D5.2 Amendment
Building Blocks of ABC Technology

Joerg Abendroth, Souheil Bcheri, Ioannis Krontiris, Vasiliki Liagkou, Ahmad Sabouri,
Eva Schlehahn, Fatbardh Veseli, Harald Zwingelberg

Editors:	*Joerg Abendroth*
Reviewers	*Hans Guldager(MCL), Ioannis Stamatiou (CTI)*
Identifier:	*D5.2A*
Type:	*Amendment*
Version:	*1.00*
Date:	*30/09/2013*
Status:	*Draft*
Class:	*Public*

Abstract

This amendment describes in more detail the high level architecture and the building blocks of a typical Privacy-ABC application. The proposed building blocks are abstractions from several examined scenarios, including the two pilots, with emphasis on generality of applicability. These scenarios (eID, Bank as ID Service Provider, eParticipation and Verifier Privacy Protection) are described with respect to their benefits from employing Privacy-ABCs, the involved actors and the instantiation of the generic building blocks. The technical details underlying the building blocks as well as the corresponding APIs can be found in D2.1 [CKLN11]. Contrary (or, rather, complementary) to this technical deliverable, the main objective of this amendment is to provide insights in Privacy-ABCs technologies to software architects and regulators of privacy critical ecosystems in order to assist them in designing, settting up and operating their own applications using these technologies.

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

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

List of Contributors

Chapter	Author(s)
Executive Summary	Joerg Abendroth (NSN)
1. Preface	Joerg Abendroth (NSN), Harald Zwingelberg (ULD)
2. Introduction	Joerg Abendroth (NSN), Eva Schlehahn (ULD), Harald Zwingelberg (ULD)
3. High Level Architecture and its ABC4Trust Building Blocks	Joerg Abendroth (NSN), Souheil Bcheri (EDOCS), Vasiliki Liagkou (CTI), Eva Schlehahn (ULD), Fatbardh Veseli (GUF), Harald Zwingelberg (ULD)
4. Some Typical Privacy ABC Scenarios	Joerg Abendroth (NSN), Ioannis Krontiris (GUF), Ahmad Sabouri (GUF), Eva Schlehahn (ULD), Harald Zwingelberg (ULD)
5. Conclusion	Joerg Abendroth (NSN)

Executive Summary

This amendment describes in more detail the high level architecture and the building blocks of a typical Privacy-ABC application. The proposed building blocks are abstractions from several examined scenarios, including the two pilots, with emphasis on generality of applicability. These scenarios, beyond the two pilots that are described in depth in the corresponding project deliverables, are the following:

- eID
- Bank as ID Service Provider
- eParticipation
- Verifier Privacy Protection by means of Trusted Third Party

These scenarios are described with respect to their benefits from employing Privacy-ABCs, the involved actors and the instantiation of the generic building blocks. The technical details underlying the building blocks as well as the corresponding APIs can be found in D2.1 [CKLN11]. Contrary (or, rather, complementary) to this technical deliverable, the main objective of this amendment is to provide insights in Privacy-ABCs technologies to software architects and regulators of privacy critical ecosystems in order to assist them in designing, setting-up and operating their own applications using these technologies. Within this context, also also organisational, legal and set-up aspects are covered such as for instance the order in which the system parameters should be generated and exchanged between system components, the legal aspects that are important and the buildings blocks that should best specific application scenarios.

The document starts with an introduction to the assumptions (legal and technical) underlying the Privacy-ABC technologies, as well as the actors and the high level building blocks. Then after a summary of the integration and setup aspects, the scenarios beyond the two pilots, are introduced. Each scenario is presented along with its specific building block instantiation and the issues that require attention. These issues are different for each scenario and it is well possible that the readers discover that the issues that beset their own application scenarios have many similarities with the discussed issues. In the final conclusions section, we summarize our discussion and point to issues for further investigation.

