D6.2 Necessary hardware and software package for the school pilot deployment

Joerg Abendroth, Souheil Bcheri, Kasper Damgaard, Hamza Ghani, Jesus Luna, Gert Læssøe Mikkelsen, Maxim Moneta, Monika Orski, Neeraj Suri, Harald Zwingelberg

Editors:	*Souheil Bcheri, Monika Orski*
Reviewers	*Ahmad Sabouri, Marit Hansen, Joerg Abendroth*[1]
Identifier:	*D6.2*
Type:	*Deliverable*
Version:	*1.0*
Date:	*29/03/2013*
Status:	*Final*
Class:	*Public*

Abstract

In this document we provide the details of the architecture and implementation of the Söderhamn school pilot system components, as well as their API mapping with the first version of the ABC4Trust reference implementation. We explain how these components interact among themselves and with the pilot users. We provide the details of their set-up, initialization, and proper operation within the servers as well as the clients installed on users' computers. We also provide an analysis of the legal aspects of the pilot, and the results of a preliminary risk analysis of the pilot system.

[1] Reviewer of the parts that were not written by himself

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.	Johann Wolfgang Goethe – Universität Frankfurt	GUF	Germany
2.	Alexandra Institute AS	ALX	Denmark
3.	Research Academic Computer Technology Institute	CTI	Greece
4.	IBM Research – Zurich	IBM	Switzerland
5.	Miracle A/S	MCL	Denmark
6.	Nokia-Siemens Networks Management International GmbH	NSN	Germany
7.	Technische Universität Darmstadt	TUD	Germany
8.	Unabhängiges Landeszentrum für Datenschutz	ULD	Germany
9.	Eurodocs AB	EDOC	Sweden
10.	CryptoExperts SAS	CRX	France
11.	Microsoft NV	MS	Belgium
12.	Söderhamn Kommun	SK	Sweden

List of Contributors

Chapter	Author(s)
Executive Summary	Souheil Bcheri (EDOC), Monika Orski (EDOC)
1. Introduction	Souheil Bcheri (EDOC), Monika Orski (EDOC)
2. School Pilot Context	Göran Hanell (SK), Emma Nilsson (SK), Monika Orski (EDOC)
3. School Pilot Architecture	Joerg Abendroth (NSN), Souheil Bcheri (EDOC), Gert Læssøe Mikkelsen (ALX), Maxim Moneta (EDOC), Monika Orski (EDOC)
4. Key Scenarios and Important Use Cases	Souheil Bcheri (EDOC), Gert Læssøe Mikkelsen (ALX), Gregory Neven (IBM), Monika Orski (EDOC)
5. Client deployment	Gert Læssøe Mikkelsen (ALX), Monika Orski (EDOC)
6. School Registration System Deployment	Joerg Abendroth (NSN), Monika Orski (EDOC)
7. Restricted Area System Deployment	Maxim Moneta (EDOC), Monika Orski (EDOC)
8. Smart Card Deployment	Kasper Damgaard (ALX), Gert Læssøe Mikkelsen (ALX)
9. API Mapping	Tsvetoslava Vateva-Gurova (TUD), Hamza Ghani (TUD)
10. Network Set Up and Operation	Maxim Moneta (EDOC), Monika Orski (EDOC)
11. Legal Aspects	Harald Zwingelberg (ULD), Eva Schlehahn (ULD)
12. Risk Management	Jesus Luna (TUD), Hamza Ghani (TUD), Neeraj Suri (TUD)
13 Conclusion	Eva Schlehahn (ULD)

Executive Summary

In this deliverable we provide the details of the implementation, set-up and operation of the system that will be employed in the Söderhamn school pilot of the ABC4Trust project: A community interaction platform with protection of the users' privacy. The design, implementation, and testing of the pilot system was based on the use cases and pilot requirements documented in deliverables D5.1, D5.2, and 6.1 as well as the first version of the ABC4Trust reference implementation of Privacy-ABCs provided by WP4.

The *architecture* of the pilot system, as explained in the deliverable, is comprised of the following main components: (i) the *School Registration System*, which is responsible for storing data (attribute values) about the users and for issuing credentials to the students, (ii) the *Restricted Areas (RA) System*, which supports the community interaction functions, (iii) the *ABC Systems*, which provide verifications of tokens created by the User Client, (iv) the *User Client*, which provide controls of user credentials, (v) the *Client Browser*, which handles browser communication with the RA System and requests to the ABC System, and (vi) the *Smart Cards* and *Smart Card Readers*, which are distributed to the users and store users' credentials. Additional two components in the pilot are the Revocation Authority, which is implemented as an Issuer-driven revocation and the Inspector Application.

In the sections that follow, we describe the deployment of each part, the key scenarios and corresponding API mappings, the legal aspects, and what has been done to mitigate risks during the pilot. In all chapters, the descriptions emphasise on the privacy aspects.

Table of Contents

Index of Figures

Index of Tables

1 Introduction

In this chapter we give a general overview of the School pilot. Moreover we present the scope and the structure of this document.

1.1 The Söderhamn School Pilot

The School Pilot will take place in Norrtullskolan, an elementary school in Söderhamn, Sweden. It is part of the ABC4Trust project to show that the realisation of applications using Privacy-ABCs preserves the anonymity of the users and offers the required level of privacy. The school pilot will use Privacy-ABC technologies to enable secure authentication in communications between pupils, guardians and school personnel. The ABC4Trust architecture takes into account the three aspects of identity, anonymity and privacy, and combines them into one single solution.

The pilot in Söderhamn will consider several types of communication needed by the school:

- Chat communication
- Political discussions
- Counselling with health personnel
- Documents access and sharing

Chat rooms to be used by pupils and school personnel, rooms for anonymous political discussions, online counselling sessions where school personnel provide counselling in a privacy preserving environment while pupils are not required to state their identity, and document areas where school personnel can share documents (e.g. grades and development plans) with pupils and their guardians. The pilot will implement a communication portal, the Restricted Areas System, using Privacy-ABCs to verify users' credentials while protecting their privacy.

The users will be pupils, school personnel, and the pupils' parents / guardians. Each user will be issued a smart card, containing all necessary components in order to download the user's credentials from IdM database (The Issuer). Card readers will be distributed to all the users, to let them use their own PC as the client device. A number of card readers will also be available in the school, connected to common computers.

1.2 Structure of the document

Chapter 2 introduces the context of the Söderhamn school pilot deployment.

Chapter 3 shows the overall architecture and how the components of the pilot fit together.

Chapter 4 contains the most important use cases, thus providing a functional view as the basis for the more technical views in the following chapters.

Chapter 5 describes the client side deployment.

Chapter 6 describes the deployment of the school registration system.

Chapter 7 describes the deployment of the restricted area system, the pilot's own server side.

Chapter 8 describes the deployment of the smart cards, where credentials are stored.

Chapter 9 provides an overview of the API mappings within the pilot.

Chapter 10 describes the network setup, at the school where the pilot's servers are located.

Chapter 11 describes the legal aspects of the pilot.

Chapter 12 provides a risk analysis and a description of how the risks are managed.

Chapter 13 provides a conclusion

Chapter 14 Glossary

Chapter 15 Acronyms

Chapter 16 Bibliography

2 School Pilot Context

The pilot site is a school in Söderhamn, a few hours north from Stockholm, with a population of almost 26 000 inhabitants. The school Norrtullskolan is educating about 580 students from the age of six to sixteen and has about 80 employees.

Norrtullskolan has students from many different nationalities; Pupils not only from Sweden, but also from Somalia, Turkey, Iraq, Iran, Afghanistan, Egypt, Eritrea and many other countries. Due to this mixture of origins, the pupils learn a lot about different cultures and life styles.

Through the years the school personnel have tried to make this school the best environment for both students and teachers. Norrtullskolan has a vision to erase all bullying, discrimination and other self-esteem lowering treatments, and the pupils are also involved in this project.

An important fact about Norrtullskolan is that they have adopted computers in their education system. Computers are used not only by the school personnel, but also by the pupils as part of their education. Thus, the school network and computer literacy are already established.

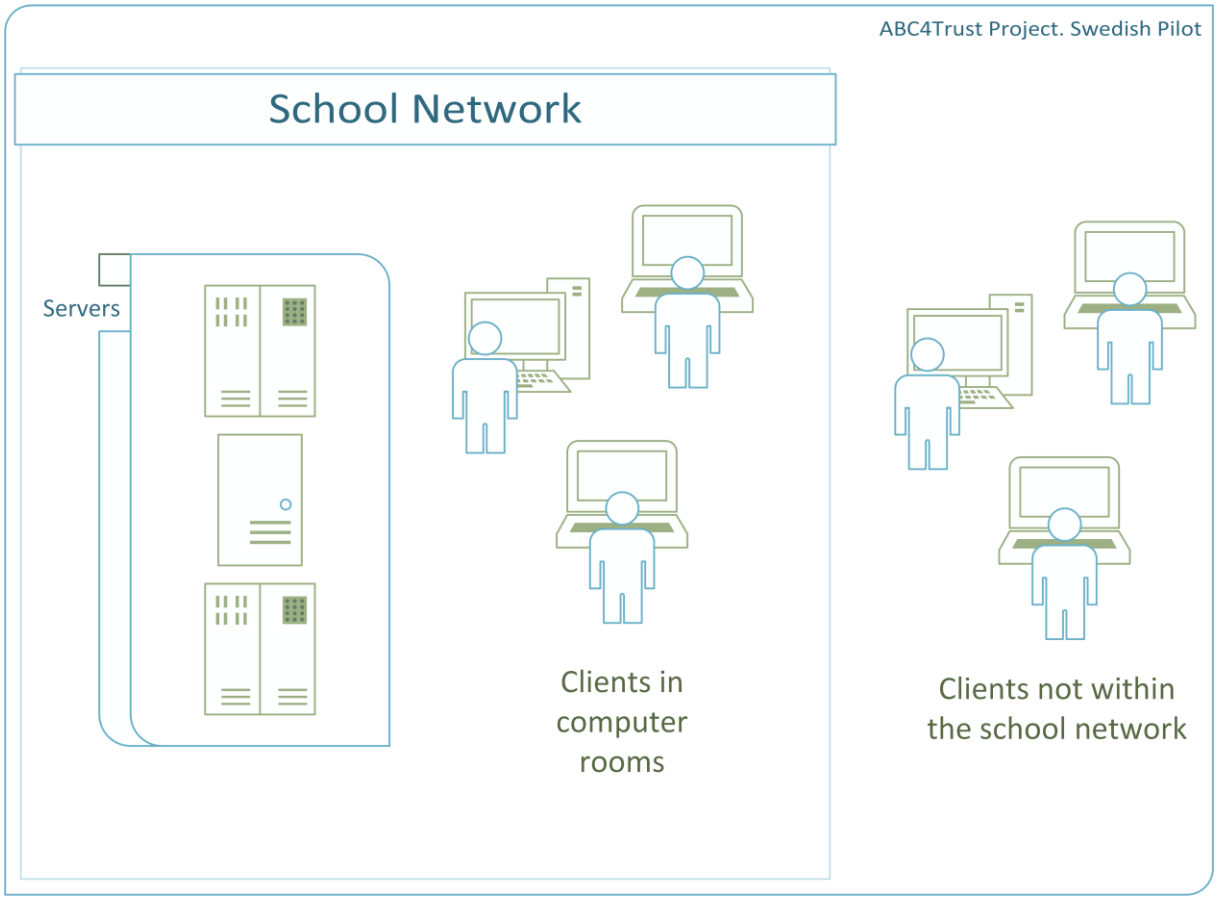

Figure 1 Overview of the school pilot

The school pilot context consists of the servers hosting the restricted area, the school registration system and the user clients. The servers will be placed at Norrtullskolan and will be part of the school's network. Clients are installed at all computers with a smart card reader attached, which includes computers within the school as well as in the pupils' homes.

Thus, the pilot includes servers within a secured network, internal to the school, with Internet access to the servers, and clients in different locations, accessing the servers through the Internet.

2.1.1 Assumptions and the Scope of the Pilot

As the focus of the pilot is to test Privacy-ABC technology in a real life situation and to give feedback about the results some assumption have been made. The trial will be conducted using other existing technologies but the main focus will remain on testing Privacy-ABC technology. This section gives a short summary of the assumptions made for the pilot and the reader can find more details in Chapter 11 Legal Aspects and Chapter 12 Risk Management.

Some aspects of privacy and anonymity are not object of the ABC4Trust research project on data protection compliant authentication for online services. The pilot provides anonymity or pseudonymity for the application level only. The connection to the computer network that involves further layers to enable the communication (cable, electromagnetic waves) is not in focus of the pilot.

Privacy-ABCs work on the application layer but malicious service providers may nevertheless identify users by analyzing information from the underlying network levels, e.g. IP-addresses. While the anonymization on network layers is not within scope of the ABC4Trust, we provide some pointers to existing privacy-enhancing technologies (PETs) for these purposes in section 11.3.

The anonymization of communication on the layers 6 to 1 is not within the scope of the ABC4Trust project. However, the broader concept of ABC4Trust foresees and references to existing technologies providing further protection of users.

On the network and transport layer, anonymization is possible by deploying mix-services such as TOR or AN.ON to effectively hide the own IP-address form the relying party – in case of the Swedish use case, hide it from the Restricted Area System and the School Registration System.

The ABC4Trust project will not provide TOR or AN.ON clients or services

The threat modelling presented in Chapter 12 is based on a series of trust assumptions derived from the current pilot's deployment, namely:

- There exists three well-differentiated trust boundaries (User, Eurodocs' premises and Inspector premises), each one under a different administrative domain. These are further explained in Section 12.1.1
- Involved entities, processes, data stores and data flows (as shown in Figure 44) are considered to be secured under the due diligence principle, that is, best effort mechanisms. Assumptions include:
    - Users protecting their smartcard's PIN/PUK and, not disclosing their real-identity to other entities via exchanged messages (e.g., while chatting in Restricted Areas).
    - All entities taking minimum countermeasures to protect their hardware/software (e.g., via antivirus/personal firewalls and, keeping updated/patched software systems).
    - Use of authenticated and encrypted communication channels (e.g. through SSL/TLS).
- Users trust in Issuers, Revocation Authority and Inspector.
- Verifiers, Revocation Authority and Inspector trust in Issuers.
- Involved processes (e.g., ABCE and IdM) are considered correct and secure.
- The attacker model (further explained in Chapter 12) considers malicious and skilled insiders/outsiders, with a finite amount of resources (e.g., they are not able to break the underlying crypto).

The threat analysis discussed in this document aims to be comprehensive under the previous set of trust assumptions, therefore might not apply to production environments where e.g., zero-day attacks and operational errors compromise the overall system.

3 School Pilot Architecture

In this chapter we provide a high level description of the systems that are deployed in the School pilot. Together, they form the ecosystem in which the ABC4Trust technology is used.

3.1 Architecture Plan

Figure 2 depicts an overview of the components of the pilot architecture. This is a more detailed and updated version of the pilot architecture set out in [BGOZ12].

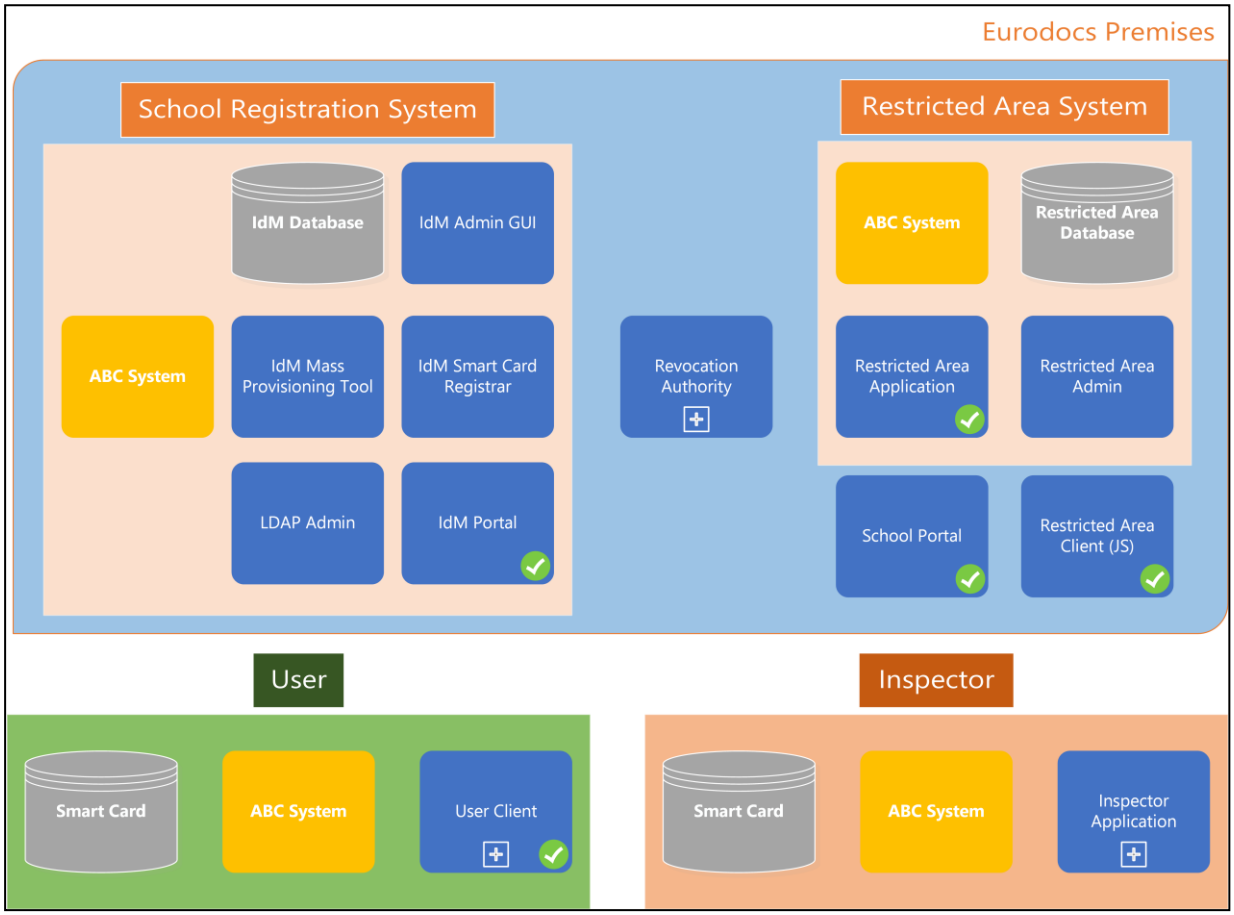

Figure 2: High Level Architecture of the School Pilot

The school pilot contains clients distributed in the school's internal network as well as to pupils' homes, and servers placed within the school network (not at EDOC's premises, as the school network is more secure). The servers are logically divided into the School Registration System (on a virtual server of its own), the Revocation Authority and the Restricted Area System. The Restricted Area System contains all the logic used within the restricted areas and a web server front. Separate instances of ABC Systems are used by the School registration system and the Restricted Area System.

The overall layout of data storage is shown in Figure 3.

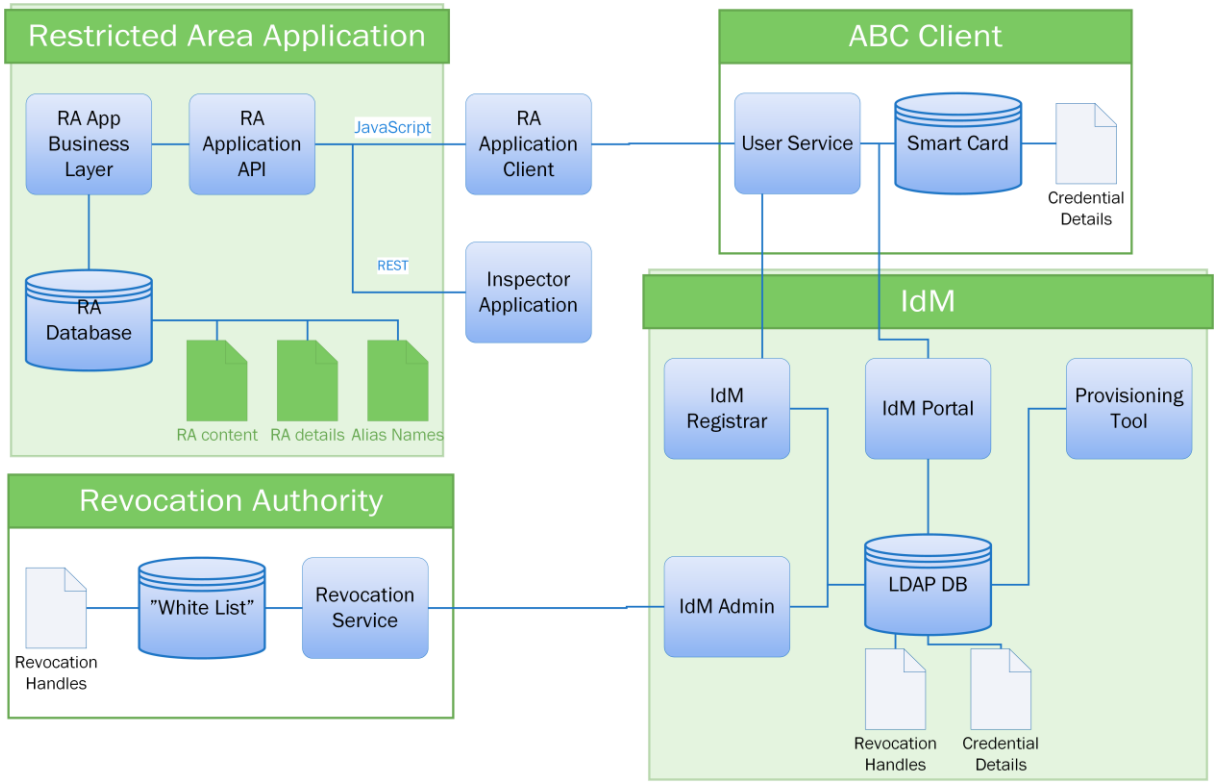

Figure 3: Data flow and storage within the architecture

3.1.1 Architecture Components

Here we provide a list of all names used in the deliverable to describe the main architecture and components used in pilot system. The main parts are the following:

- School Registration System
- Restricted Area System
- Inspector Application (can be also named Inspector Client in some cases)
- ABC System (meaning ABC Core Components)
- ABC Systems (meaning different configuration of ABC System in particular case)
- User Client

To be more specific Restricted Area System consists of:

- Restricted Area Application (sometimes is called School Application)
- Restricted Area Admin
- School Portal (website with links to different applications and help information)
- Restricted Area Client (JavaScript client coming from Restricted Areas Application incl. Dashboard, Alias Selector etc.)

School Registration System includes following:

- IdM Portal
- IdM Smart Card Registrar

- IdM Admin GUI
- IdM LDAP Admin
- IdM Mass Provisioning Tool

When the ABC System or ABC Core Components are mentioned it means the System with following components:

- ABCE
- Crypto Engine
- Revocation Proxy
- Key Manager
- External Device Crypto Interface
- External Device Data Interface

ABC Systems is implemented and used such as Verifier ABC System, Issuer ABC System, User ABC System, Inspector ABC System and Revocation ABC System.

User Client includes:

- Browser Plugin (replaces the term of Firefox Plugin, Browser Client)
- User ABC System (is an entity of ABC System with configuration for usage on client side; in some context can be named as User Service, ABC Client or ABCE+CE)
- User ABCE GUI (replaces the term of Identity Selector)
- Smart Card Software (APDU command interpreter)
- PCSC (incl. driver for reader etc. provided by the PC OS)

3.2 Server Side Architecture

3.2.1 School Registration System

The main component of the school registration system runs in java servlet container, such as Tomcat and is based on a Linux system. The school registration system includes the ABC System, which runs on its own virtual machine that is based on windows operating system. A smart card reader component is also included in the client part of the ecosystem.

3.2.2 Restricted Area System

The Restricted Area (RA) system is built using .NET Framework 4.0 and the C# programming language, and 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, will run within a Java environment, also on Windows 2008 Server.

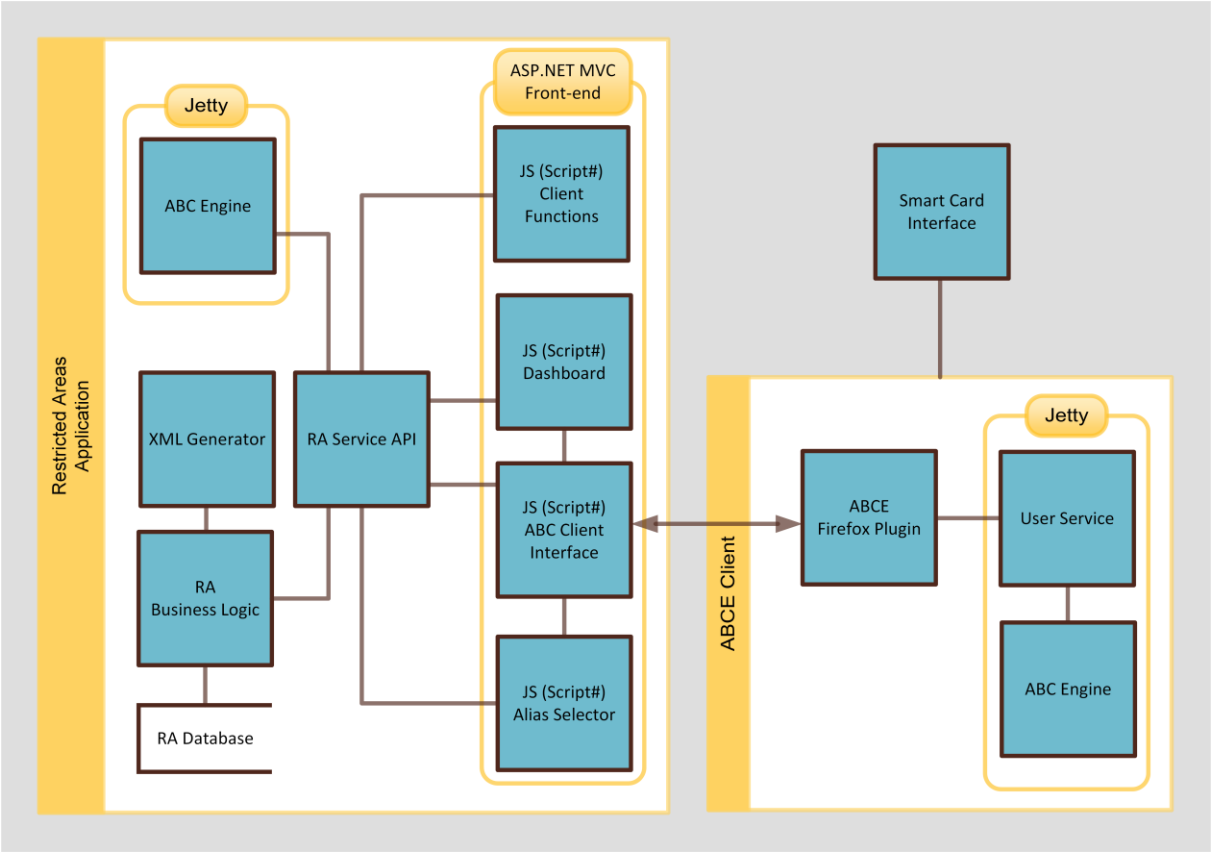

Figure 4: Restricted Area server and client test environment, with components within server and client

Figure 4 shows both the server side and the client side of the Restricted Area Application. The server side is implemented in C# and has dependencies to .NET Framework 4.0, 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 server are the business logic entities, the XML Generator and its related components, and an API to provide connections for the RA 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 server 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.

3.2.3 Revocation Authority

For various reasons the validity of issued credentials might have to end prior the initial set time, and in general it is often necessary to be able to revoke credentials. A user might lose control over her smart card, the role of a user changes, a user is no longer to be part of the system, or a user has not followed certain rules associated with a credential. In any of these cases the authority (the school administration) that issued the credential should be able to revoke it in a way that does not interfere with the privacy properties of the ABC technology. When a credential is revoked, it is invalidated system wide, and no verifier should trust this credential anymore. This type of revocation is known as *issuer driven* revocation. For this to be implemented there is a Revocation Authority in the system.

The Revocation Authority will generate revocation handles and send these to the issuer (the IdM), which will embed a revocation handle in each revocable credential. This communication between the issuer (the IdM) and the Revocation Authority is done directly between the two ABC engines, and does not involve the above application layer. The Revocation Authority maintains a list of the valid revocation handles, and makes it possible for the users to obtain a non-revocation evidence from which the user anonymously can prove to a verifier (in the case of this pilot, the Restricted Area Application) that the credential has not been revoked. The user's ABC engine obtains the non-revocation evidence directly from the ABC engine of the Revocation Authority, without involving the above application layer. Revocation of credentials is performed by the school administrator using the IdM Admin GUI to select which user and which credential to revoke. The IdM Admin GUI sends the revocation handle of the credential to the revocation authority, which removes the revocation handle from the list of valid revocation handles and updates the non-revocation evidences for all users. The revocation process is described in more details in chapter 4.15 **Revocation**.

In practice in the Söderhamn pilot, the IdM will be administered using the IdM Admin GUI but it will also be possible to use LDAP tools to change the attributes values in the IdM database and to revoke credentials. LDAP tools are the "LDAP Mass Provisioning Tool" and the "LDAP Admin tool". Only school administration officials will be able to revoke credentials. When the school administrator uses the IdM Admin GUI (or LDAP tool) to change the value of one attribute, the IdM will automatically perform a revocation of the corresponding credential containing the old attribute value. This ensures that any valid credential always contains the same attribute value as the IdM.

3.3 Client Side Architecture

3.3.1 User ABC System

The User ABC System software consists of two parts: the ABC System and the User Service. The ABC System, consisting of the ABC engine and a crypto engine, is responsible for all lower layers, including handling credentials, smart cards, policies etc. and if possible (if the user have credentials that fulfils the policy) given the users credentials, providing presentation tokens fulfilling the requested policies. The User Client is made as a browser (Firefox and Internet Explorer) plugin and the ABC System is made as an application server executed locally on the user's computer. The User Service is supplying a user interface, making the user capable of choosing between different credentials if more than one fulfils the requested policy. Moreover, registration, backup/restoration and other management of smart cards are also possible for the user through the browser plugin. This software both the browser plugin and the ABC System needs to be installed locally on the computers the users use for the pilot, however it is combined into a single installation package. For a more thorough description of the ABC System we refer to deliverable D4.1 of this project [IRI2012].

3.3.2 Browser Plugin

All client side usage will be through a web browser on the user's computer.

To enhance the user experience the functionality of an alias selector and a dashboard has been developed. And as to avoid linkability between the user's different aliases and to avoid the risk of traceability between the user's usages of different restricted areas, the alias selector and the dashboard functionalities are implemented as client components. In the pilot they are implemented in the Restricted Area application as browser client part. This browser client provides the following two important features:

- Alias Selector
- Dashboard

The Alias Selector handles the list of aliases owned by the user. It is designed to create new aliases, delete old ones and switch between them. Alias information is stored on the user's smart card. When the Alias Selector has to be rendered, the client makes a call to the ABCE user service via a plugin, to get the content of BLOB area of the card. After retrieving the list of alias IDs and names, the corresponding aliases are rendered into UI element. Whenever an alias operation is performed, e.g. a switch between aliases, the client sends a request to the ABCE to generate a Presentation Token and compares it to the one saved in the Restricted Area database during creation. The user's aliases are saved and retrieved from the user's own smart card.

The Dashboard is the part of the client that allows a user to see the Restricted Areas she recently accessed, or marked as favourite, as well as private Restricted Areas. To avoid linkability this has to be done in a separate requests to the database for each alias. A list of aliases is taken from the Alias Selector, to avoid extra requests to the smart card. The Dashboard loads alias IDs and makes calls to the RA server to retrieve the list of Restricted Areas for the active alias. Then, the Restricted Areas are rendered on Dashboard as UI elements. Thus, the Dashboard is not performing operations on the card content itself, it's just there to create the dynamic type of output which lets the user have a personalised start page view.

To avoid any additional modules and plugins, client-side JavaScript is used. In the pilot, this JavaScript code is generated during the build from Script# (Script Sharp). Using Script# made it easier to organize the JS code. UI elements use the jQuery framework, and some JavaScript code from Twitter Bootstrap. Twitter Bootstrap is a CSS framework, based on LESS. KnockoutJS has been used to tie everything within the client together.

The browser client needed to access the Restricted Area Admin interface (web page) during the pilot is Internet Explorer version 8 and later. It is developed using ASP.NET WebForms and has simple and easy to use interface.

The browser client will use plugins to communicate with the ABC client (see above).

3.3.3 Inspector Application

The inspector application is a special case of the client use. The inspection tokens are encrypted with the inspector's public key. They can be retrieved from the database by Eurodocs staff and transferred to the Inspector. Since the Inspector ABC System includes Inspector Service by default, the inspector uses his/her private key and Inspector client with a regular ABC System to inspect the token.

The Inspector service part of the client is a simple .NET WinForms application that is able to contact ABC System, and pass the private key and the inspection token retrieved from the Restricted Area database to the Inspector Service. After getting the decrypted reply from the Inspector Service, it is shown to the Inspector on the screen.

4 Key Scenarios and Important Use Cases

Some basic functions of the school pilot will be:

- School registration and connection to the ABC systems.
- Creation of Restricted Areas of different kinds. The RA has a set of restrictions (access policies), specifying who is allowed to enter it. There are RAs for chat rooms, counselling, political discussions, and sharing documents.
- Matching of credentials and restrictions (access policies) to determine who is allowed to access a certain RA.
- Identity disclosure in some situations. A function to let the user disclose her identity when she chooses to do so. School personnel (the counsellors) are automatically disclosing their identity within the management of counselling sessions.
- Inspection: A user's identity can be revealed, but only by a set of mostly manual steps, if the content of a message would contain serious threats. An inspection can take place only if it is compliant with the inspection grounds.

Below, these key scenarios are detailed by important use cases, with a reasonable amount of technical details.

Use Cases:

4.1 ABC System Setup

4.2 Smart Card Registration using One-Time-Password

4.3 Subsequent Logins to the School Registration System

4.4 Obtaining the School Credential

4.5 Other School Credentials

4.6 Viewing User's Data

4.7 Login to Restricted Area Application

4.8 Choose or Create Alias

4.9 Instantiating a Restricted Area

4.10 Access to a Restricted Area

4.11 Counselling

4.12 Restricted Chat Room

4.13 Political Discussions

4.14 Sharing Documents

4.15 Revocation

4.16 Emergency Situation (Inspection)

4.17 Viewing / Deleting Credentials

4.18 Changing PIN

4.19 Unlocking the Smart Card with a PUK

4.20 reIssuance of U-Prove Tokens

Involved Parties:

School Administrators

Eurodocs Administrators

School Inspection Board

School Personnel (Teachers, Counsellors), Guardians, Pupils

Restricted Area System

School Registration System (IdM Application, IdM Portal, ABC System)

IdM Smart Card Registrar

IdM Admin GUI

User Client

4.1 ABC System Setup

This use case describes the prior pilot setup. The Eurodocs administrators use smart card registration to create all needed parameters (cryptographic and non-cryptographic). In this context they also initialize the cards with the trusted parameters and trigger the pupils', guardians', and personnel's cards to release a scope-exclusive pseudonym (via the IdM Smart Card Registrar), which will be used later on to identify the authorised cards. They also initialize the inspector's card with an inspector key to release an inspector public key.

Prerequisite: School maintains a list/records of all participating pupils, personnel, guardians.

School has obtained non-initialized smart cards.

