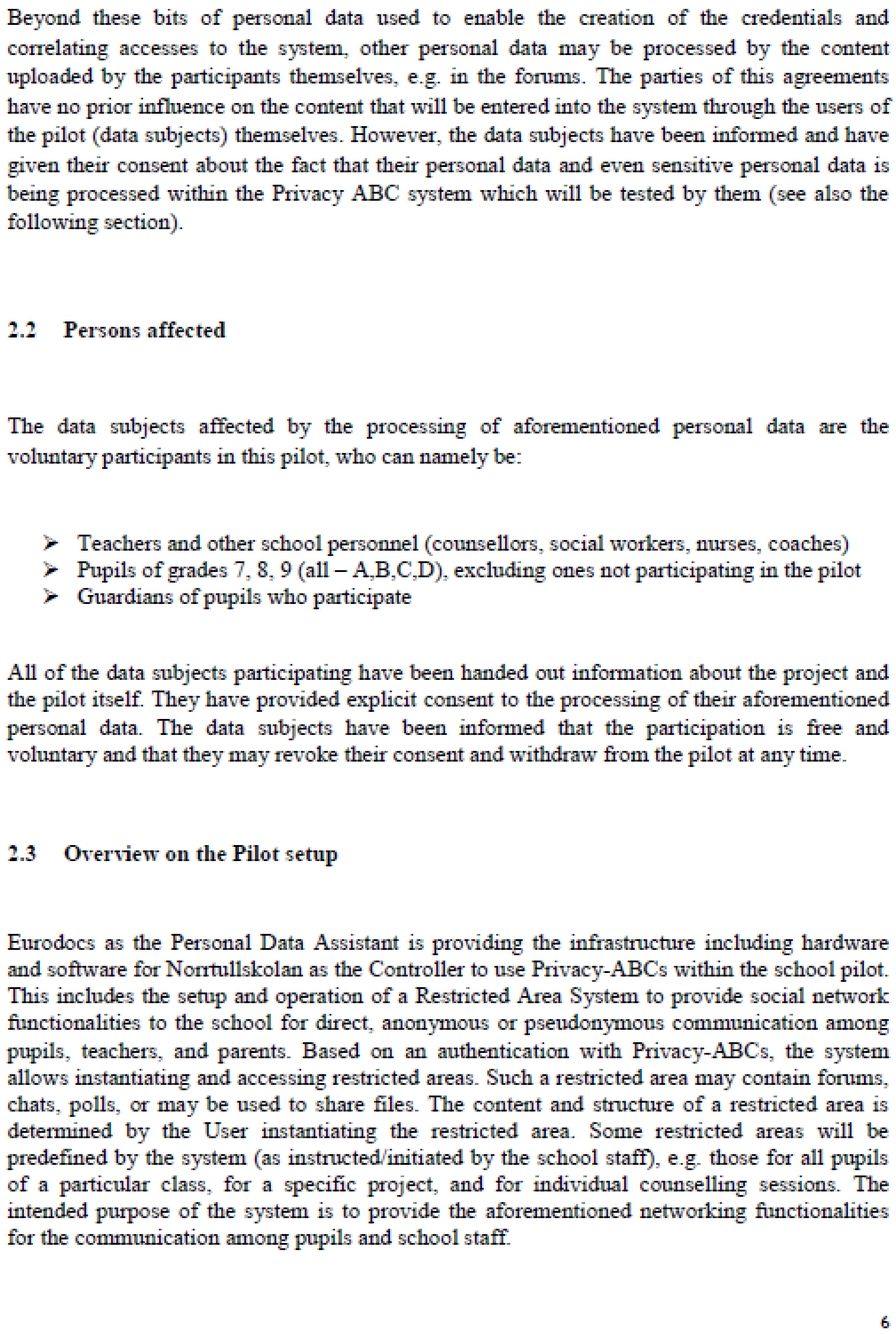
2.1 Nature of the personal data involved

The following categories of personal data related to the participating data subjects will be collected and processed and are thus subject-matter of this agreement:

- First name
- Last name
- School
- Class
- Gender
- Date of birth
- Subject (meaning individual school courses, such as maths, English, physics etc.)

Moreover, log files of the Privacy ABC system set up specifically for this pilot may include additional information, e. g. regarding the time of access to the system by data subjects, including the type of request and purpose of the user's access (e.g. obtaining a credential, executing the right of access).

5



Beyond these bits of personal data used to enable the creation of the credentials and correlating accesses to the system, other personal data may be processed by the content uploaded by the participants themselves, e.g. in the forums. The parties of this agreements have no prior influence on the content that will be entered into the system through the users of the pilot (data subjects) themselves. However, the data subjects have been informed and have given their consent about the fact that their personal data and even sensitive personal data is being processed within the Privacy ABC system which will be tested by them (see also the following section).

2.2 Persons affected

The data subjects affected by the processing of aforementioned personal data are the voluntary participants in this pilot, who can namely be:

- Teachers and other school personnel (counsellors, social workers, nurses, coaches)
- Pupils of grades 7, 8, 9 (all – A,B,C,D), excluding ones not participating in the pilot
- Guardians of pupils who participate

All of the data subjects participating have been handed out information about the project and the pilot itself. They have provided explicit consent to the processing of their aforementioned personal data. The data subjects have been informed that the participation is free and voluntary and that they may revoke their consent and withdraw from the pilot at any time.

2.3 Overview on the Pilot setup

Eurodocs as the Personal Data Assistant is providing the infrastructure including hardware and software for Norrtullskolan as the Controller to use Privacy-ABCs within the school pilot. This includes the setup and operation of a Restricted Area System to provide social network functionalities to the school for direct, anonymous or pseudonymous communication among pupils, teachers, and parents. Based on an authentication with Privacy-ABCs, the system allows instantiating and accessing restricted areas. Such a restricted area may contain forums, chats, polls, or may be used to share files. The content and structure of a restricted area is determined by the User instantiating the restricted area. Some restricted areas will be predefined by the system (as instructed/initiated by the school staff), e.g. those for all pupils of a particular class, for a specific project, and for individual counselling sessions. The intended purpose of the system is to provide the aforementioned networking functionalities for the communication among pupils and school staff.

2.4 Integration of the data processing in the pilot

The school pilot itself resides on a server in Sweden, which Users can access through thin clients (web browsers), Thereby, Users also need a local ABC client installed on their device, to be used together with smart cards and readers handed out to them. The communication channels for the transfer of personal data are shown in the visual depiction of the technical high level architecture of the Söderhamn pilot below (figure 1).

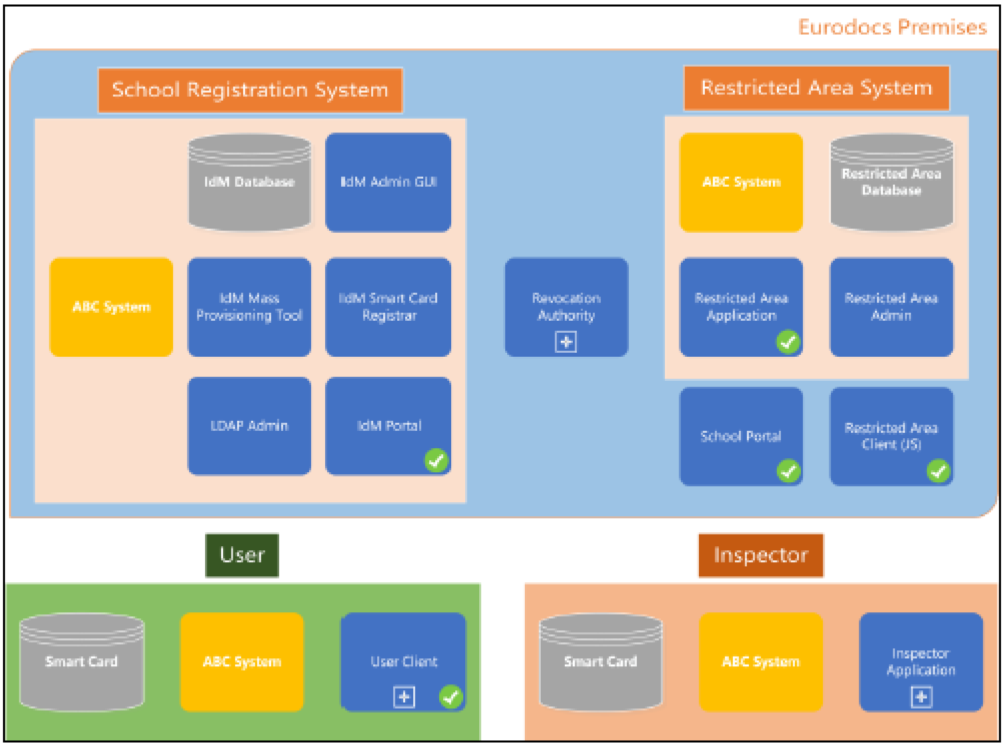
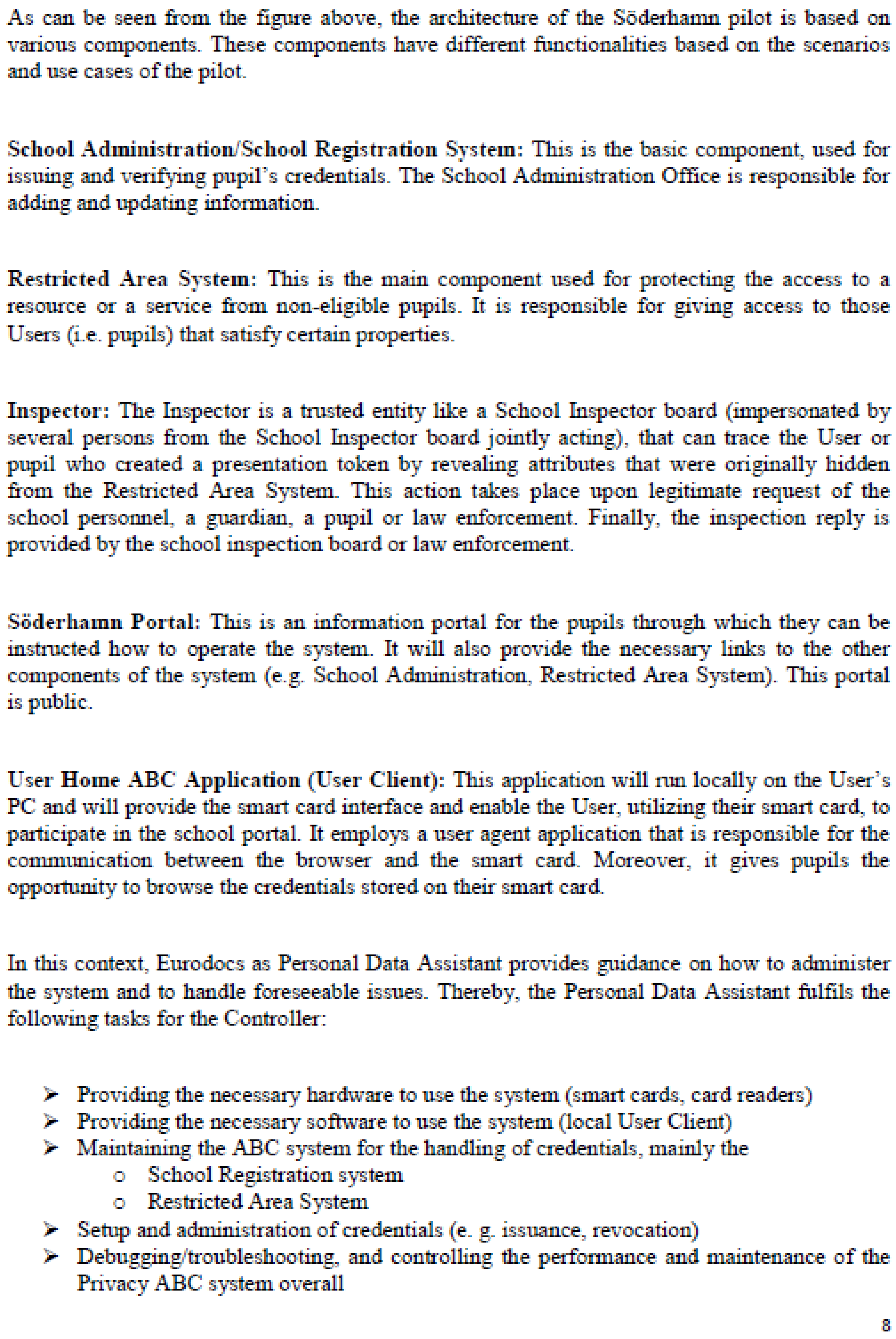


Figure 1: High level architecture of the Söderhamn pilot



As can be seen from the figure above, the architecture of the Söderhamn pilot is based on various components. These components have different functionalities based on the scenarios and use cases of the pilot.

School Administration/School Registration System: This is the basic component, used for issuing and verifying pupil's credentials. The School Administration Office is responsible for adding and updating information.

Restricted Area System: This is the main component used for protecting the access to a resource or a service from non-eligible pupils. It is responsible for giving access to those Users (i.e. pupils) that satisfy certain properties.

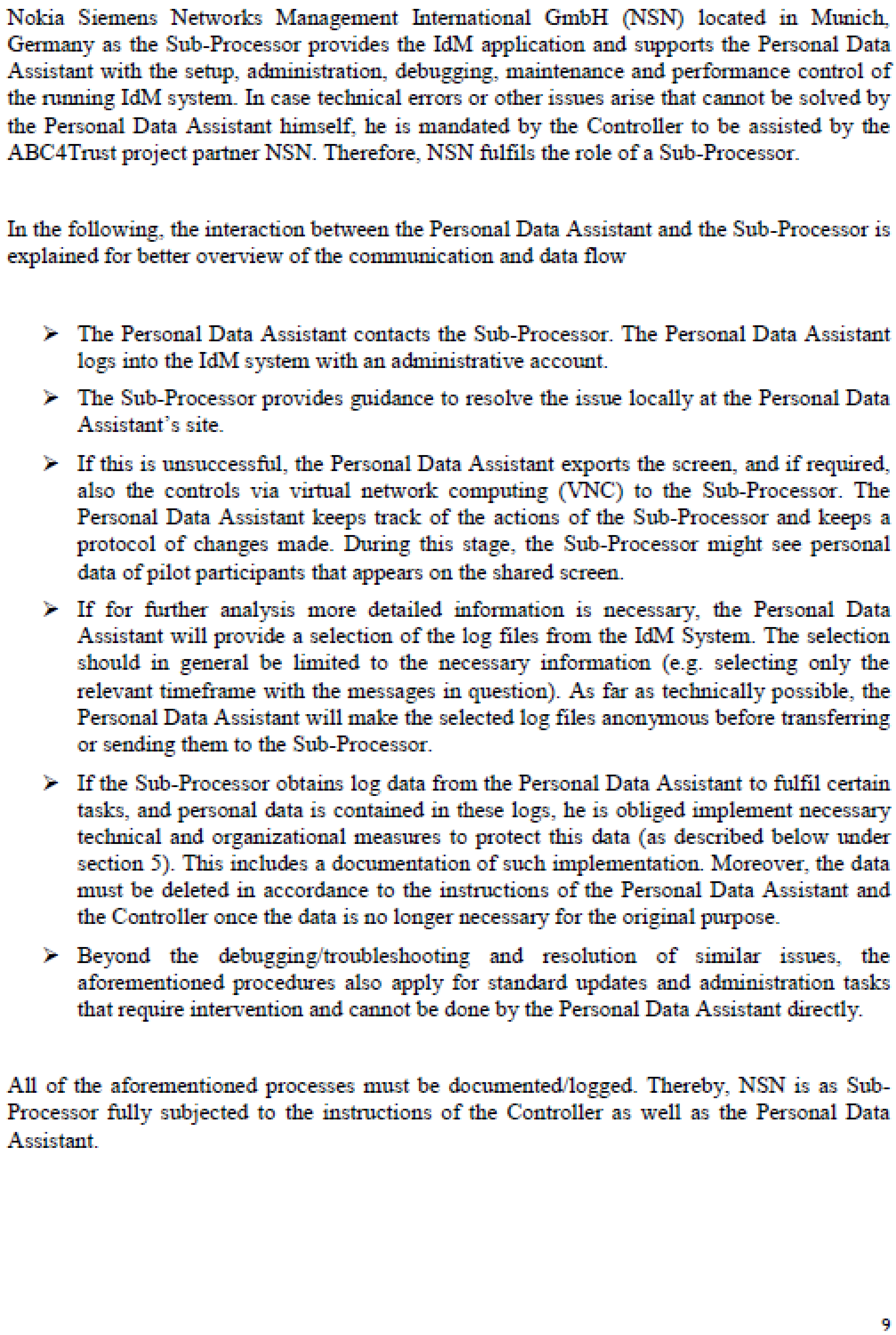
Inspector: The Inspector is a trusted entity like a School Inspector board (impersonated by several persons from the School Inspector board jointly acting), that can trace the User or pupil who created a presentation token by revealing attributes that were originally hidden from the Restricted Area System. This action takes place upon legitimate request of the school personnel, a guardian, a pupil or law enforcement. Finally, the inspection reply is provided by the school inspection board or law enforcement.