Table of Contents

Index of Figures

Index of Tables

1 Preface

This document is an amendment to D5.2 to discuss in more detail the high level architecture and associated building blocks of a generic Privacy-ABCs based application. It would be beneficial if the reader were familiar with the general ABC4Trust project objectives as well as the state of the art in IDM (Identity Management) applications as described, for example, in Chapter 1 of Deliverable D5.2. Knowledge of the genera context of the pilots would also be beneficial, as described in Chapter 2 of Deliverable D5.2 as well as, in more detail, Deliverables D6.1 and D7.1. No technical knowledge is required for following the arguments in this amendment although knowledge of some details, as given in Sections 2.1.2 and 2.2.2 of Deliverable D5.2 would also be beneficial.

1.1 Other available ABC4Trust Deliverables[1]

While this document is intended to provide a high level overview of how to compose and set up a Privacy-ABC enabled system, it naturally does not cover all necessary aspects and details. This would, moreover, contradict the purpose of this document and also repeat what has already been written in other deliverables and reports.

With respect to readership, a reader with technical knowledge in cryptography and security applications may also wish to go through Deliverable D2.1 while a business owner wishing to start up, for instance, his privacy respecting discussion forum would be more interested in Deliverables D6.1 and D6.2 related to the Söderhamn pilot of the ABC4Trust project). There are no dependencies in these deliverables, other than the fact that D5.1, D5.2 and D5.2a, and this document form a logical sequence, content-wise, and should be best read in this order.

The reader is referred to the description of the ABC4Trust deliverables and publications below in order to locate the material best suited to her needs:

D2.1 [CKLN11] this deliverable focuses on the (low level) aspects of the architecture of attribute based credential technologies. It is planned to be revised into an updated version called H2.1 (heartbeat document). Many technical details are included in this deliverable, such as API specifications, example presentation policies, credential formats etc.

D4.1[2] is the initial reference implementation. The published software packages contain the sources of the crypto libraries, the token and claims handlers, as well as a summary of the reference implementation documentation. Once published, it will be available on the ABC4Trust website. For already existing releases or updates in relation to D4.1 please refer to the already existing Github repository (see reference at the end of this list).

D5.1 [BGL12] In Deliverable D5.1 we discussed the tasks of the individual entities and components in applications and mapped them to technical and legal roles. Legal counseling Companies planning to use Privacy-ABCs might benefit from this background information for understanding the considerations, specifically those concerning data protection, that lead to the legal documents prepared

[1] All ABC4Trust project deliverables may be obtained from: https://abc4trust.eu/index.php/pub

[2] This deliverable is the source code on the Gihub repository: https://github.com/p2abcengine/p2abcengine/wiki

for the pilots. For the latter please refer to the respective D6.x and D7.x documents referred to in the paragraphs below.

D6.1 [BGOZ12] and **D7.1** [ALPRSSSZ12] provide an in depth description of the details of the two pilots. The interested system architect can find there additional, instructive information on how to realize similar scenarios as the ones on which the pilots were based. If, however, the reader finds differences between these documents and this amendment, it should be borne in mind those documents have been written early in the project, before the realization of the pilots, while this amendment is being written later, incorporating newer insights and different views to previous considerations.

D6.2 [ABDG12] This document describes the considerations that lead to the current set up of the ABC4Trust school pilot. The Chapter titled "4.1 ABC System Setup" describes the different functionalities of the pilot and their relation to the Privacy-ABC architecture. This chapter should be of interest to stakeholders planning to set up an ABC-enabled system. Furthermore, Chapter 9 on the API Mapping and Chapter 11 on legal considerations with a focus on data protection issues that may arise in connection with the Inspection feature can provide insights that cannot possibly be covered in this high level amendment document.

D7.2 [DGGL12] This description of the first round of the Patras pilot, which was based on scenarios related to the operation of a privacy respecting course evaluation system for university students, should be helpful to stakeholders targeting a system that does not require advanced functionalities such as, for instance, Inspection but only the basic Privacy-ABCs capabilities. Of particular interest may be the Appendixes of the document, which contain a user manual explaining the use of the pilot system in an easy to understand manner, complemented with screenshots from the pilot subsystems, guiding users step by step to the use of the pilot functionalities. Furthermore, in the deliverable there is also the user consent form which the pilot participants signed and which may be of interest to stakeholders from the legal sector.