- Generate Privacy-ABC parameters:
  - Credential specification school credential (credSchool), class credential (credClass), guardian credential (credGuardian), child credential (credChild), role credential (credRole), and subject credential (credSubject)
  - System parameters (trusted groups, generators for commitments, pseudonyms....)
  - Issuer parameters and issuer secret key for each of the above credential specifications (i.e., issuerSchool, issuerClass, issuerGuardian, issuerChild, issuerRole, issuerSubject)
  - Revocation authority parameters
  - Inspectors' public & secret keys

- Initialize & register pupil/guardian/personnel smart cards
  - Personalize the card by printing name, logotype etc.

- Initialize card which triggers the generation of the secret key in the trusted part of the device using a unique device ID, and retrieve PIN and PUK from the card Trigger the generation of the secret key in the trusted part of the device
- Set either U-Prove or Identity Mixer cryptographic parameters on card.
- Invoke the card to generate a scope-exclusive pseudonym for the scope "*urn:soderhamn:registration*" (see Figure 5, Figure 6 and Figure 7) →Requires knowledge of PIN
- Store the scope-exclusive pseudonym along with the smart card ID and an 'unregistered' flag in the IdM database's list of 'established pseudonyms'

- Distribute User smart cards, together with PIN & PUK and one-time-password
- Distribute Inspector smart cards, together with PIN & PUK

- Package system parameters, issuer parameters, revocation authority parameters, and inspection public key into customized version of ABC client software

Figure 5: Smart Card Registration I

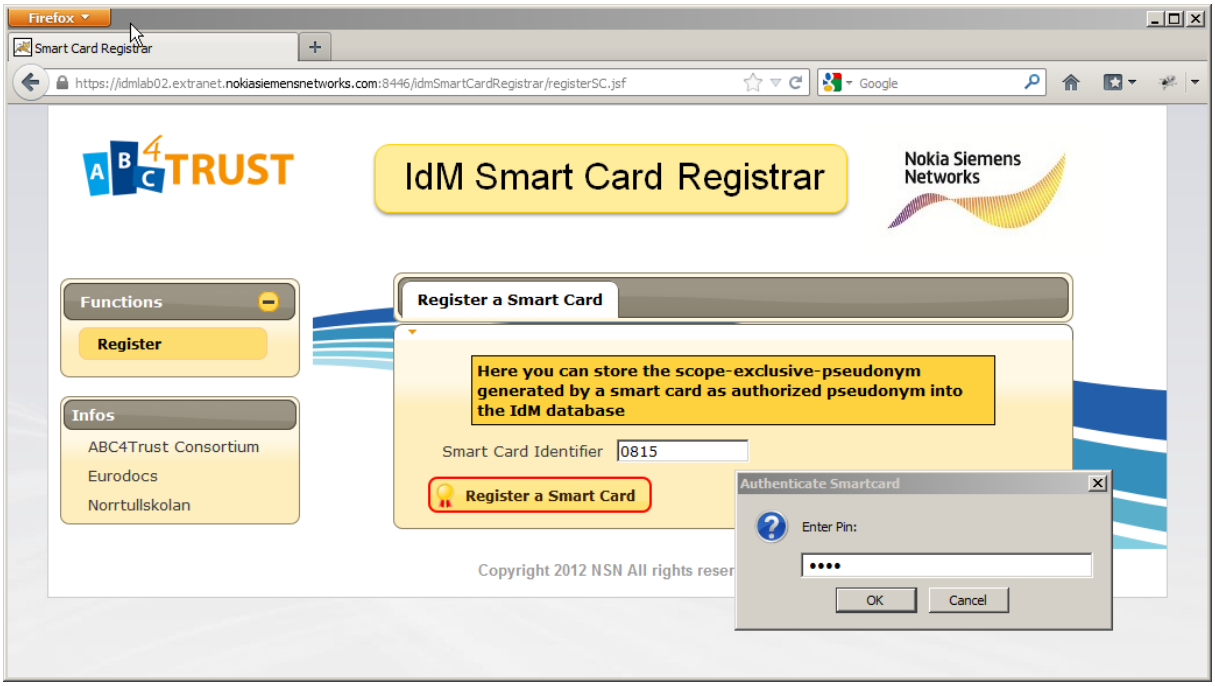

Figure 6: Smart Card Registration II

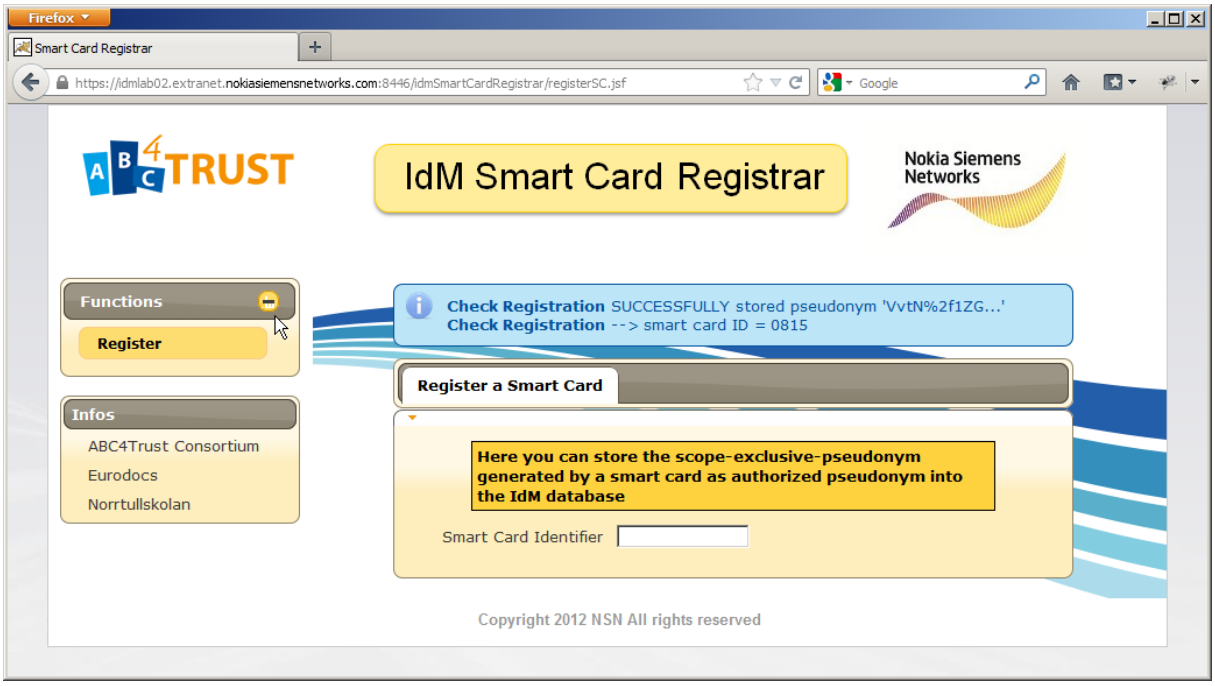

Figure 7: Smart Card Registration III

4.2 Smart Card Registration using One-Time-Password

This use case describes the steps needed for a user to claim ownership of an authorized/established scope-exclusive pseudonym.

Prerequisite: The User (pupils/guardians/personnel) has received her smart card with PIN & PUK and her one-time-password.

The School Registration System has stored the authorized scope-exclusive pseudonym from the User's card in the IdM database during system setup.

- The User presses the login button in the IdM Portal and is automatically forwarded to the IdM Application (see Figure 8 and Figure 9). Here, the User authenticates herself via her one-time-password. After successfully authenticating herself, the User is forwarded back to the IdM Portal. Now the User has only the options to either register her smart card or to log out (see Figure 10). If the User chooses to register her smart card, the IdM Portal forwards a policy asking to forward a token containing a scope-exclusive pseudonym based on the scope "urn:soderhamn:registration" to the User. This policy is the same for all Users:

```
<abc:PresentationPolicyAlternatives
xmlns:abc="http://abc4trust.eu/wp2/abcschemav1.0"  Version="1.0">
    <abc:PresentationPolicy PolicyUID="urn:soderhamn:policies:loginPseudonym">
        <abc:Message>
            <abc:Nonce>some fresh nonce</abc:Nonce>
        </abc:Message>
        <abc:Pseudonym Exclusive="true" Scope="urn:soderhamn:registration"/>
    </abc:PresentationPolicy>
</abc:PresentationPolicyAlternatives>
```

- The User invokes the `UserABCE.createPresentationToken()` method with the received presentation policy to obtain the presentation token containing the requested pseudonym, which will be automatically generated if it does not exist already. The generation of the presentation token requires presence of the smart card.
- The IdM Portal verifies the token by calling the `VerifierABCE.verifyTokenAgainstPolicy()` method of the ABCE. In particular it checks whether the presented pseudonym is among the established ones from the system setup phase and is marked as "unregistered". The latter check of the status of the pseudonym will be handled by the IdM Portal (i.e., not by the ABCE).).
- The IdM Portal copies the scope-exclusive pseudonym to the User's data set and marks this pseudonym as 'claimed'. During the verification of the presentation token, the IdM Portal queries the ABCE, which type of smart card has been used and adds the cryptoEngine parameter value to the User's data set. Note: The cryptoEngine parameter must be provided to the ABCE during Issuance.
- After successfully registering the User's smart card, the one-time-password is disabled.

For any subsequent login of the user to the IdM Portal, the user must authenticate herself in the IdM Application using Privacy-ABC technology only.
After successful registration of the User's smart card, the IdM Portal now offers other menu items to the User (see Figure 12)

- to obtain credentials

- view the data stored in the School Registration System about her

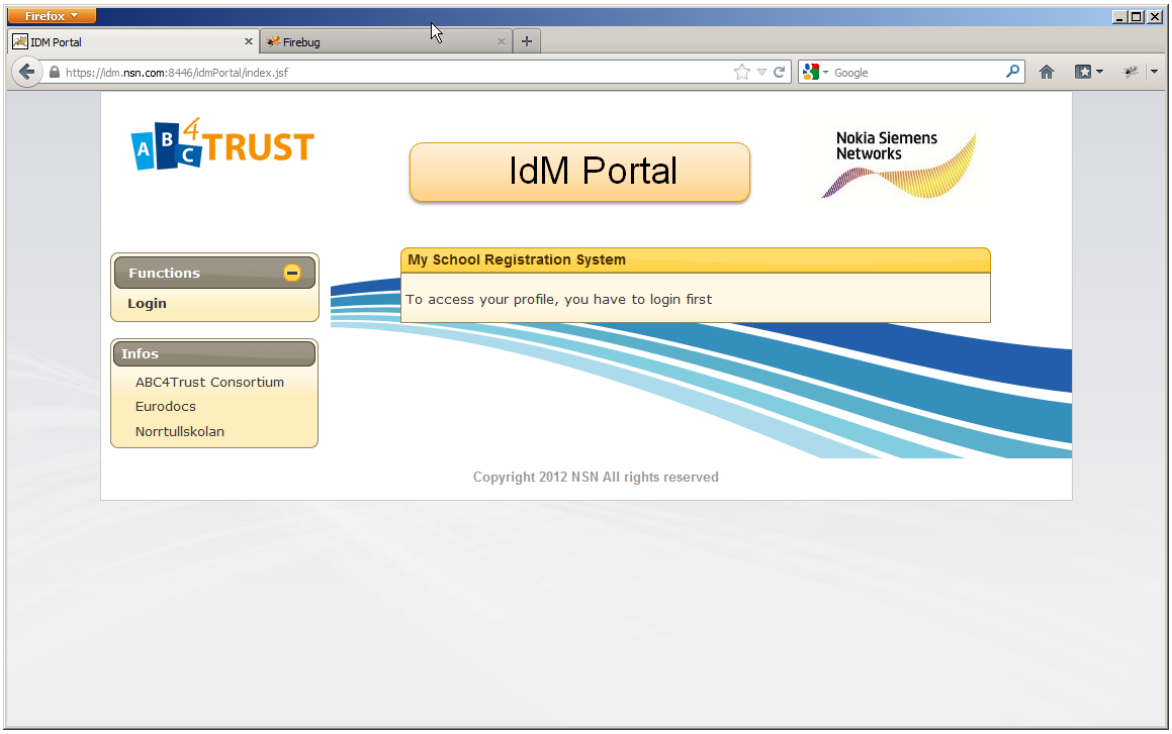

Figure 8: IDM Portal

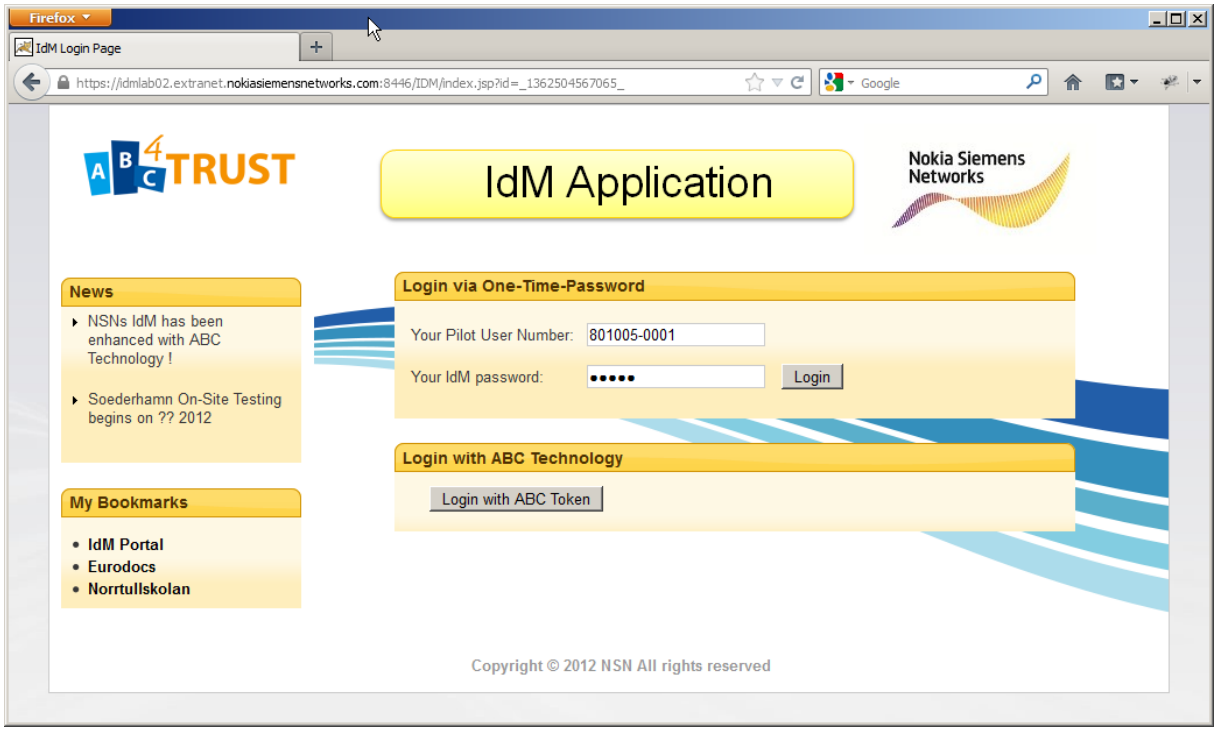

Figure 9: IDM Application Login Page

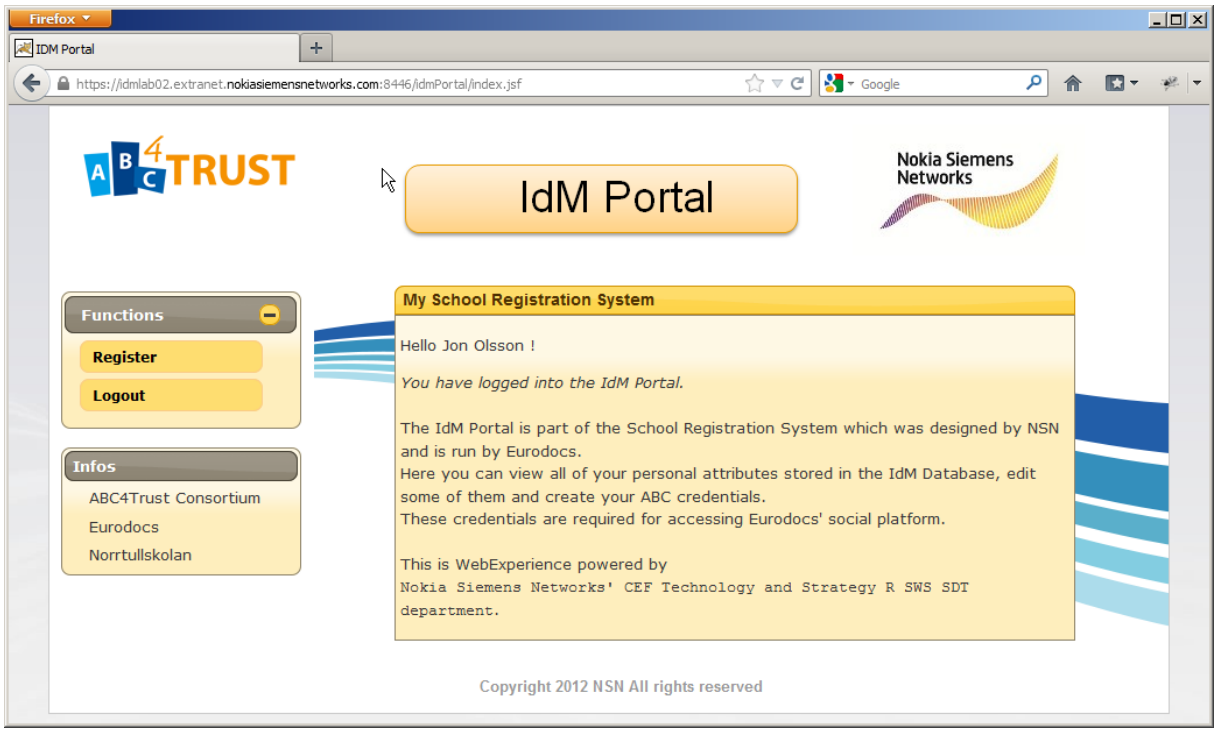

Figure 10: Successful login into IdM Portal

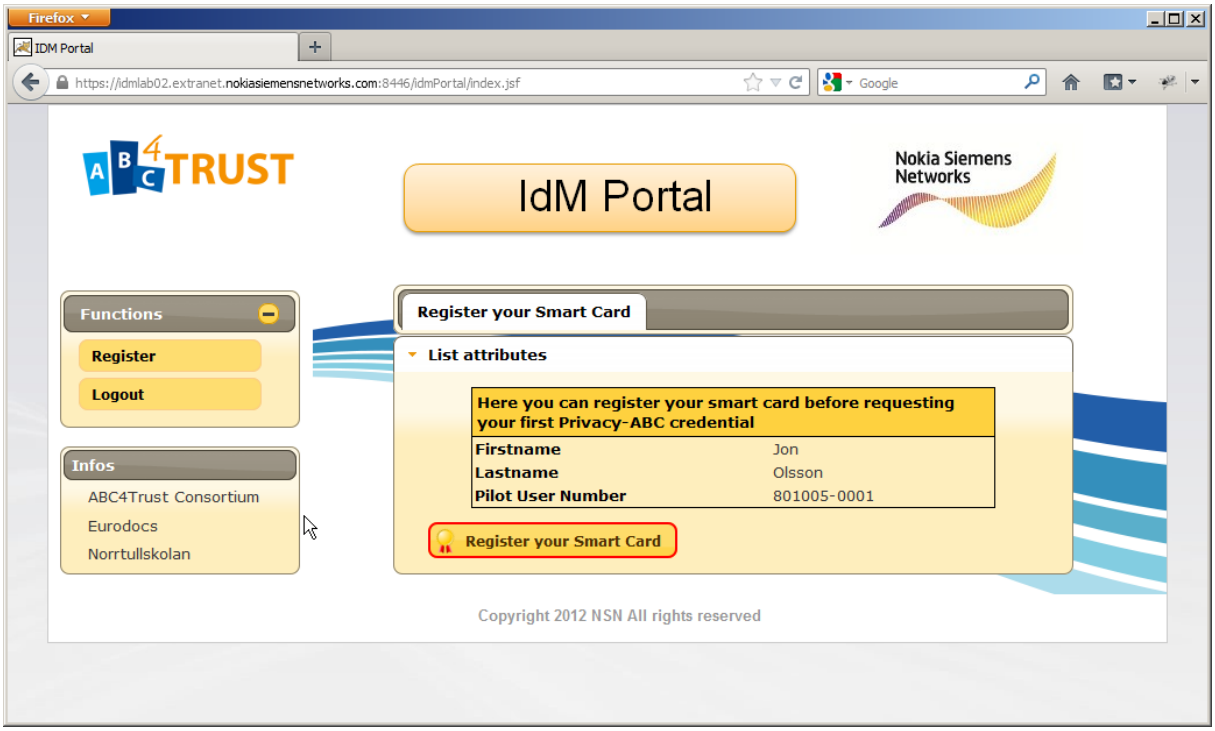

Figure 11: Register User's Smart Card in IdM Portal to his account

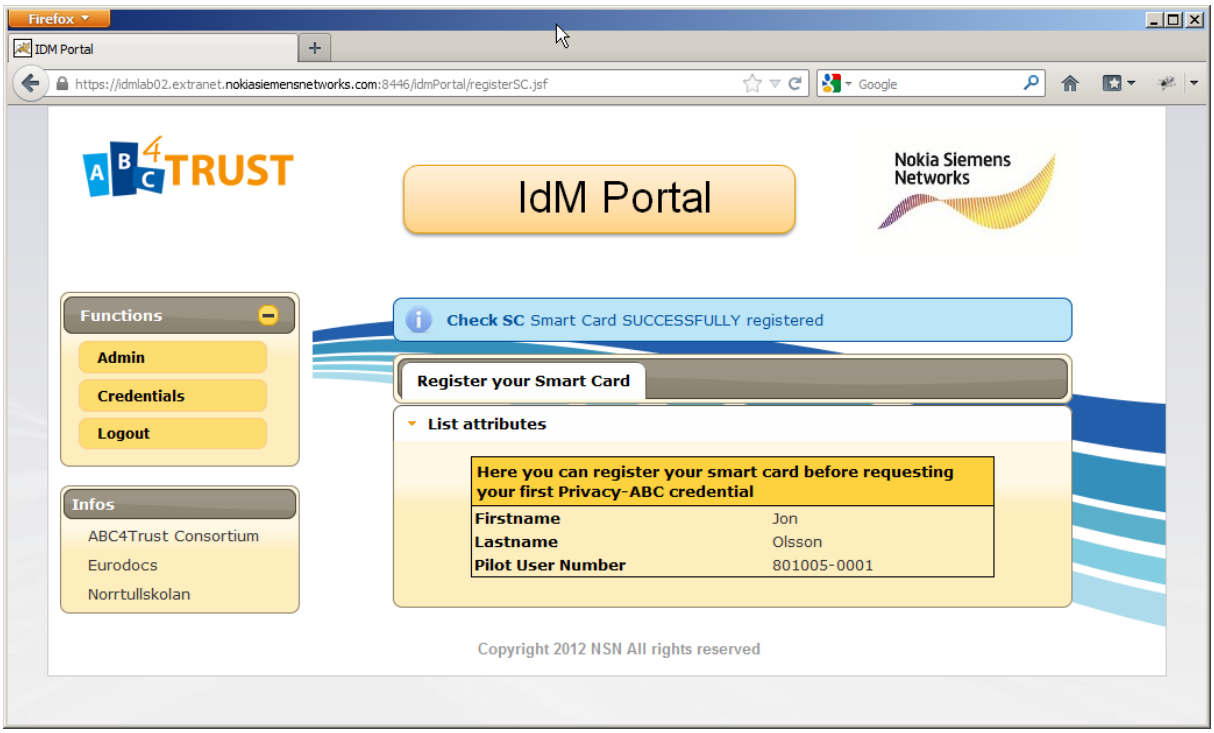

Figure 12: Successful Registration of Smart Card via IdM Portal

4.3 Subsequent Logins to the School Registration System

This use case describes the steps needed for a User to login to the IdM Portal after successfully registering her smart card (see previous chapter, use case **Smart Card Registration using One-Time-Password**).

Prerequisite: The User (pupils/guardians/personnel) has received her smart card with PIN & PUK.

The User has registered her smartcard (see 4.2).

- The User presses the login button in the IdM Portal and is automatically forwarded to the IdM Application. Here, the User authenticates herself via ABC technology. Logging in via ABC technology requires that the User enter her PIN. After successfully authenticating herself, the User is forwarded back to the IdM Portal. The IdM Application forwards the same policy which the IdM Portal forwarded to the User when she registered her smart card. Please note that the "Nonce" is a random value generated on the fly during every login procedure.
- See previous chapter for details.

4.4 Obtaining the School Credential

This use case describes the steps needed for a user to receive the main credentials that includes the identity information and can be used to prove towards the Restricted Areas that she is a registered user in the school pilot.

Prerequisite: The User (pupils/guardians/personnel) has received her smart card with PIN & PUK.

The User has logged in to the IdM Portal using her registered smart card.

- The User selects the "Credentials" button and is forwarded to the menu item "School"
- Now the User can see all attribute values which will be added to her credSchool (see Figure 13).
- After pressing the "Get Credential" button, the IdM Portal will forward the issuance policy to the User which is the same for all Users.
- The School Registration System forwards a policy asking to forward a token containing a scope-exclusive pseudonym based on the scope "urn:soderhamn:registration". This policy is the same for all Users:

```
<abc:IssuancePolicy Version="1.0"
xmlns:abc="http://abc4trust.eu/wp2/abcschemav1.0">
    <abc:PresentationPolicy PolicyUID="urn:soderhamn:policies:issuance">
        <abc:Message>
            <abc:FriendlyPolicyName lang="en">Policy: Authorized Users
only</abc:FriendlyPolicyName>
            <abc:FriendlyPolicyDescription lang="en">This policy will request the
pupil to present the established scope-exclusive Pseudonym with the scope
"urn:soderhamn:registration".No Privacy ABCs are required for this
step.</abc:FriendlyPolicyDescription>
        </abc:Message>
        <abc:Pseudonym Exclusive="true" Scope="urn:soderhamn:registration"
Established="false" Alias="#nym"/>
    </abc:PresentationPolicy>
    <abc:CredentialTemplate SameKeyBindingAs="#nym">

<abc:CredentialSpecUID>urn:soderhamn:credspec:credSchool</abc:CredentialSpecUID>

<abc:IssuerParametersUID>urn:soderhamn:issuer:credSchool</abc:IssuerParametersUID>
    </abc:CredentialTemplate>
</abc:IssuancePolicy>
```

- The issuer invokes the IssuerABCE.initIssuanceProtocol() method with the above policy, the User's attribute values and her cryptoEngine type, and finally sends the resulting IssuanceMessage to the User.
- **Important:** During the issuance protocol, the issuer will check that the IssuanceToken uses the same PseudonymValue as was registered with the pupil's record. Otherwise, cheating users could obtain credentials that are not bound to a smart card, and thereby create software-only credentials that can be shared with other users. In future versions of the ABCE, we will make sure that the presentation policy can impose one particular PseudonymValue to be used, but this is currently not supported.
- The user feeds the IssuanceMessage to the `UserABCE.issuanceProtocolStep()` method to obtain a new IssuanceMessage that is sent back to the issuer.
- The IssuanceMessage ping-pong continues until the user's side returns a credential description and the issuer's side returns the URI of an issuance log entry. The issuance log entry contains the revocation handle amongst others. It is stored only on the issuer side and not given to the user. It is there since it is technology independent and will work the same way for both Idemix and U-Prove and any other similar technologies.

The User and the IdM Portal interact in a similar manner to issue all other credentials (credClass, credGuardian, credChild, credCourse, credRole, credSubject) that the User is entitled to receive.

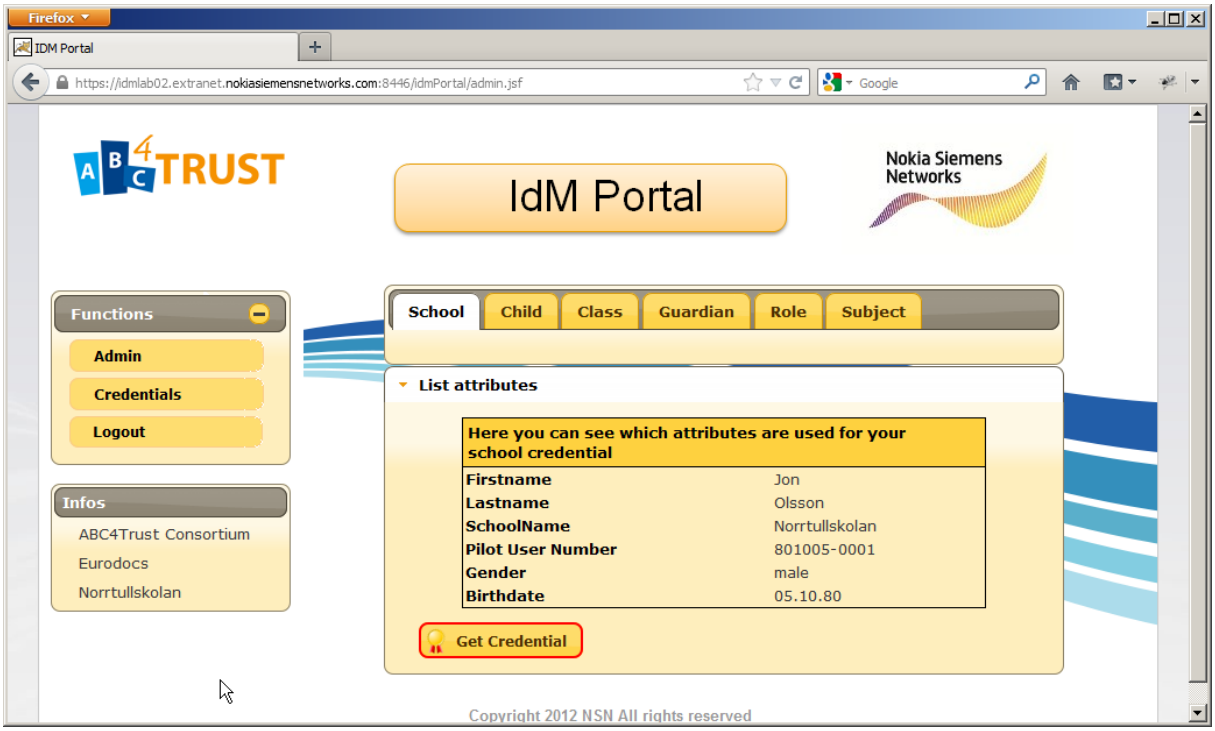

Figure 13: IdM Portal List of Attributes

4.5 Other School Credentials

This use case describes the steps needed for a user to receive or update their auxiliary credentials (e.g. the credentials that show parent-child relationships).

"Other school credentials" are issued analogously to the school credential. Pre-requisite for obtaining these auxiliary credentials is that the User's smart card is registered. This allows the User to login to the IdM Portal via the IdM Application with ABC technology. Please note that for obtaining credentials from the IdM Portal does not require the User to have any valid credentials beforehand.

4.6 Viewing User's Data

This use case describes the steps needed for viewing the User's own data (the attribute values at the IdM).

Prerequisite: The User (pupils/guardians/personnel) has received her smart card with PIN & PUK.

The User has registered her smartcard (see 4.2)

- The User presses the login button in the IdM Portal and is automatically forwarded to the IdM

Application. Here, the User authenticates herself via ABC technology. After successfully authenticating herself, the User is forwarded back to the IdM Portal. The IdM Application forwards the same policy which the IdM Portal forwarded to the User when she registered her smart card. Please note that the "Nonce" is a random value generated on the fly during every login procedure.

- Now the User can select the "Admin" button to view her data (see Figure 14)

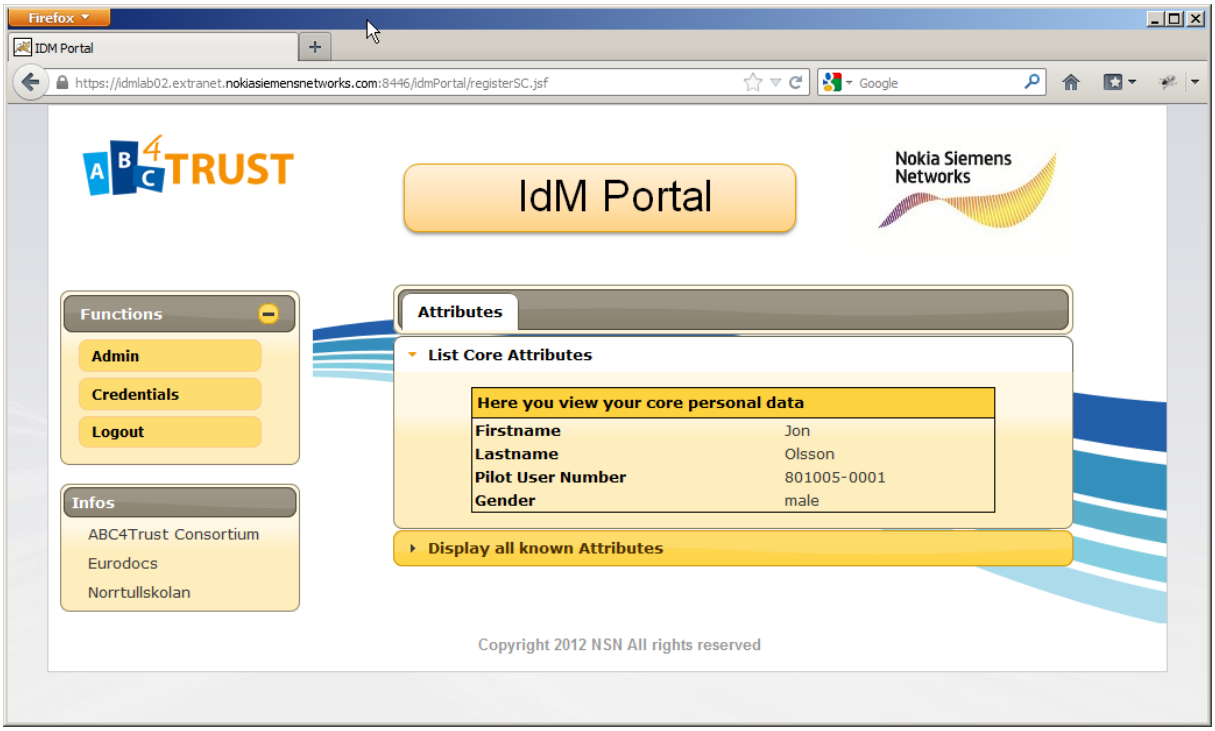

Figure 14: IdM Viewing User Data

4.7 Login to Restricted Area Application

This use case describes the steps when a user logs in to Restricted Area Application using the School Credential (credSchool). To enter the application the user has to prove that she has a valid credSchool. Entering (via a successful login) the RA application does not mean that the user will get access to any content (messages, documents etc.). A successful login means that the user will be able to see lists of Restricted Areas. In order to enter a Restricted Area the user still has to prove that her credentials satisfy the corresponding access policy (See 4.10 **Access to a Restricted Area**).

Prerequisite: User has obtained the school credential on her smart card from the IdM Portal and it is still valid (see use cases 4.4 **Obtaining the School Credential**).