Söderhamn Portal: This is an information portal for the pupils through which they can be instructed how to operate the system. It will also provide the necessary links to the other components of the system (e.g. School Administration, Restricted Area System). This portal is public.

User Home ABC Application (User Client): This application will run locally on the User's PC and will provide the smart card interface and enable the User, utilizing their smart card, to participate in the school portal. It employs a user agent application that is responsible for the communication between the browser and the smart card. Moreover, it gives pupils the opportunity to browse the credentials stored on their smart card.

In this context, Eurodocs as Personal Data Assistant provides guidance on how to administer the system and to handle foreseeable issues. Thereby, the Personal Data Assistant fulfils the following tasks for the Controller:

- Providing the necessary hardware to use the system (smart cards, card readers)
- Providing the necessary software to use the system (local User Client)
- Maintaining the ABC system for the handling of credentials, mainly the
  - School Registration system
  - Restricted Area System
- Setup and administration of credentials (e. g. issuance, revocation)
- Debugging/troubleshooting, and controlling the performance and maintenance of the Privacy ABC system overall



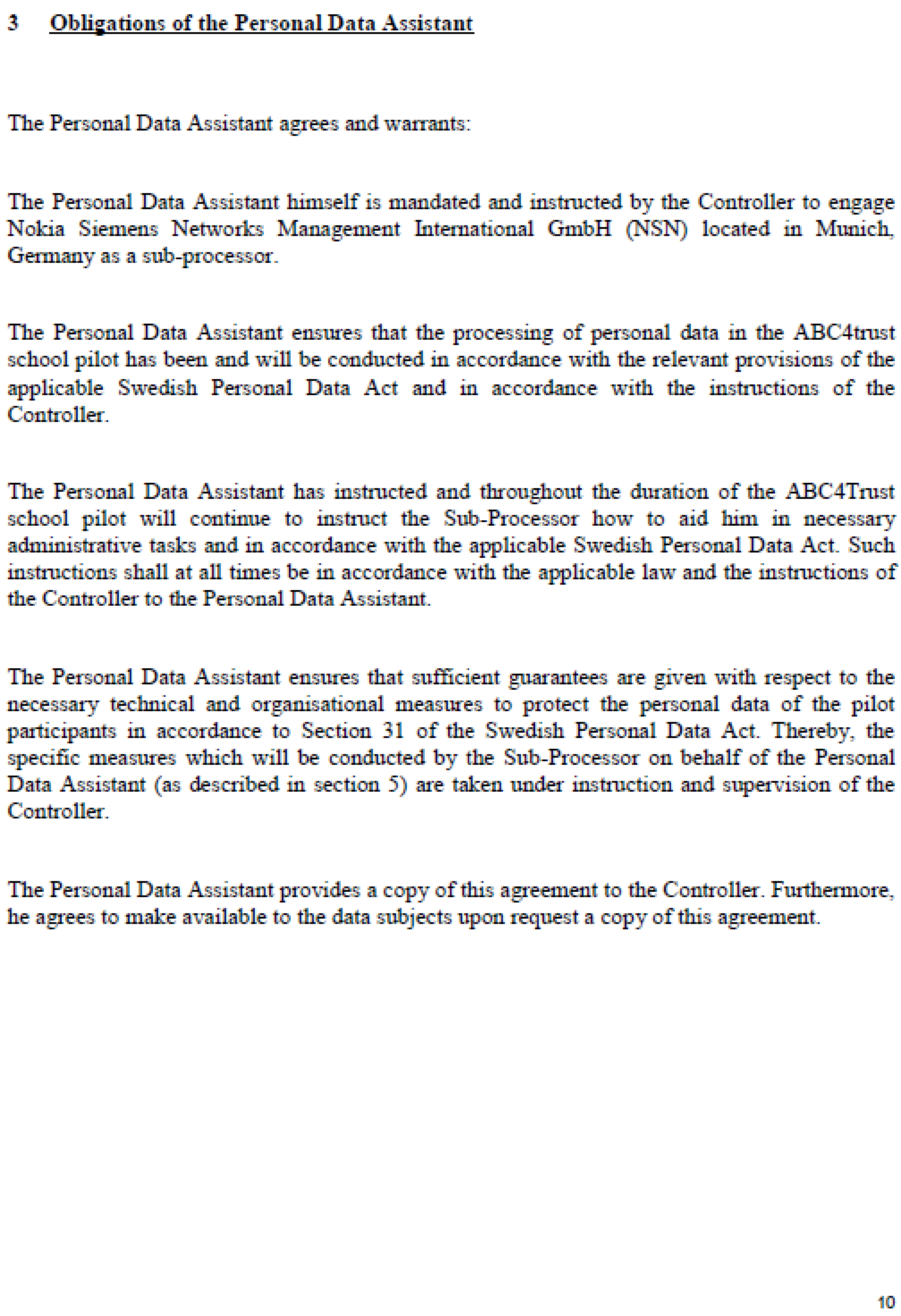
Nokia Siemens Networks Management International GmbH (NSN) located in Munich, Germany as the Sub-Processor provides the IdM application and supports the Personal Data Assistant with the setup, administration, debugging, maintenance and performance control of the running IdM system. In case technical errors or other issues arise that cannot be solved by the Personal Data Assistant himself, he is mandated by the Controller to be assisted by the ABC4Trust project partner NSN. Therefore, NSN fulfils the role of a Sub-Processor.

In the following, the interaction between the Personal Data Assistant and the Sub-Processor is explained for better overview of the communication and data flow

- ➢ The Personal Data Assistant contacts the Sub-Processor. The Personal Data Assistant logs into the IdM system with an administrative account.
- ➢ The Sub-Processor provides guidance to resolve the issue locally at the Personal Data Assistant's site.
- ➢ If this is unsuccessful, the Personal Data Assistant exports the screen, and if required, also the controls via virtual network computing (VNC) to the Sub-Processor. The Personal Data Assistant keeps track of the actions of the Sub-Processor and keeps a protocol of changes made. During this stage, the Sub-Processor might see personal data of pilot participants that appears on the shared screen.
- ➢ If for further analysis more detailed information is necessary, the Personal Data Assistant will provide a selection of the log files from the IdM System. The selection should in general be limited to the necessary information (e.g. selecting only the relevant timeframe with the messages in question). As far as technically possible, the Personal Data Assistant will make the selected log files anonymous before transferring or sending them to the Sub-Processor.
- ➢ If the Sub-Processor obtains log data from the Personal Data Assistant to fulfil certain tasks, and personal data is contained in these logs, he is obliged implement necessary technical and organizational measures to protect this data (as described below under section 5). This includes a documentation of such implementation. Moreover, the data must be deleted in accordance to the instructions of the Personal Data Assistant and the Controller once the data is no longer necessary for the original purpose.
- ➢ Beyond the debugging/troubleshooting and resolution of similar issues, the aforementioned procedures also apply for standard updates and administration tasks that require intervention and cannot be done by the Personal Data Assistant directly.

All of the aforementioned processes must be documented/logged. Thereby, NSN is as Sub-Processor fully subjected to the instructions of the Controller as well as the Personal Data Assistant.

9



3 Obligations of the Personal Data Assistant

The Personal Data Assistant agrees and warrants:

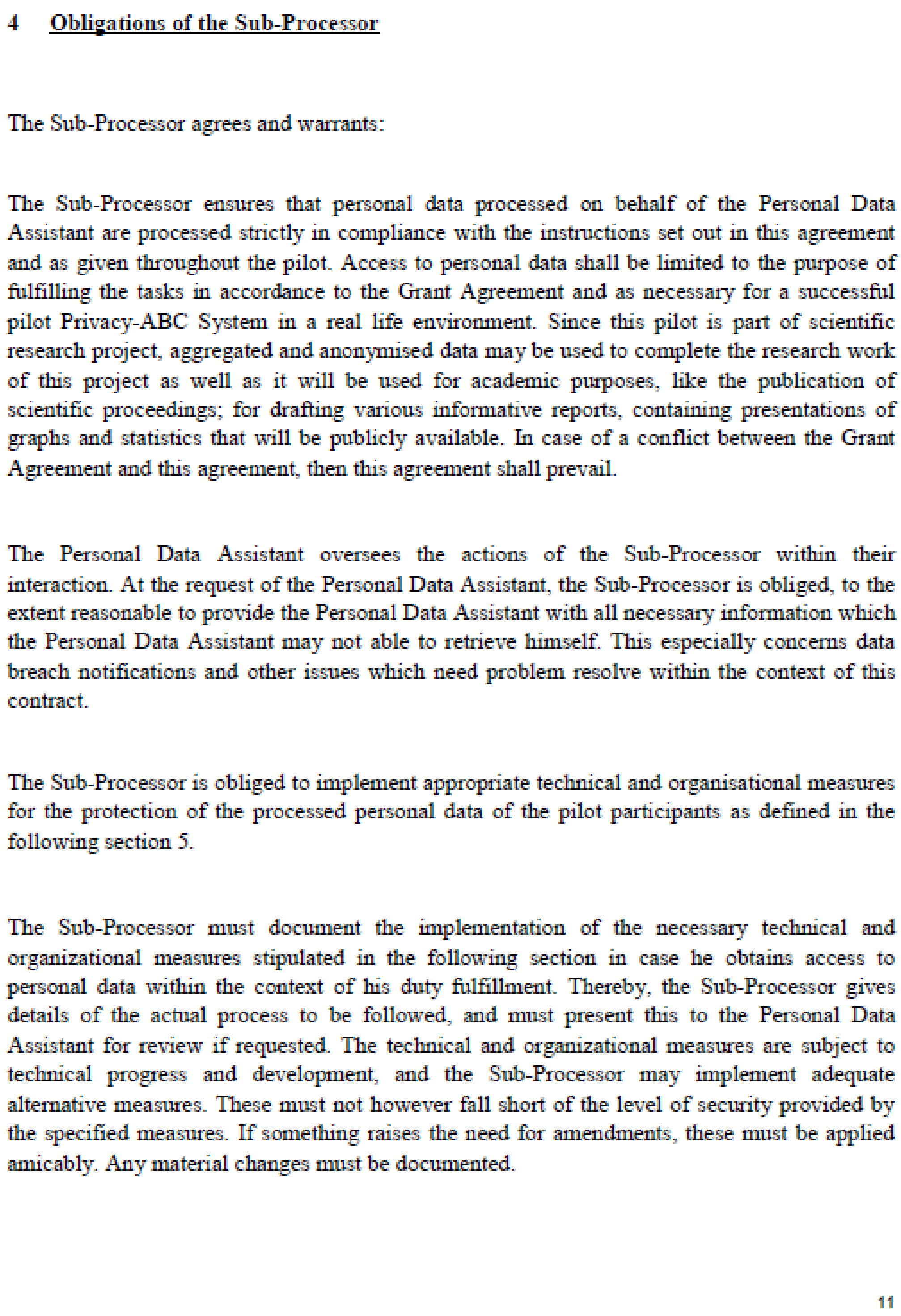
The Personal Data Assistant himself is mandated and instructed by the Controller to engage Nokia Siemens Networks Management International GmbH (NSN) located in Munich, Germany as a sub-processor.

The Personal Data Assistant ensures that the processing of personal data in the ABC4trust school pilot has been and will be conducted in accordance with the relevant provisions of the applicable Swedish Personal Data Act and in accordance with the instructions of the Controller.

The Personal Data Assistant has instructed and throughout the duration of the ABC4Trust school pilot will continue to instruct the Sub-Processor how to aid him in necessary administrative tasks and in accordance with the applicable Swedish Personal Data Act. Such instructions shall at all times be in accordance with the applicable law and the instructions of the Controller to the Personal Data Assistant.

The Personal Data Assistant ensures that sufficient guarantees are given with respect to the necessary technical and organisational measures to protect the personal data of the pilot participants in accordance to Section 31 of the Swedish Personal Data Act. Thereby, the specific measures which will be conducted by the Sub-Processor on behalf of the Personal Data Assistant (as described in section 5) are taken under instruction and supervision of the Controller.

The Personal Data Assistant provides a copy of this agreement to the Controller. Furthermore, he agrees to make available to the data subjects upon request a copy of this agreement.



4 Obligations of the Sub-Processor

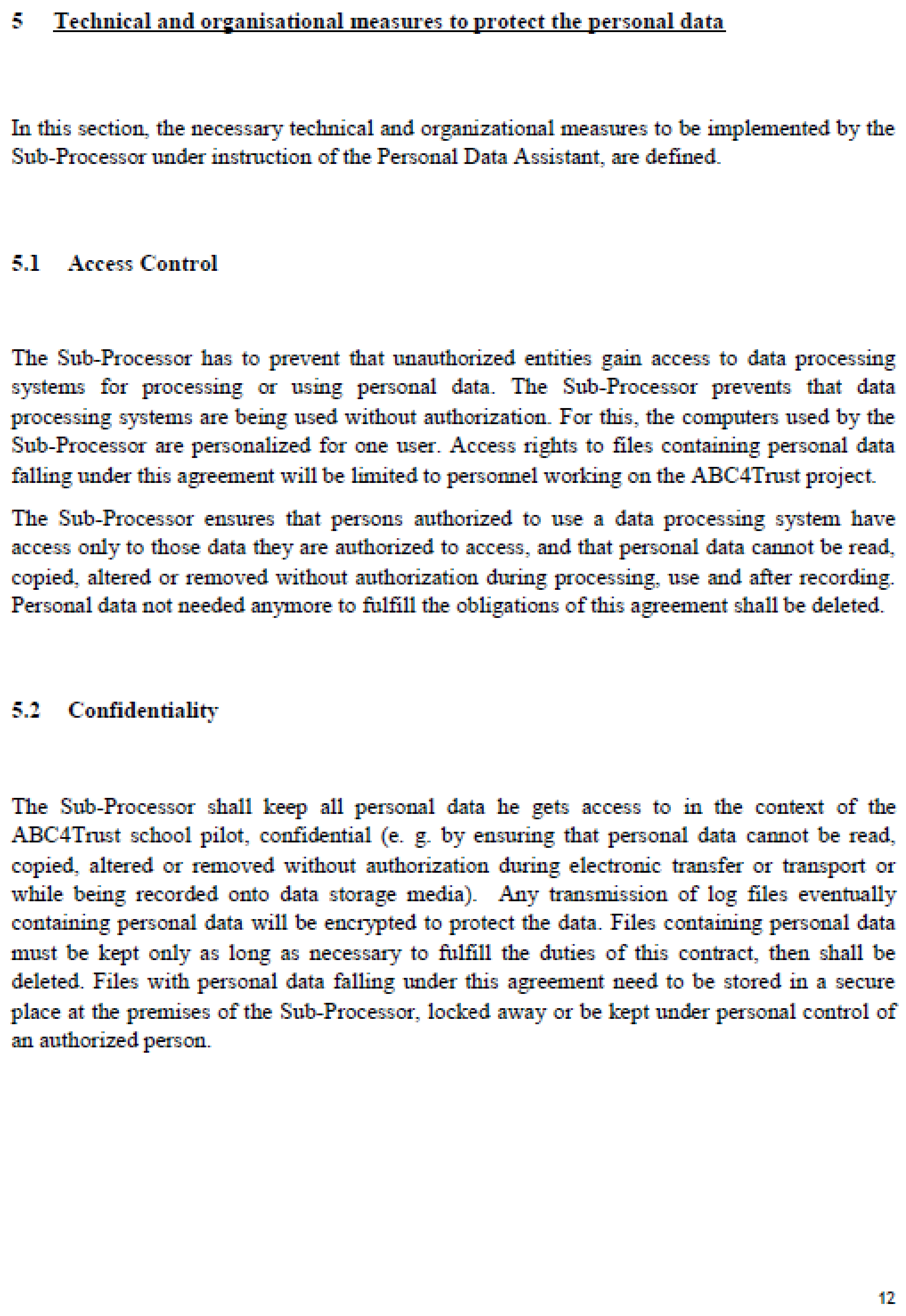
The Sub-Processor agrees and warrants:

The Sub-Processor ensures that personal data processed on behalf of the Personal Data Assistant are processed strictly in compliance with the instructions set out in this agreement and as given throughout the pilot. Access to personal data shall be limited to the purpose of fulfilling the tasks in accordance to the Grant Agreement and as necessary for a successful pilot Privacy-ABC System in a real life environment. Since this pilot is part of scientific research project, aggregated and anonymised data may be used to complete the research work of this project as well as it will be used for academic purposes, like the publication of scientific proceedings; for drafting various informative reports, containing presentations of graphs and statistics that will be publicly available. In case of a conflict between the Grant Agreement and this agreement, then this agreement shall prevail.