- The User navigates to the Restricted Area Application.
- The RA server prepares a presentation policy that requests the user to prove ownership of a valid school credential (credSchool). It sends the policy to the user:

```
<?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 Client displays a message to the User asking to plug her smart card in the reader device and confirm the check against her credentials.
- The User Client invokes the `UserABCE.createPresentationToken()` method with the received presentation policy to obtain the presentation token containing the requested pseudonym. It sends the presentation token to the school system.
- The school system verifies the token by calling the `VerifierABCE.verifyTokenAgainst-Policy()` method of the ABCE. If verification fails, a failed page is displayed and access is refused.
- User is authorized and redirected to the main page of Restricted Area Application.

4.8 Choose or Create Alias

This use case describes the steps needed for a user to choose a previously created alias or to create a new alias. The user can select one of the aliases that she has created previously. Those aliases are saved in a secure and privacy friendly way on the users' smart card. If the user creates a new alias the system will make sure that the chosen alias is unique system wide. The new alias is also added to the smart card and can be used later.

Prerequisite: User has logged in to Restricted Area Application (See 4.7 **Login to Restricted Area Application**)

- Restricted Area Application shows user a popup with ability to choose existing or create a new alias.
- 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 server checks whether the alias *aliasname* is already taken. If so, it asks the

user to choose a different alias.

- The RA server prepares a presentation policy that requests the user 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:alias AliasID">
    <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 Client displays a message to the User to plug her smart card in the reader device and confirm the check against her credentials.
- The User Client invokes the `UserABCE.createPresentationToken()` method with the received presentation policy to obtain the presentation token containing the requested pseudonym, which will be automatically generated based on the given scope. It sends the presentation token to the school system.
- The school system verifies the token by calling the `VerifierABCE.verifyTokenAgainst-Policy()` method of the ABCE. If verification fails, a failed login window 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 Client sends *alias ID* to the RA Application.
  - The presentation policy will look like in the example above.
  - The User Client invokes the `UserABCE.createPresentationToken()` method with the received presentation policy to obtain the presentation token containing the requested pseudonym. It sends the presentation token to the RA Application.
  - The RA Application verifies the token by calling the

`VerifierABCE.verifyTokenAgainst-Policy()` method 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.
- Otherwise, the main screen of the RA Application is displayed, giving an overview of the relevant information, RAs, etc… that the school system maintains for alias *aliasname*.

4.9 Instantiating a Restricted Area

Administrators and in some cases 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.

Prerequisite: The user has received a smart card and has obtained credentials from the IdM (see use cases 4.4 **Obtaining the School Credential**).

The user has logged in to the RA Application.
The user has an active alias (not anonymous); chosen in alias selector and proved as described in Chapter 4.8 **Choose or Create Alias**

- The user will be asked for defining the access policy that must be applied when other users want to enter this restricted area through a GUI.
- The RA Application creates the restricted area, converts the access policy 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.

4.10 Access to a Restricted Area

This use case describes how a user that already has entered the Restricted Area Application (see use case 4.7 **Login to Restricted Area Application**) can enter a Restricted Area. In order to enter a Restricted Area the user still has to prove that her credentials satisfy the access policy (See 4.10 **Access to a Restricted Area**).

Prerequisite: The user has received a smart card and has obtained all needed credentials from the IdM (see use cases 4.4 **Obtaining the School Credential** and 4.5 **Other School Credentials**).

The user has logged in to the RA Application.
The user has an active alias, chosen in alias selector and proved as described in section 4.8 **Choose or Create Alias**.

To access a restricted area (RA), the user must generate presentation tokens (generated from her credentials) that proves the attribute values in her credential match (fulfil) the access policy of the RA. For example, the user can prove that she is "a 12 years old girl".

- The user logs in to the system (see use case 4.7 **Login to Restricted Area Application**).
- The user chooses a previously established alias or creates a new one (4.8 **Choose or Create Alias**)
- The user navigates and select a Restricted Area to enter.
- If this alias was previously used to access only political discussions or other non-inspectable areas and this is the first time that the user uses this alias to access an inspectable restricted area, then a warning pops up saying that accessing the RA under this alias makes her messages in the non-inspectable areas inspectable too.
- If the user has previously accessed a specific RA using an alias and the predefined session timeout of this RA (see use case 4.9 **Instantiating a Restricted Area**) using the same alias has not exceeded, then the school system grants access to the RA without any further authentication.
- Otherwise, if this is the first time that the user accesses this specific RA or session doesn't

contain tokens, then the Restricted Area System sends the presentation policy that was associated to the restricted area. Example of XML containing access policy alternatives can be found in Chapter 4.9 **Instantiating a Restricted Area**

- The policy is sent from Restricted Area Application to the User Client.
- 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 handled by the client.
- The client prepares the presentation token and sends it to the RA system.
- The RA system 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 system updates the timestamp of the last successful presentation.
- If it is set as inspectable during creation, the RA server stores the token identifier of the presentation token description returned by the verifyTokenAgainstPolicy() method and associates it to the all messages written during this session within, so that it can later fetch the token back from the database in case inspection is needed. (The token itself is stored by default by the verifyTokenAgainstPolicy method.)

4.11 Counselling

A pupil gets help and advice from school personnel.

Prerequisite: The user has received a smart card and has obtained all needed credentials from the IdM (see use cases 4.4 **Obtaining the School Credential** and 4.5 **Other School Credentials**).
The user has logged in to the RA Application.
The user has an active alias, chosen in alias selector and proved as described in Chapter 4.8 **Choose or Create Alias**

- The pupil browses to the counselling section.
- When creating counselling Restricted Area the user chooses the counsellor by either name or by role. Access Policy allows the user to access this RA herself by her active alias as an "alternative". Alias is not included to the XML because it is checked against active alias. On every creation of the counseling session new alias is created by the application to avoid linkability of aliases. XML will contain the policy alternative for the teacher identified by either First name - Last name or Role.
- To access counselling session this access policy is generated in XML format as shown in the below example. The login process is the same as use case 4.10 **Access to a Restricted Area**.

```
<abc:PresentationPolicyAlternatives>
  <abc:PresentationPolicy PolicyUID="urn:soderhamn:policies:counselling:no-
attributes">
    <abc:Message>
         <abc:Nonce>some fresh nonce</abc:Nonce>
    </abc:Message>
    <abc:Pseudonym Scope="urn:soderhamn:alias:aliasname" Exclusive="true"
Alias="#nym"/>
    <abc:Credential SameKeyBindingAs="#nym" Alias="#schoolcred">
      <abc:CredentialSpecAlternatives>
        <abc:CredentialSpecUID>urn:soderhamn:credSchool</abc:CredentialSpecUID>
      </abc:CredentialSpecAlternatives>
      <abc:IssuerAlternatives>

<abc:IssuerParametersUID>urn:soderhamn:issuerSchool</abc:IssuerParametersUID>
      </abc:IssuerAlternatives>
      <abc:DisclosedAttribute AttributeType="urn:soderhamn:credSchool:
```

```
pilotUserNumber”>
        <abc:InspectorAlternatives>

<abc:InspectorPublicKeyUID>urn:soderhamn:inspectorpk</abc:InspectorPublicKeyUID>
        </abc:InspectorAlternatives>
        <abc:InspectionGrounds>
          Description of circumstances and process under which token may be
inspected.
        </abc:InspectionGrounds>
        </abc:DisclosedAttribute>
  </abc:Credential>
  </abc:PresentationPolicy>
</abc:PresentationPolicyAlternatives>
```

- If any of the above tests fails, access is denied. Otherwise, the user enters the counselling area.
- As alias names used for counselling should not be used for other purposes within the system the user will be warned if she uses such an alias in an inspectable RA and vice versa.
- The counselling session is a one-by-one session by default. The counsellor or the user (pupil or parent) may add other person to a policy.

4.12 Restricted Chat Room

Chat rooms are Restricted Areas with chat functionality activated. This use-case describes two use-cases which are almost identical, one where a person chats with other users by entering a chat room (group), another where a person that wants to chat with another person in a private chat room (one-to-one).

- User enters a Restricted Area with chat enabled. The technical details for logging in are identical to use case 4.10 **Access to a Restricted Area**.
- User proceeds to the chat section of the Restricted Area.
- In the chat room, the user can see the list of online aliases in that room and type in messages that are visible to everybody in the RA.
- In case the user wants to chat with another alias owner, she chooses that alias from the list of online aliases of the current RA and selects the option to start a chat.

4.13 Political Discussions

Political discussions are anonymous chats, with no option to reveal the identities. This use-case describes two use-cases which are almost identical, one where a person chats with other users by entering a chat room (group), another where a person that wants to chat with another person in a private chat room (one-to-one).

Technical details of this use case are similar to the 4.12 **Restricted Chat Room**. So we will describe the details again. The major difference is that Political Discussions **cannot be inspected** and this is reflected in the presentation policy. It does not contain any inspection grounds or requests to disclose attributes.

4.14 Sharing Documents

The school is producing documents (exam results, grades, individual development plans, absence reports etc.) that need to be shared with or distributed to the pupils and their parents.
Some of these documents are strongly confidential and have to be shown only to the recipient. A personal Restricted Area exists for every user in the system, where these documents can be found. The access policy for such a restricted area usually includes the pupil, teachers and the legal guardians. In this use-case the authentication of the user is performed using Privacy ABC technology.

In general case, every Restricted Area, where the Documents functionality is enabled, can contain documents uploaded by the user or the RA owners.
To access the RA with shared documents user has to follow the steps described below:

- Enter the RA with documents as described in use case 4.10 **Access to a Restricted Area**.
- Proceed to the documents section of the Restricted Area.
- Upload document if the user has sufficient access rights.
- Download available documents by clicking on the corresponding "download" links.

4.15 Revocation

A user's credentials can be made invalid when the user is forced to leave the school, he or his guardians revoke consent to participate in the pilot, an attribute value is no longer valid, or a user reports his smart card as lost or stolen.

- The revocation requestor could be the user herself who goes to the school office to report a lost card, or could be the school administration because an issued credential should no longer be valid (e.g., the pupil changed classes, a course has finished, or a pupil has left the school).
- The school administrator used the IdM Admin GUI (see Figure 15) to launch revocation
- The administrator selects the corresponding credential type she needs to revoke (see Figure 18)
- If multiple credentials of the same type are available (only in the case of credGuardian or credChild), the administrator can select a specific credential based on its attribute values (i.e. the Pilot User Number).
- The IdM Admin GUI software creates a revocation request containing the revocation handle of the credential and sends it to the Revocation Authority.
- The Revocation Authority server software receives the request, extracts the revocation handle, and revokes the given revocation handle using the revoke() method of the ABCE.

If the administrator changes an attribute value which is mapped to an already issued credential, the issued credential will automatically be revoked prior to writing the new attribute value into the IdM database.

If the revocation takes place due to a pupil leaving the school, additional steps will be taken:

- The user gets a possibility to backup personal RAs such as the personalized RA for sharing documents, counselling sessions etc. afterwards these RAs will be deleted from the RA system within one month. In order to delete a user's counselling RA the user has to prof that he is the owner of the alias that the counselling RA belongs to. This means that a request for deletion of a RA for counselling has to be done before the revocation of the user's credentials takes place.
- The user may decide to request the deletion of content that he posted in other RAs she has contributed to under her real name.

When it comes to revocation it's important to know that, as per default, a revocation check is performed whenever a revocable credential is used (included in a presentation policy). There is no way (no option in the presentation policy definition) to indicate not to perform a revocation check.

In the Swedish pilot all credentials are revocable which means that a revocation check is performed whenever a credential is used (included in the presentation policy).

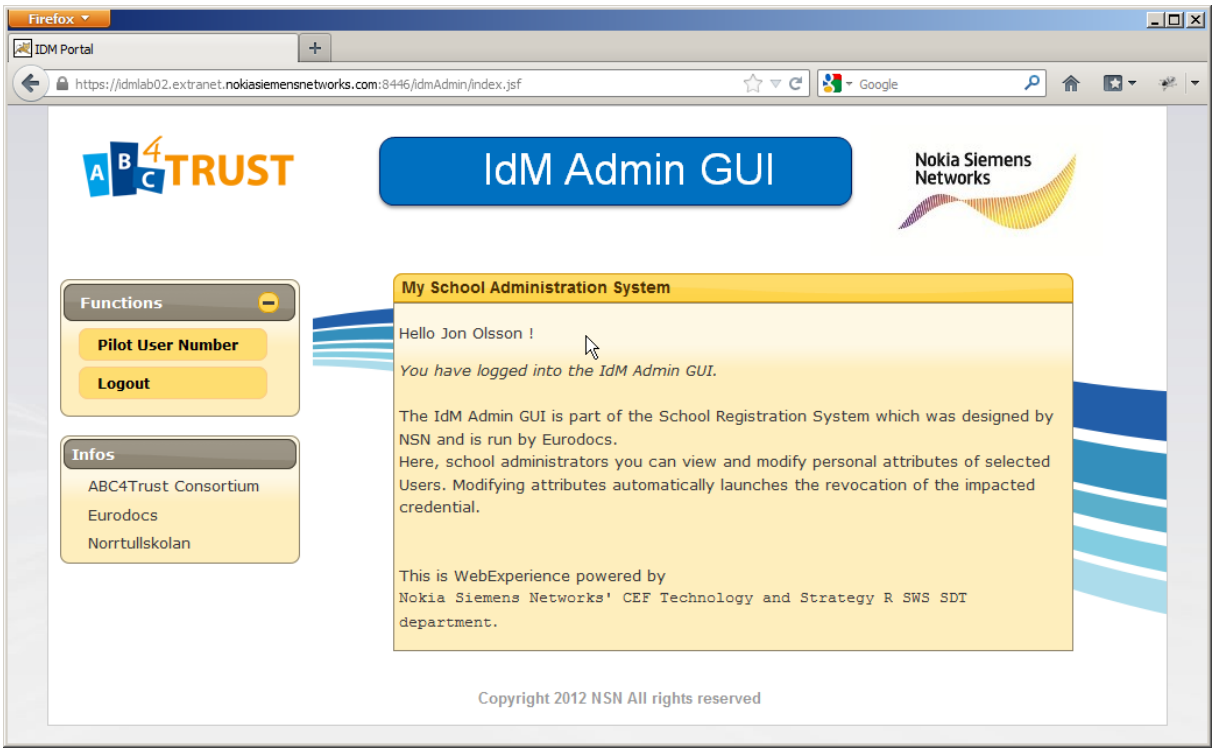

Figure 15: IdM Admin GUI I

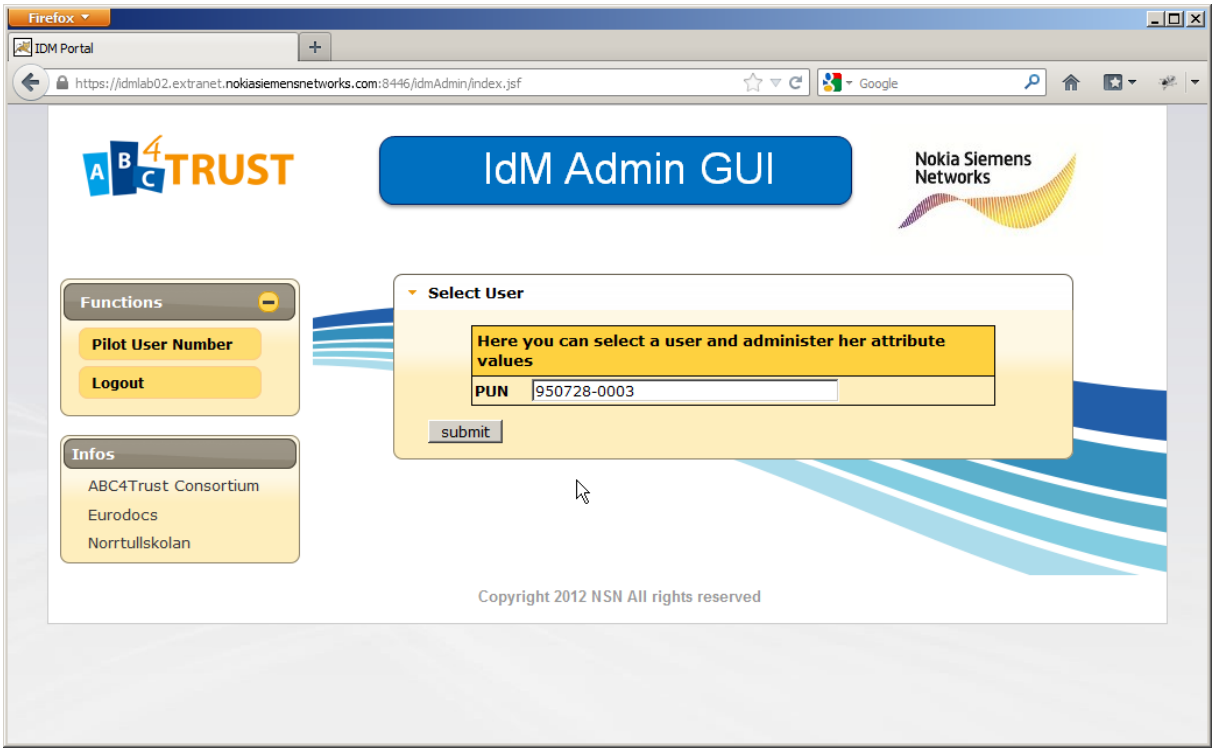

Figure 16: IdM Admin GUI II

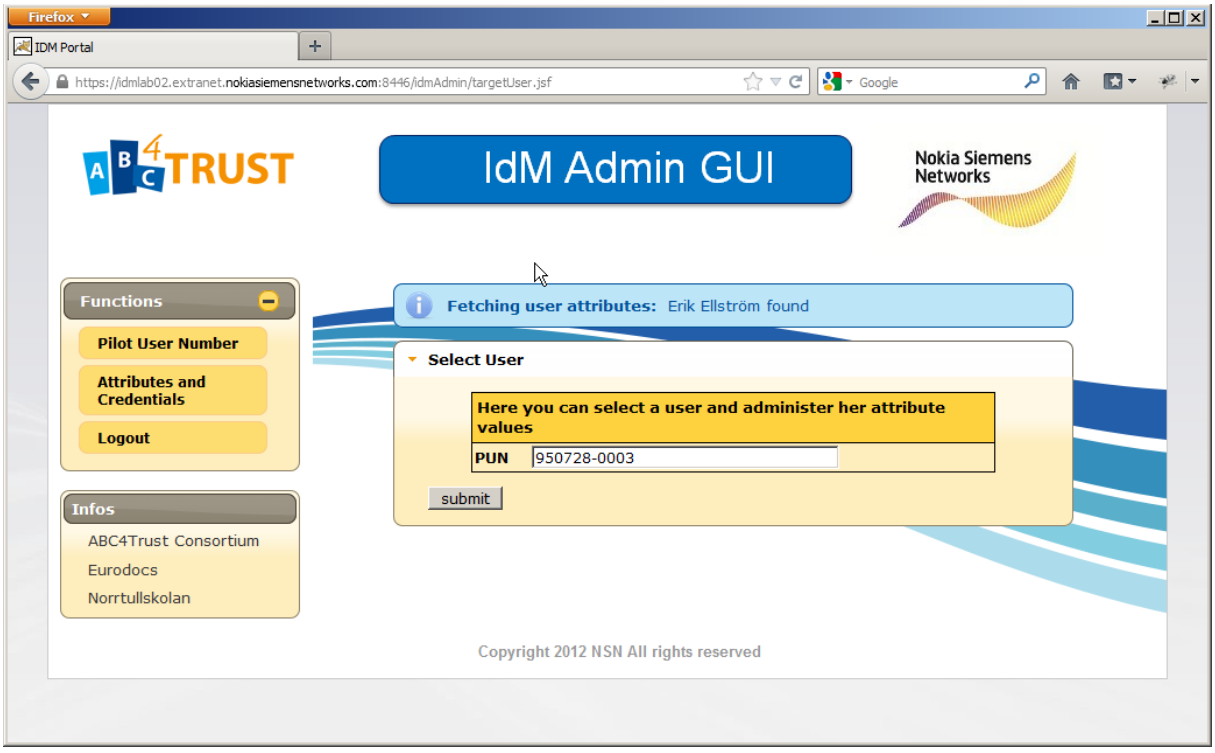

Figure 17: IdM Admin GUI III

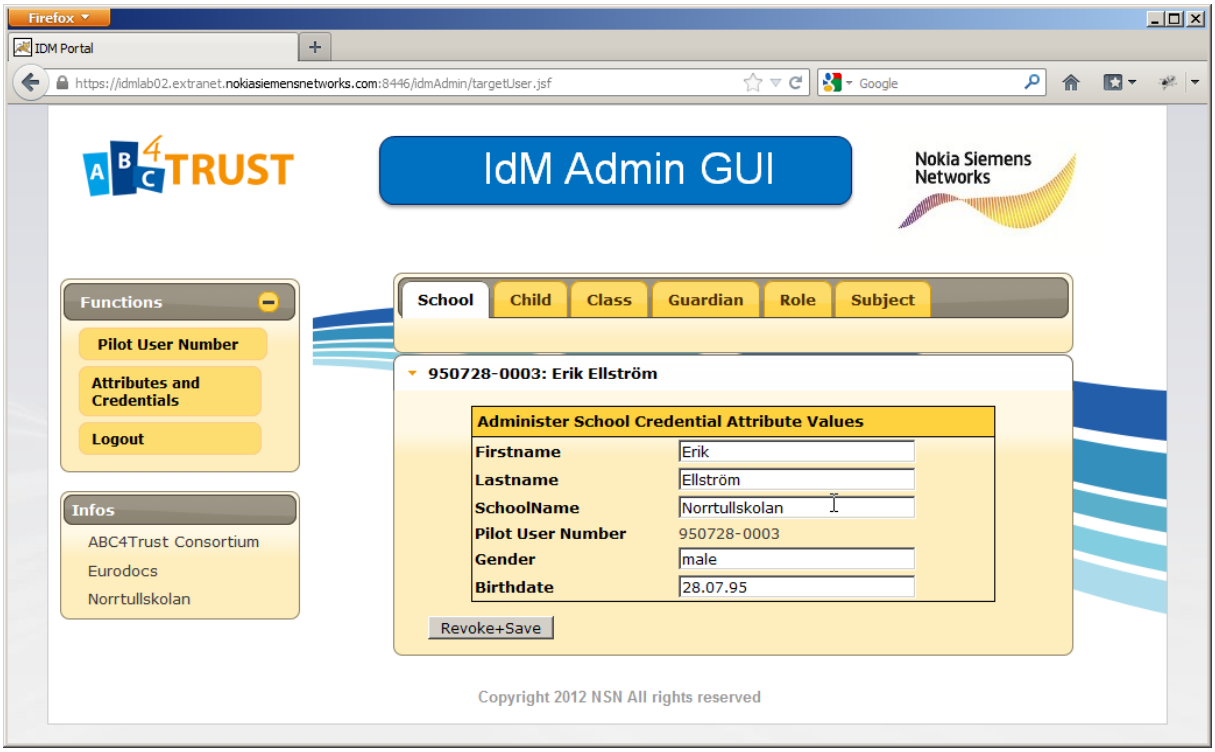

Figure 18: IdM Admin GUI Revocation

4.16 Emergency Situation (Inspection)

When an emergency situation occurs that causes the RA Application and the School Inspection Board to start the inspection procedure for a particular communication, Inspector with an approval from Board can reveal attributes from the corresponding presentation token.

Inspection is done using the Inspector Application which is a separate client application that is not connected directly to Restricted Area database.

Prerequisite: Restricted Area where the reported content was found is inspectable.
School Inspection Board approves the Inspection

- A user (possibly a teacher, parent, or counsellor, but possibly a fellow pupil) reports an emergency situation, specifying details of the communication via reporting content. Any post of the wall, uploaded document or chat message can be reported.
- The potentially offending content is copied by the RA Application and forwarded to the members of the School Inspection Board.
- The School Inspection Board and other users responsible investigate the case and decide whether it mandates triggering inspection. If the board decides that the case does not require identification of the user, it either closes the case (ignore) or may decide to delete the content and write a warning to the respective RA as a substitute to reported content.
- If inspection is seen as necessary, the school administration starts a user interface to look up the offending communication and clicks a button to initiate the inspection procedure. The software looks up the identifier of the presentation token that was linked to the offending

communication in the school RA Application sends the presentation token to the inspector using secure channel.

- The inspector starts the inspection client software and inserts his smart card containing the inspection key.
- The inspection client retrieves the presentation token from the inspection request and calls the InspectorABCE.inspect() method on the presentation token to get back the encrypted attribute value (in case of the example policies above, the Pilot User Number).
- The ABCE asks the inspector to enter his/her smart card PIN code.
- If the PIN is correct, the User Service fetches the inspection decryption key from the smart card storage and decrypts the inspectable attributes.

  Note: The current implementation does not support performing the decryption operation on the smart card itself. The inspector must take care to perform the inspection on a safe (i.e., trustworthy and virus-free) computing environment, since the inspector's private key will be loaded in the memory of the computer.
- The inspector informs the School Inspection Board about the decrypted attribute value.
- The School Inspection Board takes appropriate action.
- Depending on the situation, once appropriate, the user concerned will be informed that her identity has been revealed.
- In any case the School Inspection Board makes its activity visible in the concerned RA by adding a post stating that including the inspection time. If required by the circumstances, e.g. in order to not interfere with official investigations etc., this information may be postponed for an adequate time but not omitted entirely.

4.17 Viewing / Deleting Credentials

This use case describes the steps needed for a user to view/delete her credentials stored on her smart card.

- The user starts the credential management part of the User Client application (the browser plugin) on her local client.
- The User Client asks to insert the smart card and enter the PIN code.
- The User Client obtains the list of descriptions of stored credentials by calling the UserABCE.listCredentials() and UserABCE.getCredentialDescription() methods.
- When the user deletes a credential, the credential management application calls the UserABCE.deleteCredential() method.

4.18 Changing PIN

The user can change the PIN protecting her smart card, this use case describes the process of doing that.

- The user selects the Change PIN option in the user interface of the User Client
- She enters the old PIN and a new PIN twice
- The User Client sends the `CHANGE PIN(oldpin, newpin)` instruction to the card, and if the old PIN was correct the card changes its PIN to the new one

4.19 Unlocking the Smart Card with a PUK

To protect the data on the smart card, the card will enter locked mode if the user, or someone else, has input in the wrong PIN code in three consecutive tries. In locked mode the card cannot be used. To get the smart card back to working mode the user has to type a PUK code, this use case describes this process.

- The user select unlock card in the user interface of the User Client.
- She types in the correct PUK code and a new PIN code.
- The User Client sends the `RESET PIN(puk, newpin)` instruction to the smart card, and it the PUK code is correct the card will return to working mode, with the new PIN code.

In case an incorrect PUK code is inserted ten times the card enters dead mode, and no data on the card can get restored. The user will have to get a new card and restore a backup from the old card to the new one.

4.20 reIssuance of U-Prove Tokens

Contrary to Idemix technology, U-Prove technology does not support the feature that issued credentials can be used unlimited times for generating un-linkable tokens.

Basically it can be said that U-Prove credentials are mapped to a limited amount of tokens and the entire "package" is made available to the User during issuance.

Contrary to U-Prove, Idemix credentials are issued to the User without tokens. The User can generate as many on-the-fly Idemix tokens (without the Issuer having to be online) as she requires when interacting with the Verifiers.

Typically, issuance always requires the User to visit the IdM Portal, the Issuer GUI. In order to increase the User experience when using U-Prove smart cards, a new feature has been implemented to allow GUI-less re-issuance of U-Prove tokens.

This use case describes the steps needed for a User holding a U-Prove smart card to receive a new batch of U-Prove tokens mapped to the same U-Prove credential.

Prerequisite: The User has obtained a U-Prove credential.

The U-Prove credential is still valid, i.e. it has not been revoked

The IdM Portal, the Revocation Authority and the IdM ABC System are online and running.

- The User Client detects that the number of unused U-Prove tokens is running low.
- Without interaction of the User, the User application contacts the same 'start issuance' and 'continue issuance 'URLs it contacted for obtaining the initial batch of U-Prove tokens.
- The URLs point to servlets of the IdM Portal.
- The IdM Portal checks if the User application forwards an established cookie when visiting these URLs.
- Since the User application will not forward any cookies in this case, the IdM Portal is assuming
  - That the User owns a U-Prove smart card.
  - That the User is requesting for re-issuance of the credential the specific URL it is servicing.

- Now the IdM Portal will trigger the ABC System to use a special policy (see the XML listing below) mapped to the original credential specification
- The ABC System will check if the underlying credential, which generated the issuance token, was not revoked.
- The policy also forces the User to reveal all attribute values (including the revocation handle) mapped to the original credential.
- After verifying the policy in the course of the issuance messages, the User will obtain a credential and new tokens containing the same revocation handle and the attesting the same attribute values.
- If a credential was revoked, the User is forced to visit the IdM Portal and use the GUI to obtain a new credential with a new revocation handle.

```
<abc:IssuancePolicy Version="1.0"
xmlns:abc="http://abc4trust.eu/wp2/abcschemav1.0">
    <abc:PresentationPolicy PolicyUID="urn:soderhamn:policies:issuance">
        <abc:Message>
            <abc:FriendlyPolicyName lang="en">Policy: Authorized Users
only</abc:FriendlyPolicyName>
            <abc:FriendlyPolicyDescription lang="en">This policy will request the
pupil to present the established scope-exclusive Pseudonym with the scope
"urn:soderhamn:registration".No Privacy ABCs are required for this
step.</abc:FriendlyPolicyDescription>
        </abc:Message>
        <abc:Pseudonym Exclusive="true" Scope="urn:soderhamn:registration"
Established="false" Alias="#nym"/>
        <abc:Credential SameKeyBindingAs="#nym" Alias="#schoolcred">
            <abc:CredentialSpecAlternatives>

<abc:CredentialSpecUID>urn:soderhamn:credspec:credSchool</abc:CredentialSpecUID>
            </abc:CredentialSpecAlternatives>
            <abc:IssuerAlternatives>

<abc:IssuerParametersUID>urn:soderhamn:issuer:credSchool</abc:IssuerParametersUID>

<abc:IssuerParametersUID>urn:soderhamn:issuer:credSchool:idemix</abc:IssuerParame
rsUID>

<abc:IssuerParametersUID>urn:soderhamn:issuer:credSchool:uprove</abc:IssuerParame
rsUID>
            </abc:IssuerAlternatives>
            <abc:DisclosedAttribute
AttributeType="http://abc4trust.eu/wp2/abcschemav1.0/revocationhandle"/>
            <abc:DisclosedAttribute
AttributeType="urn:soderhamn:credspec:credSchool:firstname"/>
            <abc:DisclosedAttribute
AttributeType="urn:soderhamn:credspec:credSchool:lastname"/>
            <abc:DisclosedAttribute
AttributeType="urn:soderhamn:credspec:credSchool:pilotUserNumber"/>
            <abc:DisclosedAttribute
AttributeType="urn:soderhamn:credspec:credSchool:gender"/>
            <abc:DisclosedAttribute
AttributeType="urn:soderhamn:credspec:credSchool:schoolname"/>
            <abc:DisclosedAttribute
AttributeType="urn:soderhamn:credspec:credSchool:birthdate"/>
        </abc:Credential>
    </abc:PresentationPolicy>
    <abc:CredentialTemplate SameKeyBindingAs="#nym">

<abc:CredentialSpecUID>urn:soderhamn:credspec:credSchool</abc:CredentialSpecUID>

<abc:IssuerParametersUID>urn:soderhamn:issuer:credSchool</abc:IssuerParametersUID>
    </abc:CredentialTemplate>
```