The Personal Data Assistant oversees the actions of the Sub-Processor within their interaction. At the request of the Personal Data Assistant, the Sub-Processor is obliged, to the extent reasonable to provide the Personal Data Assistant with all necessary information which the Personal Data Assistant may not able to retrieve himself. This especially concerns data breach notifications and other issues which need problem resolve within the context of this contract.

The Sub-Processor is obliged to implement appropriate technical and organisational measures for the protection of the processed personal data of the pilot participants as defined in the following section 5.

The Sub-Processor must document the implementation of the necessary technical and organizational measures stipulated in the following section in case he obtains access to personal data within the context of his duty fulfillment. Thereby, the Sub-Processor gives details of the actual process to be followed, and must present this to the Personal Data Assistant for review if requested. The technical and organizational measures are subject to technical progress and development, and the Sub-Processor may implement adequate alternative measures. These must not however fall short of the level of security provided by the specified measures. If something raises the need for amendments, these must be applied amicably. Any material changes must be documented.



5 Technical and organisational measures to protect the personal data

In this section, the necessary technical and organizational measures to be implemented by the Sub-Processor under instruction of the Personal Data Assistant, are defined.

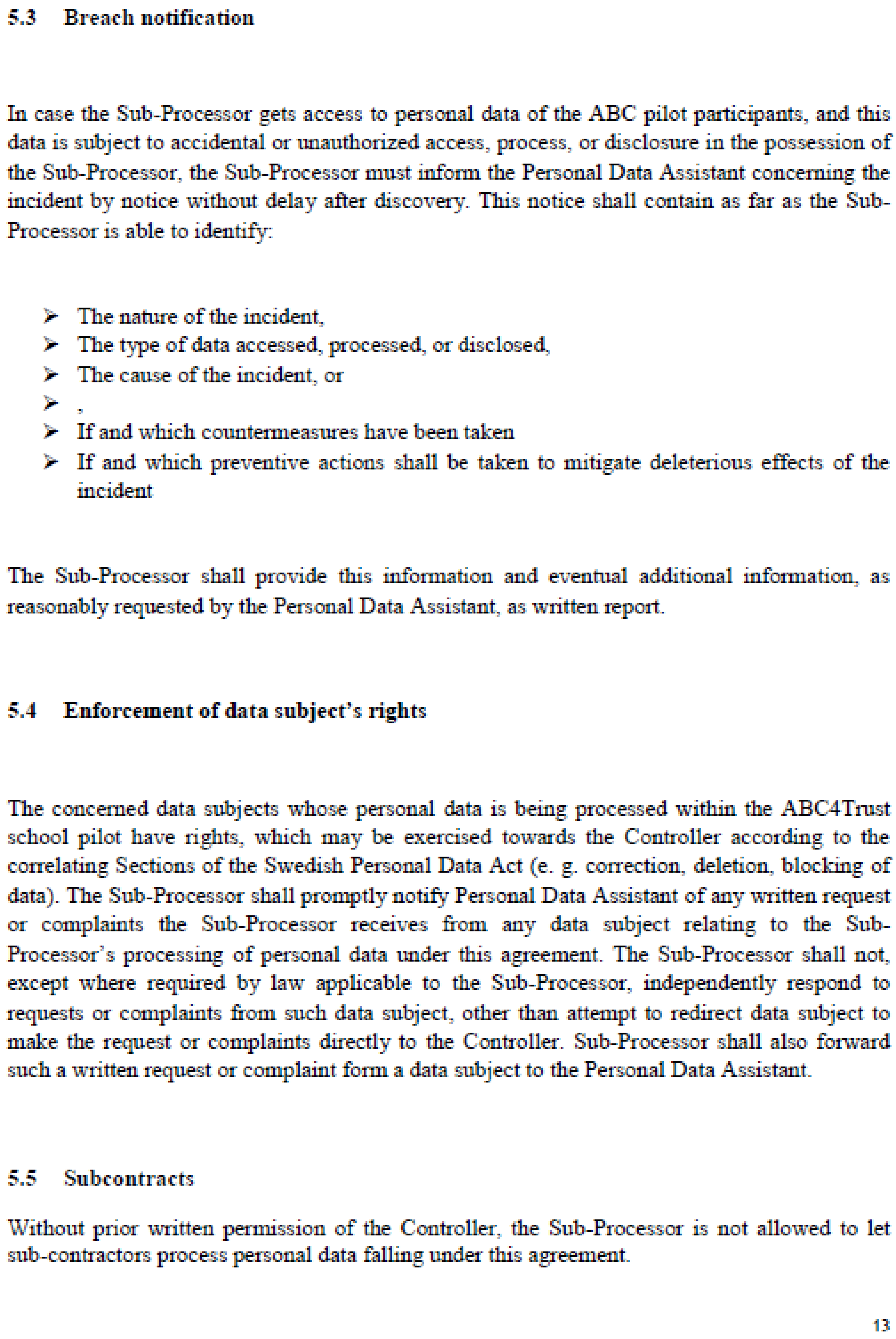
5.1 Access Control

The Sub-Processor has to prevent that unauthorized entities gain access to data processing systems for processing or using personal data. The Sub-Processor prevents that data processing systems are being used without authorization. For this, the computers used by the Sub-Processor are personalized for one user. Access rights to files containing personal data falling under this agreement will be limited to personnel working on the ABC4Trust project.

The Sub-Processor ensures that persons authorized to use a data processing system have access only to those data they are authorized to access, and that personal data cannot be read, copied, altered or removed without authorization during processing, use and after recording. Personal data not needed anymore to fulfill the obligations of this agreement shall be deleted.

5.2 Confidentiality

The Sub-Processor shall keep all personal data he gets access to in the context of the ABC4Trust school pilot, confidential (e. g. by ensuring that personal data cannot be read, copied, altered or removed without authorization during electronic transfer or transport or while being recorded onto data storage media). Any transmission of log files eventually containing personal data will be encrypted to protect the data. Files containing personal data must be kept only as long as necessary to fulfill the duties of this contract, then shall be deleted. Files with personal data falling under this agreement need to be stored in a secure place at the premises of the Sub-Processor, locked away or be kept under personal control of an authorized person.



5.3 Breach notification

In case the Sub-Processor gets access to personal data of the ABC pilot participants, and this data is subject to accidental or unauthorized access, process, or disclosure in the possession of the Sub-Processor, the Sub-Processor must inform the Personal Data Assistant concerning the incident by notice without delay after discovery. This notice shall contain as far as the Sub-Processor is able to identify:

- The nature of the incident,
- The type of data accessed, processed, or disclosed,
- The cause of the incident, or
- ,
- If and which countermeasures have been taken
- If and which preventive actions shall be taken to mitigate deleterious effects of the incident

The Sub-Processor shall provide this information and eventual additional information, as reasonably requested by the Personal Data Assistant, as written report.

5.4 Enforcement of data subject's rights

The concerned data subjects whose personal data is being processed within the ABC4Trust school pilot have rights, which may be exercised towards the Controller according to the correlating Sections of the Swedish Personal Data Act (e. g. correction, deletion, blocking of data). The Sub-Processor shall promptly notify Personal Data Assistant of any written request or complaints the Sub-Processor receives from any data subject relating to the Sub-Processor's processing of personal data under this agreement. The Sub-Processor shall not, except where required by law applicable to the Sub-Processor, independently respond to requests or complaints from such data subject, other than attempt to redirect data subject to make the request or complaints directly to the Controller. Sub-Processor shall also forward such a written request or complaint form a data subject to the Personal Data Assistant.

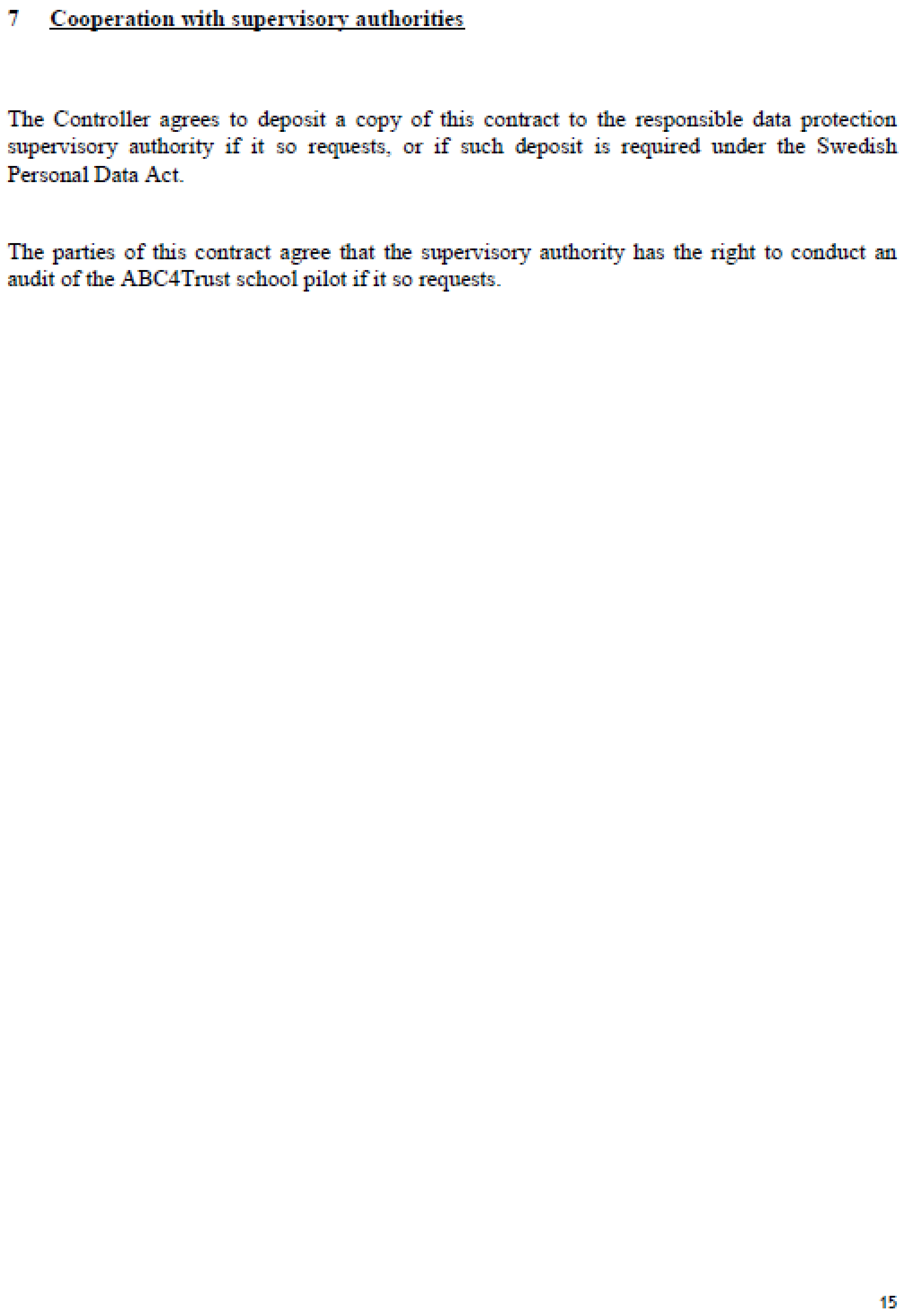
5.5 Subcontracts

Without prior written permission of the Controller, the Sub-Processor is not allowed to let sub-contractors process personal data falling under this agreement.

13

6 Third-party beneficiary clause

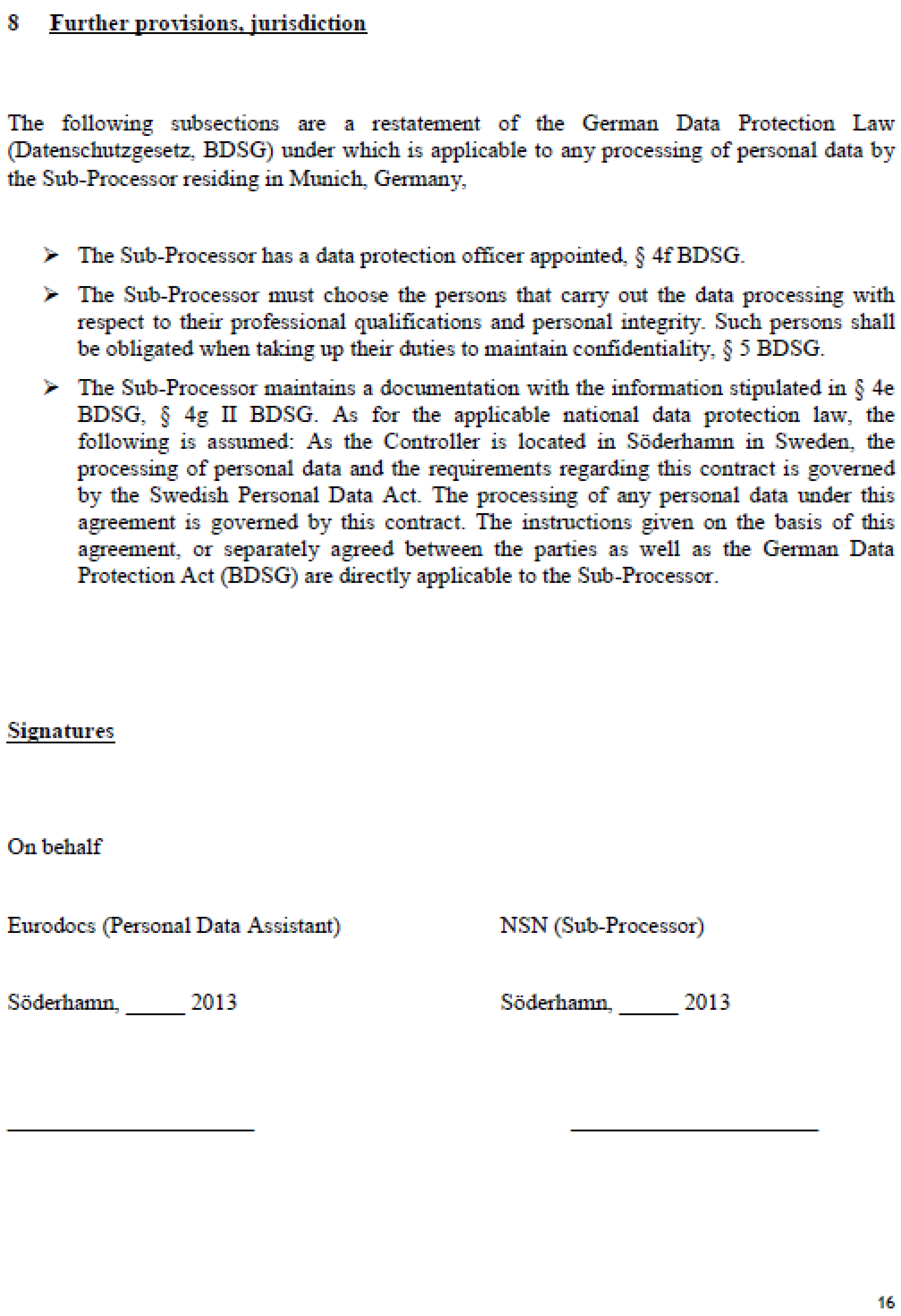
The Controller can enforce the above described obligations of this agreement against the Sub-Processor in place of the Personal Data Assistant as third-party beneficiary in case the Personal Data Assistant has factually disappeared or has ceased to exist in law.



7 Cooperation with supervisory authorities

The Controller agrees to deposit a copy of this contract to the responsible data protection supervisory authority if it so requests, or if such deposit is required under the Swedish Personal Data Act.

The parties of this contract agree that the supervisory authority has the right to conduct an audit of the ABC4Trust school pilot if it so requests.



8 Further provisions, jurisdiction

The following subsections are a restatement of the German Data Protection Law (Datenschutzgesetz, BDSG) under which is applicable to any processing of personal data by the Sub-Processor residing in Munich, Germany,

- The Sub-Processor has a data protection officer appointed, § 4f BDSG.
- The Sub-Processor must choose the persons that carry out the data processing with respect to their professional qualifications and personal integrity. Such persons shall be obligated when taking up their duties to maintain confidentiality, § 5 BDSG.
- The Sub-Processor maintains a documentation with the information stipulated in § 4e BDSG, § 4g II BDSG. As for the applicable national data protection law, the following is assumed: As the Controller is located in Söderhamn in Sweden, the processing of personal data and the requirements regarding this contract is governed by the Swedish Personal Data Act. The processing of any personal data under this agreement is governed by this contract. The instructions given on the basis of this agreement, or separately agreed between the parties as well as the German Data Protection Act (BDSG) are directly applicable to the Sub-Processor.

Signatures

On behalf

Eurodocs (Personal Data Assistant) NSN (Sub-Processor)

Söderhamn, _____ 2013 Söderhamn, _____ 2013

________________________ ________________________

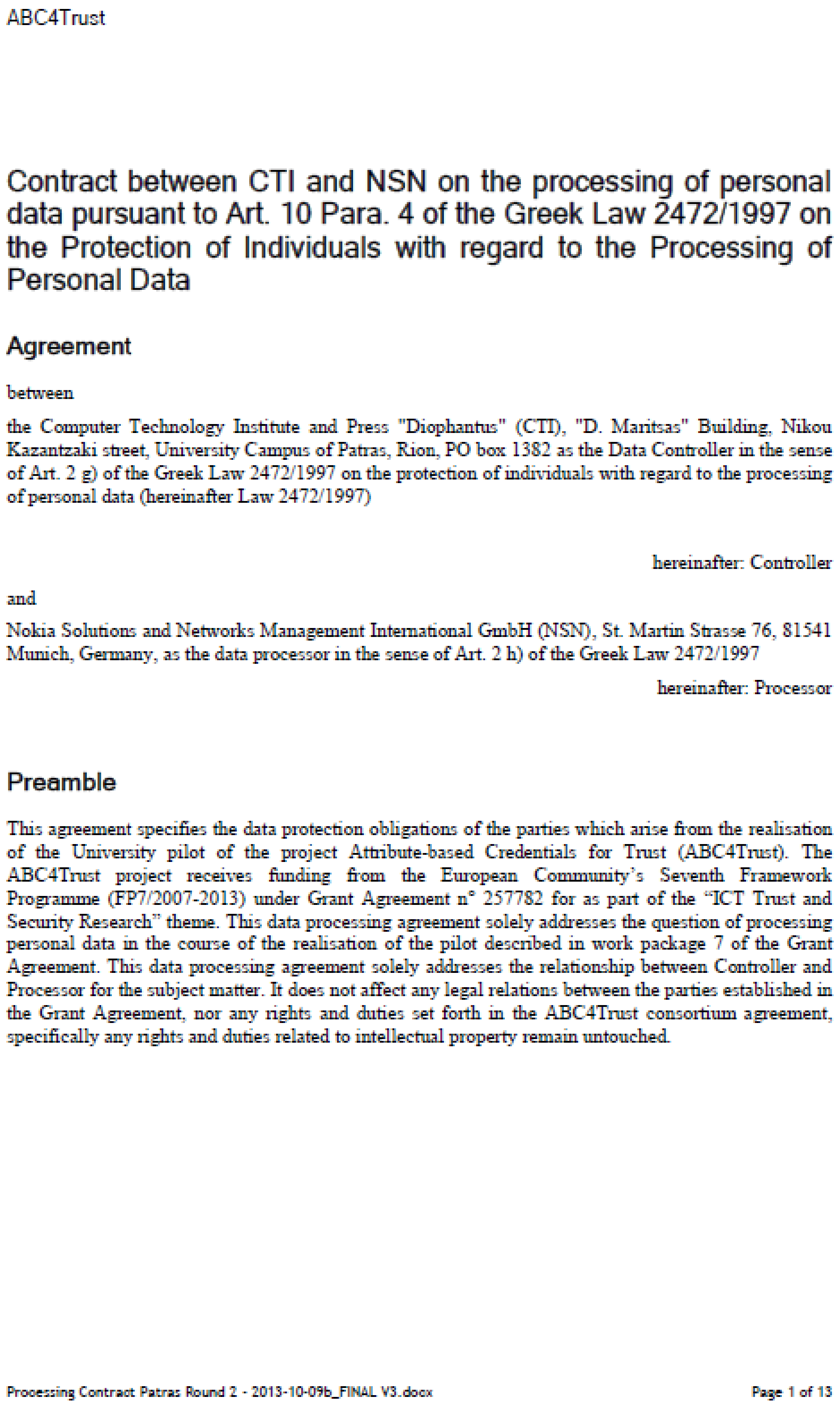
16

Attachments:

Appendix : Information sheet and consent form for parents/pupils + for school staff

17

A.3 Processing Contract of Patras Pilot



Contract between CTI and NSN on the processing of personal data pursuant to Art. 10 Para. 4 of the Greek Law 2472/1997 on the Protection of Individuals with regard to the Processing of Personal Data

Agreement

between

the Computer Technology Institute and Press "Diophantus" (CTI), "D. Maritsas" Building, Nikou Kazantzaki street, University Campus of Patras, Rion, PO box 1382 as the Data Controller in the sense of Art. 2 g) of the Greek Law 2472/1997 on the protection of individuals with regard to the processing of personal data (hereinafter Law 2472/1997)

hereinafter: Controller

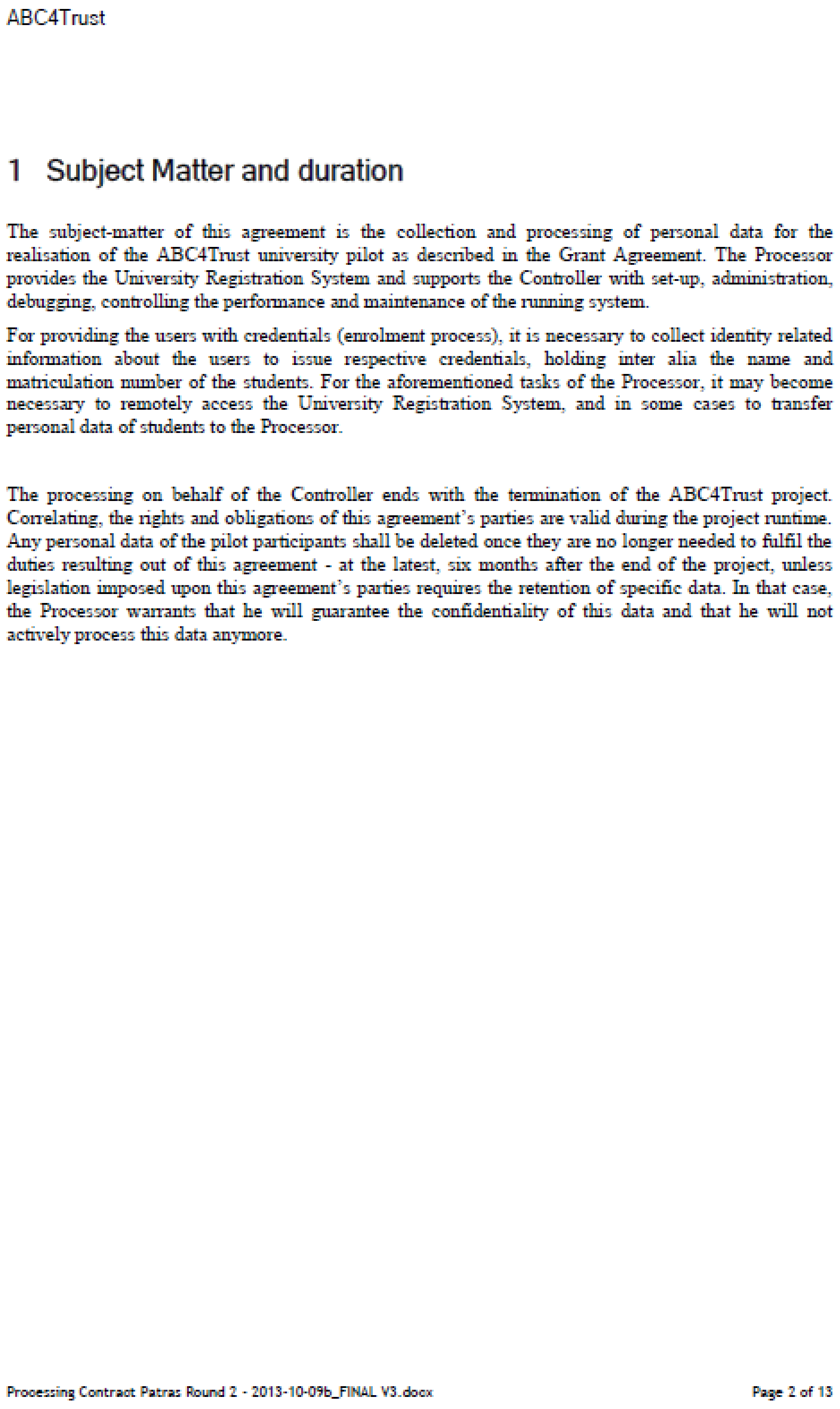
and

Nokia Solutions and Networks Management International GmbH (NSN), St. Martin Strasse 76, 81541 Munich, Germany, as the data processor in the sense of Art. 2 h) of the Greek Law 2472/1997

hereinafter: Processor

Preamble

This agreement specifies the data protection obligations of the parties which arise from the realisation of the University pilot of the project Attribute-based Credentials for Trust (ABC4Trust). The ABC4Trust project receives funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement n° 257782 for as part of the "ICT Trust and Security Research" theme. This data processing agreement solely addresses the question of processing personal data in the course of the realisation of the pilot described in work package 7 of the Grant Agreement. This data processing agreement solely addresses the relationship between Controller and Processor for the subject matter. It does not affect any legal relations between the parties established in the Grant Agreement, nor any rights and duties set forth in the ABC4Trust consortium agreement, specifically any rights and duties related to intellectual property remain untouched.



ABC4Trust

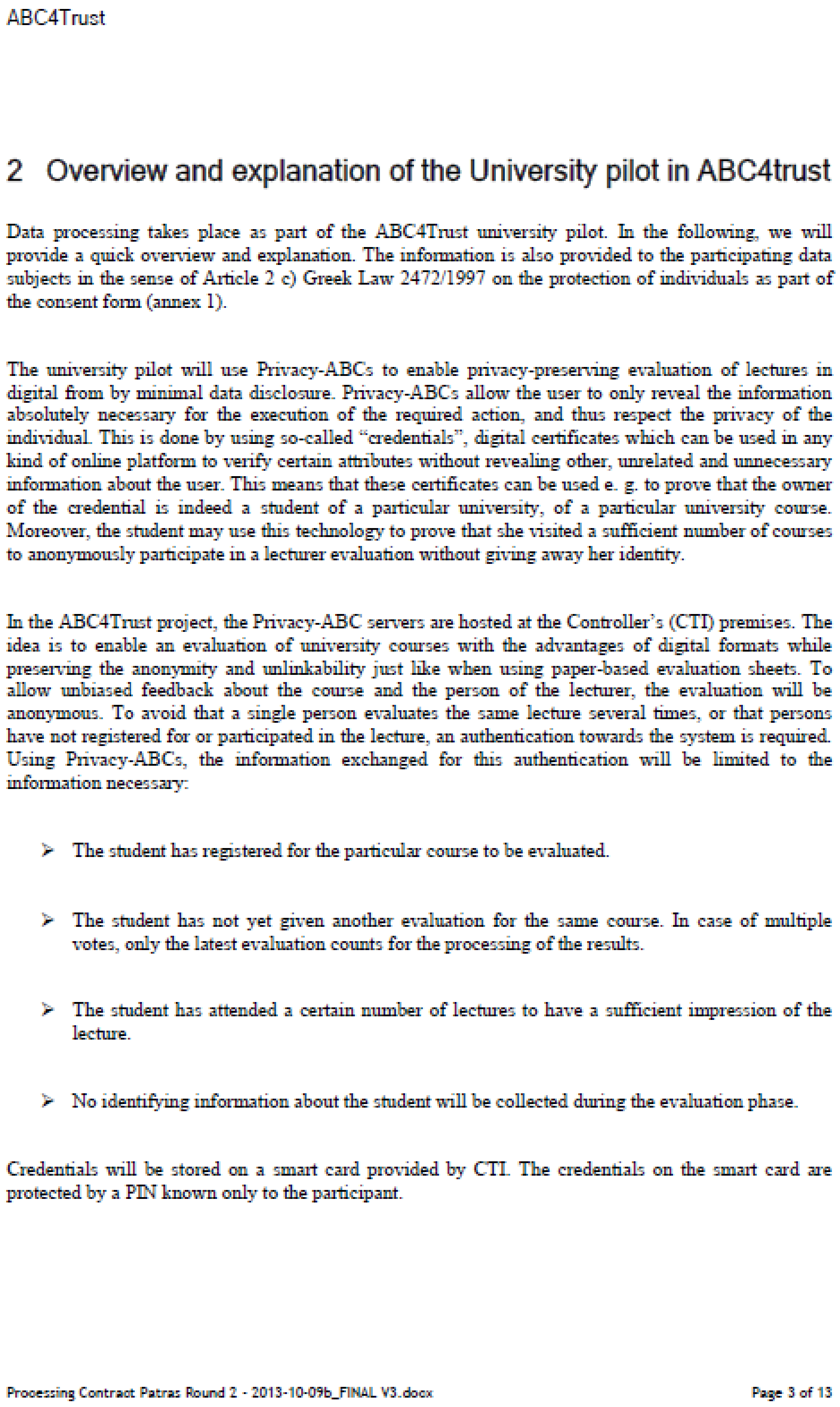
1 Subject Matter and duration

The subject-matter of this agreement is the collection and processing of personal data for the realisation of the ABC4Trust university pilot as described in the Grant Agreement. The Processor provides the University Registration System and supports the Controller with set-up, administration, debugging, controlling the performance and maintenance of the running system.