```
</abc:IssuancePolicy>
```

5 Client Deployment

5.1 User ABC System

The User ABC System has to be installed both as a local service (the ABC Engine and the crypto engine) and as a browser plugin locally on all machines the users want to use for the Pilot. This is done by a supplied Windows installer especially made for this pilot. The installer also contains the cryptographic keys, such as the public key of the issuer, the inspector and the revocation authority, needed for participating in the system.

5.2 Browser Plugin

The browser plugin (see above) will be used in all communications with the local User ABC System. As plugins are available for Firefox and Internet Explorer, those are the browsers supported within the pilot. In this pilot some parts (e.g. Alias selector) are implemented using JS originating from the RA.

5.3 Inspector Application

The inspector application is deployed as a client application, which uses the Inspector Service. Inspector Service is coming with Inspector ABC System. The application can be run on Windows operating system. Inspector application gets information needed for inspection from the RA Admin application so it should be able to reach it. Security is assured by usage of HTTPS and authentication using admin login from RA Admin application.

6 School Registration System Deployment (IdM)

According to the pilot architecture that has been described in Figure 2, the School Registration System is mainly responsible for issuing credentials to the users. Potential users of this system are pupils, parents and school personnel who are able to collect Privacy-ABCs that certify their role at the Restricted Area System. Also administrators, who can populate the system's database with the users' attributes, register pilot smart cards, and change or revoke credentials.

6.1 Server Hardware and Environment

The servers used within the school system have the following configuration:

Hostname	Hardware	OS	IP	Description
VHOST01	PE2950 III Quad-Core Xeon X5460 3.16GHz / 2x6MB 1333FSB / 8Gb RAM / 2x74Gb HDD	Ubuntu Server 12.04 LTS +KVM	81.94.#.#	Host machine for RA Application Web Server
VHOST02	PE2950 III 2 x Quad-Core Xeon X5460 3.16GHz / 2x6MB 1333FSB / 8Gb RAM / 6x74Gb HDD	Ubuntu Server 12.04 LTS +KVM	81.94.#.#	Host machine for IdM and ABC System

Virtual machines that are set up on the hosts for School Registration System are the followings:

Machine name	Host machine	IP	FQDN	OS
IdM	VHOST02	81.94.#.#	idm.abc4trust.se	Ubuntu Server 12.04 LTS
ABCE	VHOST02	81.94.#.#	abce.abc4trust.se	Windows Server 2008

6.2 Registration System

Deployment of the registration system is done in two phases:

(1) Setup of the Hardware, Network and Virtual Servers

(2) Import of data and generation of cryptographic material

The phase one setup is standard and not use case specific. Coordination is to be done with the network department to get IP addresses and setup port forwarding. The hardware or virtual server containers are also being setup. This phase is generic and can be compared to other software projects.

Phase two is use case specific and requires both understanding of the ABC technology, the IdM and the Portal, as well as good user-level knowledge of cryptography. Cryptographic material for securing the registration system portal (HTTP server based) and IdM has to be obtained or created.

The project decided to host the Registration System on 2 different systems (see Figure 19). This is due to the fact that NSNs IdM is customised for Linux operating systems whereas Microsoft's Crypto-Engine (U-Prove CE) requires .NET which by itself does not run natively on Linux.

The Registration System runs on a 32-bit Ubuntu Linux system, version 10.04 (Lucid Lynx) LTS. Ubuntu is an open source operating system distributed under the GNU General Public License [GNU GPL].

The operating system of the Registration ABC System is a 32-bit Windows Server 2008 standard edition with Service Pack 1.

Adapting the IdM to fit into a Windows operating system would be possible, but this would mean that, next to additional customisation efforts, NSNs local test labs and the Eurodocs installation differ in their operating systems which could make debugging more difficult.

The ABCE itself and IBM's Crypto-Engine are java-based applications, which can easily run in a Windows or in a Linux environment.

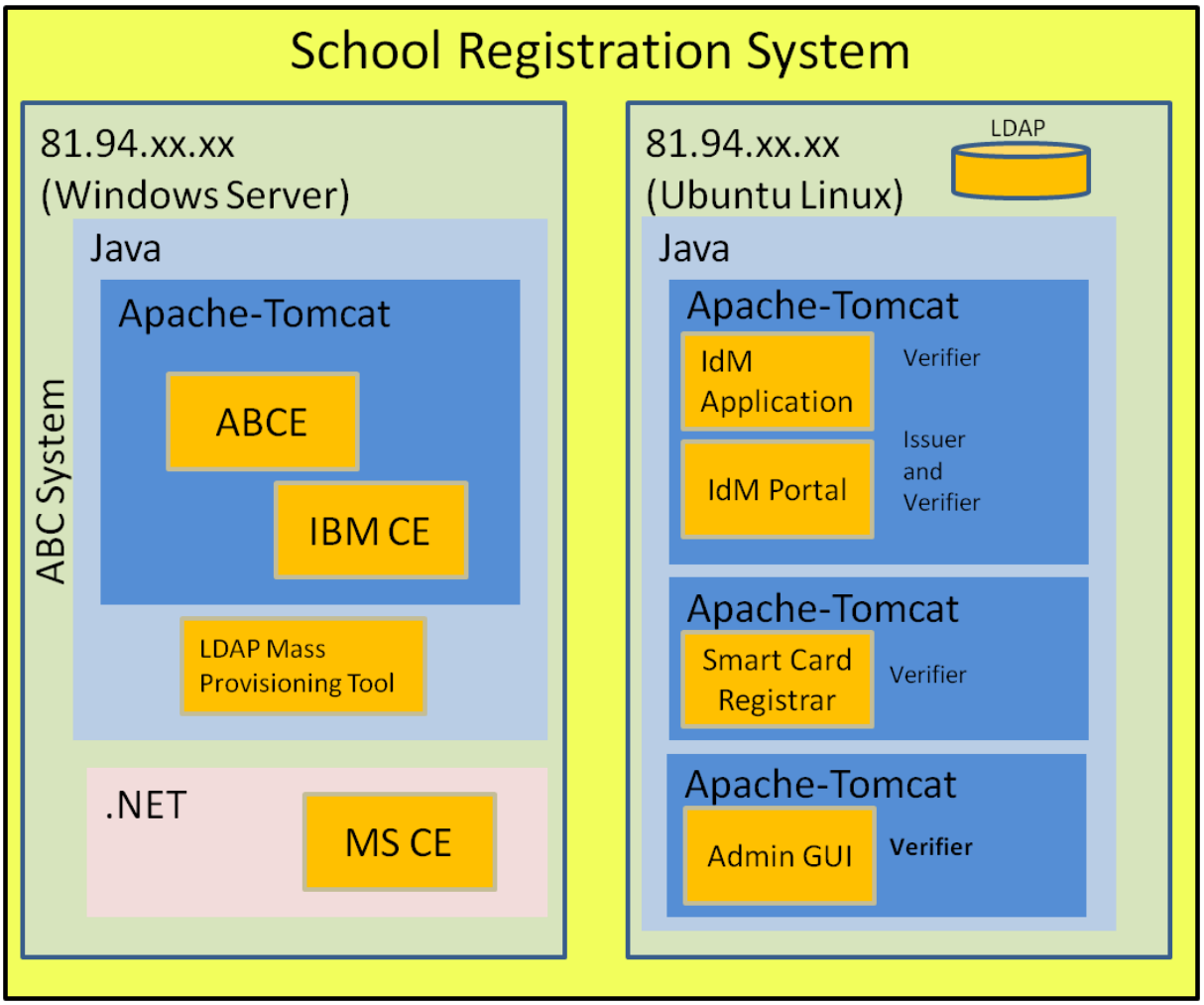

Figure 19. Application Overview of the School Registration System

Instead of hosting all applications on a Windows system, one could consider hosting them on a Linux system. But since the U-Prove CE has not been thoroughly tested on Mono

(http://en.wikipedia.org/wiki/Mono_%28software%29), NSN decided to host the entire ABC core components on a Windows system.

The host "81.94.#.#" runs on a 32-bit Ubuntu Linux system, version 10.04 (Lucid Lynx) LTS. Ubuntu is an open source operating system distributed under the GNU General Public License [GNU GPL].

The operating system of "81.94.#.#" is a 32-bit Windows Server 2008 standard edition with Service Pack 1.

6.2.1 Software Deployment of Registration ABC System

The following programs/applications required for the pilot are installed on the Windows server:

1. jdk1.6.0_35
2. apache-tomcat-6.0.35
3. Microsoft .NET Framework 4.5
4. freeSSHd 1.2.6
5. LDAP Admin 1.1.0.0
6. Microsoft Crypto-Engine
7. ABC4TrustSytem.war web-service (contains the IBM Crypto-Engine and the ABCE)
8. LDAP Mass-Provisioning Tool (java-based)

As can be seen in Figure 19, the ABC System contains an Apache Tomcat web server. This server hosts the ABC4TrustSystem web-service, which is configured to listen on port 8080. The IdM System will address this port to proxy all ABC technology related traffic between the User and the Issuer and Verifier ABCE. The ABCE of the ABC System therefore is responsible for handling 2 ABC roles concurrently.

Microsoft's Crypto-Engine is an independent executable which must be "run as administrator" to listen on port 32123. The U-Prove CE will be addressed by the ABC4TrustSystem web-service in case U-Prove crypto actions need to be performed.

The LDAP Admin program is used to manually inspect and modify the contents of the IdM Database hosted on idm.abc4trust.se (i.e. IdM System).

For provisioning a larger number of Users, the LDAP Mass-Provisioning Tool facilitates the tasks of the administrator. This tool can read comma-separated CSV files and transfer their contents to the IdM database.

Several manual configuration settings were necessary to make the system run. Next to setting the environment variable JAVA_HOME to point to the java installation, the administrator must verify that Apache Tomcat is configured to listen on port 8080 by customizing the server.xml file.

For the U-Prove CE, the environment variable "PathToUProve" must be set to point to the U-Prove executable.

Finally, Windows firewall must be configured to allow traffic to the SSH port and to the HTTP port.

6.2.2 Software Deployment of IdM Portal

The following software has been installed on the Linux system:

1. jre1.7.0_07
2. apache-tomcat-6.0.35

3. LDAP library 2.4-2
4. schemas required by the IdM for the Söderhamn pilot
5. an initial data-set not containing User data
6. IdM application (stored as directory tree)
7. idmPortal.war
8. idmSmartCardRegistrar.war

Contrary to the ABC System, the IdM System hosts 2 instances of the Apache Tomcat server. The reason for this is to allow the idmSmartCardRegistrar to listen on a port different to the IdM Portal and the IdM Application. In this pilot, the latter 2 listen on port 8443 whereas the registrar listens on 8444. This way, the network administrators can protect the registrar from unauthorized access from the Internet.

The IdM Application represents the backend of the IdM that authenticates the Users. The IdM Portal is the GUI, which allows Users to inspect the attributes the IdM stored about them. Next to that, Users visit the IdM Portal for registering their smart cards (i.e. "claiming authorized scope-exclusive pseudonyms") and for gathering Privacy-ABCs (i.e. credentials).

The IdM Application is basically a SAML server. The IdM Portal is a "trusted third party" of the IdM Application. The IdM Portal uses the IdM Application to authenticate its users.

During the course of the project, the necessity to extract the scope-exclusive pseudonyms from the smart cards and to store them (next to a "smart card ID") in the IdM database prior to distributing the smart cards to the students became clear. The pseudonyms stored in the IdM database represent a set of authorized values. The reason for this measure is to guarantee that no other smart cards are allowed to communicate with the IdM System. The Smart Card Registrar has been implemented to deal with these tasks.

The administrator of the pilot network must protect the IdM database and the Smart Card Registrar from unauthorized access.

Analogue to the ABC System, the IdM System must be manually configured in several areas. Next to setting the JRE_HOME variable to point to the java installation, the Apache Tomcat ports must be customized to use 8443 for the 1st instance hosting the IdM Application and the IdM Portal, and to use 8444 for the 2nd instance hosting the Smart Card Registrar. Finally, a "keystore" file must be stored on idm.abc4trust.se in order to enable the HTTPS access. The server.xml files of Apache Tomcat must be adapted to use the certificate installed in the keystore.

6.2.3 Software Deployment of IdM Application

SSH, LDAP, RFP (VNC: Remote Framebuffer Protocol), RDP (Microsoft: Remote Desktop Protocol) access points are reserved for authorized administrators only.

HTTPS allows access to public web-services.

And finally, the HTTP access is reserved for communication between "81.94.#.#" and "81.94.#.#".

6.2.4 Software Deployment of IdM Admin GUI

The IdM Admin GUI is a tool to be used by the school administrators to change attribute values in the database which are relevant for credentials and to revoke credentials. No specific setup phase is needed, the IdM Admin GUI can be used any time the IdM Application is running.

If the school administrators want to alter an attribute value that is relevant for credentials, the IdM Admin GUI allows them to select a specific user account and view the associated attributes. There is

also a button to save changes. This results in (a) a revocation of the former credential and, if the revocation was successful, (b) a change of the attribute value in the LDAP Database. The network communications for the revocation are handled by the ABC System thus the IdM Admin GUI does not need additional network communication.

7 Restricted Area System Deployment

7.1 Server Hardware and Environment

The servers used within the school system have the following configuration:

Hostname	Hardware	OS	IP	Description
VHOST01	PE2950 III Quad-Core Xeon X5460 3.16GHz / 2x6MB 1333FSB / 8Gb RAM / 2x74Gb HDD	Ubuntu Server 12.04 LTS +KVM	81.94.#.#	Host machine for RA Application Web Server
VHOST02	PE2950 III 2 x Quad-Core Xeon X5460 3.16GHz / 2x6MB 1333FSB / 8Gb RAM / 6x74Gb HDD	Ubuntu Server 12.04 LTS +KVM	81.94.#.#	Host machine for IdM and ABCE

Virtual machine which is set up to host the Restricted Area System is the following:

Machine name	Host machine	IP	FQDN	OS
Mainframe	VHOST01	81.94.#.#	-	Windows Server 2008

7.2 Restricted Area Server

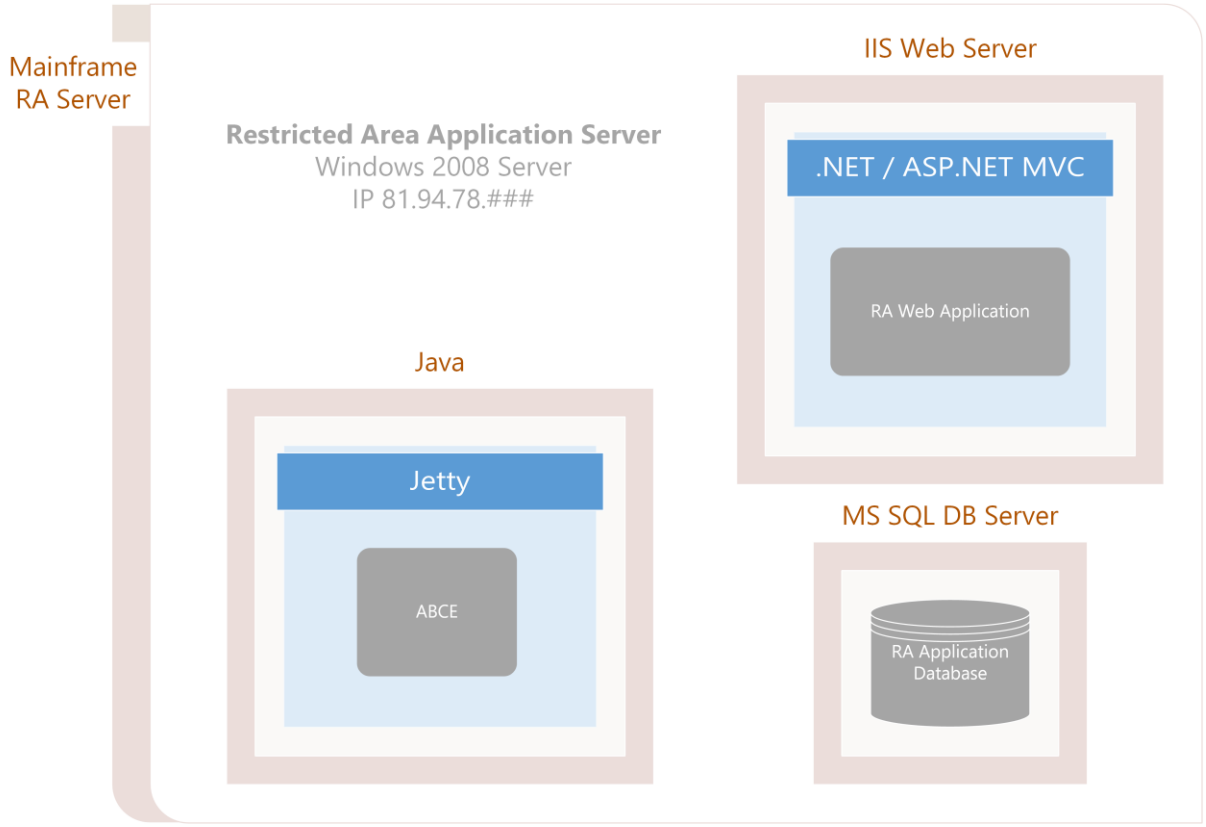

Figure 20: Application overview for the Restricted Area System

Restricted Area server is a Windows Server machine with IIS web server, .NET framework and MS SQL Server installed. Also ABC engine java binaries should be launched within Maven environment and U-Prove service started.

The server listens to HTTP on port 80 and HTTPS on port 443.

SSH and RDP access is available to server administrators only.

The following programs/applications required for the pilot are installed on the Windows server:

1. Microsoft .NET Framework 4.5
2. MS SQL Server 2008 R2
3. IIS 7.5
4. freeSSHd 1.2.6
5. jdk1.6.0_35
6. Maven
7. Microsoft Crypto-Engine (U-Prove service)
8. ABC4TrustSytem.war web-service (contains also an ABC System)

The installation of the ABC System follows the same procedure as in 6.2.1 above. The RA application is deployed as an ASP.NET MVC web application on .NET and displayed via Web Server to users through HTTP and HTTPS connections.

8 Smart card Deployment

This chapter describes the management functionalities of the smart cards. The use cases of the pupils and the other end users are described earlier in sections 4.17 to 4.20.

The smart cards also provide more advanced functionalities such as firmware update and factory reset, these functionalities are not described here. For a description of these functionalities we refer to the smart card manual [BDP12].

8.1 Smart cards Initialization

Before being used, the smart card has to be initialized with the cryptographic parameters that it has to use. These values include the public key of the issuer, and other cryptographic parameters being used in issuance of credentials and proofs using the credentials. The initialization of the card also includes PIN code generation and key generation inside the card itself.

When initializing the card, the first thing being done is putting the card from virgin mode into root mode using an access code that changes with each firmware update of the cards software. Then the root authority, the authority responsible for the smart cards generates an RSA key-pair. These keys are used for making the smart card and the computer capable of sending some secret values securely between the two, and can at later stages also be used for root-authorized commands such as deleting or adding an issuer. For efficiency reasons not all communication is secured, however, secret values such as PIN, PUK etc. are communicated securely, this is to enable outsourcing of the initialization of the cards, as the secret data is encrypted, but can be decrypted by the root authority. After this the smart card generates a random master secret used for secure issuance of credentials and proofs. A random PIN code and a random PUK code are also generated. The PIN code is used whenever the user wants to use some of the security related functionalities of the smart card. The PUK code is used to re-enable the card if a wrong PIN code has been typed more than three times. The PIN and the PUK codes are sent to the user client and are stored in a file on the root authority computer used to initialize the card.

The smart card is also initialized with a set of cryptographic parameters, specifying which algorithmic groups should be used for computing cryptographic values, and the card is initialized with parameters including the public keys of the issuers, which the card should be able to receive credentials from. These values vary depending on whether it is an Identity Mixer smart card or a U-Prove smart card.

The last initialization step is that the card is changed from being in root mode to be in working mode, and a scope exclusive pseudonym is generated on the card. The generated scope exclusive pseudonym can be stored on the card to make the card capable of proving that it is the correct card, when later communicated with.

9 API Mapping

In this chapter we describe the API mapping of the ABCE for the following Söderhamn Pilot Use Cases described in Chapter 4. The focus is on the ABCE method calls. The Use Cases have also been defined and described in the previous deliverable D6.1 (see Chapter 2 of D6.1 [BGOZ12])

9.1 System Setup

Table 1 and Figure 21 show the API mapping for the first use case System Setup. All the parties of the system need to go through an initialization phase first in order to become functional. For more details about this use case we refer to Section 3.1 of D6.1 [BGOZ12].

Step	Explanation
1	The Eurodocs administrator creates the school registration system wide parameters by invoking the method *setupSystemParameters(keylength:integer, mechanism:anyURI)*
2	The Eurodocs administrator creates issuer parameters and issuer secret key *for credSchool (issuerSchool), credSubject (issuerSubject), credClass (issuerClass), credGuardian (issuerGuardian), credChild (issuerChild)* by invoking the method: *setupIssuerParameters(credspec:CredentialSpecification, syspars:SystemParameters, uid:anyURI, hash:anyURI, revparsuid:anyURI)*
3	The Eurodocs administrator creates the revocation authority parameters by invoking the method: *setupRevocationAuthorityParamter(keylength:integer, mechanism:anyURI, uid:anyURI, inforef:RevocationReference, evidenceref:NonRevocationEvidenceReference, updref:RevocationUpdateReference)*
4	The Eurodocs administrator generates a list of PIN/PUK code values for all issued cards.
5	The Eurodocs administrator initializes each smart card by **(5.1)** setting PIN/PUK, **(5.2)** triggering the generation of the secret key in the trusted part of the device, invoking the method *setupUser(klength:integer, muid: anyURI)*, and **(5.3)** invoking the card to obtain a scope-exclusive pseudonym for the scope "urn:soderhamn:registration" by calling the method *getPseudonymsWithMetaData(urn:soderhamn:registration)*. These steps are repeated for each smart card which has to be initialized.
6	The inspector's card is initialized and inspector's public key is issued by invoking the method *setupInspectorPublicKey(klength:integer, muid:anyURI, uid:anyURI).*

Table 1: ABC System Setup

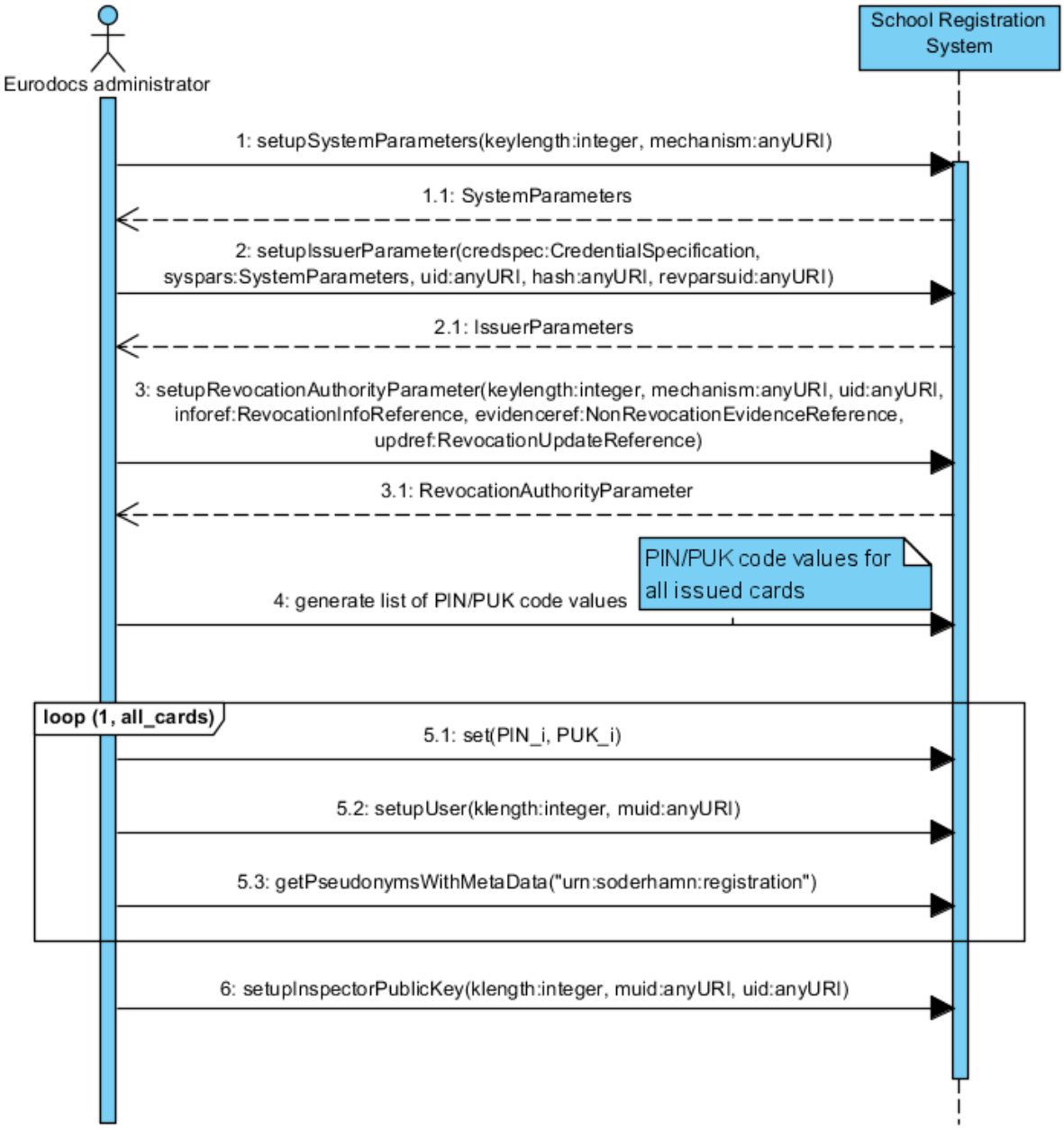

Figure 21. ABC System Setup

9.2 Smart Card Registration using One-Time-Password

Here we present the API mapping for the use case *Smart Card Registration using One-Time-Password.* This use case describes the steps needed for a user to receive the main credential that includes the identity information and can be used to prove that she is a registered user in the school pilot.

Step	Explanation

1	The user logs into the school registration system and authenticates using user ID and OTP: *Login(ID, OTP).* She gets *PresentationPolicy PolicyUID="urn:soderhamn:policies:loginPseudonym"* in return by the school registration system **(1.1).**
2	The user invokes the *createPresentationToken(p:PresentationPolicyAlternatives)* method using the received presentation policy from **(1)** in order to obtain the presentation token containing the requested pseudonym. This token is then sent to the school registration system **(2.1)**.
3	The school registration system verifies the token by invoking the *verifyTokenAgainstPolicy(p:PresentationPolicyAlternatives, t:PresentationToken, store:boolean)* method **(3.1).** If the verification is successful, the school registration system updates user's data set and registers the smart card **(3.2).**
4	The One-Time-Password is disabled.

Table 2: Smart Card Registration using One-Time-Password

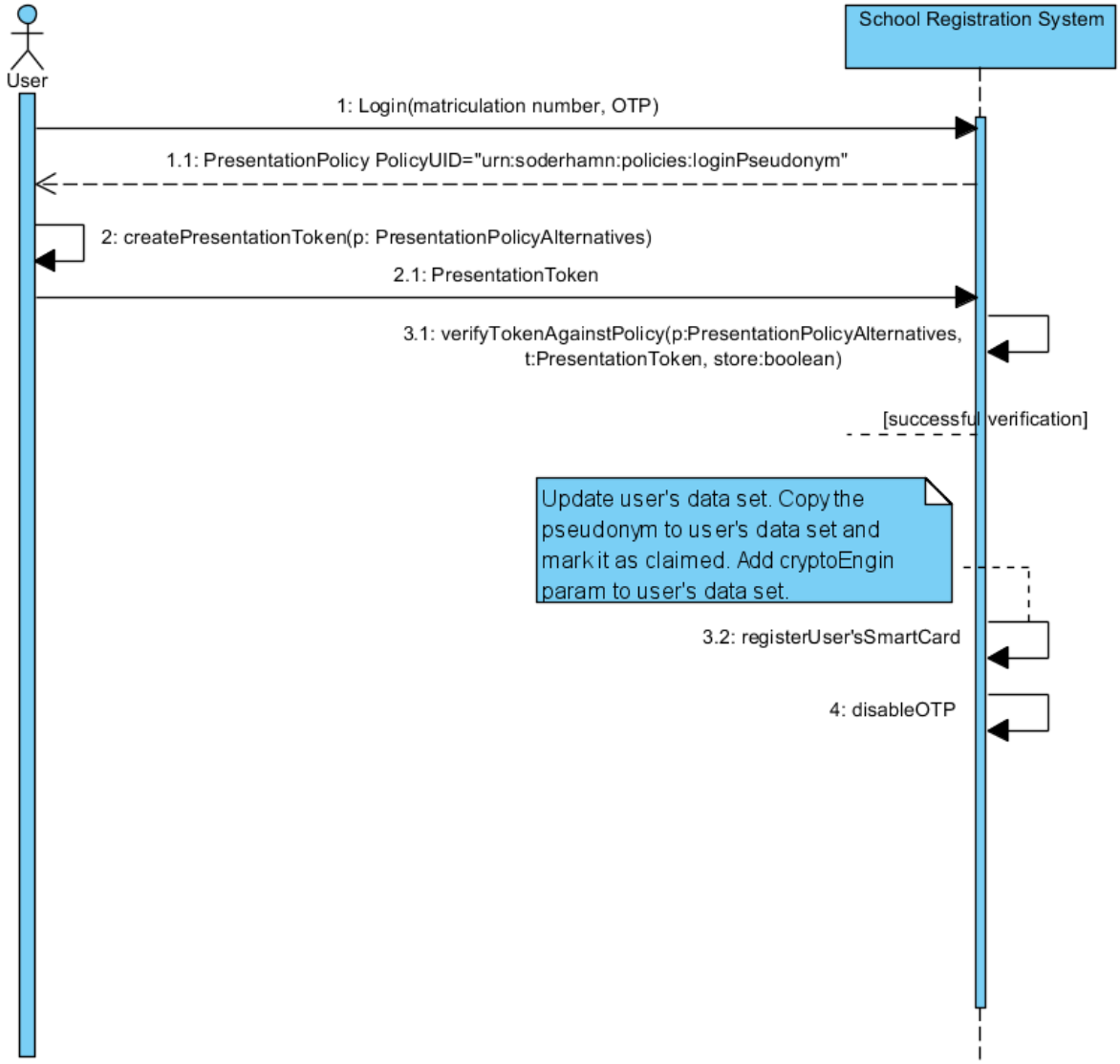

Figure 22. Smart Card Registration using One-Time-Password

9.3 Subsequent Logins to the School Registration System

Here we present the API mapping for the use case *Subsequent Logins to the School Registration System*. Table 3 and Figure 23 depict the steps of receiving a new credential, which contains updated attributes.

Step	Explanation
1	The user logs in to the school registration system and authenticates via ABC technology which requires entering correct PIN. She gets *PresentationPolicy PolicyUID="urn:soderhamn:policies:loginPseudonym"* as a return.
2	The user invokes the *createPresentationToken(p:PresentationPolicyAlternatives)* method using the received presentation policy from **(1)** in order to obtain the presentation token

	containing the requested pseudonym. This token is then sent to the school registration system.
3	The school registration system verifies the token by invoking the *verifyTokenAgainstPolicy(p:PresentationPolicyAlternatives, t:PresentationToken, store:boolean)* method.

Table 3: Subsequent Logins to the School Registration System

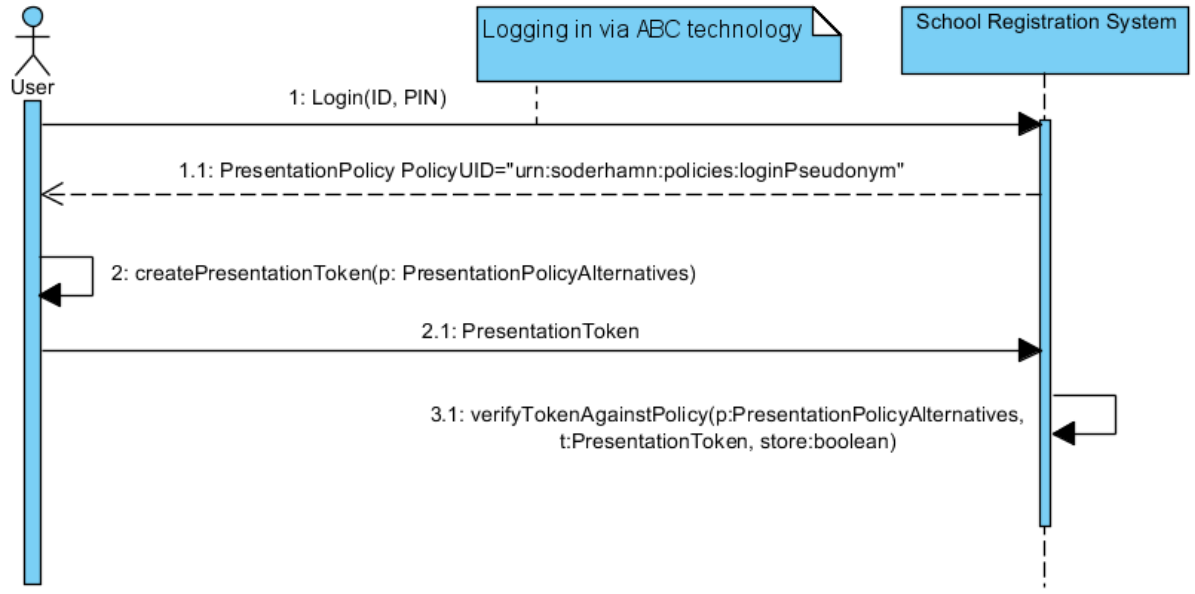

Figure 23. Subsequent Logins to the School Registration System

9.4 Obtaining the School Credential

Here we present the API mapping for the use case *Obtaining the School Credential*. It describes which steps are needed for a user to receive or update their school.

Step	**Explanation**
1	The user is logged in to the school registration system using ABC technology and requests obtaining the school credential **(1)**. She gets *PresentationPolicy PolicyUID="urn:soderhamn:policies:loginPseudonym"* as a return **(1.1)**.
2	The user invokes the *createPresentationToken(p:PresentationPolicyAlternatives)* method using the received presentation policy from Step 1 in order to obtain the presentation token containing the requested pseudonym **(2)**. This token is then sent to the school registration system **(2.1)**.
3	The school registration system verifies the token by invoking the *verifyTokenAgainstPolicy(p:PresentationPolicyAlternatives, t:PresentationToken, store:boolean)* method.

4	Upon successful verification, the school registration system fetches the attribute values of the user attached to the presented pseudonym and starts the issuance protocol for the *school* credential by invoking the method*: initIssuanceProtocol(ip:IssuancePolicy, atts:Attribute[])* **(4.1)**. The returned *IssuanceMessage* is sent to the user **(4.2)**.
5	The user invokes the method *issuanceProtocolStep(m:IssuanceMessage)* with the received IssuanceMessage **(5)** to obtain a new *IssuanceMessage* that has to be sent back to the issuer **(5.1)**.
6	The issuer gets the IssuanceMessage from the user and feeds it to the *issuanceProtocolStep(m:IssuanceMessage)* method **(6)** in order to obtain a new *IssuanceMessage* which has to be sent back to the user **(6.1)**. Steps 5-6 are repeated until the user's side returns a *CredentialDescription*.

Table 4: Obtaining the School Credential

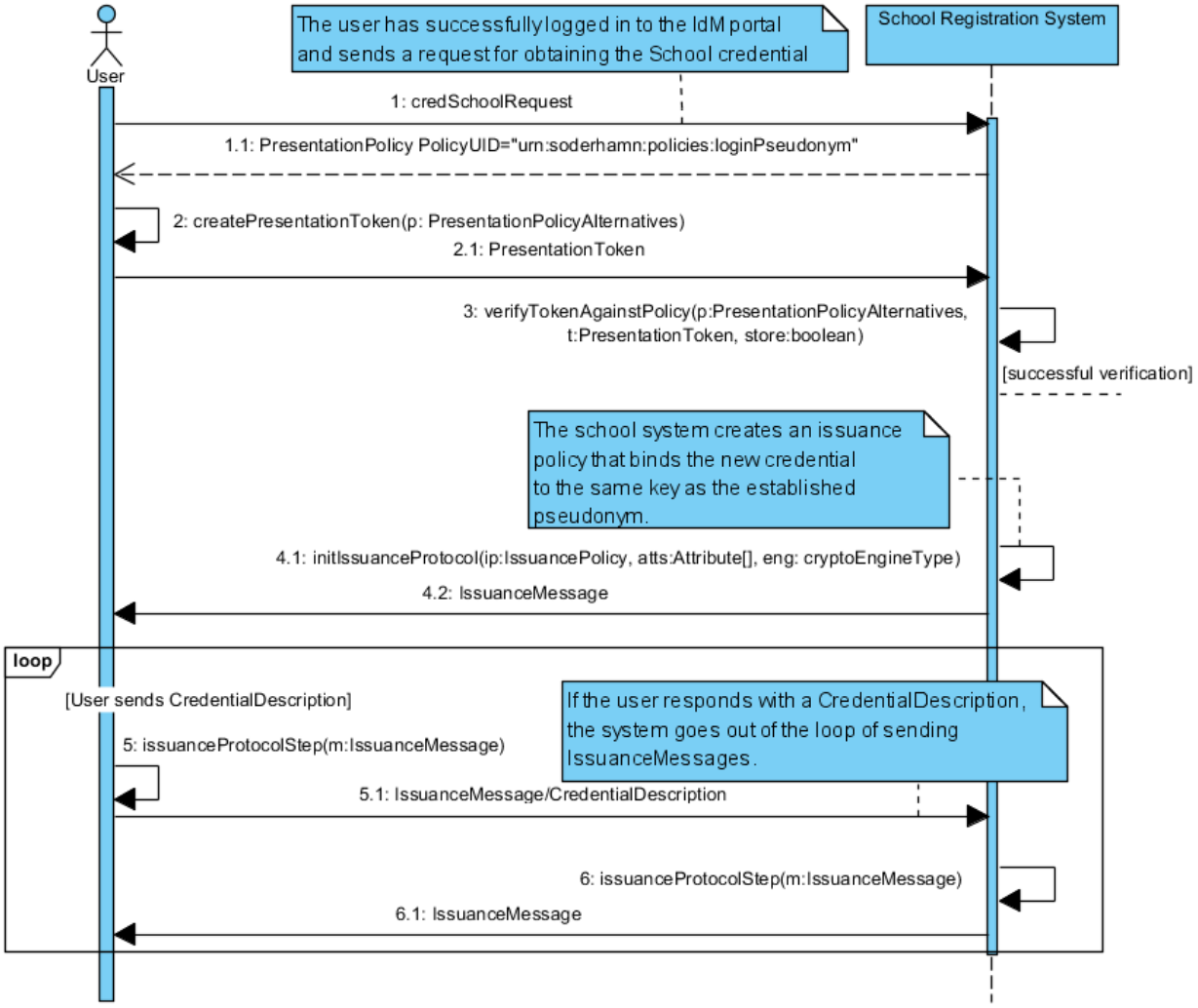

Figure 24. Obtaining the School Credential

9.5 Other School Credentials

Here we present the API mapping for the use case *Other School Credentials*. It describes which steps are needed for a user to receive or update their auxiliary credentials (e.g. the credentials showing parent-child relationship).

Step	Explanation
1	The user logs in to the school registration system using the ABC technology and request auxiliary credentials **(1)**. She gets *PresentationPolicy PolicyUID="urn:soderhamn:policies:loginPseudonym"* as a return **(1.1)**.
2	The user invokes the *createPresentationToken(p:PresentationPolicyAlternatives)* method using the received presentation policy from Step 1 in order to obtain the presentation token containing the requested pseudonym **(2)**. This token is then sent to the school registration system **(2.1)**.
3	The school registration system verifies the token by invoking the *verifyTokenAgainstPolicy(p:PresentationPolicyAlternatives, t:PresentationToken, store:boolean)* method.
4	Upon successful verification, the school registration system fetches the attribute values of the user attached to the presented pseudonym and starts the issuance protocol for a *other school* credential by invoking the method*: initIssuanceProtocol(ip:IssuancePolicy, atts:Attribute[])* **(4.1)**. The returned *IssuanceMessage* is sent to the user **(4.2)**.
5	The user invokes the method *issuanceProtocolStep(m:IssuanceMessage)* with the received IssuanceMessage **(5)** to obtain a new *IssuanceMessage* that has to be sent back to the issuer **(5.1)**.
6	The issuer gets the IssuanceMessage from the user and feeds it to the *issuanceProtocolStep(m:IssuanceMessage)* method **(6)** in order to obtain a new *IssuanceMessage* which has to be sent back to the user **(6.1)**. Steps 5-6 are repeated until the user's side returns a *CredentialDescription*.

Table 5: Other School Credentials

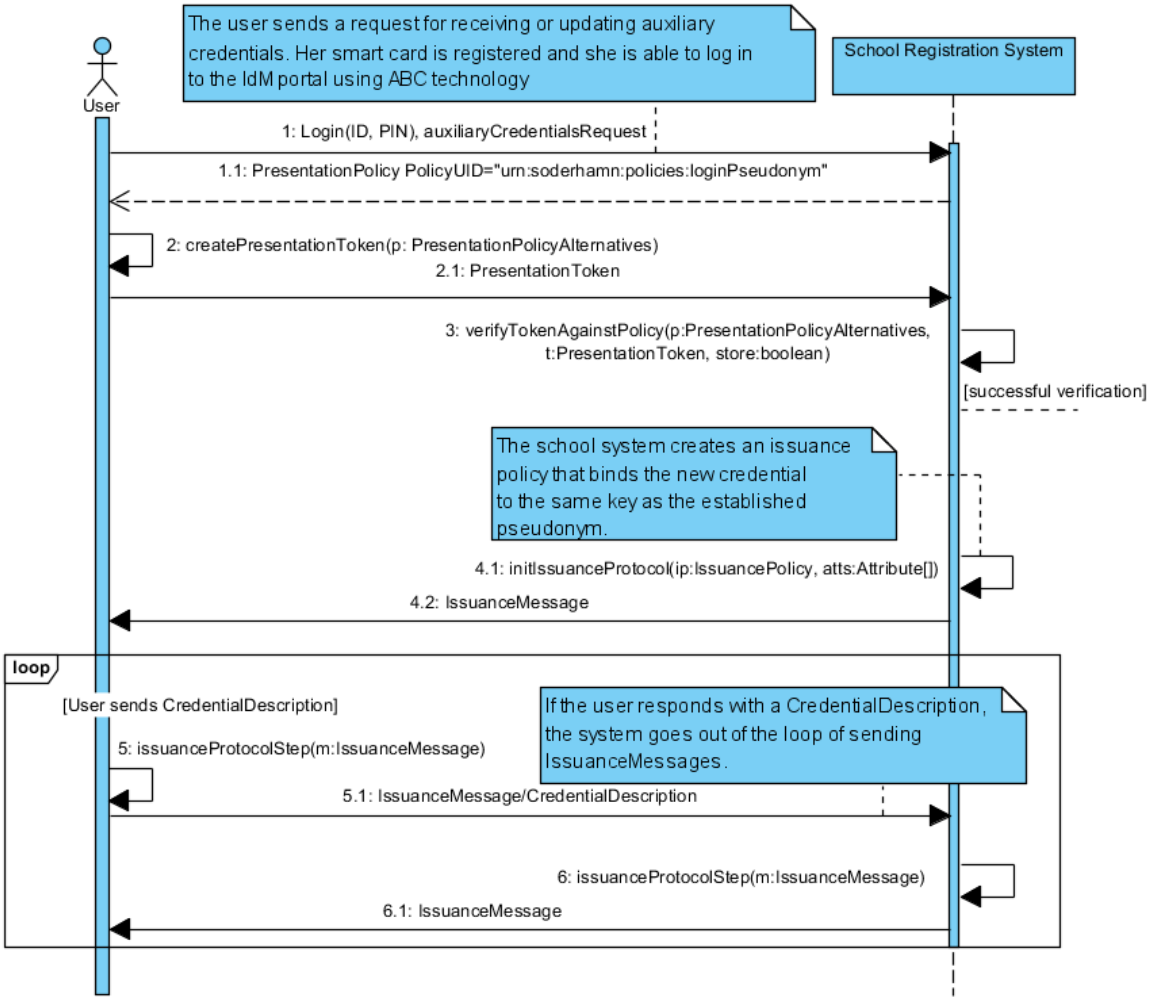

Figure 25. Other School Credentials

9.6 Viewing User's Data

The API mapping for the use case *Viewing of the User's Data* is presented in Table 6 and Figure 26. The use case describes the steps needed for viewing the user's profile.

Step	Explanation
1	The user logs into the school registration system and authenticates using ABC technology **(1)**. The school registration system asks the user to prove the possession of a credential, i.e. satisfying the returned *PresentationPolicy PolicyUID="urn:soderhamn:policies:loginPseudonym"***(1.1)**.
2	The user invokes the *createPresentationToken(p:PresentationPolicyAlternatives)* method using the received presentation policy from Step 1 in order to obtain the presentation token containing the requested pseudonym **(2)**. This token is then sent to the school registration system **(2.1)**.