For providing the users with credentials (enrolment process), it is necessary to collect identity related information about the users to issue respective credentials, holding inter alia the name and matriculation number of the students. For the aforementioned tasks of the Processor, it may become necessary to remotely access the University Registration System, and in some cases to transfer personal data of students to the Processor.

The processing on behalf of the Controller ends with the termination of the ABC4Trust project. Correlating, the rights and obligations of this agreement's parties are valid during the project runtime. Any personal data of the pilot participants shall be deleted once they are no longer needed to fulfil the duties resulting out of this agreement - at the latest, six months after the end of the project, unless legislation imposed upon this agreement's parties requires the retention of specific data. In that case, the Processor warrants that he will guarantee the confidentiality of this data and that he will not actively process this data anymore.

Processing Contract Patras Round 2 - 2013-10-09b_FINAL V3.docx Page 2 of 13



2 Overview and explanation of the University pilot in ABC4trust

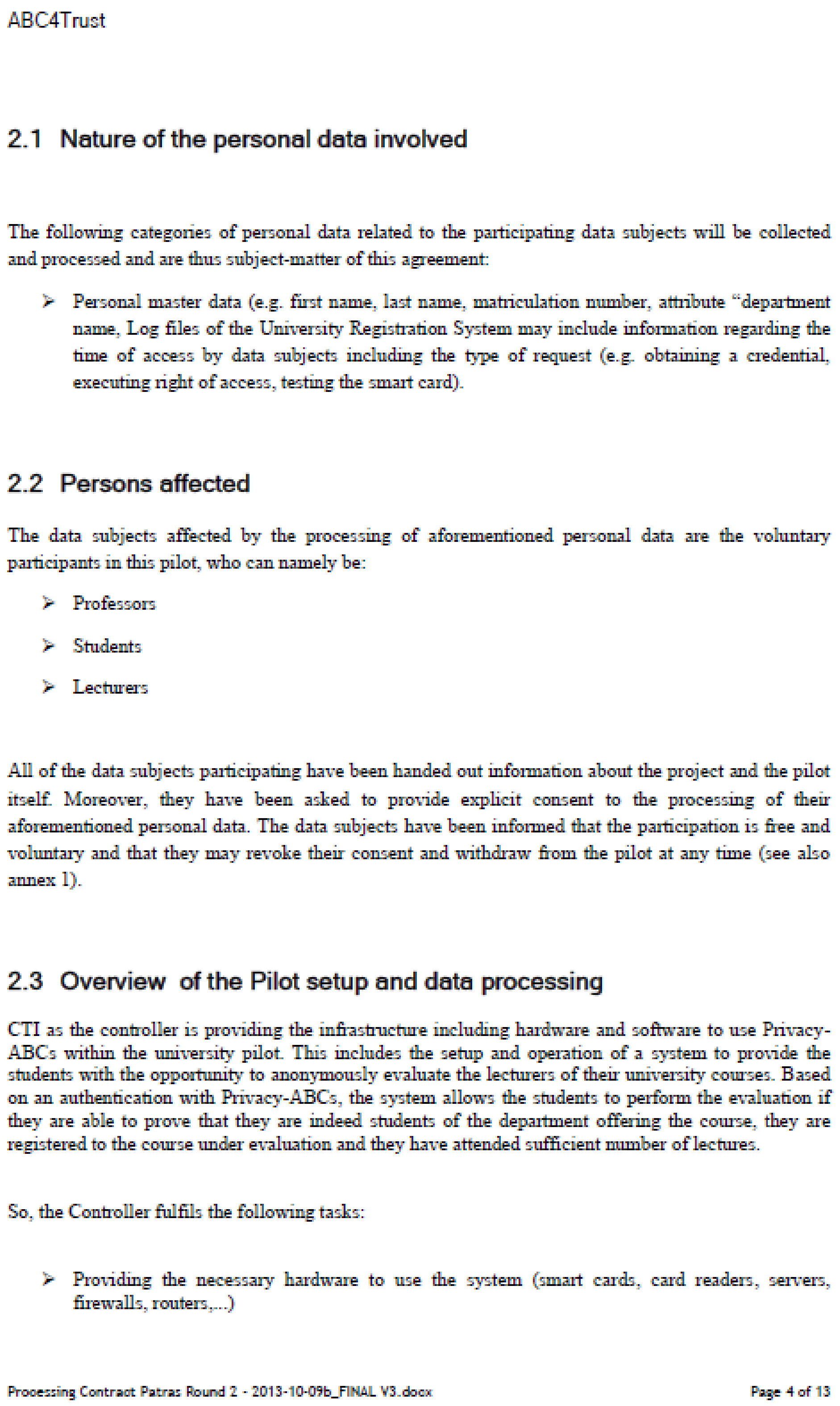
Data processing takes place as part of the ABC4Trust university pilot. In the following, we will provide a quick overview and explanation. The information is also provided to the participating data subjects in the sense of Article 2 c) Greek Law 2472/1997 on the protection of individuals as part of the consent form (annex 1).

The university pilot will use Privacy-ABCs to enable privacy-preserving evaluation of lectures in digital from by minimal data disclosure. Privacy-ABCs allow the user to only reveal the information absolutely necessary for the execution of the required action, and thus respect the privacy of the individual. This is done by using so-called "credentials", digital certificates which can be used in any kind of online platform to verify certain attributes without revealing other, unrelated and unnecessary information about the user. This means that these certificates can be used e. g. to prove that the owner of the credential is indeed a student of a particular university, of a particular university course. Moreover, the student may use this technology to prove that she visited a sufficient number of courses to anonymously participate in a lecturer evaluation without giving away her identity.

In the ABC4Trust project, the Privacy-ABC servers are hosted at the Controller's (CTI) premises. The idea is to enable an evaluation of university courses with the advantages of digital formats while preserving the anonymity and unlinkability just like when using paper-based evaluation sheets. To allow unbiased feedback about the course and the person of the lecturer, the evaluation will be anonymous. To avoid that a single person evaluates the same lecture several times, or that persons have not registered for or participated in the lecture, an authentication towards the system is required. Using Privacy-ABCs, the information exchanged for this authentication will be limited to the information necessary:

- The student has registered for the particular course to be evaluated.
- The student has not yet given another evaluation for the same course. In case of multiple votes, only the latest evaluation counts for the processing of the results.
- The student has attended a certain number of lectures to have a sufficient impression of the lecture.
- No identifying information about the student will be collected during the evaluation phase.

Credentials will be stored on a smart card provided by CTI. The credentials on the smart card are protected by a PIN known only to the participant.



2.1 Nature of the personal data involved

The following categories of personal data related to the participating data subjects will be collected and processed and are thus subject-matter of this agreement:

- Personal master data (e.g. first name, last name, matriculation number, attribute “department name, Log files of the University Registration System may include information regarding the time of access by data subjects including the type of request (e.g. obtaining a credential, executing right of access, testing the smart card).

2.2 Persons affected

The data subjects affected by the processing of aforementioned personal data are the voluntary participants in this pilot, who can namely be:

- Professors
- Students
- Lecturers

All of the data subjects participating have been handed out information about the project and the pilot itself. Moreover, they have been asked to provide explicit consent to the processing of their aforementioned personal data. The data subjects have been informed that the participation is free and voluntary and that they may revoke their consent and withdraw from the pilot at any time (see also annex 1).

2.3 Overview of the Pilot setup and data processing

CTI as the controller is providing the infrastructure including hardware and software to use Privacy-ABCs within the university pilot. This includes the setup and operation of a system to provide the students with the opportunity to anonymously evaluate the lecturers of their university courses. Based on an authentication with Privacy-ABCs, the system allows the students to perform the evaluation if they are able to prove that they are indeed students of the department offering the course, they are registered to the course under evaluation and they have attended sufficient number of lectures.

So, the Controller fulfils the following tasks:

- Providing the necessary hardware to use the system (smart cards, card readers, servers, firewalls, routers,...)

- Providing the necessary software to use the system (User software, web applications, ...)
- Maintaining the system for the handling of credentials,
- Setup and administration of credentials (e. g. issuance, revocation)
- Debugging/troubleshooting, and controlling the performance and maintenance of the system overall

NSN as the Processor is administering the University Registration System running on 2 machines which reside on the Premises of the Controller. In the general overview of the Pilot, the IdM Portal is part of the University Registration System used for issuing the necessary cryptographic credentials to the students (Users).

The Processor provides the applications of the University Registration System and supports the Controller with their set-up, their administration, their debugging, controlling their performance and their maintenance. In doing so, the Processor is fully subjected to the instructions of the Controller.

The communication channels for transfer of data are shown in Figure 1 below.

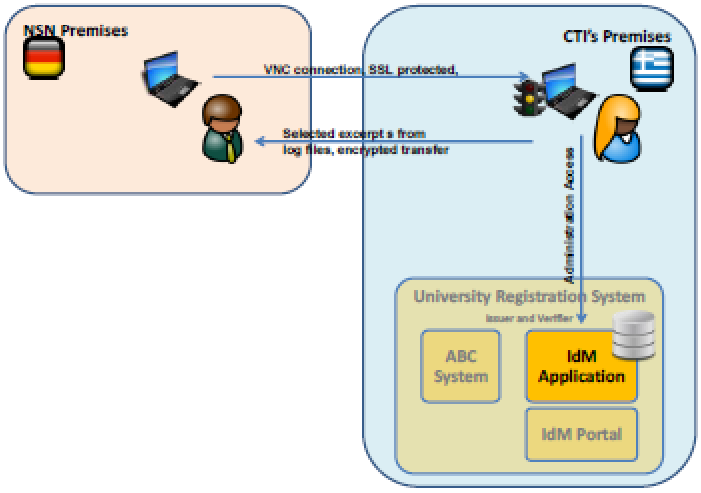
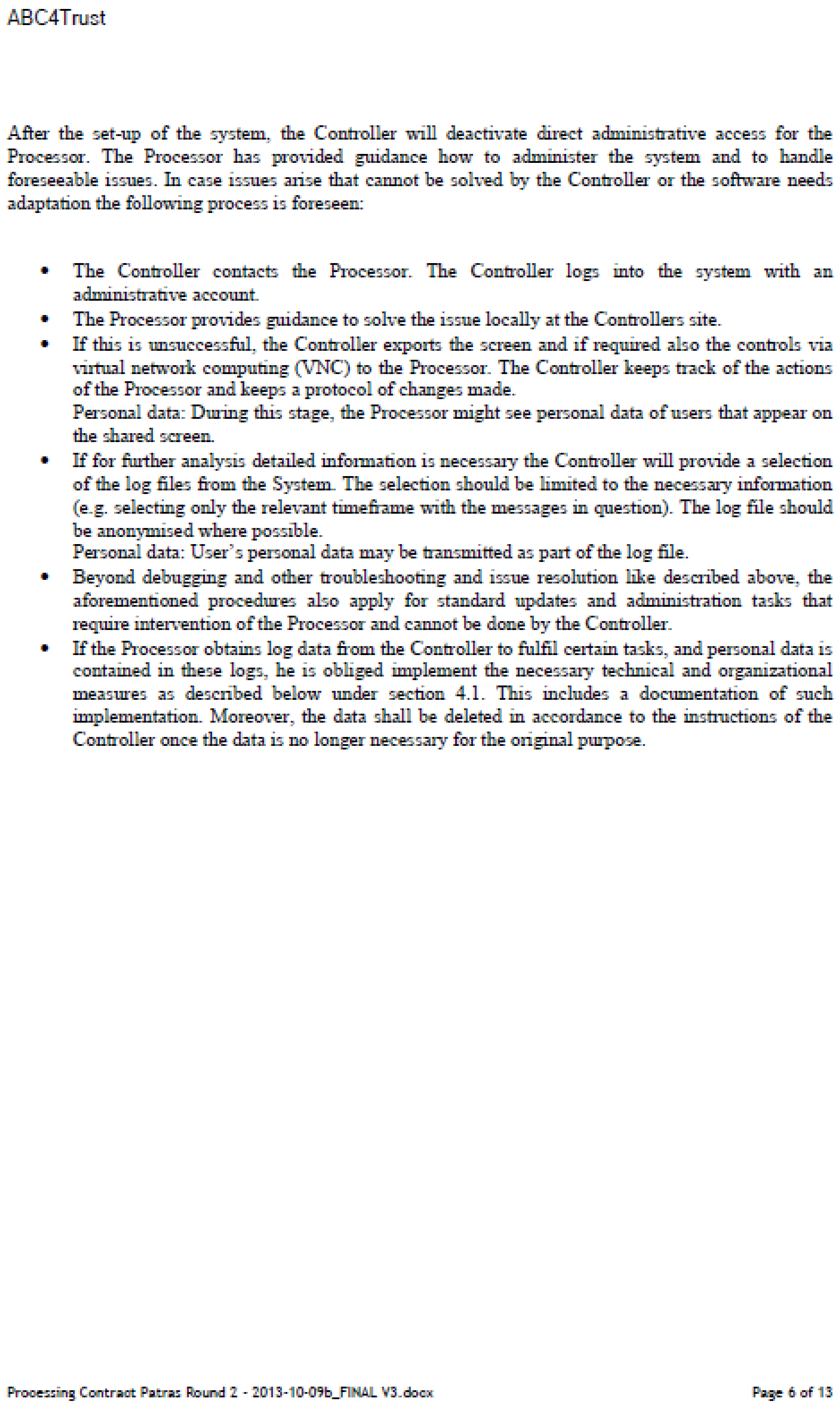
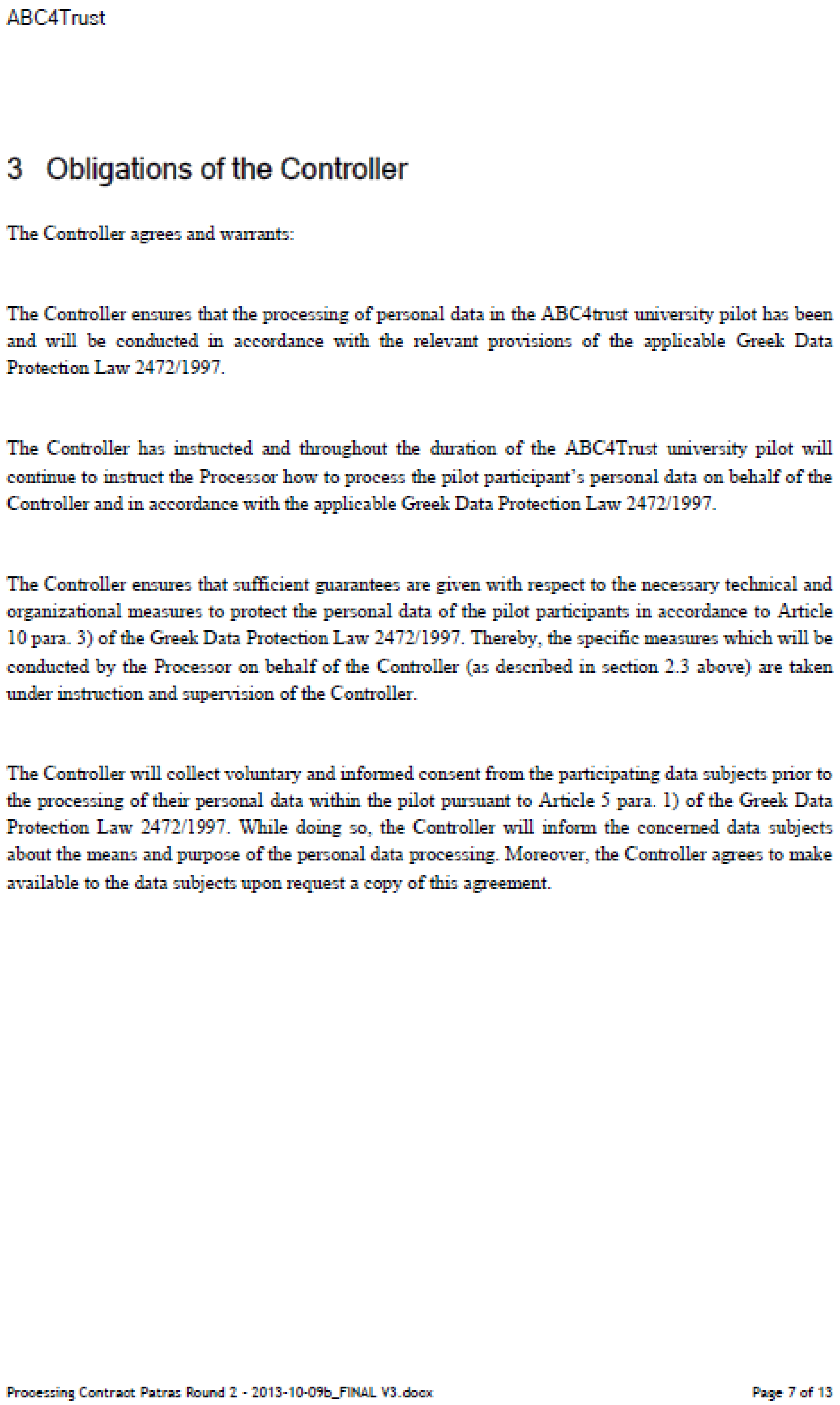