3	The school registration system verifies the token by invoking the *verifyTokenAgainstPolicy(p:PresentationPolicyAlternatives, t:PresentationToken, store:boolean)* method . If the verification is successful, the profile of the user is displayed **(3.2)**.
4	In case of successful verification, the user can select the Admin button to view her data **(4.1)**. User's profile is displayed **(4.2)**.

Table 6: Viewing User's Data

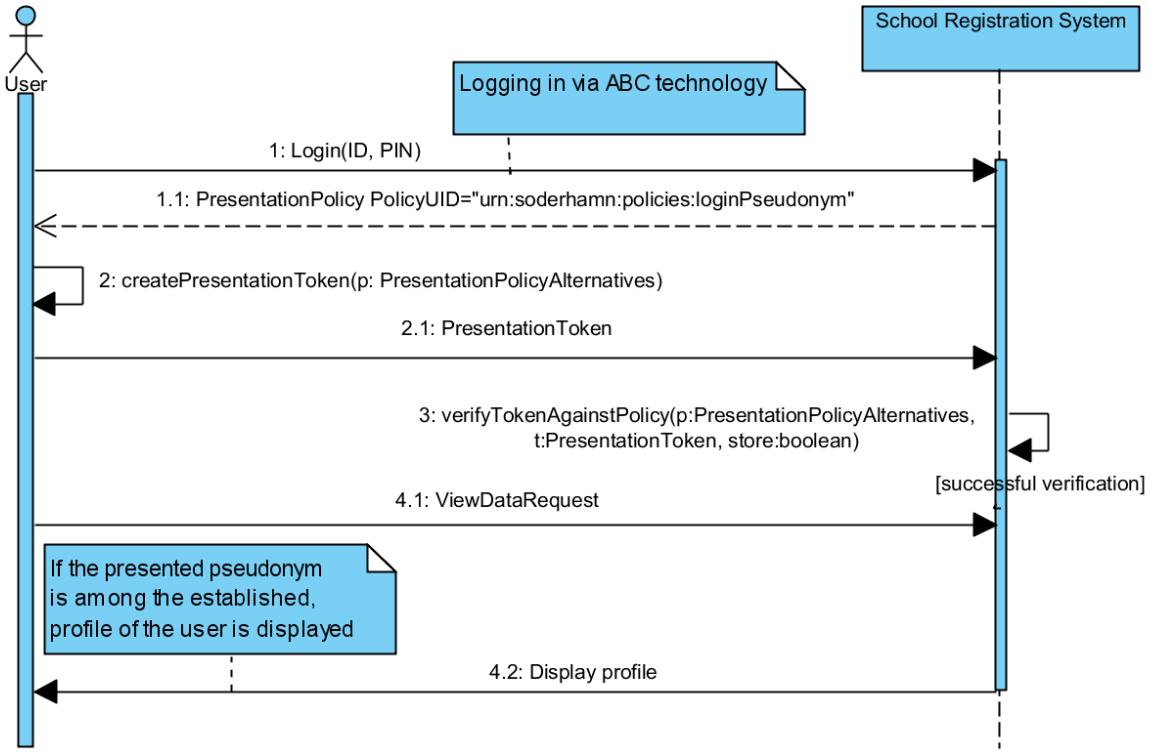

Figure 26. Viewing User's Data

9.7 Login to Restricted Area Application

Here we present the API mapping for the use case *Instantiating a Restricted Area*. Table 7 and Figure 27 describe the steps needed for the creation of a Restricted Area, i.e., a discussion board with a custom access control policy.

Step	Explanation
1	The user is logged into the RA system using ABC technology and requests accessing a restricted area application **(1)**. The system asks the user to prove the possession of certain attributes (e.g. member of staff, etc.), i.e. satisfying the returned policy **(1.1)**.
2	The user invokes the *createPresentationToken(p:PresentationPolicyAlternatives)* method using the received presentation policy from Step 1 in order to obtain the presentation token

	containing the requested pseudonym.
3	The PresentationToken is then sent to the RAA.
4	The Restricted Area Application verifies the token by invoking the *verifyTokenAgainstPolicy(p:PresentationPolicyAlternatives, t:PresentationToken, store:boolean)* method **(3.1)**.
5	Upon successful verification the user is granted access to the restricted area application.

Table 7: Login to Restricted Area Application

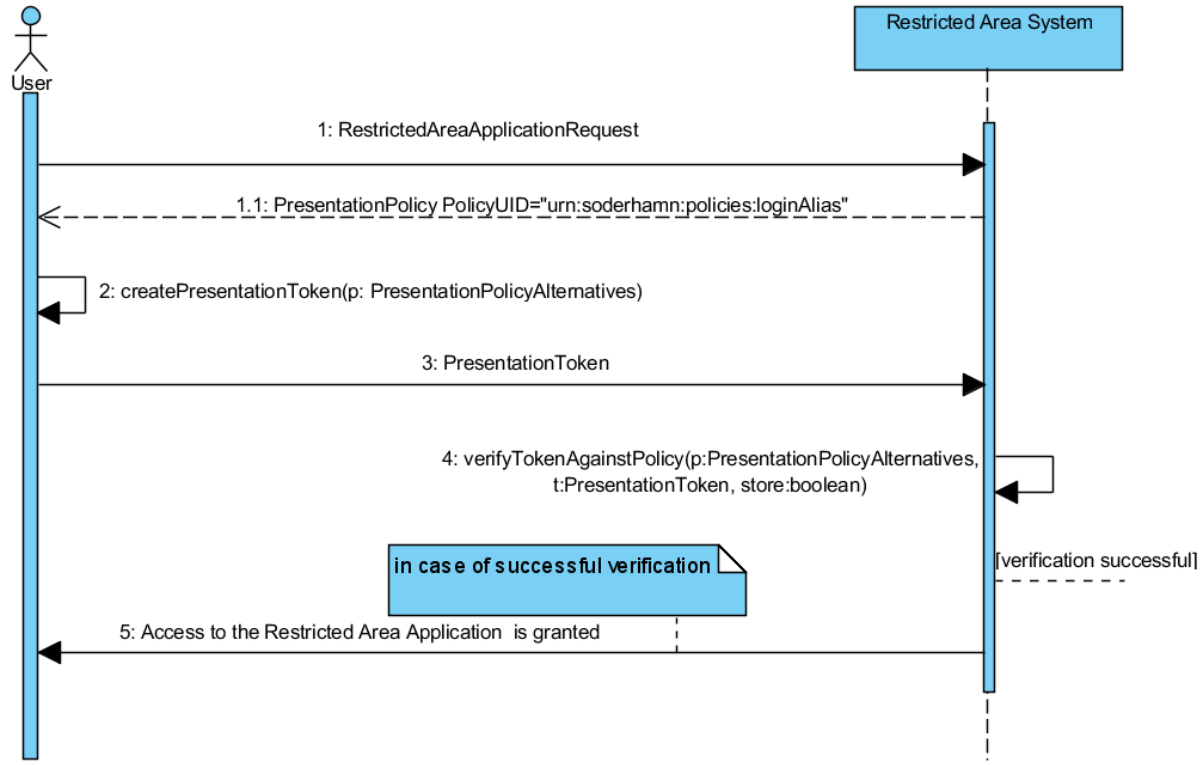

Figure 27. Login to Restricted Area Application

9.8 Choose or Create Alias

Here we present the API mapping for the use case *Choose or Create Alias.* Table 8 and Figure 28 describe the steps needed for a user to choose a previously established alias or to create a new one.

Step	**Explanation**
1	The user logs into the RA system using ABC technology and chooses to create a new alias. She sends a request for creating a new alias with a chosen aliasname to the school registration system **(1)**. The school registration system checks the availability of the aliasname **(1.1)**. If the aliasname is not taken, the RA server send a PresentationPolicy to the user requesting her to present a scope-exclusive pseudonym for scope string urn:soderhamn:alias:aliasname and a valid school credential **(1.2)**.
2	The user invokes the *createPresentationToken(p:PresentationPolicyAlternatives)* method

	using the received presentation policy from Step 1 in order to obtain the presentation token containing the requested pseudonym **(2)**. This token is then sent to the school registration system **(2.1)**.
3	The Restricted Area Application verifies the token by invoking the *verifyTokenAgainstPolicy(p:PresentationPolicyAlternatives, t:PresentationToken, store:boolean)* method.
4	Upon successful verification the chosen aliasname is associated to the pseudonym and the pseudonym is stored in the database of the Restricted Area Application.
5	The user is granted access.
6	If the user chooses to log in using an established alias, she sends her alias ID and the existing aliasname to the Restricted Area Application **(6)**. The Restricted Area Application responds with the respective presentation policy **(6.1)**.
7	The user invokes the createPresentationToken(p:PresentationPolicyAlternatives) method using the received presentation policy from Step 1 in order to obtain the presentation token containing the requested pseudonym **(7)**. This token is then sent to the school registration system **(7.1)**.
8	The Restricted Area Application verifies the token by invoking the verifyTokenAgainstPolicy(p:PresentationPolicyAlternatives, t:PresentationToken, store:boolean) method.
9	Upon successful verification the user is granted access.

Table 8: Choose or Create Alias

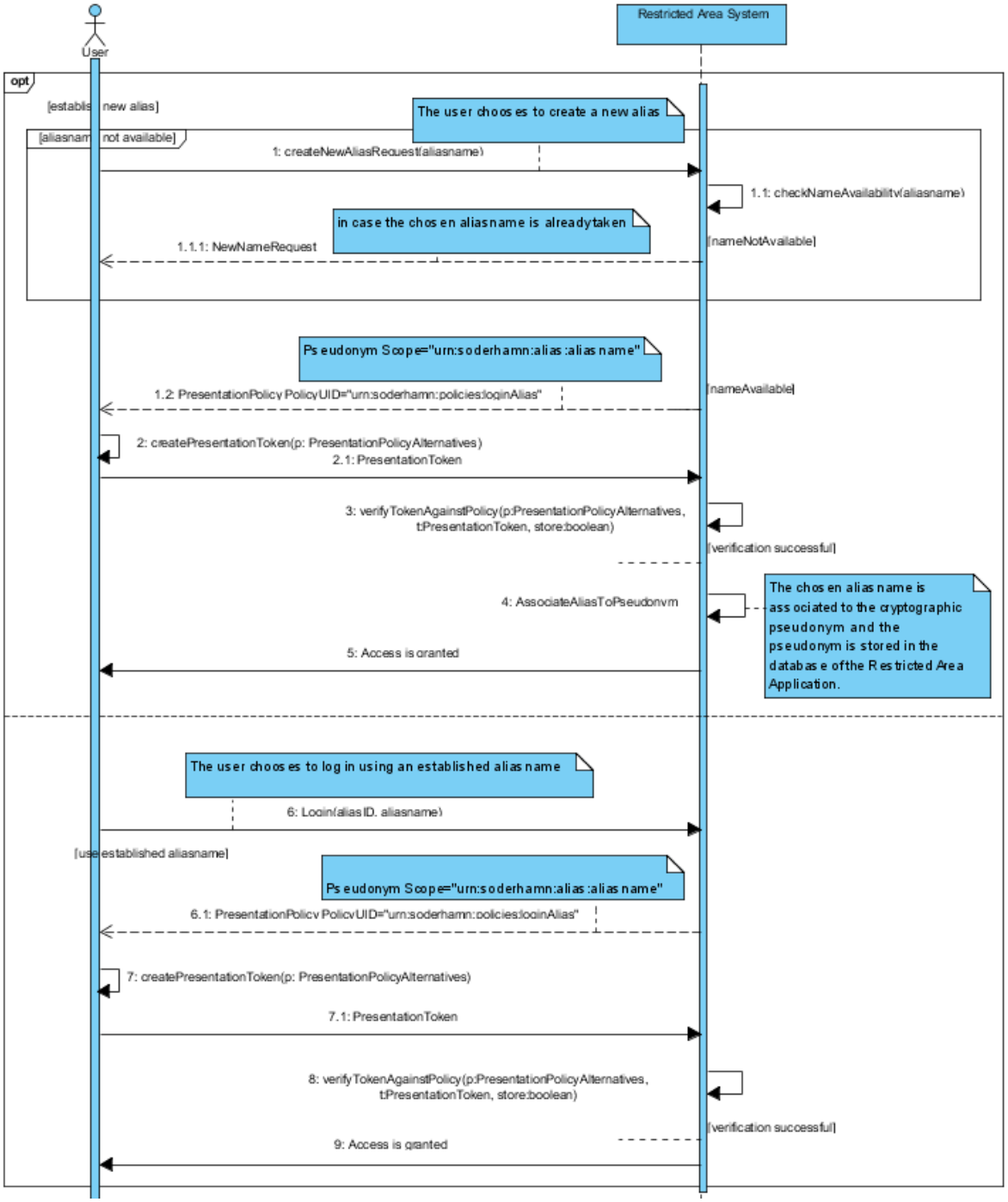

Figure 28. Choose or Create Alias

9.9 Instantiating a Restricted Area

Here we present the API mapping for the use case *Instantiating a Restricted Area* Table 9 and Figure 29 describe the steps needed for the creation of a Restricted Area, i.e., a discussion board with a custom access control policy.

Step	Explanation

1	The user is logged into the RA system using ABC technology and sends an InstantiateRestrictedArea request.
2	The user is prompted by the Restricted Area Application to define the access policy that must be applied when other users want to access the RA **(2.1)**. The user responds with the defined access policy **(2.2)**.
3	The received access policy is represented as PresentationPolicyAlternatives and associated to the created RA.

Table 9: Instantiating a Restricted Area

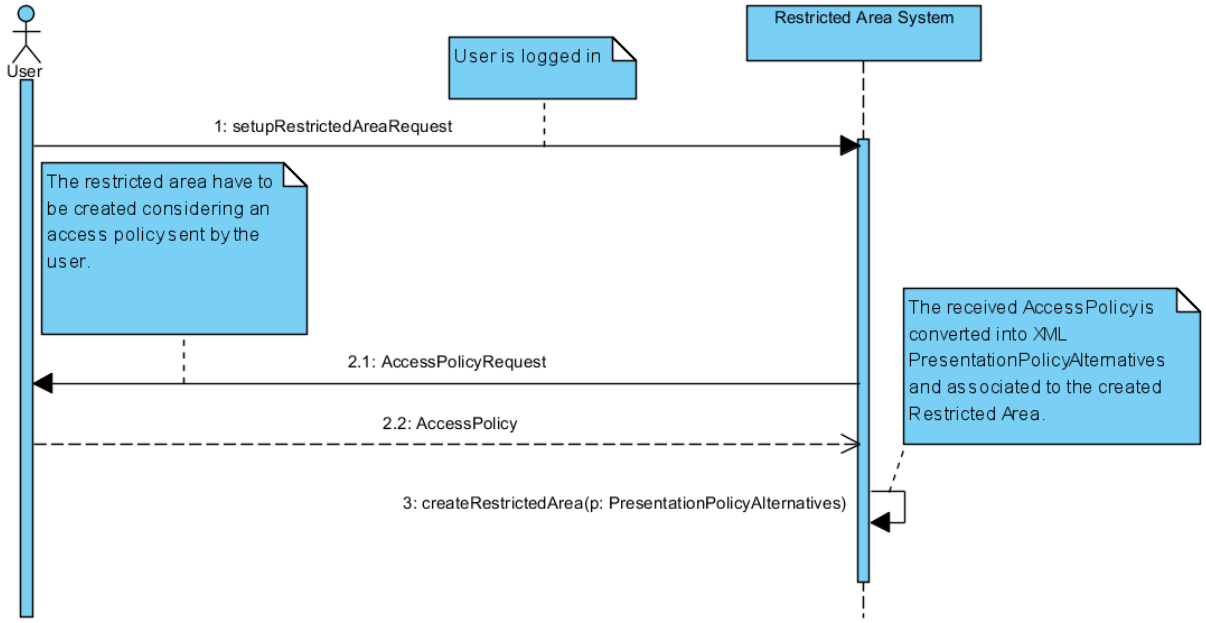

Figure 29. Instantiating a Restricted Area

9.10 Access to a Restricted Area

In Table 10 and Figure 30 we present the API mapping for the use case *Access to a Restricted Area*. In order to access the restricted area, the user must possess a number of credentials that fulfil the access policy of this area.

Step	Explanation
1	The user requests access to a restricted area **(1)** on the RA server using either a previously established alias or a new one. If the chosen alias was previously used in non-inspectable restricted areas and this is the first time it is used in an inspectable restricted area, a warning pops up. The user is warned that the messages issued by this alias in the non-inspectable restricted areas become inspectable, too **(1.1)**. If this was not the first time the alias accesses this restricted area and the last session is still open, no further authentication is needed **(1.2)**. If this is the first time the alias accesses the RA or the session expired, the user is sent the PresentationPolicy associated with this restricted area **(1.3)**.
2	The ABC User System side generates a list of PresentationTokenDescriptions using the *PresentationPolicyAlternatives* as input. Then the custom identity selector is invoked to ask

	the user what information should be revealed **(2)**. The respective PresentationToken is sent to the school `registration` system **(2.1)**.
3	The Restricted Area Application verifies the *PresentationToken* by invoking the *verifyTokenAgainstPolicy(p:PresentationPolicyAlternatives, t:PresentationToken, store:boolean)* method and extracts the cryptographic pseudonym from the PresentationToken **(3)**. In case of successful verification, the alias is registered **(3.1)**.
4	In case the RA is inspectable, the messages written during the session are associated to the token identifier.
5	If the verification was successful, the user is granted access to the restricted area.

Table 10: Access to a Restricted Area

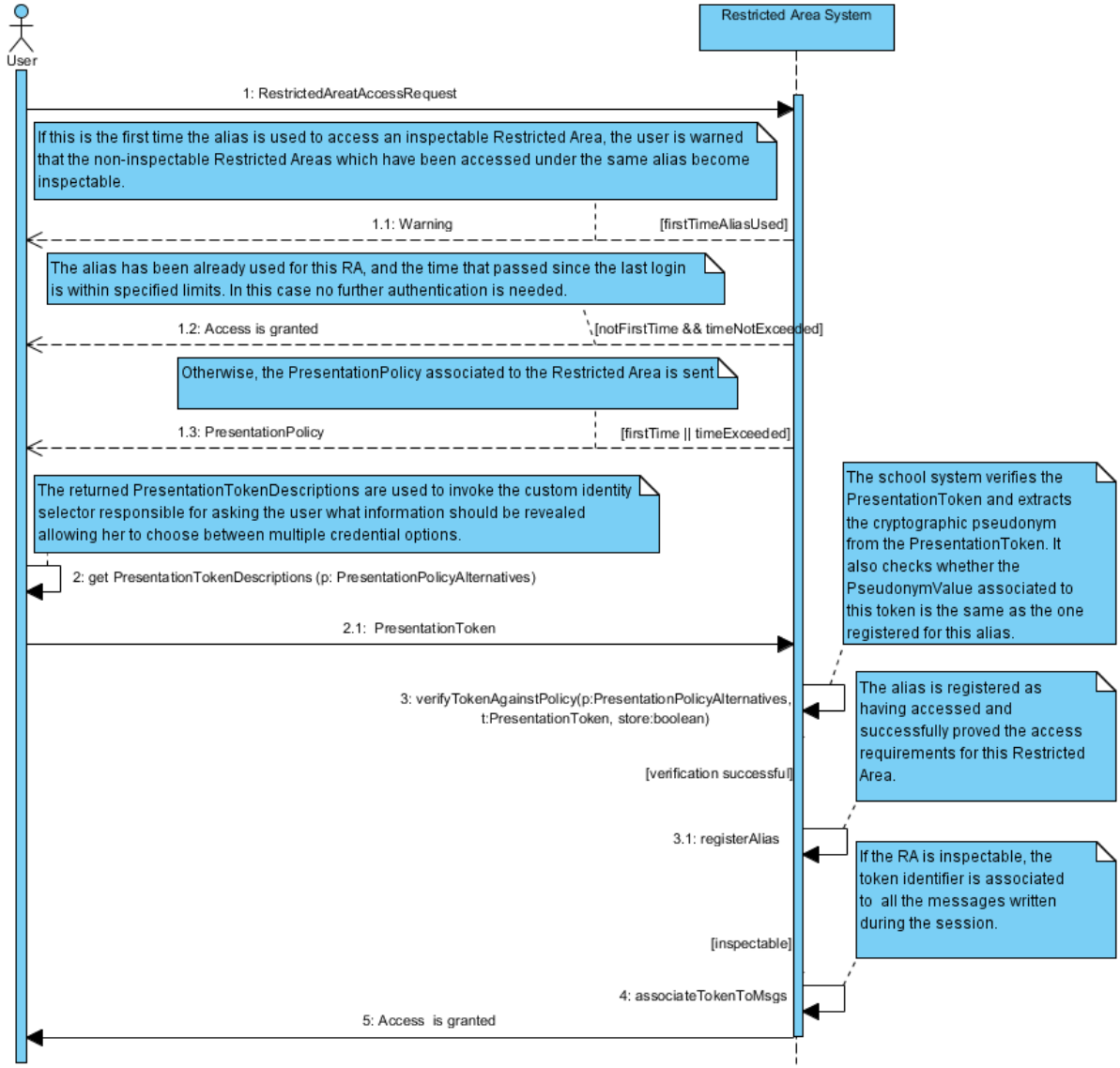

Figure 30. Access to a Restricted Area

9.11 Counselling

Here we present the API mapping for the use case *Counselling* . It describes the way a pupil gets help and advice from school personnel using the school registration system.

Step	Explanation
1	The pupil is logged in to the Restricted Area Application using an existing or a new alias and browses the counselling section triggering a counselling request **(1)**. The School Registration System responds sending the PresentationPolicy corresponding to the way the user intends to access the counselling section **(1.1)**.
2	The user invokes the method *createPresentationToken(p:PresentationPolicyAlternatives)* in order to obtain a credential complying with the received PresentationPolicy. A list of PresentationTokenDescriptions is generatedusing the PresentationPolicyAlternatives as input. The custom identity selector is invoked to ask the user about the information which should be revealed.
3	The obtained PresentationToken is sent to the school registration system.
4	The Restricted Area Application verifies the PresentationToken received by the user by triggering the method *verifyTokenAgainstPolicy(p:PresentationAlternatives, t:PresentationToken, store:boolean).*
5	If the verification was successful, the pupil gets help from the school personnel.
6	In case a third party should be involved in the counselling process, the user is asked for consent **(6)**. Upon the receipt of the consent **(6.1)**, the third party is involved in the discussion **(6.2)**.

Table 11: Counselling

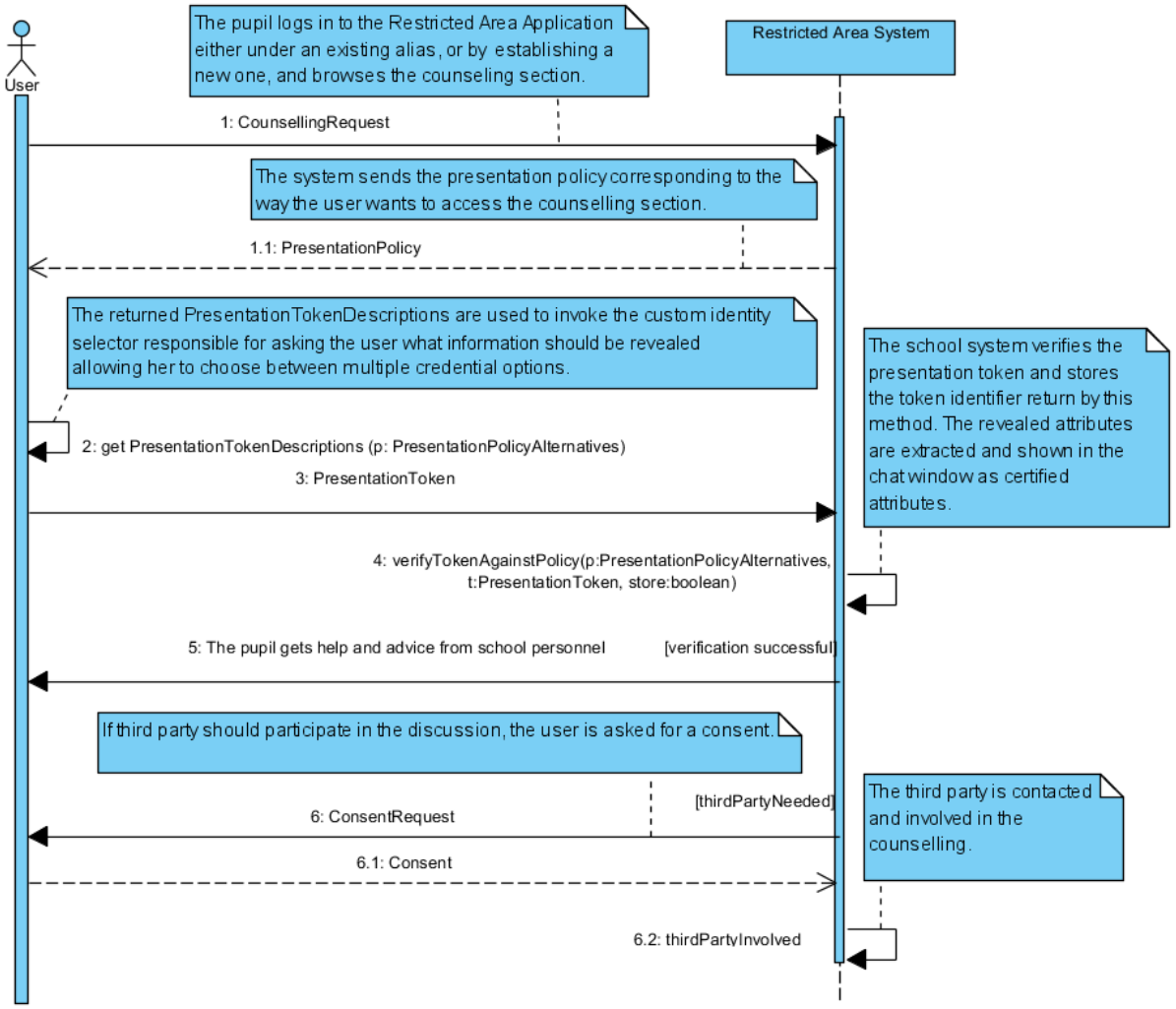

Figure 31. Counselling

9.12 Restricted Chat Room

Here we present the API mapping for the use Restricted Chat Room. The use case describes two use cases which are almost identical, one where a person chats with other users by entering a chat room (group), another where a person that wants to chat with another person in a private chat room (one-to-one).

Step	Explanation
1	The user requests access to a (private) chat room (a kind of restricted area with chat functionality) **(1).** As a response, she gets the PresentationPolicyAlternatives associated with this restricted area **(1.1).**
2	The ABC user system side generates a list of PresentationTokenDescriptions using the PresentationPolicyAlternatives as input. Then the custom identity selector is invoked to ask the user about the credential option she wants to choose **(2)**. The respective PresentationToken is sent to the school registration system **(2.1)**.

3	The Restricted Area Application verifies the PresentationToken by invoking the *verifyTokenAgainstPolicy(p:PresentationPolicyAlternatives, t:PresentationToken, store:boolean)* method of the ABCE and extracts the cryptographic pseudonym from the PresentationToken.
4	If the verification is successful the user is granted access to the (private) chat room. The user is able to see a list of online aliases, post a message visible to all participants or chat with other alias owners

Table 12: Restricted Chat Room

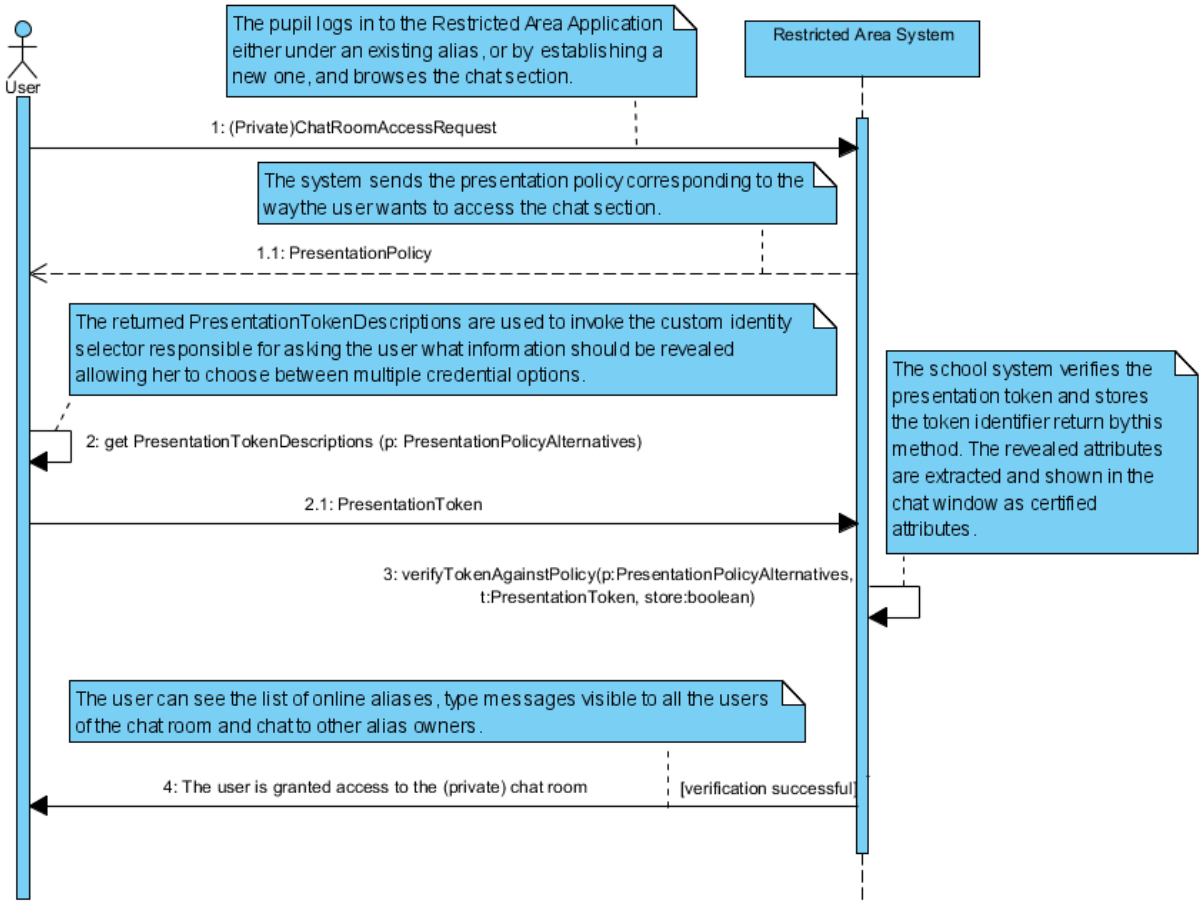

Figure 32. Restricted Chat Room

9.13 Political Discussions

Here we present the API mapping for the use case *Political Discussions*. The technical details are almost identical to use case 9.12.

Step	Explanation
1	The user is requesting access a chat room for political discussions **(1)**. The RA System

	responds sending the PresentationPolicy corresponding to the requested area **(1.1)**. The political discussions are intended to be anonymous and therefore the policy contains no option for inspecting the token. It does not contain a block asking to disclose the PUN number to the inspector.
2	The user invokes the method *createPresentationToken(p:PresentationPolicyAlternatives)* in order to obtain a PresentationToken corresponding to the received PresentationPolicy **(2)**. The obtained PresentationToken is sent to the School Registration System **(2.1)**.
3	The Restricted Area Application verifies the token considering the policy triggering the method *verifyTokenAgainstPolicy(p:PresentationPolicyAlternatives, t:PresentationToken, store:boolean).*
4	If the verification was successful, the user is granted access to the (private) chat room for political discussions.

Table 13: Political Discussions

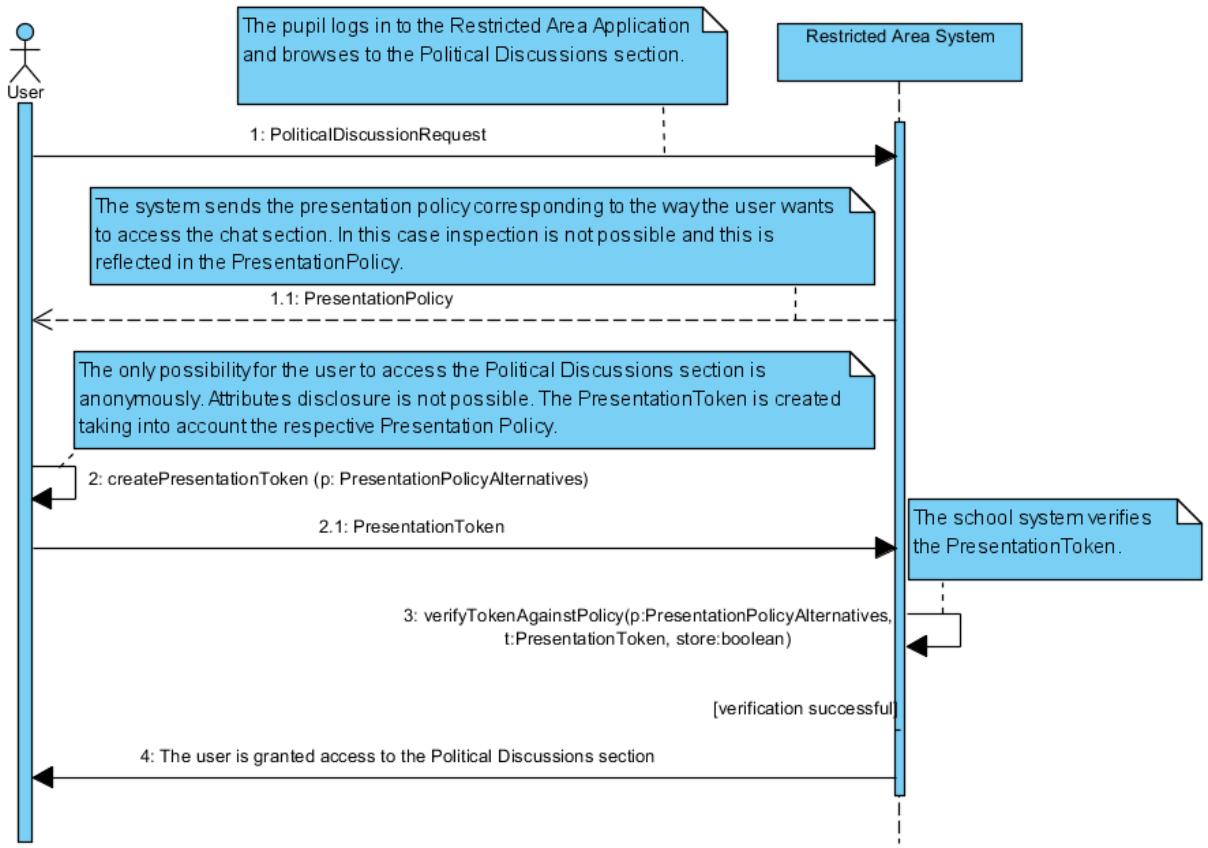

Figure 33. Political Discussions

9.14 Sharing Documents

Here we present the API mapping for the use case *Sharing Documents.* Technically, this use case is a special case of a restricted area where users are completely identified.

Step	Explanation
1	The user is requesting access to the shared documents **(1).** As a response, a PresentationPolicy is sent by the Restricted Area Application **(1.1)**.
2	The user generates a presentation token using the method *createPresentationToken(p:PresentationPolicyAlternatives)* and taking into consideration the provided PresentationPolicy **(2)**. The PresentationToken is sent to the school registration system **(2.1)**.
3	The Restricted Area Application verifies the token triggering the method *verifyTokenAgainstPolicy(p:PresentationPolicyAlternatives, t:PresentationToken, store:boolean*
4	If the verification is successful, the user is granted access to the shared documents.

Table 14: Sharing Documents

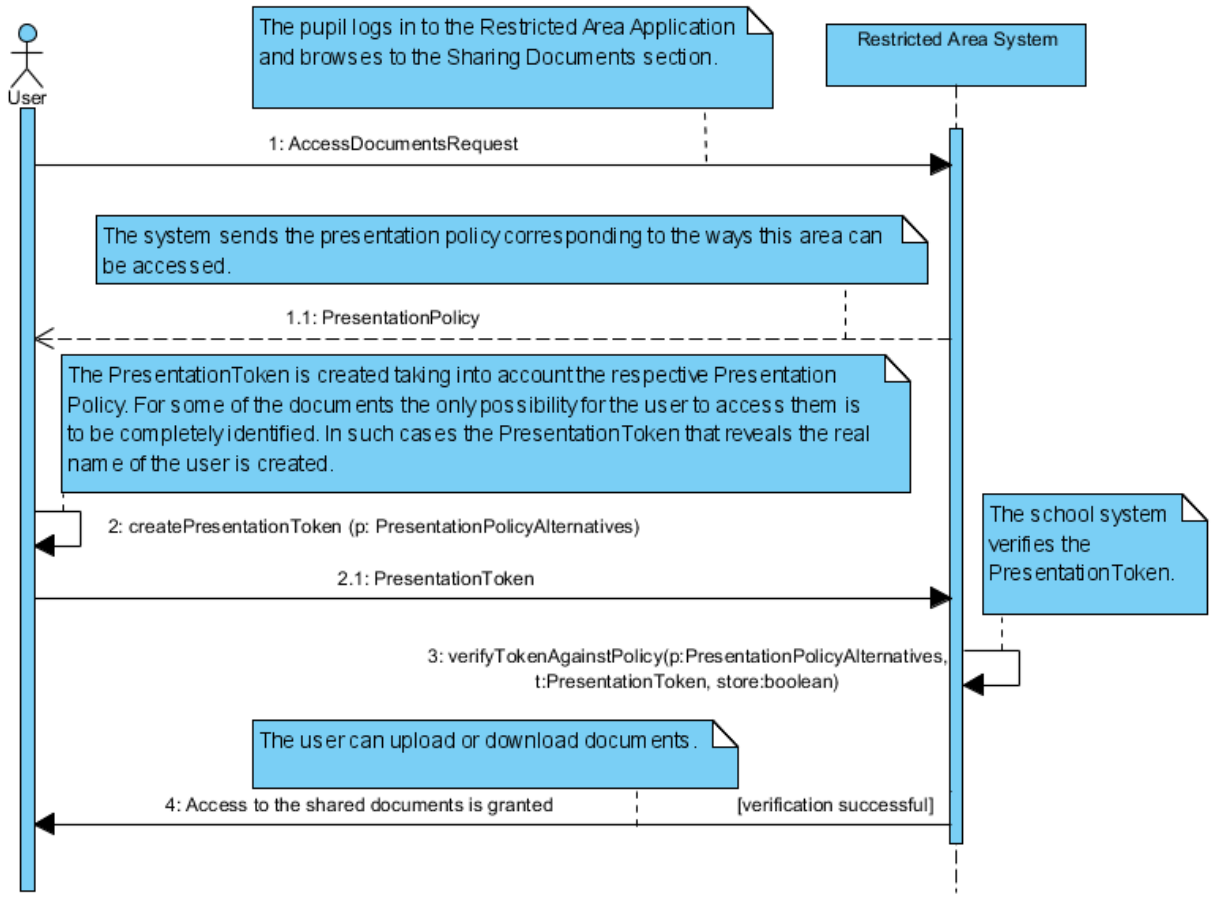

Figure 34. Sharing Documents

9.15 Revocation

In this section the use case *Revoking a User's Credential* is described. The steps involved in the revocation of a credential are shown in Table 15 and Figure 35.

Step	Explanation
1	A revocation requestor sends a request to revoke a credential to the System Administrator.
2	The System Administrator verifies the condition for revoking the credential.
3	In case the revocation condition is fulfilled, the user is given the possibility to backup her data or to delete her contributions from Restricted Areas where she used her real name. These possibilities are presented to the user **(3)**. The user responds choosing to delete/backup some personal data **(3.1)**. The system administrator takes actions corresponding to user's choice **(3.2)**.
4	If the revocation condition is fulfilled, the System Administrator looks up the revocation handle associated to the credential which has to be revoked and send it to the Revocation Authority.
5	The School Registration System revokes the given revocation handle triggering the method *revoke(rparsuid:anyURI, atts:Attribute[]).*
6	The revocation information is published.

Table 15: Revocation

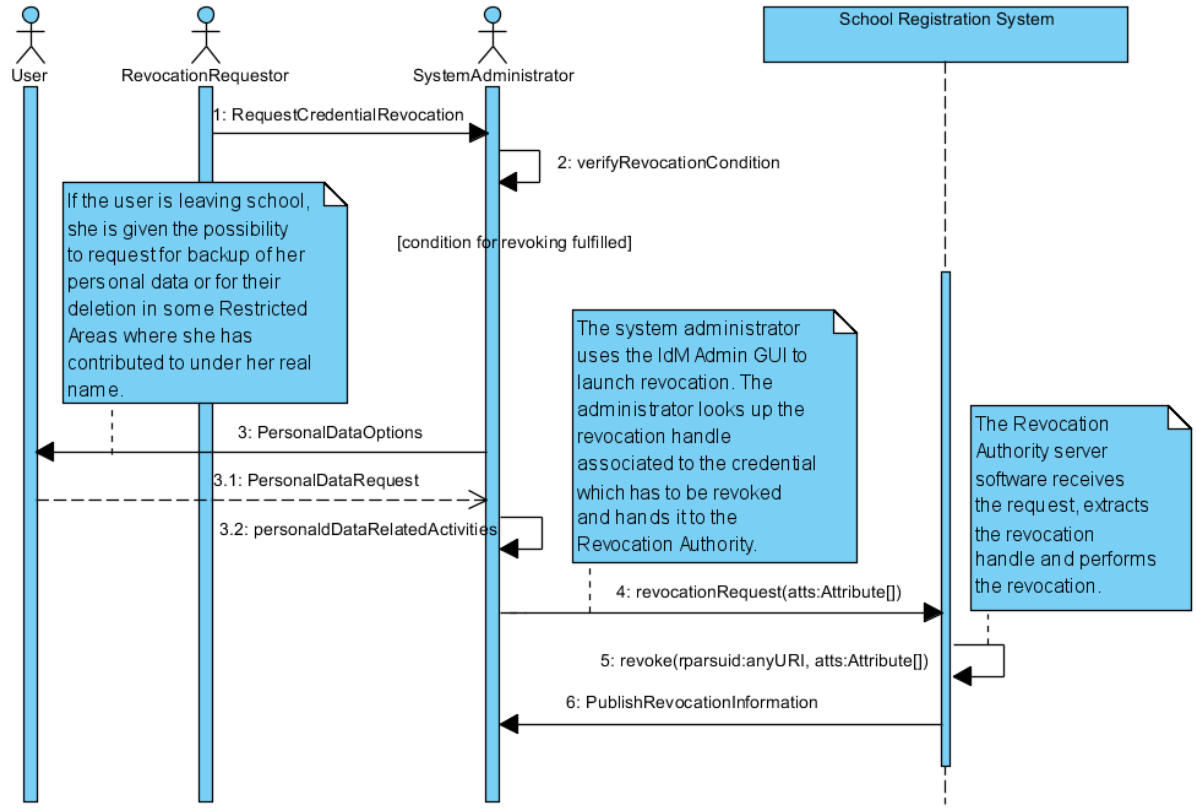

Figure 35. Revocation

9.16 Emergency Situation (Inspection)

We present the API mapping for the use case Emergency Situation. This use case describes which steps are made in case of an emergency situation that causes the school registration system to start the inspection procedure for a particular communication.

Step	Explanation
1	A user reports an emergency situation to the School Inspection Board.
2	The School Inspection Board investigates the case and if the board decides to trigger inspection, an InvestigationRequest is sent to the Restricted Area Syste.
3	The Restricted Area System looks up the identifier of the presentation token that was linked to the offending communication, and fetches the corresponding presentation token by invoking the method *getToken(tokenuid:anyURI)*.
4	The *PresentationToken* returned by the *getToken(tokenuid:anyURI)*-method is sent to the ABC Inspector via a secure channel.
5	The Inspector triggers the method *inspect(t:PresentationToken)*.
6	The Inspector sends the decrypted attribute back to the School Inspection Board.
7	The School Inspection Board takes some appropriate actions.

Table 16: Emergency Situation (Inspection)

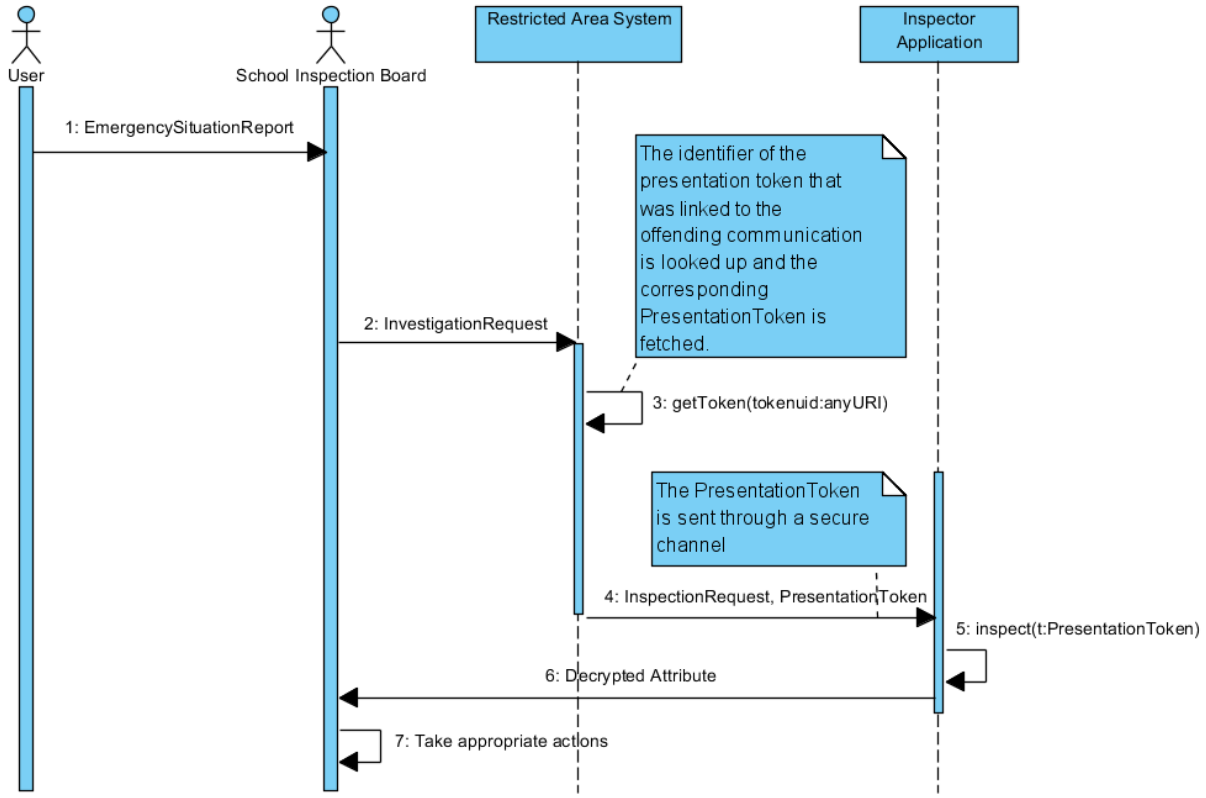

Figure 36. Emergency Situation (Inspection)

9.17 Viewing/Deleting Credentials

Table 17 and Figure 37 present the API mapping for the use case Viewing/Deleting Credentials. This use case describes the steps needed for a user to view/delete her credentials stored on her smart card.

Step	Explanation
1	The user uses the browser plugin to request to view her credentials. The browser plugin will prompt her to insert her smart card and enter PIN code.
2	User ABC System invokes the method *listCredentials()* to get a list of the credentials associated with the user **(2)**. This list of credentials is then displayed to the user **(2.1)**.
3	The user sends a request to obtain a *CredentialDescription* for a specific credential.
4	User ABC System invokes the method *getCredentialDescription(creduid:anyURI)* **(4)** and the returned CredentialDescription is sent to the user **(4.1)**.
5	The user sends a request to delete a credential.
6	User ABC System triggers the method *deleteCredential(creduid:anyURI)* to delete the specified credential **(6)**. The user is informed that the credential was deleted as requested **(6.1)**.

Table 17: Viewing/Deleting Credentials

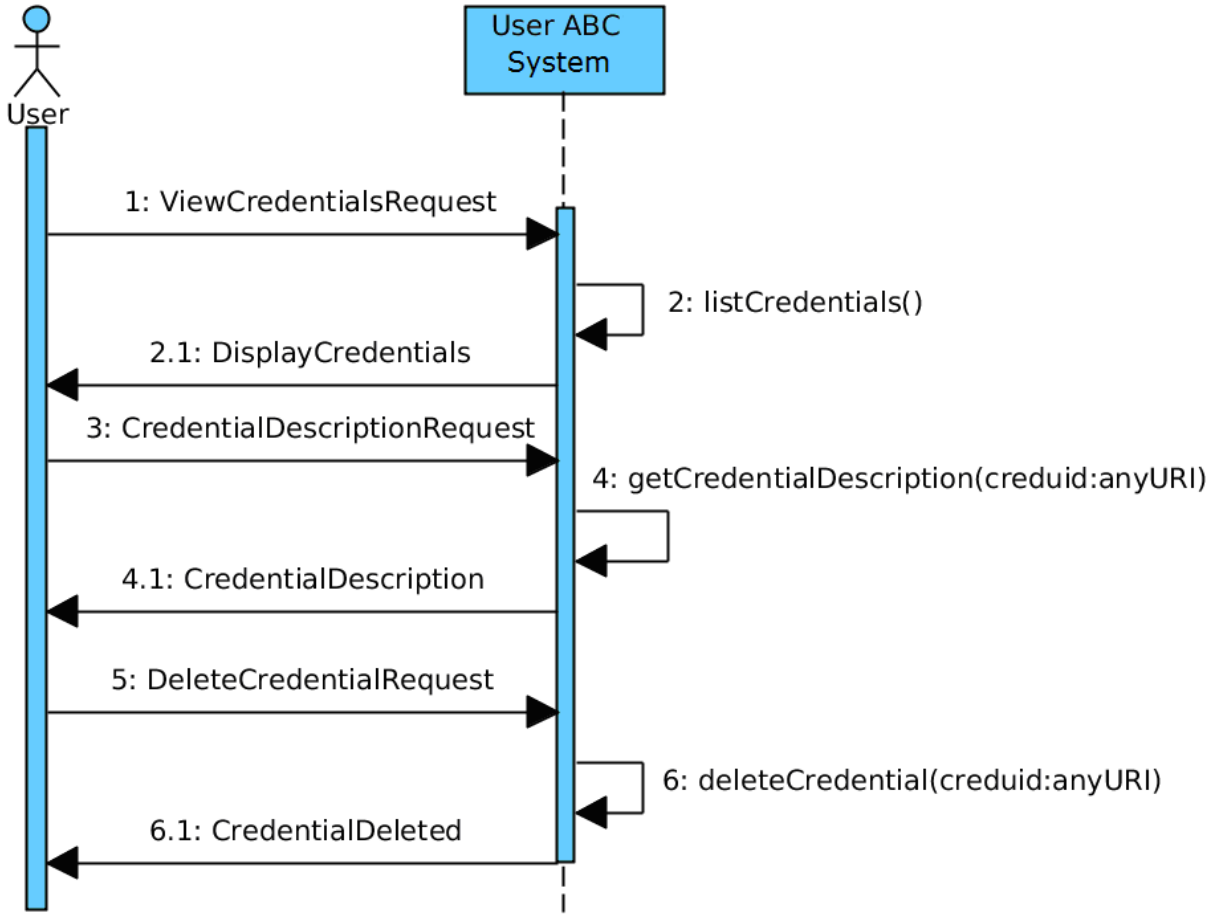

Figure 37. Viewing/Deleting Credentials

9.18 Changing PIN

Table 18 and Figure 38 present the API mapping for the use case Changing PIN. This use case describes the steps needed for a user to change the PIN code protecting her smart card.

Step	Explanation
1	The user selects the Change PIN option in the user interface of the User Client triggering the method CHANGE PIN(oldpin, newpin).
2	The User Client verifies the oldpin the user entered.
3	In case the oldpin is correct, the PIN is successfully changed.
4	The user is informed that the change of the PIN code was successful.
5	The user sends a request to delete a credential.
6	User ABC System triggers the method *deleteCredential(creduid:anyURI)* to delete the specified credential **(6)**. The user is informed that the credential was deleted as requested **(6.1)**.

Table 18: Changing PIN

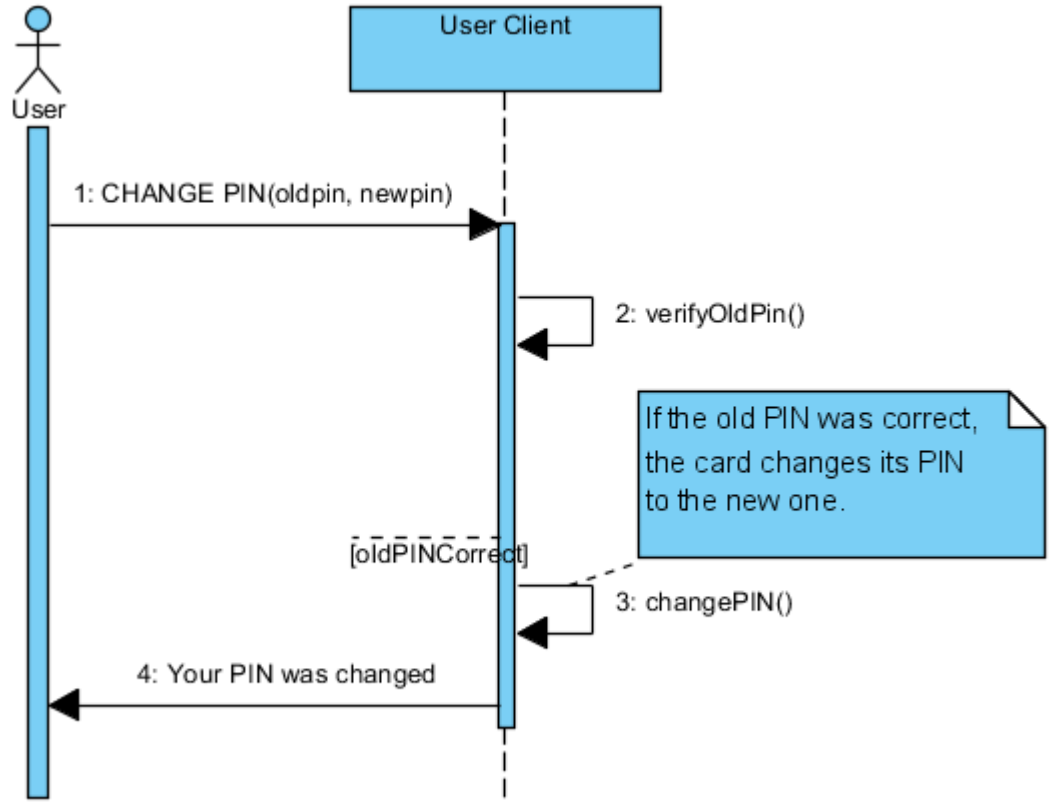

Figure 38. Changing PIN

9.19 Unlocking the Smart Card with a PUK

Table 19 and Figure 39 present the API mapping for the use case Unlocking the Smart Card with a PUK. This use case describes the steps needed for a user to unlock the smart card.

Step	Explanation
1	The user selects unlock card in the user interface of the User Client invoking the method RESET PIN(puk, newpin). The user types the PUK code and the new PIN code.
2	The User Client verifies the entered PUK code.
3	If the PUK code is correct, the PIN code is successfully changed.
4	The user is informed about the successful change of the PIN code.
5	In case the PUK code is wrong, the user is requested to enter it again triggering the RESET PIN(puk, newpin) method in **(5.1).**
6	In case the user types an incorrect PUK 3 times, the smart card enters dead mode.

Table 19: Unlocking the Smart Card with a PUK

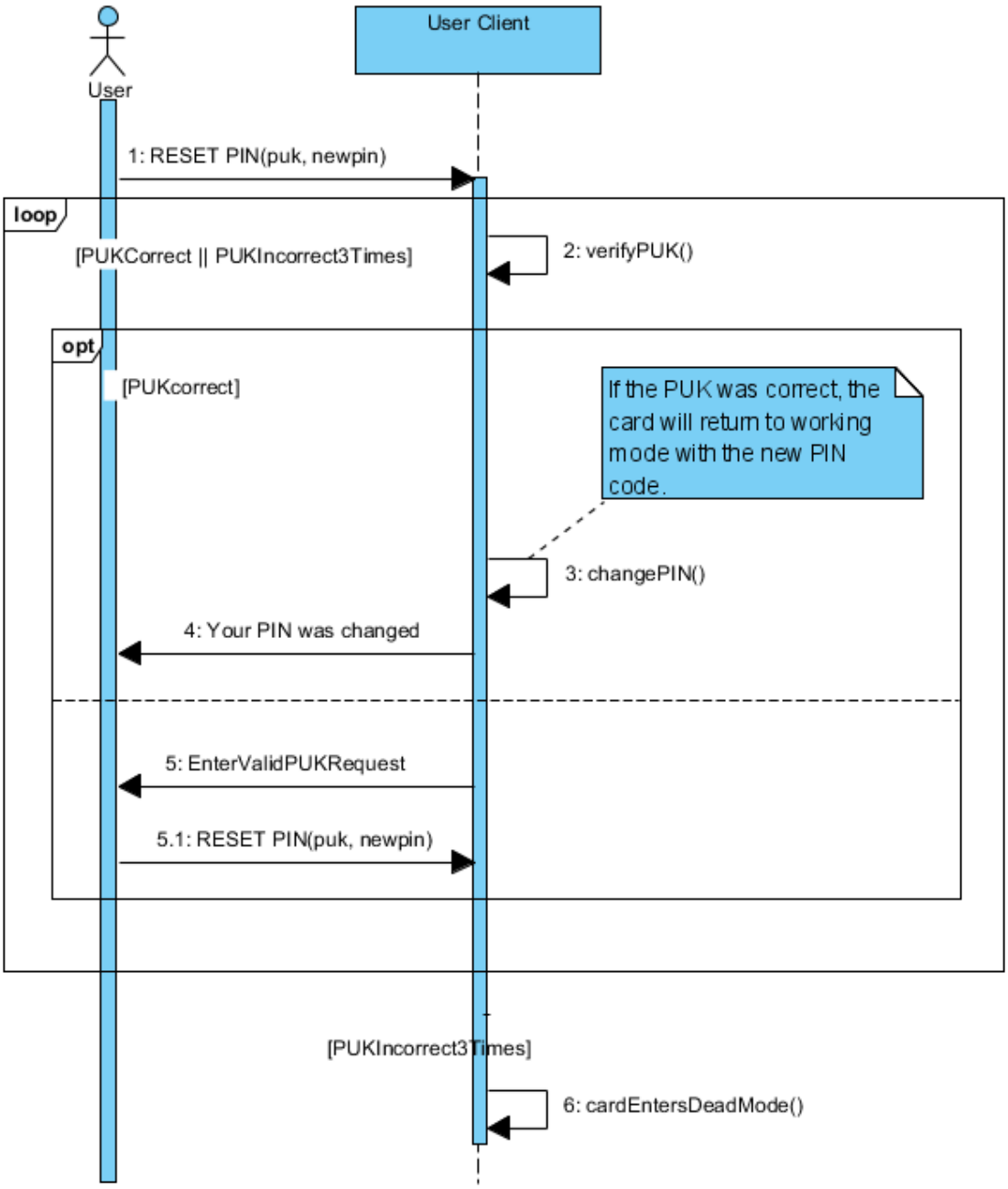

Figure 39: Unlocking the Smart Card with a PUK

9.20 reIssuance of U-Prove Tokens

Table 20 and Figure 40 present the API mapping for the use case reissuance of U-Prove Tokens. This use case describes the steps needed for a user holding a U-Prove smart card to receive a new batch of U-Prove tokens mapped to the same U-Prove credential.

Step	Explanation
1	The user application detects that the number of unused U-Prove tokens is low and sends a U-Prove reissuance request **(1).** The ABC System responds with a special policy forcing the user

	to reveal all the credentials **(1.1).**
2	The user generates a presentation token using the method *createPresentationToken(p:PresentationPolicyAlternatives)* and taking into consideration the provided PresentationPolicy **(2)**. The PresentationToken is sent to the school registration system **(2.1)**.
3	The ABC system checks if the underlying credential which generated the issuance token was not revoked and verifies the policy,
4	If the verification was successful and the credential is not revoked, the user will obtain a credential and new tokens containing the same revocation handle and attesting the same attribute values.

Table 20: reIssuance of U-Prove Tokens

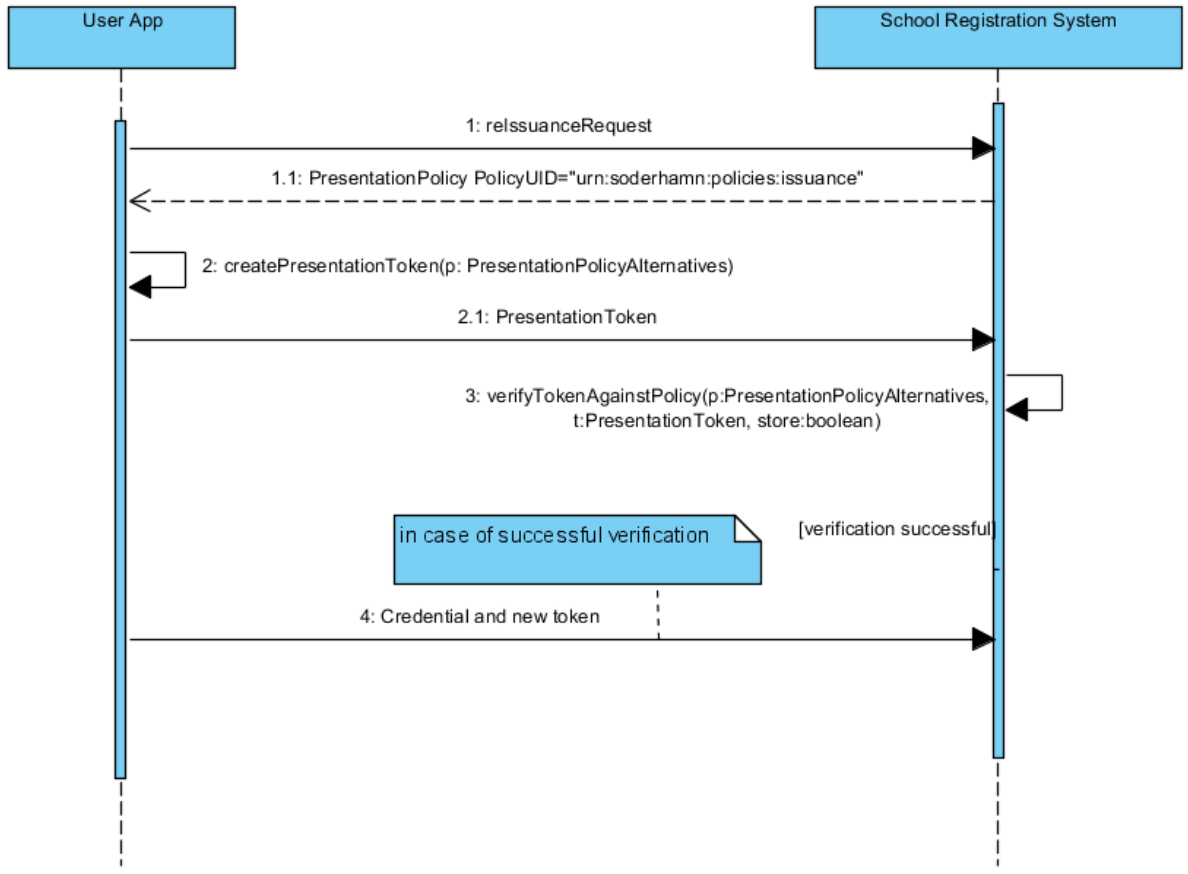

Figure 40: reissuance of U-Prove Tokens

10 Network Set up and Operation

The school network is set up with private internal addresses on the network 10.175.0.0. They all communicate with the outside / Internet via a combined firewall and NAT. The address exposed to the external world is 81.94.#.#.

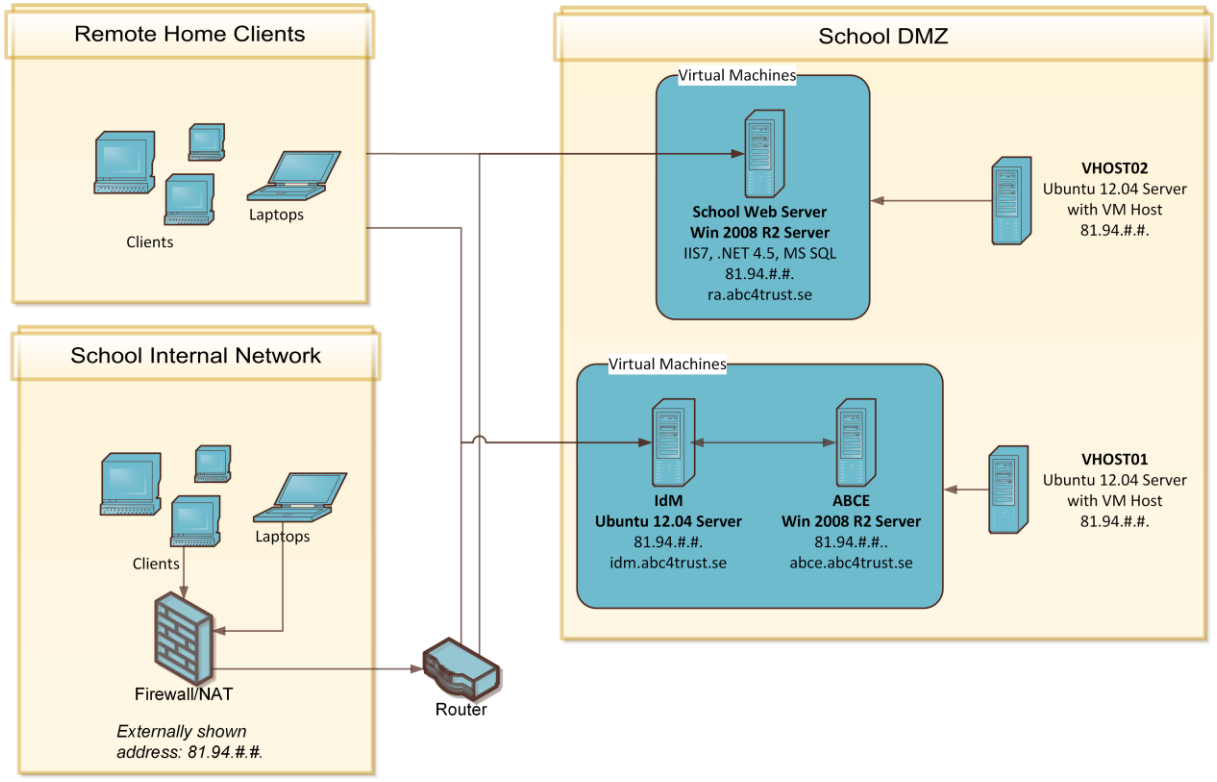

Figure 41: Network topography

Virtual server	***IP-address***
IdM server (including ABC engine)	81.94.#.#
Restricted Areas server / web server	81.94.#.#
ABC engine server (used by RA server)	81.94.#.#

Table 21: Servers used on school DMZ

Servers used on school DMZ

The servers reside on a section of the school DMZ, with the following characteristics:

Net: 81.94.#.#

Subnet mask: 255.255.255.240

Default Gateway: 81.94.#.#

Addresses allocated for use: 81.94.#.# -- 81.94.#.#
Broadcast address: 81.94.#.#

Clients access the servers from pupils' homes, as well as from the school internal network.

11 Legal Aspects

The Söderhamn pilot demonstrates a series of solutions to privacy problems inherent in existing systems typically used for the authentication of Users. Compared to other spheres of network communication (like online forums equipped with classic authentication by real name), some selected features of Privacy-ABCs on software and hardware level improve the protection of the User's personal data considerably. The reasons for such enhanced protection will be explained in section 11.1 below, while taking into account the above described details of the implementation, set-up and operation of the Privacy ABC system for the Söderhamn school pilot. A feature special to the ABC4Trust school pilot is inspection. In case it becomes necessary to identify a User at a later stage, inspection allows this while at the actual authentication the User does not need to provide her identifying information in clear text. The pure existence of this feature raised concerns of privacy experts, and we agree that the inspection functionality may trigger this impression at the first look. However, the authors of this chapter, being familiar with the technical design and realization of this functionality, take the opportunity to explain inspection and its benefits for a system deploying Privacy-ABCs. The chapter authors even see the chances in the inspection feature as an enabler for Privacy-ABCs for a variety of use cases where otherwise relying parties would be well advised to insist on full identification of Users. Thereby, inspection might bring a quantitative and qualitative improvement in data protection to many applications compared to already existing forms of User authentication. For this more in-depth explanation regarding these considerations, see section 11.2 below. As Privacy-ABCs work on the application layer, malicious service providers may nevertheless identify users by analyzing information from the underlying network levels, e.g. IP-addresses. While the anonymization on network layers is not within scope of the ABC4Trust, we provide some pointers to existing privacy-enhancing technologies (PETs) for these purposes in section 11.3.

11.1 Privacy features enabled in the school pilot

Privacy-ABCs are a new way to approach addressing privacy issues, thereby significantly improving the current state of the art in the industry. In the following, we present the most relevant privacy benefits generally provided by the deployment of attribute-based credentials:

- For many use cases, the current state of the art for authentication processes perform the desired functionality, but mostly come along with a full identification of the user. In this context, Privacy-ABCs enable a minimization of personal data being disclosed. For the exchange of opinions, an anonymous and unauthenticated forum can create a sphere for users to express themselves free of fear from identification and repression. Likewise, the right to inform oneself by access to publicly available information on websites can be made possible by anonymous access to those websites.
- Privacy-ABCs allow verifying certain attributes such as age, residence, and nationality without revealing the real information itself. They enable various functionalities for minimal data disclosure, such as the verification of the same entity acting as on a previous occasion without collecting further identifying information. For example, this can be done by assigning a cookie, or sharing a secret such as username and password. Moreover, Privacy-ABCs, as used in the Söderhamn Pilot, go further by not relying on a shared secret, but being cryptographically bound to the credential issued by the school. This way, the necessary access token cannot be easily passed on to someone else without risking impersonation – providing an additional reason for relying parties to trust this authentication without giving away any more information.
- Pseudonymous use: Privacy-ABCs may support the inspection feature, allowing the identification of a user that pseudonymously authenticated her towards the system. To this the authentication part remains anonymous. In addition, an encrypted token is provided with the identifying information (making it pseudonymous per legal definition), but the service provider does not learn the true identity yet.

The encrypted identifying information will only be sent with consent of the user. In some applications, this may be necessary due to legal obligations for the relying parties, such as for the school in the pilot. For details and possibilities to preserve a maximum of privacy even in cases where inspection is enabled, please see the dedicated section in this the following section 11.2.

- Identified use: Though being the exception case of Privacy-ABCs, it is also possible to configure a content sphere in such a way that attributes verified allow a direct linkage to a specific person, e.g. the real name, matriculation number or other unique identifier. However, this possibility of linking to the user is made transparent prior to submitting her personal data by the user client. In the Swedish pilot, this feature is used to regulate access to the file shares containing personal information made available for the pupils and their parents by the school, e.g. absence reports. In this case, a clear identification not only in line with data protection law, but is required to protect the data from unauthorized access.

These benefits are already provided the current state of the project's system implementation and set-up as well as through the operational means as defined so far. Beyond these benefits, the pilot is generally designed to fully comply with the legal data protection requirements of the European Data Protection Directive as well as with its national implementation in Sweden through the Swedish Personal Data Act:

According to Section 30 of the Swedish Personal Data Act, the processing of personal data by a third party on behalf of the data controller may occur under supervision and within the boundaries of the data controller's instructions (data assistant). This also applies if this data assistant is located in another European country. Thus, the legal framework would allow a hosting of the IdM server at NSN in Munich, Germany under strict preconditions. However, it would mean that NSN would have access to all personal data contained in clear text. Moreover, in contrast to the Patras pilot (as described e. g. in the scenario definition of this pilot in D5.1[2]), we have a different dimension regarding the personal data processed: Much more personal data are stored and processed, and personal data of minors are concerned. Since it cannot be expected of the school to set up a community interaction platform all by itself, it is naturally dependent on an external service provider. To resolve these issues and to avoid a full disclosure of all personal data, it was decided to host the IdM server in Sweden. Still, it was necessary to establish a formal data processing relationship since the IdM experts located in Munich need to be able to perform system maintenance and error troubleshooting in case it is needed. For this purpose, we set up a three-step process for these tasks. In a first step, simple telephone support occurs. If this does not aid in the debugging attempt, step two foresees the transmission of selected sanitized log files containing the needed information. Only if these two procedures are of no avail for the error treatment, a third step foresees the granting of direct system access by NSN under the control of the Swedish partner.

According to Section 22 of the Swedish Personal Data Act, the processing of personal identity number may be allowed under strict preconditions only. The Swedish civic number is widely used in Swedish public and private life, and is also used as a standard identifier in Swedish schools. Therefore, it was initially intended to use the civic number as source for the birthdate and as unique identifier. Generally, such a unique identifier used in a series of contexts – for the Swedish number basically everywhere from public entities, e-government, tax authorities, and health systems or in employment contexts – bears a high risk of cross-context profiling. The particular risk of such a nationwide unique identifier used in many different contexts is that a person may easily be profiled across all these contexts and data from a variety of different sources may be easily linked. Due to this fact, the project decided to only keep the syntax of the number and the part representing the birthdate. The other digits

[2] https://abc4trust.eu/index.php/pub/119-d5-1-scenario-definition-for-both-pilots

are replaced with a placeholder (Pilot User Number, PUN). To allow identification in the inspection cases and within the school IdM system for issuing credentials, the school retains a list of the modified numbers linking these to the identifying information for each participant. The scope of application of the PUN is limited to the specific Privacy-ABC system set up in Söderhamn and prevents such cross-context linkage.

11.2 Inspection

In the following section, we provide some considerations with regard to the inspection functionality of the Privacy-ABCs from the data protection view. For a general description of inspection from the perspective of technical realization, please refer to [CKLNPR12] and the respective sections in the foregoing chapters of this document. Moreover, for more details on the implementation of the feature in the Söderhamn pilot, please refer to the prior chapters of this document, especially to section 4.16 **Emergency Situation (Inspection)**.

11.2.1 The inspection feature

The presentation tokens used in Privacy-ABCs are fully anonymous by default – unless the verified content of the attributes allows identification of the user, e.g. as it contains the combination of name and birthdate. To enable the inspection feature, the presentation policy must clearly inform the user that inspection is enabled and usually contain a reference to the conditions under which the user may be identified (inspection grounds). The presentation token will then include the information necessary for a potential identification of the user if an inspection ground is fulfilled. This additional information is added to the token but encrypted under the public key of the inspector. The relying party, in the pilot the RA server, is then able to cryptographically verify that the presentation token could be inspected without being able to reveal the actual value. If circumstances occur that allow an inspection according to the previously published inspection grounds, the relying party may turn to the inspector requesting the identification of the user. The inspector eventually verifies that the inspection grounds are fulfilled and may decrypt the data. A denial of the identity revelation is also possible if the inspector comes to the conclusion that the claimed inspection ground is not fulfilled.

The inspection may be requested by other parties than the relying party such as criminal prosecutors, other users of the system, or third parties whose right has been infringed. These parties, however, need to contact the relying party first as this is the responsible data controller. In particular, a court order or an inquiry of competent public authority binding for the relying party may allow inspection.

A core role in the inspection process resides in the entity of the Inspector. Depending on the use case, this may be any entity preferably trusted by both involved parties. A malicious Inspector acting collusive with or forced by the relying party could decrypt all identifying information in a system and thus create linkability of the whole content to the respective users. While technical countermeasures exist to split the secret key in various ways, e.g. so that only 3 out 5 Inspectors may decrypt the information, this approach has not been chosen for the current pilot. However, the Inspector is expected to discuss the inspection grounds together with colleagues in an inspection board.

For more detailed information on the architecture used in inspection processes, please refer to [CKLNPR12].

11.2.2 Technical and organisational requirements for inspection

Besides setting up the system correctly (including all cryptographic elements), some technical and organizational measures need to be taken to ensure that the inspection process can be undergone in a privacy-respecting manner. In addition to a minimum level of transparency, a well-defined process is necessary to clearly state the legal and factual reasons allowing inspection (inspection grounds).

Transparency of inspection should be provided throughout all stages of the process, which also includes the general requirements for the hardware and software used in the pilot. During the authentication process, the policy must clearly state the demand for the inspection information to be included in the presentation token and provide the indication of the inspection grounds. The user client should also warn the users that the presentation token she is about to present would allow potential inspection. For the time of the inspection itself, it should be made known to the user, that the identity was revealed. While the data has actually been obtained directly from the user (data subject), she is not aware that the relying party has obtained the identifying information in clear-text. According to an analogous application of Art. 11 Para. 1 of the Directive 95/46/EC (Data Protection Directive), the relying party is required to provide the requested information. This includes the reasons for the inspection and a reference to the contextual information that are now linked to the identifying information as well. For example, in the Söderhamn pilot, a notification shall be set including information about which RA has been inspected. Exceptions to this rule may be permissible the exemptions and restrictions enlisted in Art. 13 of Directive 95/46/EC apply. These are mainly related to individual cases concerned with national security, defence, public security, the prevention, investigation and detection of criminal offences, etc. Consequently, the necessary notification of the user may be waived in such specific cases only.

So, informing the user without culpable delay is the standard procedure which usually should not occur later than 3 to 6 months after the incident. In case of risks for public security, the information may be delayed until after consultation of the responsible security authority.

The inspection process needs to be defined beforehand. The process is triggered by some entity requesting inspection. In a next step, the necessary information for evaluating whether the inspection grounds are fulfilled must be obtained. This process needs to be defined further in detail. Depending on the expected number of inspections, this process may be automated to the extent possible. The inspection result then needs to be communicated. In case of a rejection, the communication goes towards the entity requesting the inspection; in case of a success, the affected user needs to be informed for transparency reasons and a designated inspection receiver gets the information from the inspector. The inspection receiver is not necessarily the same person as the entity asking for inspection. E.g. in the Swedish pilot any user may ask a check of potentially offending entries in Restricted Areas, but it is not this complaining pupil that is likely to receive the uncovered information but rather the principal or the class teacher of the offender for the purpose of taking further actions. Where possible, the inspection process should foresee additional fine-grained reactions. So for example in the Söderhamn use case it will be foreseen that the School Inspection Board may simply comment on offending entries or delete those, thereby foregoing the possibility to identify the author. The inspection process should be protocolled somehow to allow later evaluation or supervision.

Finally, the inspection grounds need to clearly specify the reasons allowing an inspection. Usually, not all kinds of possible circumstances may be listed as inspection grounds since it is simply not feasible to foresee and cover all eventualities. Rather, more generic clauses may be defined as long as they are sufficiently precise to be verified against a given case. An example for such a generic clause would be to allow inspection in cases of a severe and sufficiently concrete threat to the life, liberty of physical integrity of a person. Moreover, an existing court order, or an administrative request which would force the relying party to reveal the identity may count as inspection ground, as long as they provide corresponding legal obligations to the responsible parties. More use case specific reasons should be made evident for users. In the Söderhamn pilot, references are also made to the already existing and valid school policies on discrimination and degrading treatment and the corresponding rules resulting of them.

11.2.3 Evaluation of the inspection features under data protection aspects

The inspection feature of Privacy-ABCs has been criticized inter alia by members of the ABC4Trust Reference Group – a board of external experts providing early feedback and suggestions to the project consortium. In particular, the inspection feature was seen as the introduction of a backdoor or key

escrow into this privacy preserving technology. While we understand these concerns and the criticism, seeing the ambivalent nature of this feature, we would like to elaborate why we disagree with this view. The inspection feature indeed allows the identification of users using a privacy preserving technology. However, unlike backdoors in other software, the inspection feature does not come secretly, but must be openly communicated at the time of credential presentation. This shall be done prior to the first use and authentication of the user towards the system. This transparency requirement for the user is enforced by the user client, thus software under the control. Such a user client should preferably be available under an open source code license allowing a review or third party audits of the code for potential backdoors. It is conceivable that this user client may also be configured to deny presentation policies with foreseen inspection to protect the identity of the user. But such a function would render inspectable Restricted Areas generally inaccessible for the user. However, all participants of the Swedish pilot will receive an explanation of the feature, sign a correlating consent form, and the user client provided by ABC4Trust clearly displays which information for a potential later inspection will be transferred to the RA system prior to the transmission.

So in relation to the concerns expressed, we take the opportunity to summarize the drawbacks as well as the benefits of the inspection feature in the next two subsections followed by an overall evaluation.

Aspects and specifics of inspection eventually posing a risk to privacy

- If set up incorrectly, an omnipotent single "system-administrator-inspector" could arbitrarily decrypt all inspection tokens and create linkability between all bits of information.
- The inspector is yet another entity involved with the processing of personal data.
- The inspection would not work for telecommunication providers in Europe. Inspection is not sufficient to fulfill the legal requirements of the European Data Retention Directive[3] (Directive 2006/24/EC hereinafter: DRD). The articles 3 and 5 of the Directive requires from telecommunication companies to retain a fixed set of data (calling and called phone-, IMSI- and IMEI numbers)[4]. The DRD does not foresee exceptions to this rule for cases in which the user is clearly identifiable for the telecommunication service provider by other means such as an inspectable token stored for each communication process. Here, the inspection would add just another layer of personal data to being processed unless the legislative framework is changed accordingly.
- The existence of the inspection functionality may reduce the user's trust into the whole system.

However, none of these aspects mentioned above infringes the privacy of individuals more than the alternative procedures currently deployed in most authentication systems. Such procedures nowadays rely primarily on a prior real name revelation, or the collection of IP-addresses for later linkage to an individual. In addition, most of these ostensible drawbacks of inspection may be prevented or remedied with precautions in the system set up as well as with organizational measures defining the precise inspection grounds and regulating the inspection process.

The potential issue with the required data retention for telecommunication providers is not a problem of the ABC technology as such, but results from the requirements stated in the Data Retention Directive, which focuses on aspects of the existing mobile communication technology. If the DRD had been kept technology neutral, just demanding telecommunication providers to be able to identify the

[3] Directive 2006/24/EC of the European Parliament and of the Council of 15 March 2006 on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications networks and amending Directive 2002/58/EC, online: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:105:0054:0063:EN:PDF.

[4] See Art. 5 Para. 1 lit (e) of Directive 2006/24/EC.

person responsible for one or more particular calls, this requirement may have been fulfilled with Privacy-ABCs and inspection.

Features eventually enabling privacy – compared to the current state of the art

- Inspection even may function as a privacy enabler: The inclusion of such a feature makes it possible for a variety of businesses to deploy Privacy-ABCs technology in the first place. Then, the current state of the art in most ICT systems will be significantly improved, since these almost exclusively rely on personal identification of the user before a login or service use. Such procedures so far leave users without any anonymity at all. This also may apply not only for businesses, but also for public and private entities. Privacy-ABCs with inspection functionality will thereby protect the identity of the majority of the user who comply with terms of use and legal restrictions. Only the identity of those users who violate these will be revealed.
- Inspectable tokens remain unlinkable unless all tokens have been decrypted. Thus, one incident triggering a necessary inspection provides identification for that case but allows no linkage to the same person in other usage contexts. It must be noted though that all other visits to Restricted Areas may be linked if an already inspected Alias was used to visit them. So if Users want to use an inspectable or inspected Alias for other Restricted Areas, they will get notified, which will enable them to choose a different Alias.
- In inspection cases on legitimate inspection grounds, the data controller benefits from a more complete set of issuer-verified identifying information. This is an advantage over having only IP-addresses that need to be resolved or forcing the user to disclose her real name before using any service. Thereby, not only administrative effort will be less, but also practicability will increase.
- The predefinition of the inspection procedure and the legitimate inspection grounds shape a clear framework which will not only make the system transparent and understandable for the users, but also establish an objective and controllable process in the context of the Privacy ABC system.
- Transparency and controllability are also established by the inspector; provided that an independent external party is chosen for this task and that the inspector objectively evaluates the claimed inspection grounds and assessing their fulfillment. Besides ensuring that the service provider may not identify the users without control of the Inspector as a third party the inspector may also ensure transparency about any inspection done.
- Further safeguards are still possible, e. g. by means of log files adequately documenting the inspection process and automated information of users.
- Inspection may be used to actually enable privacy preserving use cases. If an online service requires payment but handles this payment via a service provider, then the credit card details may be encrypted to the secret key of this payment provider. The online service then only needs the verification that the payment has been processed from the payment provider and does not learn the identity of the user.
- In the long run inspection may act as a privacy enabler via its transparency features. At the moment apparently an assumption exists that whenever identifying information is available it should be collected for e.g. liability purposes and likewise it is demanded to be able to identify acting persons in all types of processes. The inspection feature may be set up in a way that the inspection process is transparent and the inspector as independent party enforces this transparency. This way we may gain empirical evidence whether the use of inspection was effective against the nominated reason for using, whether inspection was “abused” for

additional purposes or whether one could just waive the requirement as it is either not used or not successful.

Evaluation of the risks

Altogether, inspection may be another privacy threat if deployed incorrectly or/and in addition to existing identification and tracking methods involving yet another entity – the inspector – with the processing of personal data. However, done correctly with appropriate safeguards, inspection may even be a privacy enabler especially for businesses so far denying the Privacy-ABCs technology completely due to concerns of legal and factual needs to reveal the identity of a user. This counts especially for all the cases in which the normal turn of business does not require a clear identification of the acting party, but one nevertheless needs the identity just for the case something goes wrong such as missing payment, liability cases, illegal content upload on websites or the like. Here, inspection may even be the reason to allow these service providers as relying parties to deploy Privacy-ABCs in the first place and deactivate other types of tracking such as logging IP-addresses for such uses. Due to this prospect, inspection may become a privacy enabler in the long run. Privacy-ABCs could provide a viable alternative for relying parties to the collection of IP-addresses or the identification of users as a precautionary measure. However, as elaborated above, this may only be truly effective if the privacy preserving requirements are mapped to all stages of the processing of the User's personal data, taking all hardware and software components of the whole Privacy ABC system into account.

In fact, if the inspection feature is used responsibly and with the foreseen safeguards, the use of Privacy-ABCs with this feature enabled is still much more privacy preserving than current practices deployed in most systems.

11.2.4 Project pilot - Söderhamn specific aspects of inspection

Given that the school is responsible towards the parents, the teachers as its staff, the public, and furthermost towards its pupils, it must be accepted that the school needs to retain some control over the services offered. This includes the community interaction platform in the ABC4Trust pilot. In particular, the municipality of Söderhamn, having the official responsibility for the school, is a public entity and thus is closely bound to the applicable Swedish public law. So in the following, any reference to the entity of "the school" refers to the Norrtullskolan legally represented by the municipality of Söderhamn.

After detailed communication with the school, taking into consideration the property of the school being a public entity and the responsibility for the pupils as well as the legality of the service, it was decided to have most Restricted Areas set up as potentially inspectable areas. Besides most Restricted Areas set up for the school being inspectable, also all Restricted Areas set up by pupils will include inspection. An exception is made for the areas of political discussions by allowing fully anonymous authentication. These discussions will be moderated by the teacher setting up the political discussion, thus providing for sufficient supervision. Inappropriate content, e.g. assaults against pupils with foreign origin or different race, may then be used as a basis for a discussion in class. Other offending content may be blocked or deleted.

To sum it up, to fulfill the requirements in relation to the organizational measures required by section 11.2.2 above, the project has foreseen the inspection functionality for a series of Restricted Areas, and established a workflow for inspection as described in the following subsections.

11.2.4.1 Inspection Grounds