Figure 1 Communication between Controller and Processor



After the set-up of the system, the Controller will deactivate direct administrative access for the Processor. The Processor has provided guidance how to administer the system and to handle foreseeable issues. In case issues arise that cannot be solved by the Controller or the software needs adaptation the following process is foreseen:

- The Controller contacts the Processor. The Controller logs into the system with an administrative account.
- The Processor provides guidance to solve the issue locally at the Controllers site.
- If this is unsuccessful, the Controller exports the screen and if required also the controls via virtual network computing (VNC) to the Processor. The Controller keeps track of the actions of the Processor and keeps a protocol of changes made.
  Personal data: During this stage, the Processor might see personal data of users that appear on the shared screen.
- If for further analysis detailed information is necessary the Controller will provide a selection of the log files from the System. The selection should be limited to the necessary information (e.g. selecting only the relevant timeframe with the messages in question). The log file should be anonymised where possible.
  Personal data: User's personal data may be transmitted as part of the log file.
- Beyond debugging and other troubleshooting and issue resolution like described above, the aforementioned procedures also apply for standard updates and administration tasks that require intervention of the Processor and cannot be done by the Controller.
- If the Processor obtains log data from the Controller to fulfil certain tasks, and personal data is contained in these logs, he is obliged implement the necessary technical and organizational measures as described below under section 4.1. This includes a documentation of such implementation. Moreover, the data shall be deleted in accordance to the instructions of the Controller once the data is no longer necessary for the original purpose.



3 Obligations of the Controller

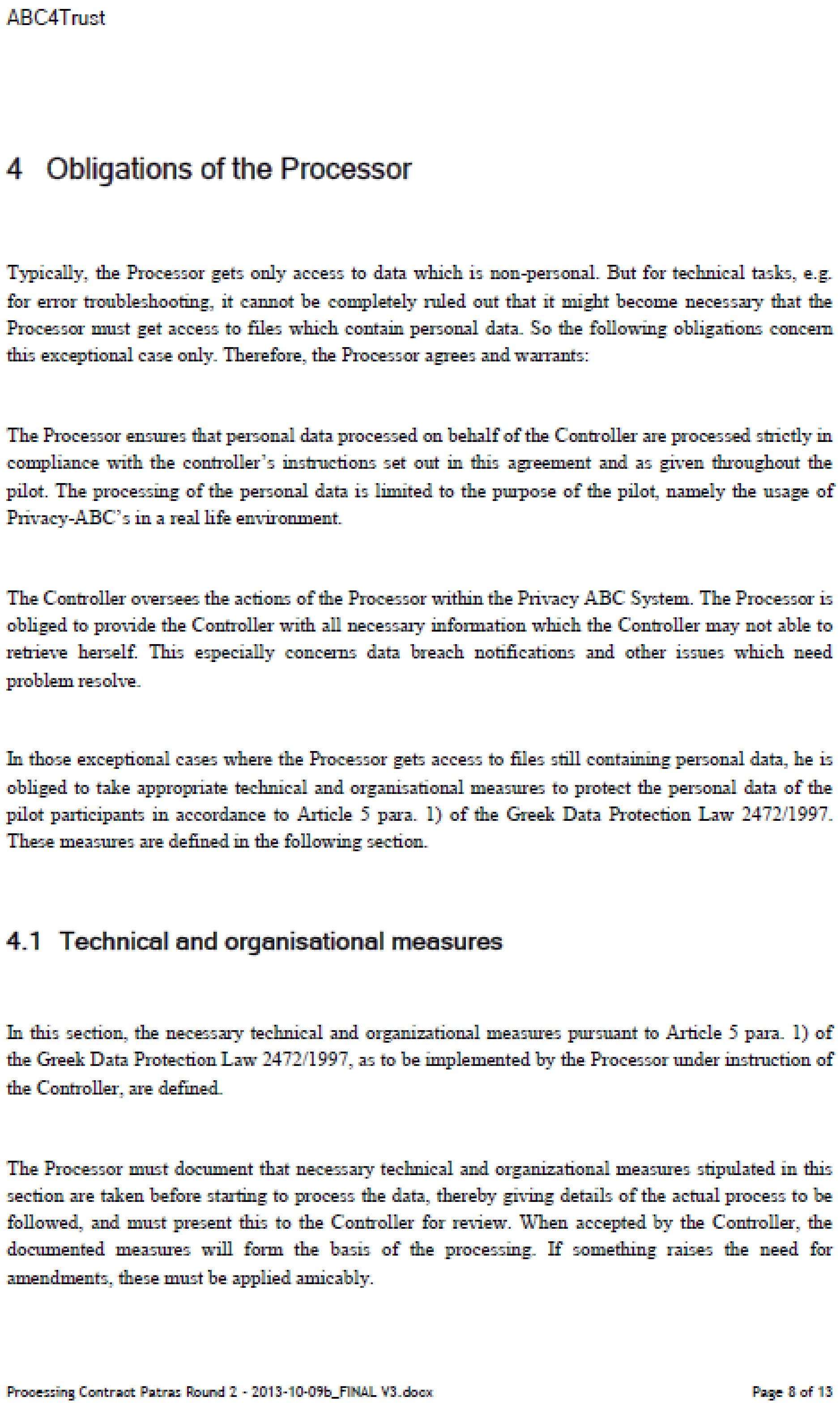
The Controller agrees and warrants:

The Controller ensures that the processing of personal data in the ABC4trust university pilot has been and will be conducted in accordance with the relevant provisions of the applicable Greek Data Protection Law 2472/1997.

The Controller has instructed and throughout the duration of the ABC4Trust university pilot will continue to instruct the Processor how to process the pilot participant's personal data on behalf of the Controller and in accordance with the applicable Greek Data Protection Law 2472/1997.

The Controller ensures that sufficient guarantees are given with respect to the necessary technical and organizational measures to protect the personal data of the pilot participants in accordance to Article 10 para. 3) of the Greek Data Protection Law 2472/1997. Thereby, the specific measures which will be conducted by the Processor on behalf of the Controller (as described in section 2.3 above) are taken under instruction and supervision of the Controller.

The Controller will collect voluntary and informed consent from the participating data subjects prior to the processing of their personal data within the pilot pursuant to Article 5 para. 1) of the Greek Data Protection Law 2472/1997. While doing so, the Controller will inform the concerned data subjects about the means and purpose of the personal data processing. Moreover, the Controller agrees to make available to the data subjects upon request a copy of this agreement.



4 Obligations of the Processor

Typically, the Processor gets only access to data which is non-personal. But for technical tasks, e.g. for error troubleshooting, it cannot be completely ruled out that it might become necessary that the Processor must get access to files which contain personal data. So the following obligations concern this exceptional case only. Therefore, the Processor agrees and warrants:

The Processor ensures that personal data processed on behalf of the Controller are processed strictly in compliance with the controller's instructions set out in this agreement and as given throughout the pilot. The processing of the personal data is limited to the purpose of the pilot, namely the usage of Privacy-ABC's in a real life environment.

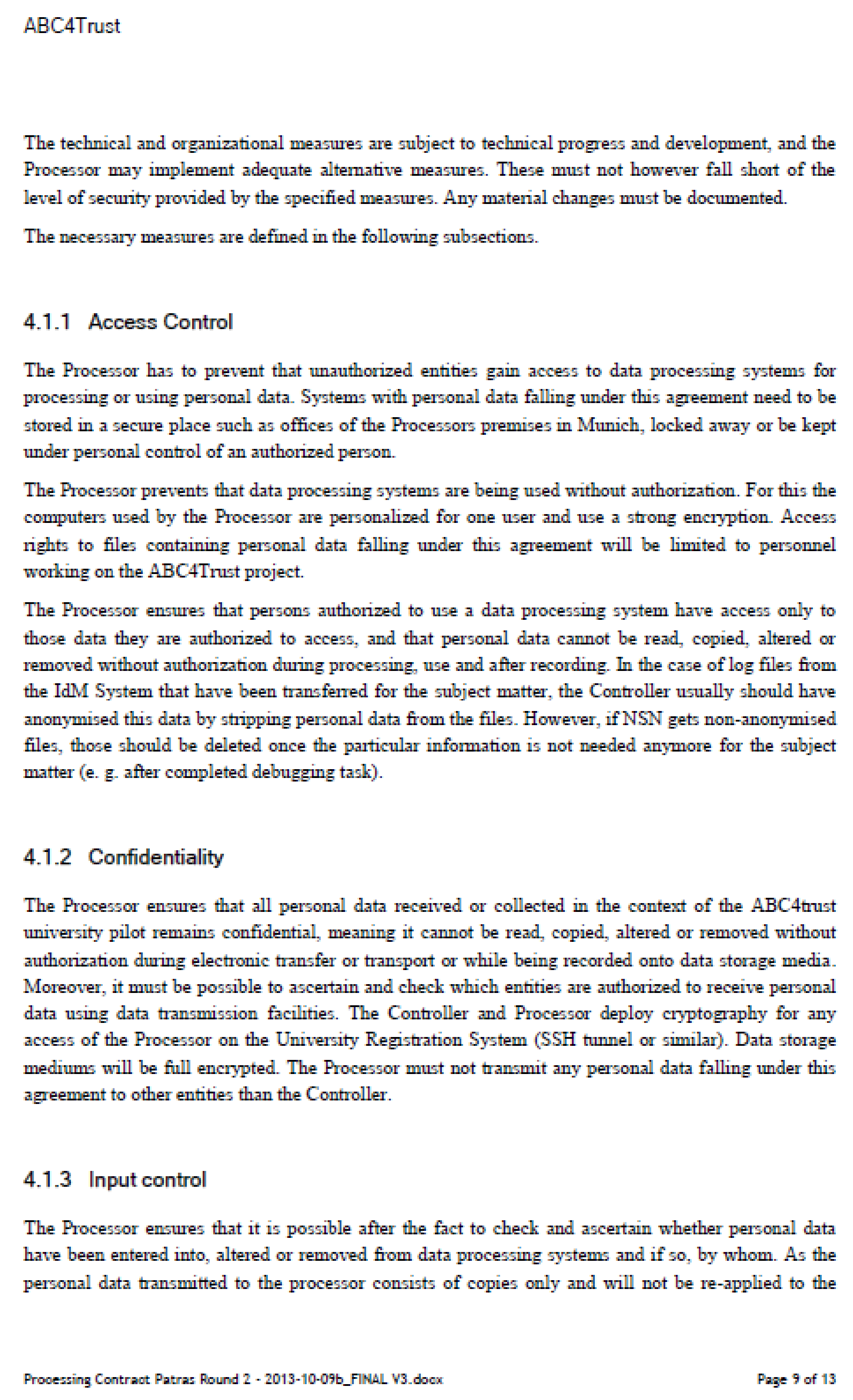
The Controller oversees the actions of the Processor within the Privacy ABC System. The Processor is obliged to provide the Controller with all necessary information which the Controller may not able to retrieve herself. This especially concerns data breach notifications and other issues which need problem resolve.

In those exceptional cases where the Processor gets access to files still containing personal data, he is obliged to take appropriate technical and organisational measures to protect the personal data of the pilot participants in accordance to Article 5 para. 1) of the Greek Data Protection Law 2472/1997. These measures are defined in the following section.

4.1 Technical and organisational measures

In this section, the necessary technical and organizational measures pursuant to Article 5 para. 1) of the Greek Data Protection Law 2472/1997, as to be implemented by the Processor under instruction of the Controller, are defined.

The Processor must document that necessary technical and organizational measures stipulated in this section are taken before starting to process the data, thereby giving details of the actual process to be followed, and must present this to the Controller for review. When accepted by the Controller, the documented measures will form the basis of the processing. If something raises the need for amendments, these must be applied amicably.



The technical and organizational measures are subject to technical progress and development, and the Processor may implement adequate alternative measures. These must not however fall short of the level of security provided by the specified measures. Any material changes must be documented.

The necessary measures are defined in the following subsections.

4.1.1 Access Control

The Processor has to prevent that unauthorized entities gain access to data processing systems for processing or using personal data. Systems with personal data falling under this agreement need to be stored in a secure place such as offices of the Processors premises in Munich, locked away or be kept under personal control of an authorized person.

The Processor prevents that data processing systems are being used without authorization. For this the computers used by the Processor are personalized for one user and use a strong encryption. Access rights to files containing personal data falling under this agreement will be limited to personnel working on the ABC4Trust project.

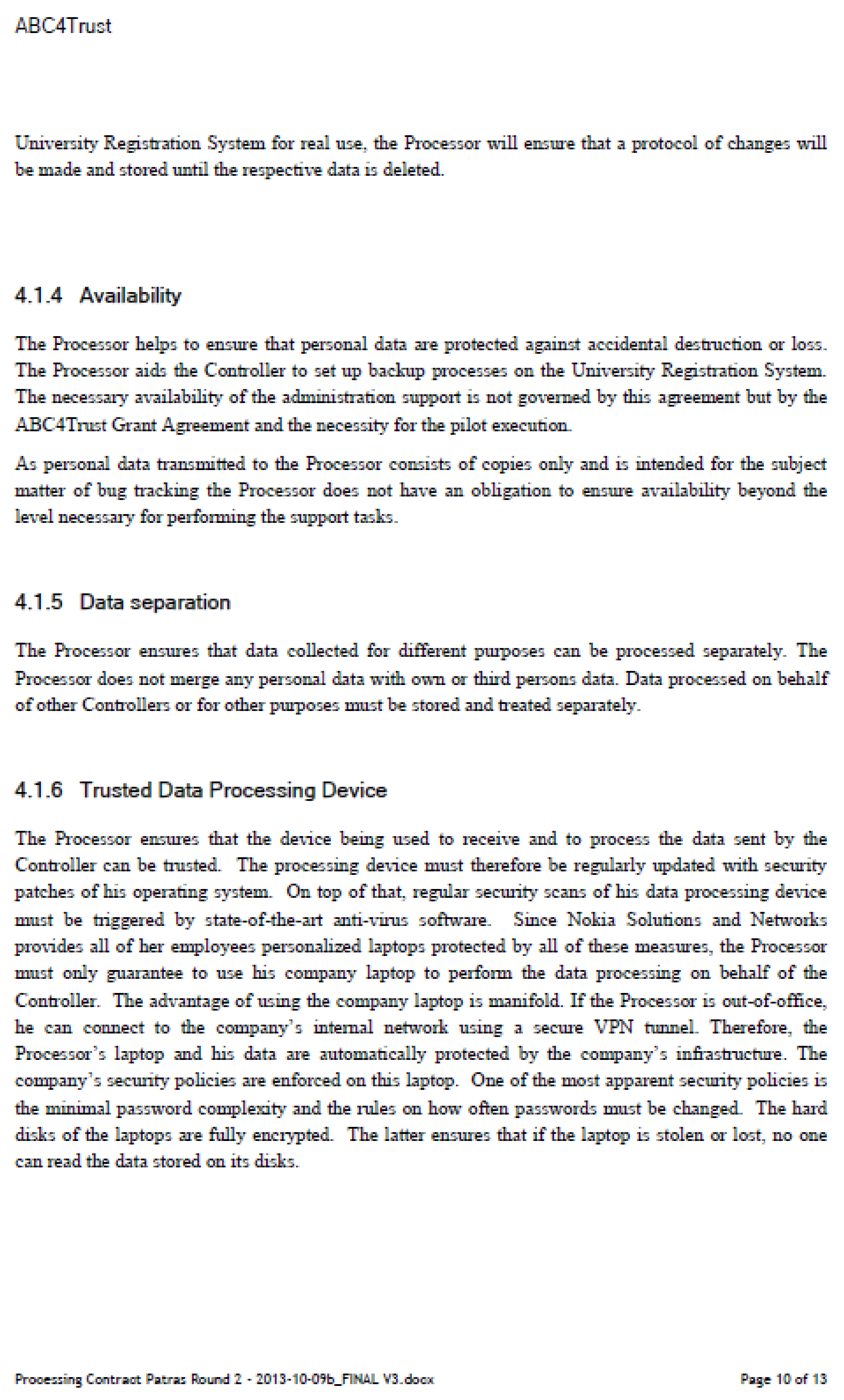
The Processor ensures that persons authorized to use a data processing system have access only to those data they are authorized to access, and that personal data cannot be read, copied, altered or removed without authorization during processing, use and after recording. In the case of log files from the IdM System that have been transferred for the subject matter, the Controller usually should have anonymised this data by stripping personal data from the files. However, if NSN gets non-anonymised files, those should be deleted once the particular information is not needed anymore for the subject matter (e. g. after completed debugging task).