The most relevant reason to implement the inspection feature is by the request of the school that has to comply with its official policy against discrimination and degrading treatment[5]. Moreover, the partners from Norrtullskolan expressed the wish to be able to reveal identities of pupils in case of emergencies, like a suicide threat etc. to react accordingly and prevent damages. The inspection process was clearly predefined to provide a transparent and controlled handling in case an inspection is requested during the trial. All participants have been made aware of the inspection process and how it works in the user manual and as part of the consent form.

The material reasons for allowing an inspection are laid down in the inspection grounds made available before participation in the pilot and each time an inspectable Restricted Area is accessed. Besides permitting an inspection for cases of severe threat to life, liberty of physical integrity of a person, an existing court order or administrative request, a series of school related inspection grounds follow from the official Norrtullskolan policy against discrimination and degrading treatment. The Norrtullskolan policy against discrimination and degrading treatment is a document binding for all attendees of the school and responsible persons (e.g. principal, educators, and parents). The policy aims at ensuring the fair and equal treatment of school attendees regardless of sex, gender identity and sexual orientation, ethnic background, religion and creed, disabilities or age. Moreover, any other kind of abusive and degrading treatment, harassment and other dangers to the safety of pupils (mostly covering cases of offensive language among pupils, but not exclusively) are stated as the legitimate case to initiate adequate countermeasures. In the context of this policy, involved parties (principal, educators, students, parents) are expected to respect the aforementioned objectives of the document and partially pose concrete obligations. Furthermore, countermeasures involve the possibility to request the aid of work team security agents, mentors and representatives of student health groups. All steps taken in an individual case at hand are bound to be documented accordingly, also by using prepared fill-in forms (such as the municipality forms to document abusive treatment and to investigate violations). Cases of mobbing and harassment must be reported by the school to the school authority, including the names of the offenders.

To align the realization of the pilot to the policy requirements given by the school, we will adapt these to predefined inspection grounds on which the inspector will be allowed to execute the identity revelation with his inspection token. These inspection grounds will be available to all participants and be part of the information sheets and consent forms to be provided.

11.2.4.2 Practical realization of inspection within the Söderhamn pilot

Inspection requests will be dealt with in a multi-stage process to provide the School Inspection Board alternative approaches in proportionate relation to the individual offense at hand. The School Inspection Board consists of the principal, the school nurse and the school counselor.

The inspection can be triggered by any user of the system. Each entry to a Restricted Area contains a button to report the content. Clicking on this button link will cause to mark the content as reported for inspection, and then an e-mail is sent to the members of the inspection board informing them about the request. In addition, the content of the respective entry or file is made available in the system for the inspectors, who must authenticate towards the RA service by proving to have the role of “inspector”. The School Inspection Board may open the content and decide about the appropriate action. Depending on the content, it may be necessary to also gain knowledge of the rest of the RAs contents

[5] The policy is available online at:
http://www.soderhamn.se/download/18.12494e5813c05809128e67/Norrtullskolans+plan+mot+diskriminering+och+kr%C3%A4nkande+behandling.pdf

to evaluate the potential offence under knowledge of the full context. Depending on how severe the reason for inspection is, these resolutions may be taken:

- **Minor offense/violation level I**
    - A notification about the inspection is set visibly in the RA concerned with a warning pointing to the Norrtullskolan policy against discrimination and degrading treatment
- **Minor offense/violation level II**
    - A notification like above is set and the offending/violating forum post or thread in the RA will be deleted/made invisible.
- **Moderately severe offense/violation**
    - An inspection request is sent to be assessed by an inspector and a notification about the inspection is left in the RA for transparency reasons.
    - If the inspection grounds are fulfilled by the case the inspector identifies the author of the offending post(s). If the inspector comes to the conclusion that the inspection grounds are not fulfilled, he denies the identification.
    - The inspection board can eventually take action itself with the possibilities given above (notification, admonition, blocking and deletion) and in addition by directly confronting the identified offender giving her/him opportunity for own statements and reasoning on the topic. E.g. ordering the offender in the principal's office.
- **Severe offense or threat to life, liberty or physical integrity**
    - An inspection request is sent to be assessed by an inspector and a notification about the inspection is left in the RA for transparency reasons.
    - If the inspection grounds are fulfilled by the case the inspector identifies the author of the offending post(s). If the inspector comes to the conclusion that the inspection grounds are not fulfilled, he denies the identification.
    - Still, the board may take action itself with the possibilities given above (notification, admonition, blocking and deletion).
    - The inspection board suggests a special treatment of the case, e.g. sending the identified offender for guidance with the counselor or other appropriate measures.
    - The School Inspection Board/school administration further conducts eventually necessary measures, like formal penalties and/or notification of the offender's legal guardians.

The inspection board selects the appropriate measures and the mark in the database will be removed. If further actions with the identified participant are necessary, these steps are taken under the usual conditions of the school, e.g. by having the offender rapport to the principal's office, or to have a talk with the school counselor.

The following picture depicts the flow of the decision process. The first two levels of interaction are marked as "delete" the other two as "inspect".

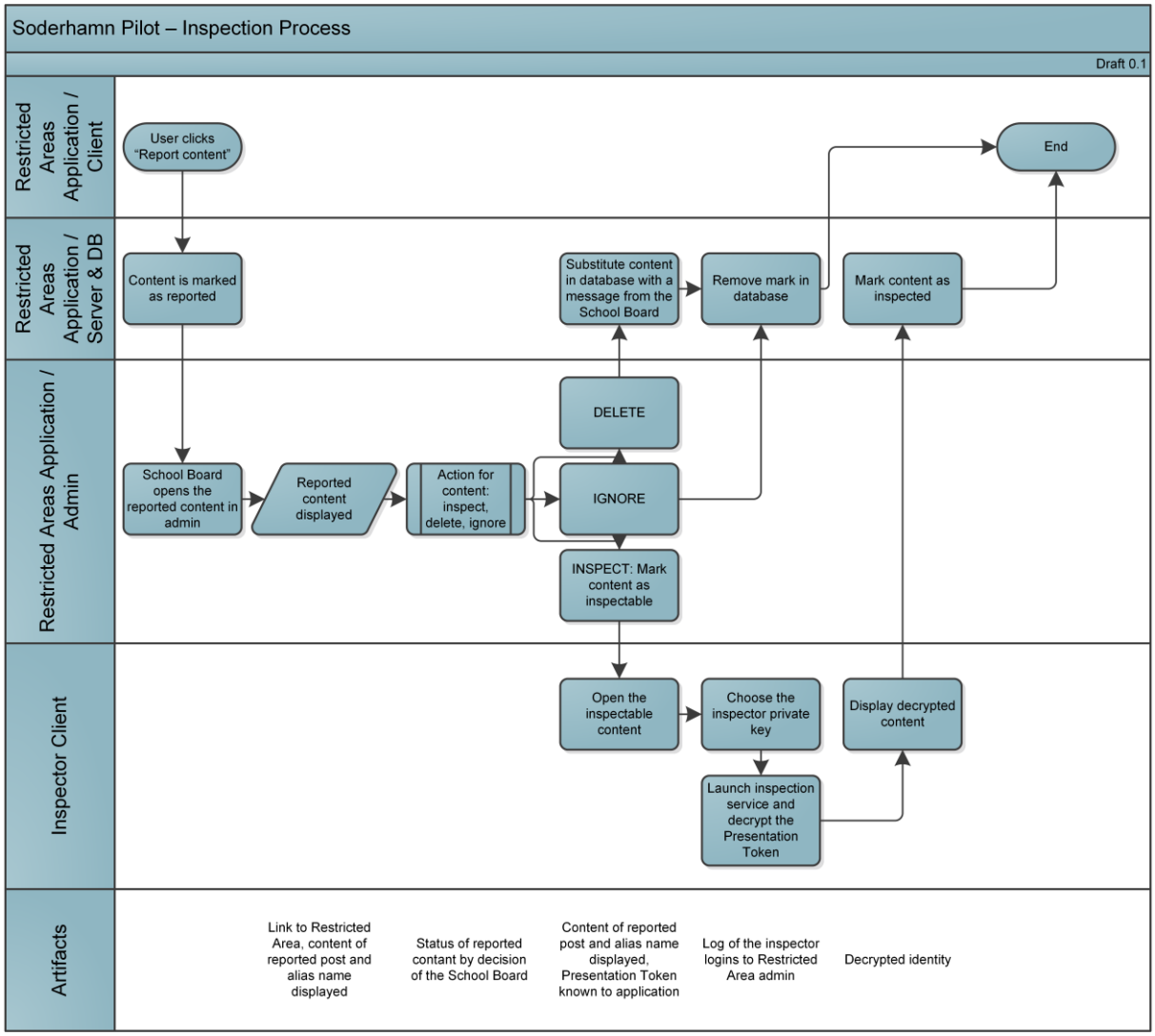

Figure 42: Inspection process

The aforementioned process might be refined during the pilot. Major adaptations, in particular any changes to the inspection grounds, will be clearly communicated to the participants.

11.3 Additional requirements to ensure anonymity

Some aspects of privacy and anonymity are not object of the ABC4Trust research project on data protection compliant authentication for online services. The pilot provides anonymity or pseudonymity for the application level only. The connection to a computer network however, involves further layers to enable the communication (cable, electromagnetic waves) still below the application level. The anonymization of communication on the layers 6 to 1 within the classical OSI model is not within the scope of the ABC4Trust project. However, the broader concept of ABC4Trust foresees and references to existing technologies providing further protection of users.

On the network and transport layer, anonymization is possible by deploying mix-services such as TOR or AN.ON to effectively hide the own IP-address form the relying party – in case of the Swedish use case, hide it from the RA system and the school registration system. Onion routing services change the IP-address by sending the users inquiries though a series of nodes. The packages of the inquiry will

then have the IP-address of the exit note contacting the destination server. Replies take the respective way back to the sender.

Using onion routing requires the installation of a respective client, which often comes bundled with a compatible browser. Using the designated browser reduces the risk of re-identifying a user according to particularities of the browser or other identifying information remaining in the normally used browser.

The use of onion routing services does not contradict European law. Quite on the contrary, Directive 2002/58/EC[6] requires Member States to encourage and foster the use of anonymous and pseudonymous data where possible, see Recital 6 of Directive 2002/58/EC. For a detailed legal analysis regarding these aspects of data protection, criminal procedures and liabilities under German law see [ULD06].

Onion routing systems that may be deployed to anonymize the communication with the school registration system and the RA system include:

- AN.ON / JonDonym:

  https://anonymous-proxy-servers.net/

- TOR

  https://www.torproject.org/

The ABC4Trust project will not provide TOR or AN.ON clients or services, but the named services may be used for free, the first one also offers premium services with notably faster online connection. It is considered to describe the possibility to the participants as part of the handbook.

[6] Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications).

12 Risk Management

In this section we present and discuss the results of applying a novel security- and privacy-aware Quantitative Threat Modeling Methodology (QTMM [QTMM12]) to the *ABC-related stages*[7] of the Söderhamn pilot, with the goals of *(i)* identifying the potential risks and, *(ii)* eliciting the adequate set of security and privacy requirements. The results presented in this chapter only contain those identified threats that could not be further mitigated in the current version of the deployed system, but that nevertheless might have a noticeable risk/impact. These results might be used to further improve the security and privacy levels of the overall pilot in the following iteration. In order to keep homogeneity with the rest of this Deliverable, the present chapter is organized according to the use cases identified for this pilot. Before presenting the results of the QTMM, the next section gives an overview of the basic concepts of this methodology.

12.1 Overview of the applied Quantitative Threat Modelling Methodology

While the general concept of "Privacy-by-Design (PbD)" is increasingly a popular one, there is considerable paucity of either rigorous or quantitative underpinnings supporting PbD. Drawing upon privacy-aware modeling techniques, this section overviews the basic concepts of a novel Quantitative Threat Modeling Methodology (QTMM) that has been proposed in the context of ABC4Trust. The QTMM was applied to the Söderhamn pilot to draw objective conclusions about the different privacy and security related attacks that might affect it. Interested readers are referred to [QTMM12] for further details about the QTMM.

The QTMM comprises the five steps shown in Figure 43, where an informal use case description is the entry point to elicit a set of optimal security and privacy requirements. The rest of this section presents, with the level of detail required by this document, each one of the QTMM's steps.

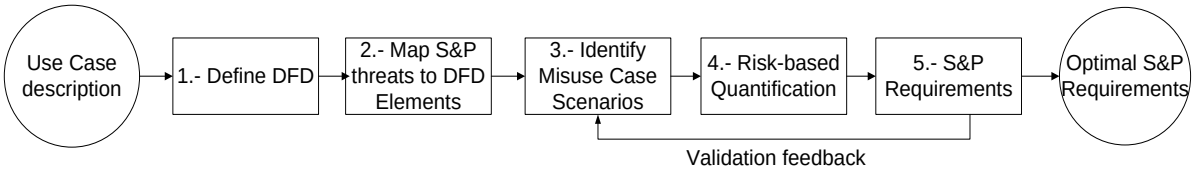

Figure 43: Overview of the applied QTMM

12.1.1 Step 1: Define Data Flow Diagrams (DFD)

In general, DFDs [DFD93] can aid the formal decomposition of a system such that the elements of Entities, Trust Boundaries, Data Flows, Data Sources and Processes are clearly identified. A DFD is a graphical representation of data flows, data stores, and relationships between data sources and destinations (entry and exit points). The guiding principle for DFDs is that an application or a system can be decomposed into subsystems, and subsystems can be decomposed into recursive lower-level subsystems. This iterative process makes DFDs useful for decomposing applications to analyze the associated threats at varied levels of detail. Typically, in a DFD only the abstract/high-level views of the interactions among the different components of a system are represented (mostly at the service-level), rather than the messages exchanged via the underlying protocol.

The DFD used for the threat analysis of the Söderhamn pilot is shown in the following Figure 44 Please notice that the level of detail shown in this DFD corresponds to the performed analysis for example, *we have not analysed the S&P threats related with the internals of the ABCE subsystem*.

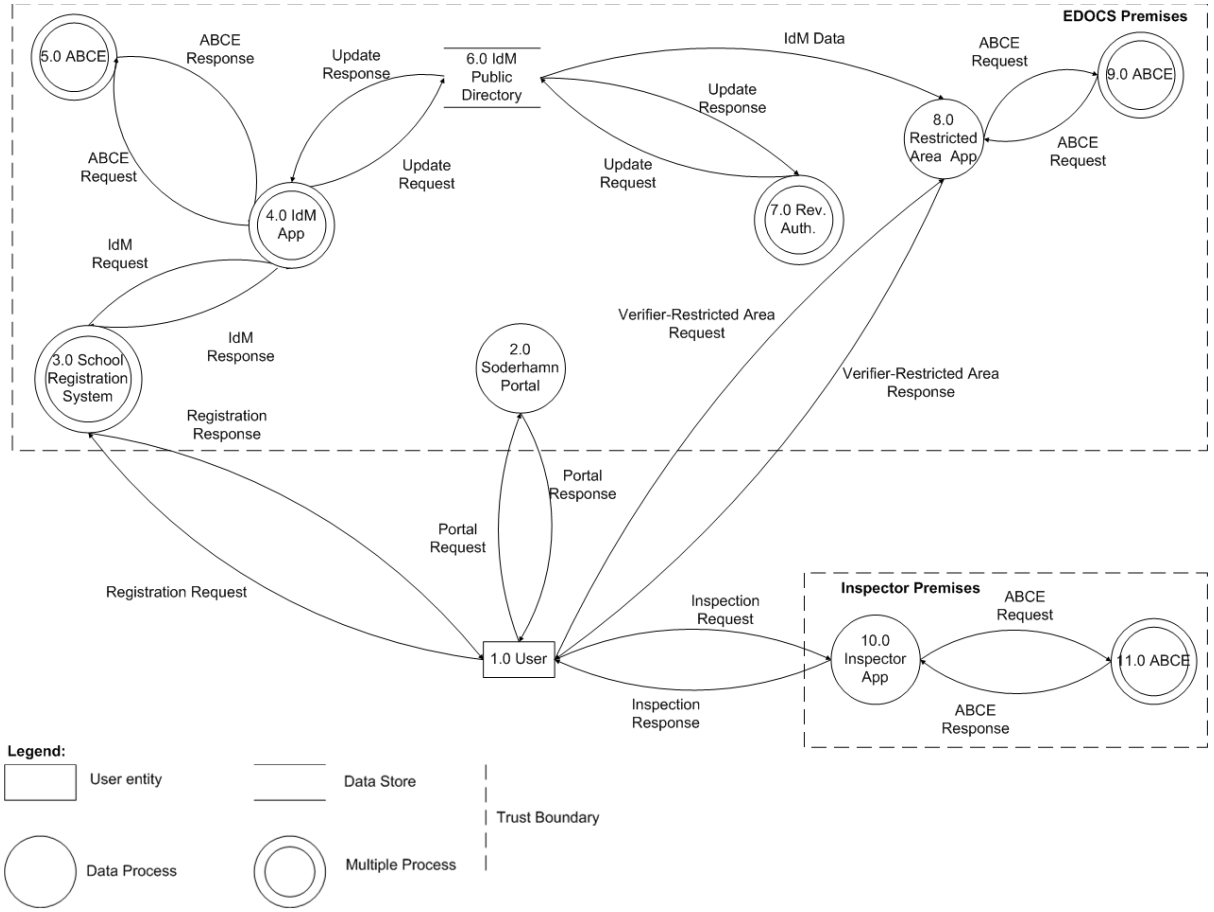

Figure 44: DFD Diagram.

12.1.2 Step 2: Map S&P threats to DFD elements

During this stage, the set of newly created DFDs are "mapped" to the threats associated with each one of the security and privacy properties to be taken into account for the QTMM. The specific properties taken into account comprise the "traditional" security ones (i.e., Confidentiality, Integrity and Availability) plus the ones proposed in [PPG12] for privacy (i.e., Unlinkability, Transparency and Intervenability). In the applied QTMM, the set of S&P threats being considered (along with their corresponding DFD mapping is shown in Table 22.). The rationale behind the proposed mapping can be found in [QTMM12].

S&P property	S&P threat	Threat explanation	DF	DS	P	E
Confidentiality	Information Disclosure	These threats expose personal information to individuals who are not supposed to have access to it.	X	X	X	
Integrity	Tampering	Tampering is the unauthorized modification of data, for example as it flows over a network between two	X	X	X	

		computers				
Availability	Denial of Service	Denial of service is the process of making a system or application unavailable.	X	X	X	
Unlinkability	Linkability	For two or more items of interest (IOIs, e.g., subjects, messages, actions, etc.) allows an attacker to sufficiently distinguish whether these IOIs are related or not within the system.	X	X	X	X
Transparency	Unawareness	Indicates that one or more parties are unaware of the conditions related with privacy-relevant data processing.				X
Intervenability	Avoidance/Non-intervenability	Indicates that the parties related with the privacy-relevant data processing, are unable to intervene.			X	X

Table 22: Mapping S&P properties to DFD elements (DF= Data Flow. DS= Data Source, P= Process, E= Entity)

For the Söderhamn pilot analysis, the S&P to DFD mapping is shown in Table 22. From this table it is worth to highlight the following assumptions about each one of the threats under analysis:

- A total of 20 potential threats were analyzed.
- No threats were analyzed wrt. ABCE-specific components (including Data Flows and Processes). This analysis is taking part in WP2 and WP3.
- Disclosure threats do not apply to the "IdM Public Directory" and, only applies to Data Flow containing personal data.
- Denial of Service threats are *system-wide* and, can compromise any of the DFD elements (in particular Data Stores and Processes).

		S&P threats					
DFD component	**Threat Target**	**I**	**T**	**D**	**L**	**U**	**A**
Data Store							
	IdM Public Directory (6.0)	X	6	0*	X		
Data Flow							
	User - Inspector App (1.0 – 10.0)	X	X	X	13		
	Restricted Area App – ABC System (8.0 - 9.0)	X	X	X	X		
	Restricted Area App – User (1.0 - 8.0)	1	7	X	14		
	School Registration System - IdM Portal (3.0 - 4.0)	X	X	X	X		
	IdM App – ABC System (4.0 - 5.0)	X	X	X	X		
	IdM App – School Registration System (4.0 - 3.0)	X	X	X	X		
	School Registration System – User (1.0 – 3.0)	2	8	X	X		

	Söderhamn Portal - User (1.0 - 2.0)	X	X	X	X		
	Public Directory – Restricted Area App (6.0 – 8.0)	X	X	X	X		
	Public Directory – IdM App (6.0 - 4.0)	X	X	X	X		
	Public Directory – Söderhamn Portal (6.0 - 2.0)	X	X	X	X		
Process							
	Inspector Application (10.0)	3	X	0*	15		X
	Restricted Area App (8.0)	4	9	0*	16		X
	Revocation Authority (7.0)	X	10	0*	X		X
	IdM Application (4.0)	X	X	0*	X		18
	School Registration System (3.0)	5	11	0*	X		X
	Söderhamn Portal (2.0)	X	12	0*	X		X
Entity							
	User (1.0)				X	17	19

Table 23: Mapping S&P threats to the Söderhamn DFD

12.1.3 Step 3: Identify Misuse Case Scenarios

It is a common practice in threat modelling to document a threat analysis as "misuse case scenarios", where details are specified about generic threats that can be posed as specific threat instances in a real system. Misuse cases can be documented in different ways, but in general the analysis should provide information about the attacker model, a summary of the attack, a set of assumptions/preconditions to launch the attack and the relevant attack tree [AT99]. Our QTMM uses attack trees in order to quantify threats and prioritize the elicitation of S&P mechanisms (this will be shown in the next section).

For the Söderhamn pilot *our first analysis identified a total of 20 misuse cases*, corresponding to the mapping shown in Chapter 9 API Mapping. The attacker model we assume for all the misuse cases are *skilled insiders or skilled outsiders* (i.e., are able to read/write data from/to DFD elements) *with a finite amount of resources* (e.g., they will not be able to break the underlying cryptography in a finite period of time). Furthermore, the first iteration of the QTMM considers that *no S&P mechanisms have been implemented*. For example, Data Flows are unencrypted; no privacy-ABC technology is integrated into the pilot, etc. Thanks to this basic assumption, it is possible to perform an iterative elicitation process, just as described in the next section. The final set of misuse cases (after eliciting the final S&P requirements) will be summarized in the following.

12.1.4 Step 4: Risk-based Quantification

The essence of our proposed QTMM is an approach to quantify the S&P risks associated with each element of an attack tree (threats and attacks). Our methodology contributes with the techniques to provide an overall quantitative score for the whole threat based on its individual attacks. This score can be used by designers and decision makers to e.g., prioritize the identified threats and begin the elicitation of the required mitigation mechanisms. Using a "score card" approach, the QTMM proposes to quantify two basic parameters on the attack trees:

- Impact: the damage potential (including affected users) of the threat/attack. Score card: (1) Insignificant,(2) Minor, (3) Moderate, (4) Major, and (5) Catastrophic.

- Risk: the likelihood of a threat/attack. Score card*: (1) Rare, (2) Unlikely, (3) Possible, (4) Likely, and (5) Certain.*

Due to space limitations, we will not explain in further details how to actually aggregate the former two parameters on the attack trees. However, interested readers are referred to [QTMM12] for more details.

12.1.5 Step 5: S&P Requirements

The final stage in traditional threat analyses is the elicitation of specific mitigation techniques. By the contrary, our QTMM approach is in fact an iterative process where elicited S&P requirements (mitigation techniques) are used to refine both the misuse cases and corresponding attack trees (quantified impacts and risks) during each iteration. This refinement process finalizes until a set of S&P requirements/mechanisms allows managing risks optimally (i.e., either to avoid, optimize or accept the resulting risk).

The rest of this chapter presents the final results (*only the final residual/accepted risks*) after iterating two times the QTMM in the Söderhamn pilot.

12.2 ABC System Setup

During the Setup stage of the Söderhamn pilot, the threats that continue having the highest impact/risk are those related with the tamper of either the setup parameters, the ABCE API or the Credential Specification (cf. Table 24).

Threat Name	Threat Class*	Comments	Impact **	Risk ***	Proposed mitigation
Unauthorized modification of Credential Specification.	T	The attacker is able to modify the Credential Specification stored in the IdM Public Directory.	4	3	Protect the integrity/ authenticity of the Credential Specification e.g., via hashes or digital signatures.
Unauthorized modification of Issuer parameters	T	The attacker can modify the Issuer's initialization parameters.	5	1	Protect the interface by allowing access only from trusted computers.
Unauthorized modification of the ABCE API.	T	The attacker can distribute a malicious/customized copy of the ABCE API.	5	2	Protect the integrity/authenticity of the ABCE API binary via e.g., hashes or digital signatures.

Table 24: QTMM results: ABC System Setup

* *(I)nformation Disclosure, (T)ampering, (D)enial of Service, (L)inkability, (U)nawareness, (A)voidance*

** *(1) Insignificant, (2) Minor, (3) Moderate, (4) Major, (5) Catastrophic*

*** *(1) Rare, (2) Unlikely, (3) Possible, (4) Likely, (5) Certain*

12.3 Smart Card Registration using One-Time-Password

For this process, information disclosure attacks constitute a relevant threat if there is no security countermeasures implemented at the Registration System (e.g., firewalls, intruder detectors, etc.).

Table 25 shows in further detail the results of our threat analysis (only threats with accepted/non-mitigated risks are shown).

Threat Name	Threat Class*	Comments	Impact **	Risk ***	Proposed mitigation
Denial of Service (DoS) against the School Registration System.	D	The attacker performs a DoS attack against the School Registration System.	3	1	As the risk is low and the impact moderate, no mitigation other than to provide direct access to the servers (for admins) if needed.
Information Disclosure by impersonating the Pupil via the login Presentation Token.	I	The attacker is able to disclose the user's information contained in the School Registration System's database, by impersonating her via login Presentation Token (replay attack).	4	1	The login Presentation Token uses a nonce (embedded into the corresponding Presentation Policy).
Information Disclosure by impersonating the user via the knowledge of the One-Time-Password.	I	The attacker is able to access the user's information contained in the School Registration System's database, by impersonating her if he gets access to her OTP before she uses it.	4	1	The OTP is distributed personally to the users (pupils/guardians/personnel).
Information Disclosure on School Registration System's Presentation Policy.	I	The attacker can modify the Presentation Policy used by the School Registration System for the registration stage, in order to disclose an unauthorized set of user's attributes.	4	3	Due diligence[8] related with the security of the School Registration System. Protect the integrity/authenticity of the Presentation Policy via e.g., hashes or digital signatures.

Table 25: QTMM results: Smart Card Registration using One-Time-Password

* *(I)nformation Disclosure, (T)ampering, (D)enial of Service, (L)inkability, (U)nawareness, (A)voidance*

** *(1) Insignificant, (2) Minor, (3) Moderate, (4) Major, (5) Catastrophic*

*** *(1) Rare, (2) Unlikely, (3) Possible, (4) Likely, (5) Certain*

12.4 Subsequent Logins to the School Registration System

In this use case information disclosure attacks continue to constitute the main potential threats. The following Table 26 shows the results of our threat analysis.

Threat Name	Threat Class*	Comments	Impact **	Risk ***	Proposed mitigation

[8] Best effort measures.

Denial of Service (DoS) against the School Registration System.	D	The attacker performs a DoS attack against the School Registration System.	3	1	As the risk is low and the impact moderate, no mitigation other than to provide direct access to the servers (for admins) if needed.
Information Disclosure by impersonating the Pupil via the login Presentation Token.	I	The attacker is able to disclose the user's information contained in the School Registration System's database, by impersonating her via login Presentation Token (replay attack).	4	1	The login Presentation Token uses a nonce (embedded into the corresponding Presentation Policy).
Information Disclosure on School Registration System's data flow.	I	The attacker obtains user's personal data by capturing the traffic between the user and the School Registration System.	4	1	Protect the traffic by using encrypted lines (https, vpn tunnel).
Information Disclosure on School Registration System's Presentation Policy.	I	The attacker can modify the Presentation Policy used by the School Registration System for the registration stage, in order to disclose an unauthorized set of user's attributes.	4	3	Due diligence related with the security of the School Registration System. Protect the integrity/authenticity of the Presentation Policy via e.g., hashes or digital signatures.