4.1.2 Confidentiality

The Processor ensures that all personal data received or collected in the context of the ABC4trust university pilot remains confidential, meaning it cannot be read, copied, altered or removed without authorization during electronic transfer or transport or while being recorded onto data storage media. Moreover, it must be possible to ascertain and check which entities are authorized to receive personal data using data transmission facilities. The Controller and Processor deploy cryptography for any access of the Processor on the University Registration System (SSH tunnel or similar). Data storage mediums will be full encrypted. The Processor must not transmit any personal data falling under this agreement to other entities than the Controller.

4.1.3 Input control

The Processor ensures that it is possible after the fact to check and ascertain whether personal data have been entered into, altered or removed from data processing systems and if so, by whom. As the personal data transmitted to the processor consists of copies only and will not be re-applied to the



University Registration System for real use, the Processor will ensure that a protocol of changes will be made and stored until the respective data is deleted.

4.1.4 Availability

The Processor helps to ensure that personal data are protected against accidental destruction or loss. The Processor aids the Controller to set up backup processes on the University Registration System. The necessary availability of the administration support is not governed by this agreement but by the ABC4Trust Grant Agreement and the necessity for the pilot execution.

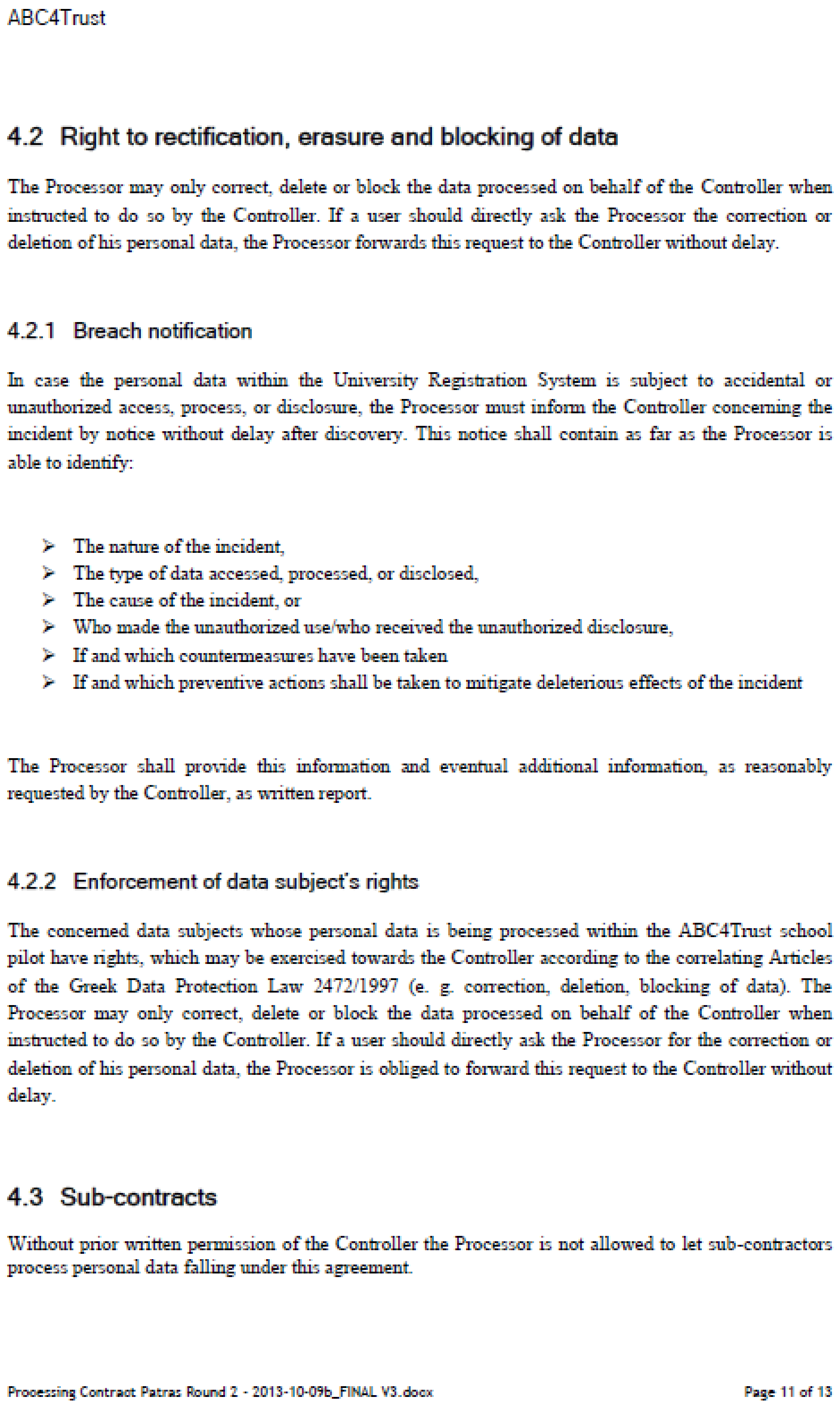
As personal data transmitted to the Processor consists of copies only and is intended for the subject matter of bug tracking the Processor does not have an obligation to ensure availability beyond the level necessary for performing the support tasks.

4.1.5 Data separation

The Processor ensures that data collected for different purposes can be processed separately. The Processor does not merge any personal data with own or third persons data. Data processed on behalf of other Controllers or for other purposes must be stored and treated separately.

4.1.6 Trusted Data Processing Device

The Processor ensures that the device being used to receive and to process the data sent by the Controller can be trusted. The processing device must therefore be regularly updated with security patches of his operating system. On top of that, regular security scans of his data processing device must be triggered by state-of-the-art anti-virus software. Since Nokia Solutions and Networks provides all of her employees personalized laptops protected by all of these measures, the Processor must only guarantee to use his company laptop to perform the data processing on behalf of the Controller. The advantage of using the company laptop is manifold. If the Processor is out-of-office, he can connect to the company's internal network using a secure VPN tunnel. Therefore, the Processor's laptop and his data are automatically protected by the company's infrastructure. The company's security policies are enforced on this laptop. One of the most apparent security policies is the minimal password complexity and the rules on how often passwords must be changed. The hard disks of the laptops are fully encrypted. The latter ensures that if the laptop is stolen or lost, no one can read the data stored on its disks.



4.2 Right to rectification, erasure and blocking of data

The Processor may only correct, delete or block the data processed on behalf of the Controller when instructed to do so by the Controller. If a user should directly ask the Processor the correction or deletion of his personal data, the Processor forwards this request to the Controller without delay.

4.2.1 Breach notification

In case the personal data within the University Registration System is subject to accidental or unauthorized access, process, or disclosure, the Processor must inform the Controller concerning the incident by notice without delay after discovery. This notice shall contain as far as the Processor is able to identify:

- The nature of the incident,
- The type of data accessed, processed, or disclosed,
- The cause of the incident, or
- Who made the unauthorized use/who received the unauthorized disclosure,
- If and which countermeasures have been taken
- If and which preventive actions shall be taken to mitigate deleterious effects of the incident

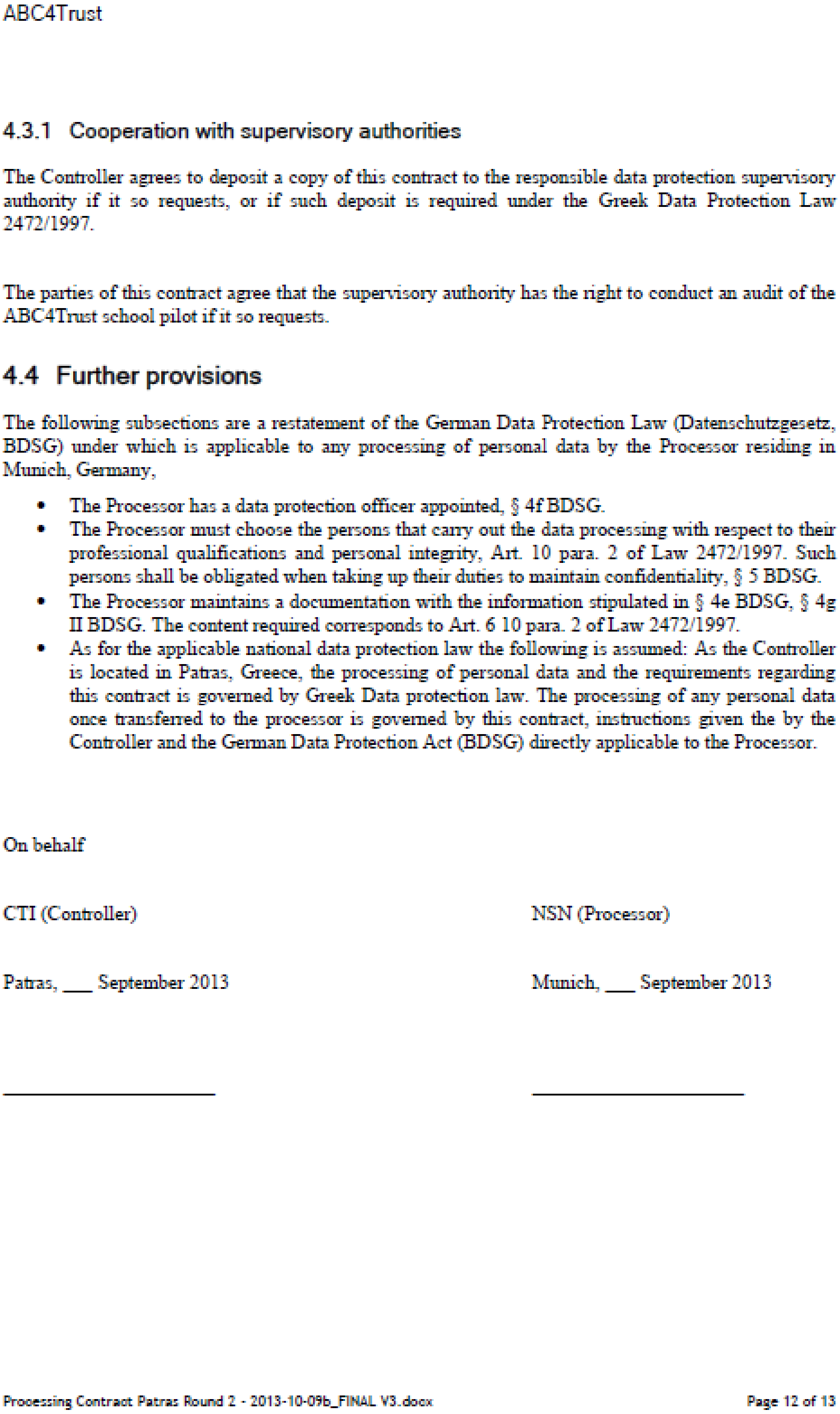
The Processor shall provide this information and eventual additional information, as reasonably requested by the Controller, as written report.

4.2.2 Enforcement of data subject's rights

The concerned data subjects whose personal data is being processed within the ABC4Trust school pilot have rights, which may be exercised towards the Controller according to the correlating Articles of the Greek Data Protection Law 2472/1997 (e. g. correction, deletion, blocking of data). The Processor may only correct, delete or block the data processed on behalf of the Controller when instructed to do so by the Controller. If a user should directly ask the Processor for the correction or deletion of his personal data, the Processor is obliged to forward this request to the Controller without delay.

4.3 Sub-contracts

Without prior written permission of the Controller the Processor is not allowed to let sub-contractors process personal data falling under this agreement.



4.3.1 Cooperation with supervisory authorities

The Controller agrees to deposit a copy of this contract to the responsible data protection supervisory authority if it so requests, or if such deposit is required under the Greek Data Protection Law 2472/1997.

The parties of this contract agree that the supervisory authority has the right to conduct an audit of the ABC4Trust school pilot if it so requests.

4.4 Further provisions

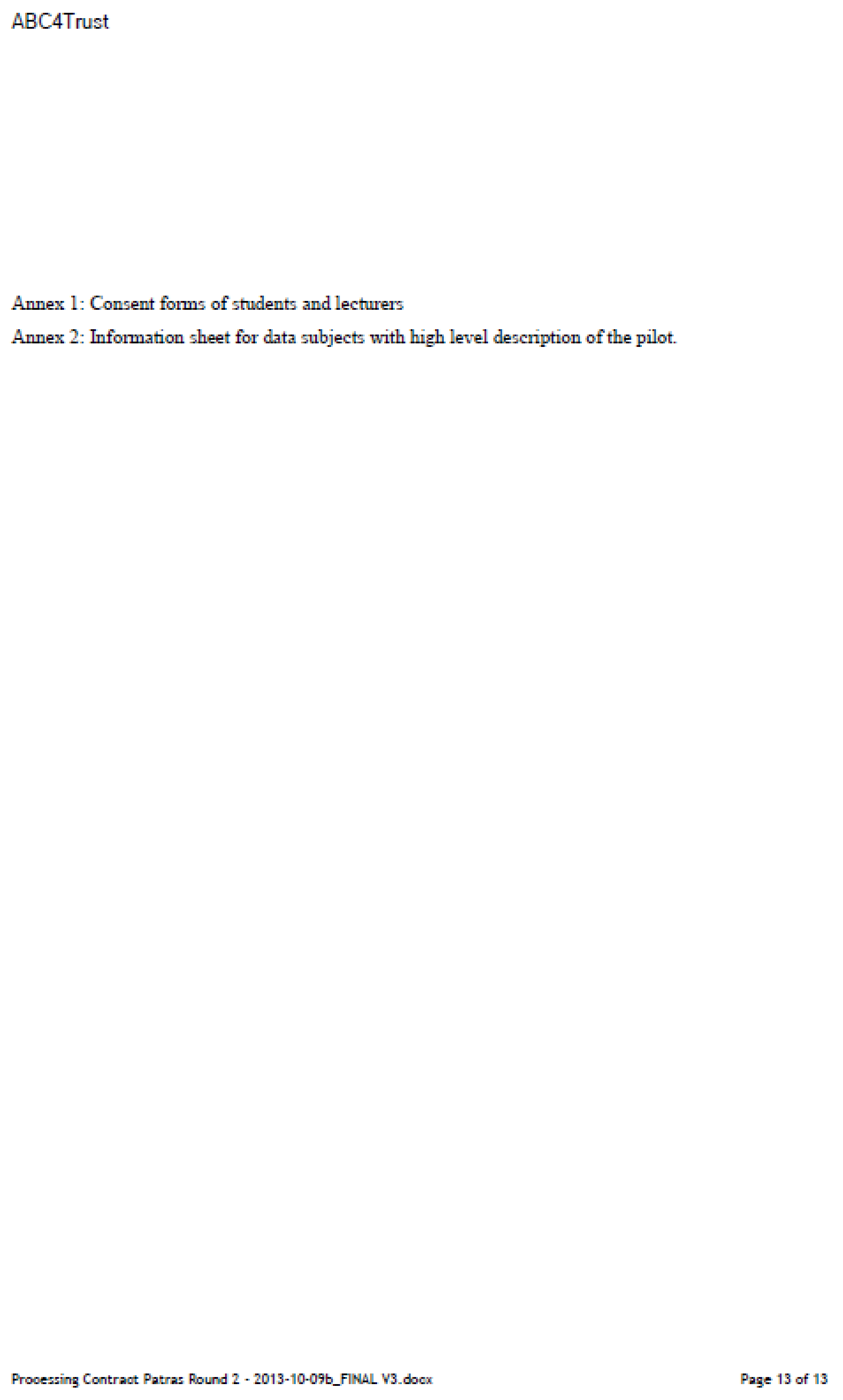
The following subsections are a restatement of the German Data Protection Law (Datenschutzgesetz, BDSG) under which is applicable to any processing of personal data by the Processor residing in Munich, Germany,

- The Processor has a data protection officer appointed, § 4f BDSG.
- The Processor must choose the persons that carry out the data processing with respect to their professional qualifications and personal integrity, Art. 10 para. 2 of Law 2472/1997. Such persons shall be obligated when taking up their duties to maintain confidentiality, § 5 BDSG.
- The Processor maintains a documentation with the information stipulated in § 4e BDSG, § 4g II BDSG. The content required corresponds to Art. 6 10 para. 2 of Law 2472/1997.
- As for the applicable national data protection law the following is assumed: As the Controller is located in Patras, Greece, the processing of personal data and the requirements regarding this contract is governed by Greek Data protection law. The processing of any personal data once transferred to the processor is governed by this contract, instructions given the by the Controller and the German Data Protection Act (BDSG) directly applicable to the Processor.