Information Disclosure on School Registration System's user database.	I	The attacker compromises the School Registration System's user database/logs to access the personal data stored there.	4	1	Protect the interface by allowing access only from trusted computers.

Table 26: QTMM results: Subsequent Logins to the School Registration System

* *(I)nformation Disclosure, (T)ampering, (D)enial of Service, (L)inkability, (U)nawareness, (A)voidance*

** *(1) Insignificant, (2) Minor, (3) Moderate, (4) Major, (5) Catastrophic*

*** *(1) Rare, (2) Unlikely, (3) Possible, (4) Likely, (5) Certain*

12.5 Obtaining the School Credential

For this use case, the set of accepted threats related with these issuance processes are similar to the ones shown in Table 27, with the exception of new impersonation threats that might appear.

Threat Name	**Threat Class***	**Comments**	**Impact****	**Risk*****	**Proposed mitigation**
Denial of Service	D	The attacker performs a	3	1	As the risk is low and the

(DoS) against the School Registration System.		DoS attack against the School Registration System.			impact moderate, no mitigation other than to provide direct access to the servers (for admins) if needed.
Information Disclosure due to compromised smart card.	I	A compromised smartcard/PIN might result in attacker disclosing the personal data of the corresponding user.	4	2	Inform the users of the risks, and that they need to protect their smart card and pin.
Information Disclosure of former user's personal data.	I	The attacker compromises historic logs files/databases of the School Registration System, to obtain personal data of former users.	3	1	Protect the interface by allowing access only from trusted computers.
Information Disclosure on the School Registration System's data flow.	I	The attacker obtains user's personal data by capturing the traffic between the user and the School Registration System (issuance protocol).	4	1	Protect the traffic by using encrypted lines (https, vpn tunnel).
Information Disclosure on School Registration System's Issuance Policy.	I	The attacker can modify the Issuance Policy used by the School Registration System, in order to disclose an unauthorized set of user's attributes.	4	1	Protect the interface by allowing access only from trusted computers.
Information Disclosure on School Registration System's user database.	I	The attacker compromises the School Registration System's user database/logs to access the personal data stored there.	4	1	Protect the interface by allowing access only from trusted computers.

Table 27: QTMM results: Obtaining the School Credential

* *(I)nformation Disclosure, (T)ampering, (D)enial of Service, (L)inkability, (U)nawareness, (A)voidance*

** *(1) Insignificant, (2) Minor, (3) Moderate, (4) Major, (5) Catastrophic*

*** *(1) Rare, (2) Unlikely, (3) Possible, (4) Likely, (5) Certain*

12.6 Other School Credentials

Since 'Other school credentials' are issued analogously to the school credential, the threat analysis related with this use case is similar to the use case "Obtaining the School Credential". The results are depicted in Table 27.

12.7 Viewing User's Data

For the use case "Viewing User's Data", which describes the steps needed for viewing the User's profile, Table 28 shows the results of our threat analysis.

Threat Name	**Threat Class***	**Comments**	**Impact****	**Risk*****	**Proposed mitigation**
Denial of Service (DoS) against the School Registration System.	D	The attacker performs a DoS attack against the School Registration System.	3	1	As the risk is low and the impact moderate, no mitigation other than to provide direct access to the servers (for admins) if needed.
Information Disclosure by impersonating the user via the login Presentation Token.	I	The attacker is able to disclose the user's information contained in the School Registration System's database, by impersonating her via login Presentation Token (replay attack).	4	1	The login Presentation Token uses a nonce (embedded into the corresponding Presentation Policy).

Table 28: QTMM results: Viewing User's Data

* *(I)nformation Disclosure, (T)ampering, (D)enial of Service, (L)inkability, (U)nawareness, (A)voidance*

** *(1) Insignificant, (2) Minor, (3) Moderate, (4) Major, (5) Catastrophic*

*** *(1) Rare, (2) Unlikely, (3) Possible, (4) Likely, (5) Certain*

12.8 Login to Restricted Area Application

The threat analysis of the use case "Login to Restricted Area Application" has shown that, similar to the previous use case "Viewing User's Data", information disclosure attacks are possible to launch if there are no minimum security countermeasures implemented within the School Registration System. Table 29 depicts the results of our threat analysis.

Threat Name	**Thre**	**Comments**	**Impac**	**Risk***	**Proposed mitigation**

	at Class*		**t****	**	
Denial of Service (DoS) against the School Registration System.	D	The attacker performs a DoS attack against the School Registration System.	3	1	As the risk is low and the impact moderate, no mitigation other than to provide direct access to the servers (for admins) if needed.
Information Disclosure by impersonating the user via the login Presentation Token.	I	The attacker is able to disclose the user's information contained in the School Registration System's database, by impersonating her via login Presentation Token (replay attack).	4	1	The login Presentation Token uses a nonce (embedded into the corresponding Presentation Policy).

Table 29: QTMM results: Login to Restricted Area Application

Login to Restricted Area Application

* *(I)nformation Disclosure, (T)ampering, (D)enial of Service, (L)inkability, (U)nawareness, (A)voidance*

** *(1) Insignificant, (2) Minor, (3) Moderate, (4) Major, (5) Catastrophic*

*** *(1) Rare, (2) Unlikely, (3) Possible, (4) Likely, (5) Certain*

12.9 Choose or Create Alias

This use case describes the steps how a user can choose a previously created alias or create a new alias. The threat analysis of the use case "Choose or Create Alias" shows that it is similar to the previous use case "Login to Restricted Area Application". The results of the threat analysis are depicted in

12.10 Instantiating a Restricted Area

Administrators and in some cases users are able to create new Restricted Areas. The threat analysis of this use case is similar to the previous use case "Login to Restricted Area Application". The results of the threat analysis are depicted in

12.11 Access to a Restricted Area

This use case describes how the user can access the Restricted Area after check of Access Policies. The threat analysis of this use case is similar to the previous use case "Login to Restricted Area Application". The results of the threat analysis are depicted in

12.12 Counselling

The threat analysis of the use cases "Counselling" has shown that, similar to the previous use case "Viewing User's Data", information disclosure attacks are possible to launch if there are no minimum security countermeasures implemented within the School Registration System. depicts the results of our threat analysis.

Threat Name	Threat Class*	Comments	Impact**	Risk***	Proposed mitigation
Denial of Service (DoS) against the School Registration System.	D	The attacker performs a DoS attack against the School Registration System.	3	1	As the risk is low and the impact moderate, no mitigation other than to provide direct access to the servers (for admins) if needed.
Information Disclosure by impersonating the user via the login Presentation Token.	I	The attacker is able to disclose the user's information contained in the School Registration System's database, by impersonating her via login Presentation Token (replay attack).	4	1	The login Presentation Token uses a nonce (embedded into the corresponding Presentation Policy).

Table 30 QTMM results: Counselling

* *(I)nformation Disclosure, (T)ampering, (D)enial of Service, (L)inkability, (U)nawareness, (A)voidance*

** *(1) Insignificant, (2) Minor, (3) Moderate, (4) Major, (5) Catastrophic*

*** *(1) Rare, (2) Unlikely, (3) Possible, (4) Likely, (5) Certain*

12.13 Restricted Chat Room

Chat rooms are Restricted Areas with chat functionality activated. This use-case describes two use-cases which are almost identical, one where a person chats with other users by entering a chat room (group), another where a person that wants to chat with another person in a private chat room (one-to-one).

As the technical details are pretty much identical to use case "Access to a Restricted Area", the result of the threat analysis is the same as depicted in Table 29.

12.14 Political Discussions

Political discussions are anonymous chats, with no option to reveal the identities. This use-case describes two main situations, one where a person chats with other users by entering a chat room (group), another where a person that wants to chat with another person in a private chat room (one-to-one).

The technical details are almost identical to use case “Access to a Restricted Area”, but without the option to inspect the token. Meaning, the policies will be similar as there, but without the block asking to disclose the pilot user number (PUN) to the inspector. The result of the threat analysis is the same as depicted in Table 29.

12.15 Sharing Documents

The school is producing many documents (exam results, grades, individual development plans etc.) that need to be shared with or distributed to the pupil’s and their parents. A personal restricted area exists for every user in the system.

Technically, this use case is a special case of a restricted area where users are completely identified. The result of the threat analysis is the same as depicted in Table 29.

12.16 Revocation

User's credentials can be made invalid when the user is forced to leave the school, he or his guardians revoke consent to participate in the pilot, an attribute value is no longer valid, or a user reports his smart card as lost or stolen. The following Table 31 depicts the threat analysis of the use case “Revocation”.

Threat Name	**Threat Class***	**Comments**	**Impact****	**Risk*****	**Proposed mitigation**
Denial of Service (DoS) against the Revocation authority.	D	The attacker performs a DoS attack against the Revocation authority.	3	1	As the risk is low and the impact moderate, no mitigation other than to provide direct access to the servers (for admins) if needed.
Denial of Service (DoS) by unauthorized revocation of valid credentials.	D	An attacker within the Revocation authority performs a revocation of valid user credentials.	3	1	Monitor access to the Revocation authority.
Denial of Service (DoS) against the user.	D	The attacker impersonates the revocation requestor to trigger the revocation of valid user credentials.	3	1	Monitor access to the Revocation authority.
Denial of Service (DoS) against the user.	D	The attacker impersonates the System administrator to trigger the revocation of valid user credentials.	3	1	Monitor access to the Revocation authority.

Table 31: QTMM results: Revocation

* *(I)nformation Disclosure, (T)ampering, (D)enial of Service, (L)inkability, (U)nawareness, (A)voidance*

** *(1) Insignificant, (2) Minor, (3) Moderate, (4) Major, (5) Catastrophic*

*** *(1) Rare, (2) Unlikely, (3) Possible, (4) Likely, (5) Certain*

12.17 Emergency Situation

The use case "Emergency situation" occurs if there is a critical situation that causes the School Registration System to start the inspection procedure for a particular communication. Table 32 shows in further detail the results of our threat analysis.

Threat Name	**Threat Class***	**Comments**	**Impact****	**Risk*****	**Proposed mitigation**
Denial of Service (DoS) against the Inspector.	D	The attacker performs a DoS attack against the Inspector.	3	1	As the Inspector uses a local ABCE on his/her computer, there is no real risk of a DoS.
Information Disclosure by collusion among the Inspector and the School inspection board.	I	Colluding internal attackers within the School inspection board and the Inspector are able to disclose the user's data and communication contents.	5	1	Carefully select the School inspection board. Inform the Inspector of the risks.
Information Disclosure by stealing the Inspector's private key.	I	The attacker can steal the Inspector's private key from the memory of the Inspector's computer. Thus, the attacker will be able to decrypt all messages sent to the Inspector.	5	1	As the risk is very low, no security countermeasures other than informing the inspectors about the risk and protecting the Inspector's computer.

Table 32: QTMM results: Emergency Situation

* *(I)nformation Disclosure, (T)ampering, (D)enial of Service, (L)inkability, (U)nawareness, (A)voidance*

** *(1) Insignificant, (2) Minor, (3) Moderate, (4) Major, (5) Catastrophic*

*** *(1) Rare, (2) Unlikely, (3) Possible, (4) Likely, (5) Certain*

12.18 Viewing/Deleting Credentials

The use case "Viewing/Deleting Credentials" can be split into 2 parts: the threat modeling of the viewing part is identical to the use case "Viewing User's Data" described above. With respect to the deleting part tampering attacks are possible to launch if there are no minimum security countermeasures implemented within the School Registration System. Table 33 depicts the results of the performed threat analysis.

Threat Name	**Threat Class***	**Comments**	**Impact****	**Risk*****	**Proposed mitigation**
Denial of Service (DoS) against the School Registration System.	D	The attacker performs a DoS attack against the School Registration System.	3	1	As the risk is low and the impact moderate, no mitigation other than to provide direct access to the servers (for admins) if needed.
Information Disclosure by impersonating the user via the login Presentation Token.	I	The attacker is able to disclose the user's information contained in the School Registration System's database, by impersonating her via login Presentation Token (replay attack).	4	1	The login Presentation Token uses a nonce (embedded into the corresponding Presentation Policy).
Tampering user credentials by allowing access to a non-valid user.	T	The attacker uses a set of non-valid credentials to view the user credentials and delete them.	3	1	Privacy-ABCs revocation feature is implemented.

Table 33: QTMM results: Viewing/Deleting Credentials

* *(I)nformation Disclosure, (T)ampering, (D)enial of Service, (L)inkability, (U)nawareness, (A)voidance*

** *(1) Insignificant, (2) Minor, (3) Moderate, (4) Major, (5) Catastrophic*

*** *(1) Rare, (2) Unlikely, (3) Possible, (4) Likely, (5) Certain*

12.19 Changing PIN/Unlocking Smart Card with a PUK

These stages do not implement any ABC feature, therefore were not considered in the threat modeling.

12.20 reIssuance of U-Prove Tokens

For this use case, the set of threats related with the process of reissuance of U-Prove tokens are depicted in the following Table 34.

Threat Name	**Threat Class***	**Comments**	**Impact****	**Risk*****	**Proposed mitigation**
Denial of Service (DoS) against the School Registration System.	D	The attacker performs a DoS attack against the School Registration System.	3	1	As the risk is low and the impact moderate, no mitigation other than to provide admins with direct access to the servers if needed.
Information Disclosure by impersonating the user via the login Presentation Token.	I	The attacker is able to disclose the user's information contained in the School Registration System's database, by impersonating her via login Presentation Token.	4	1	The login Presentation Token uses a nonce (embedded into the corresponding Presentation Policy).

Table 34: QTMM results: reIssuance of U-Prove Tokens

* *(I)nformation Disclosure, (T)ampering, (D)enial of Service, (L)inkability, (U)nawareness, (A)voidance*

** *(1) Insignificant, (2) Minor, (3) Moderate, (4) Major, (5) Catastrophic*

*** *(1) Rare, (2) Unlikely, (3) Possible, (4) Likely, (5) Certain*

13 Conclusion

In this deliverable D6.2, the details on the implementation, set-up and operation of the Privacy ABC system for the Söderhamn school pilot of the ABC4Trust project were provided. In the above chapters of this document, the deployment of the system architecture components, the pilot key scenarios, corresponding API mappings, the related and relevant legal aspects as well as actions for risk mitigation were described. In doing so, an emphasis was laid onto the privacy aspects of the school deployment, whereas a Privacy ABC system is provided for minors. For the conduction of this document, the use case descriptions and pilot requirements, as presented in the prior deliverables D5.1, D5.2, and 6.1, were taken into account. Moreover, the first version of the ABC4Trust reference implementation of Privacy-ABCs provided by WP4 formed the basis for the creation of this deliverable. Hence, the security and data protection requirements affecting the hardware and software being used for the pilots were recognized and integrated insofar as groundwork for the further work in ABC4Trust. During the following project runtime, the progress of the pilots will be closely observed and reviewed to enable a later integration of results into the ongoing development. Thereby, the practical findings of the pilots will serve to improve the performance of the Privacy ABC system as whole as well as the general requirements-compliance of the used hardware and software.

14 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 we will have the following attributes: *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. 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

Alias or nickname

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. An Alias is like a nickname chosen by the user not generated cryptographically.

Certified pseudonym

A verifiable pseudonym based on a user 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 User (i.e., to the same user secret) as the pseudonym. A pseudonym is not an alias.

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 we have the following credentials: *credSchool, credSubject, credChild, credGuardian and credRole.*

Credential specification

A data artifact specifying the list of attribute types that are encoded in a credential.

Device binding

An optional credential feature whereby the credential is bound to a strong secret embedded in a dedicated hardware device so that any presentation token involving the credential requires the presence of the device.

IdM

The Identity Management System (IdM or IdM System) is a database where all user data (attributes) needed to issue credentials are saved. In the Swedish School Pilot the IdM acts as the Issuer.

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 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, device binding, inspection, and revocation. The presentation token consists of the presentation token description, containing a technology-agnostic description of the revealed information, and the presentation token evidence, containing opaque technology-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 pseudonym she can use, or to which a new pseudonym should be associated. A single User (with a single user secret) can generate multiple verifiable or certified pseudonyms for the same scope string, but can only generate a single scope-exclusive pseudonym.

Restricted Area System (RA)

The restricted Area System is the school web application that contains all the functionality for chat, wall, documents uploading, counseling and political discussions. The restricted Area System is also an administration 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. The Revocation Authority can be an Issuer, a Relying Party, or an independent entity. 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 (PUN)

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.

PUN

See *Pilot User Number*.

Scope

See *pseudonym scope.*

Scope-exclusive pseudonym

A certified pseudonym that is guaranteed to be cryptographically unique per scope string and per user secret. Meaning, from a single user-bound credential, only a single scope-exclusive pseudonym can be generated for the same scope string.

Token

These are a building block of the underlying technology (U-Prove), that might create the need for re-fetching a credential. The ABC Engine will do that automatically, but it requires to be online to do so.

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.

User agent

The software entity that represents the human User and manages her credentials.

User binding

An optional credential feature whereby the credential is bound to an underlying user secret. By requiring multiple credentials to be bound to the same user secret, one can prevent Users from “pooling” their credentials.

User secret

A piece of secret information known to a User (either a strong random secret or a human-memorizable password or PIN code) underlying one or more issued credentials or pseudonyms. A presentation token involving a pseudonym or a user-bound credential implicitly proves knowledge of the underlying user secret.

Verifiable pseudonym

A public identifier derived from a user secret allowing a User to voluntarily link different presentation tokens created by her or to re-authenticate under a previously established pseudonym by proving knowledge of the user secret. Multiple unlinkable pseudonyms can be derived from the same user 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 a User Client 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.

15 Acronyms

ABCs
: Attribute Based Credentials

Privacy-ABCs
: Privacy Attribute Based Credentials (privacy ABCs)

ABCE
: ABC Engine

CA
: Certificate Authority

CE
: Crypto Engine

DFD
: Data Flow Diagrams

DRD
: Data Retention Directive

GUI
: Graphical User Interface

HTTP
: Hypertext Transfer Protocol

HTTPS
: HyperText Transfer Protocol Secure (HTTP secured by TLS or SSL)

HQAA
: Hellenic Quality Assurance Agency

ID
: Identifier

Idemix
: IBM Identity Mixer

IdM
: Identity Management System

ISP
: Internet Service Provider

LDAP
: Lightweight Directory Access Protocol

NFC
: Near Field Communication

ORM
: Object Relational Mapping

OSI
: Open Systems Interconnection

PbD
: Privacy-by-Design

PC
: Personal Computer

PET
: Privacy Enhancing Technologies

PIN
: Personal Identification Number

PUK
: Personal Unblocking Key

RA
: Restricted Area

RP
: Relying Party

RFP
: Remote Framebuffer Protocol (VNC)

QTMM
: Quantitative Threat Modeling Methodology

SC
: Smart Card

SCI
: Smart Card Interface

SSL
: Secure Sockets Layer

STS
: Secure Token Service

TTP
: Trusted Third Party

TLS
: Transport Layer Security

URI
: Uniform Resource Identifier

URL
: Uniform Resource Locator

WP
: Work Package

XML
: eXtensible Markup Language

16 Bibliography

[AT99] Schneier B., "Attack trees," Dr Dobb's, vol. 24, no. 12, 1999. [Online]. Available: http://www.schneier.com/paper-attacktrees-ddj-ft.html

[BDP12] Thomas Baignères, Cécile Delerablée and Pascal Paillier, Programming Privacy-ABCs on theABC4Trust Lite v1.0 Smart Card (To be published)

[BGOZ12] Souheil Bcheri, Norbert Goetze, Monika Orski, Harald Zwingelberg, D6.1 Application Description for the school deployment, version 1.1, ABC4Trust deliverable, 2012, https://abc4trust.eu/download/ ABC4Trust-D6.1-Application-Description-School.pdf

[CKLNPR12] Jan Camenisch, Ioannis Krontiris, Anja Lehmann, Gregory Neven, Christian Paquin, Kai Rannenberg, H2.1 - ABC4Trust Architecture for Developers, ABC4Trust report, 2012, online: https://abc4trust.eu/index.php/pub/149-h2-1.

[DFD93] P. Bruza and T. van der Weide , "The semantics of data flow diagrams," in Proc. of the International Conference on Management of Data. McGraw-Hill, 1993, pp. 66-78.

[GNU GPL] Official Ubuntu Installation Guide 10.04LTS Appendix F GNU General Public License see https://help.ubuntu.com/10.04/installation-guide/i386/appendix-gpl.html

[IRI2012] H. Guldage and J. Dam Nielsen, D4.1 Initial Reference Implementation, Version1, 2012.

[PPG12] H. Zwingelberg and M. Hansen, "Privacy Protection Goals and their implications for eID systems," in Proc. of the IFIP International Summer School, 2011.

[QTMM12] J. Luna, I. Krontiris and N. Suri, "Privacy-by-Design Based on Quantitative Threat Modeling" In Proc. of the IEEE International Conference on Risks and Security of Internet and Systems. 2012

[ULD06] Unabhängiges Landeszentrum für Datenschutz, "Anonymität und Recht Zur Zulässigkeit des Anonymisierungsdienstes AN.ON", 2006, online: https://www.datenschutzzentrum.de/projekte/anon/20070316-rechtliche-grundlagen.htm