On behalf

CTI (Controller) NSN (Processor)

Patras, ___ September 2013 Munich, ___ September 2013



ABC4Trust

Annex 1: Consent forms of students and lecturers

Annex 2: Information sheet for data subjects with high level description of the pilot.

Processing Contract Patras Round 2 - 2013-10-09b_FINAL V3.docx Page 13 of 13

Glossary

Attribute

A piece of information, possibly certified by a credential, describing a characteristic of a natural person or entity, or of the credential itself. An attribute consists of an attribute type determining the semantics of the attribute (e.g., first name) and an attribute value determining its contents (e.g., John).

In the Swedish School Pilot the following attributes were used: *firstname, lastname, birthdate (age), gender, class, school name, roles, subjects, children and guardians*. The attribute guardian (issued to pupils) indicates a pupil's guardians. And the attribute child (issued to guardians) indicates the children of a guardian.

Access Policy

An access policy indicates who is allowed to enter and to use the functionality (read/write messages, upload/download documents etc.) of a Restricted Area. The XML Each Restricted Area has its own access policy stating who is entitled to access/enter a Restricted Area e.g. a chat room. The administrator of the chat room (normally the one who did create the chat room) can add one or several access policies indicating the Users or groups of Users that are allowed to enter and access the chat room. Access policies can also be a mixture of individuals and groups. For example:

- Only for 12-13 years
- Only for girls 12-13 years
- Only for boys older than 12 years
- Only for class 7A
- Claudia Hugosson
- Teachers

Access policies are translated into the XML style presentation policy alternatives via the XML Generator in the RA Application.

Alias

Within Restricted Areas, in particular in Chats and Discussion boards, Users are represented by a self-chosen nickname, their alias. Each alias can be chosen only once. The alias will be bound to the User credential while preserving unlinkability allowing the User to reclaim the alias for subsequent visits.

Certified pseudonym

A verifiable pseudonym based on a device secret that also underlies an issued credential. A certified pseudonym is established in a presentation token that also demonstrates possession of a credential bound to the same device as the pseudonym.

Credential

A list of certified attributes issued by an Issuer to a User. By issuing a credential, the Issuer vouches for the correctness of the contained attributes with respect to the User.

In the Swedish School Pilot the following credentials are used: *credSchool, credSubject, credChild, credGuardian and credRole.*

Credential specification

A data artifact specifying the list of attribute types that are encoded in a credential.

Key binding

An optional credential feature whereby the credential is bound to a strong secret so that any presentation token involving the credential requires the presence of the key.

IdM Database

The Identity Management Database is a database where all User data (attributes) needed to issue credentials are saved.

Inspection

An optional feature allowing a presentation token to be de-anonymized by a dedicated Inspector. At the time of creating the presentation token, the User is aware (through the presentation policy) of the identity of the Inspector and the valid grounds for inspection.

Inspection Board

In the Swedish Pilot the inspection board consists of three persons that in emergency situations will investigate if the inspection grounds are met. The inspection board will decide whether an inspection can take place or not. The decision is forwarded to the inspector who has the inspector key needed to perform an inspection.

Inspection grounds

The circumstances under which a Verifier may ask an Inspector to trace the User who created a given presentation token.

Inspector

A trusted entity that can trace the User who created a presentation token by revealing attributes from the presentation token that were originally hidden from the Verifier.

Issuance key

The Issuer’s secret cryptographic key used to issue credentials.

Issuer

The party who vouches for the validity of one or more attributes of a User, by issuing a credential to the User.

In the Swedish School Pilot the school is the Issuer.

Issuer parameters

A public data artifact containing cryptographic and other information by means of which presentation tokens derived from credentials issued by the Issuer can be verified.

Linkability

See *unlinkability*.

Presentation policy

A policy created and published by a Verifier specifying the class of presentation tokens that the Verifier will accept. The presentation policy contains, among other things, which credentials from which Issuers it accepts and which information a presentation token must reveal from these credentials.

Presentation policy alternatives

A choice/list for presentation policies.

Presentation token

A collection of information derived from a set of credentials, usually created and sent by a User to authenticate to a Verifier. A presentation token can contain information from several credentials, reveal attribute values, prove that attribute values satisfy predicates, sign an application-specific message or nonce or support advanced features such as pseudonyms, key binding, inspection, and revocation. The presentation token consists of the presentation token description, containing a technologies-agnostic description of the revealed information, and the presentation token evidence, containing opaque technologies-specific cryptographic parameters in support of the token.

Privacy-ABC

A common name to describe privacy friendly technologies developed within the ABC4Trust project.

Pseudonym

See *verifiable pseudonym*.

Pseudonym scope

A string provided in the Verifier's presentation policy as a hint to the User which previously established a pseudonym she can use, or to which a new pseudonym should be associated. A single smart card (with a single device secret) can generate multiple verifiable or certified pseudonyms for the same scope string, but can only generate a single scope-exclusive pseudonym.

Restricted Area

See *restricted area application.*

Restricted Area Application

The restricted Area Application is the school web application that contains all the functionality for chat, wall, documents uploading, counseling and political discussions. The restricted Area Application is also an tool that offers functionality to create, delete and update different Restricted Areas. Each Restricted Area is protected by one or several Access Policies indicating who is allowed to enter and access the content within the RA.

Revocation

The act of withdrawing the validity of a previously issued credential. Revocation is performed by a dedicated Revocation Authority, which could be the Issuer, the Verifier, or an independent third party. Which Revocation Authorities must be taken into account can be specified by the Issuer in the Issuer parameters (Issuer-driven revocation) or by the Verifier in the presentation policy (Verifier-driven revocation).

Revocation Authority

The entity in charge of revoking credentials. Multiple Issuers or Verifiers may rely on the same Revocation Authority.

Revocation information

The public information that a Revocation Authority publishes every time a new credential is revoked or at regular time intervals to allow Verifiers to check that a presentation token was not derived from revoked credentials.

Revocation parameters

The public information related to a Revocation Authority, containing cryptographic information as well as instructions where and how the most recent revocation information and non-revocation evidence can be obtained. The revocation parameters are static, i.e., they do not change every time a new credential is revoked or at regular time intervals like the revocation information and non-revocation evidence (may) do.

Non-revocation evidence

The User-specific or credential-specific information that the User agent maintains, allowing it to prove in presentation tokens that the credential was not revoked. The non-revocation evidence may need to be updated either at regular time intervals or when new credentials are revoked.

Pilot User Number

Pilot User Number (PUN) is a number (10 digits) used in the pilot to uniquely identify the Users. The PUN consists of the birthdate of the User and a number (980112-XXXX). The PUN used in the pilot is not the same as the Swedish Civic Registration Number.

Scope

See *pseudonym scope.*

Scope-exclusive pseudonym

A certified pseudonym that is guaranteed to be cryptographically unique per scope string and per device secret. Meaning, from a single device bound key, only a single scope-exclusive pseudonym can be generated for the same scope string.

Traceability

See *untraceability*.

Unlinkability

The property that different actions performed by the same User, in particular different presentation tokens generated by the same User, cannot be linked to each other as having originated from the same User.

Untraceability

The property that an action performed by a User cannot be traced back to her identity. In particular, the property that a presentation token generated by a User cannot be traced back to the issuance of the credential from which the token was derived.

User

The human entity who wants to access a resource controlled by a Verifier and obtains credentials from Issuers to this end.

The Users in the Swedish School Pilot are pupils, guardians and school personnel.

The Users in the Patras Pilot are students.

User agent

The software entity that represents the human User and manages her credentials.

Device binding

An optional credential feature whereby the credential is bound to an underlying device secret. By requiring multiple credentials to be bound to the same secret, one can prevent Users from “pooling” their credentials.

Device secret

A piece of secret information known to a device (a strong random secret) underlying one or more issued credentials or pseudonyms. A presentation token involving a pseudonym or a device-bound credential implicitly proves knowledge of the underlying secret.

Verifiable pseudonym

A public identifier derived from a device secret allowing a voluntarily link to different presentation tokens or to re-authenticate under a previously established pseudonym by proving knowledge of this secret. Multiple unlinkable pseudonyms can be derived from the same device secret.

Verifier

The party that protects access to a resource by verifying presentation tokens to check whether a User has the requested attributes. The Verifier only accepts credentials from Issuers that it trusts.

In the Swedish scenarios the component that acts as a Verifier is the restricted area system. This component will interact with the IdM application and IdM Portal to grant access to those Users that satisfy the access policy for a given restricted area. The Issuer that this Verifier trusts is the school administration office – which is the only Issuer within the pilot.

List of Acronyms

ABCs	Attribute Based Credentials
ABCE	ABC Engine
admin	Short form of 'administrator'
API	Application Programming Interface
CA	Certificate Authority
CE	Crypto Engine
cred*Name*	Short form of Privacy-ABC credential of type *Name*
ENISA	European Network and Information Security Agency
FP7	Framework Programme 7
FQDN	Fully Qualified Domain Name
GUI	Graphical User Interface
HQAA	Hellenic Quality Assurance Agency
HTTP	Hypertext Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ICT	Information and Communications Technologies
ID	Identifier
Idemix	IBM Identity Mixer
IdM	Identity Manager
ISO	International Organisation for Standardisation
IdSP	Identity Service Provider
JDK	Java Development Kit
JSF	Java Servlet Faces
JSP	Java Server Pages
LDAP	Lightweight Directory Access Protocol
NFC	Near Field Communication
OTP	One Time Password
OS	Operating System
PC	Personal Computer
PET	Privacy Enhancing Technologies
PRIME	Privacy and Identity Management for Europe
PrimeLife	Privacy and Identity Management in Europe for Life
PIN	Personal Identification Number
Privacy-ABCs	Privacy Attribute Based Credentials (privacy ABCs)
PUK	Personal Unblocking Key
PUN	Pilot User Number
RA	Restricted Area
REST	Representational State Transfer
RevAuth	Revocation Authority
RP	Relying Party
SOAP	Simple Object Access Protocol

SC	Smart Card
SCI	Smartcard Interface
SSL	Secure Sockets Layer
STS	Secure Token Service
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UFW	Ubuntu Firewall
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WP	Work Package
WSDL	Web Services Description Language
XML	eXtensible Markup Language

Bibliography

[H22] Patrik Bichsel, Jan Camenisch, Maria Dubovitskaya, Robert R. Enderlein, Ioannis Krontiris, Anja Lehmann, Gregory Neven, Janus Dam Nielsen, Christian Paquin, Franz-Stefan Preiss, Kai Rannenberg, Michael Stausholm, Harald Zwingelberg

H2.2 ABC4Trust Architecture for Developers, Version 1.0, 2013

https://abc4trust.eu/download/ABC4Trust-H2.2_ABC4Trust_Architecture_for_Developers.pdf

[D41] H. Guldager and J. Dam Nielsen

D4.1 Initial Reference Implementation, Version 1, 2012

[D51] S. Bcheri, N. Götze, V. Liagkou, A. Pyrgelis, C. Raptopoulos, Y. Stamatiou, K. Storf, P. Wängmark, Harald Zwingelberg

D5.1 Scenario Definition for both Pilots, Version 1, 2012

https://abc4trust.eu/download/ABC4Trust-D5.1-Scenario-Definition.pdf

[D52] Joerg Abendroth, Souheil Bcheri, Norbert Götze, Vasiliki Liagkou, Monika Orski, Robert Seidl, Fatbardh Veseli

D5.2 Description of the “Common Denominator” Elements , Version 1, 2012

https://abc4trust.eu/download/ABC4Trust-D5.2-Description-of-the-Common-Denominator-Elements.pdf

[D52a] Joerg Abendroth, Souheil Bcheri, Ioannis Krontiris, Vasiliki Liagkou, Ahmad Sabouri, Eva Schlehahn, Fatbardh Veseli, Harald Zwingelberg

D5.2 Amendment Building Blocks of ABC Technologies, Version 1, 2013

https://abc4trust.eu/download/ABC4Trust-D5.2a_Amendment_Building_Blocks_of_ABC_Technologies.pdf

[D61] Souheil Bcheri, Norbert Goetze, Monika Orski, Harald Zwingelberg

D6.1 Application Description for the School Deployment, Version 1.1, 2012

https://abc4trust.eu/download/ABC4Trust-D6.1-Application-Description-School.pdf

[D62] Joerg Abendroth, Souheil Bcheri, Kasper Damgaard,Hamza Ghani, Jesus Luna, Gert Læssøe Mikkelsen, Maxim Moneta, Monika Orski, Neeraj Suri, Harald Zwingelberg

D6.2 Necessary hardware and software package for the school pilot deployment

https://abc4trust.eu/download/ABC4Trust-D6.2.Hard-and-Software-Package-for-School-Pilot.pdf

[D71] J. Abendroth, V. Liagkou, A. Pyrgelis, C. Raptopoulos, A. Sabouri, E. Schlehahn, Y. Stamatiou and H. Zwingelberg

D7.1 Application Description for Students, Version 1, 2012

https://abc4trust.eu/download/ABC4Trust-D7.1-Application-Description-Students.pdf

[D72] Kasper Damgaard, Hamza Ghani, Norbert Goetze, Anja Lehmann, Vasiliki Liagkou, Jesus Luna, Gert Læssøe Mikkelsen, Apostolos Pyrgelis, Yannis Stamatiou

D7.2 Necessary hardware and software package for the student pilot deployment

https://abc4trust.eu/download/ABC4Trust-D7.2.Hard-and-Software-Package-for-Student-Pilot.pdf

[Mono] Mono Project http://mono-project.com/Main_Page

[DoW] SEVENTH FRAMEWORK PROGRAME

THEME [ICT-2009.1.4]

[Trustworthy ICT]

Grant Agreement no. 257782

2012