The Standard "ISO/IEC 29101-Information technology - Security techniques - Privacy architecture framework", to be published soon, provides a high level overview of privacy technologies. The concepts of Privacy-ABCs were included in the standard (explained in more detail in Annex C of that standard).

A number of core components for authentication with Privacy-ABC systems and the relevant documentation for the reference implementation modulesare already available on Github. The available code includes components for supporting all the basic Privacy-ABC entities: Verifiers, Issuers, Revocation Authorities, Inspectors, and Users. For the latest publicly available code base and details on implementation please refer to: https://github.com/p2abcengine/p2abcengine/wiki

1.2 Structure of the document

This document, being an amendment to D5.2, does not aim at being fully independent on its own. However, we have taken care of, first, describing the assumptions, then the actors and finally the high level building blocks that will help a new reader approach the topic, on a high level in order to be able to read the more technical descriptions of Privacy-ABCs and the systems based on them. The reader is, also, advised to consult Chapters 1 and 2 from D5.2 as they include further motivating introductory text to this amendment document.

In Chapter 3, Section 3.3 is included to facilitate the reader to understand the issues involved in the adoption of the Privacy-ABCs technology and the planning of the integration of the different Ecosystem components. It, also, includes the experiences that we have collected throughout our collaboration with the ABC4Trust partners. The section’s content can, however, can be skipped if the reader is more interested in understanding the technology, without planning to adopt or develop a Privacy-ABCs system in the near future.

Chapter 4 starts with a general description of a generic Privacy-ABCs Ecosystem and Section 3.2 describes the high level composition of such an Ecosystem, including all optional elements. Then in the four typical scenarios described in Sections 4.1, 4.2, 4.3 and 4.4 the generic Ecosystem and the generic high-level building blocks are mapped to the particular features of the described scenarios. Then the motivation of using Privacy-ABCs and the resulting benefits are discussed, as well as the issues that may arise from their deployment and how can be handled. The reader should note that while each scenario points at one specific privacy issue for solution based on Privacy-ABCs, the scenario will most likely involve, in real life, a combination of such issues. Moreover, these issues will, most probably, be manageable, simultaneously, using Privacy-ABCs or some other supporting security technology. However, we chose to focus on a single issue for clarity of exposition.

Chapter 5 discusses our conclusions, also summarizing the different scenarios discussed in the corresponding sections. The details on the High Level Building Blocks, as presented in Section 3.2, are only touched upon in the conclusions section. The interested reader should, better, consult this Section for finding the relevant details.

Finally, much as Section 3.3 provides details on integration, planning and set-up of Privacy-ABCs systems, Appendix A presents examples of XML files for the description of credentials and policies. These examples do not eliminate the need to consult the technical content of Deliverable D2.1 or the example code modules in the Github, but they can be used as quick reminders of the XML file structure as well as a starting point of discussion in integration planning sessions among the stakeholders of a privacy-ABC domain.

2 Introduction

2.1 Scenarios requiring privacy respecting technologies

Several real life scenarios, not related to the Internet and ICT services, are almost by default "anonymous", in nature, and thus privacy respecting. For instance, one can walk into a shop to buy clothes and the shop assistant will estimate the clothes' size without the customer having to reveal his name or, say, her history of weight loss. Or, as another example, during a traditional election process, a voter is first verified with respect to voting eligibility and, then, casts secretly her vote. After the votes have been casted, it is not possible the voters' preferences. It is even impossible for the organizer of the election to know whether the voters exchanged the sealed ballots before putting them in the ballot box. Furthermore, in another domain, a driver has her birth date (personal data) on his driving license. Consequently, age verification at a night club entrance can be, reliably, done using the information in the driver's licence, since it has been issued (and, this, confirmed) by a trusted party (e.g. the Ministry of Transport). Yet, a police officer checking the driving license later cannot know when and for which night clubs this driving license was used to gain entry. In other words, the issuer of the driving license cannot know towards which services (night club entrance, in this case) a signed certificate (driver's licence) has been used.

On the other hand, in the uncontrollable and global Internet world, the need for authentication and identification is greater than in the customary real world scenarios stated above since it is easier for a person to create several fake eIdentities, dispersed over distant services, and use them for illegal purposes. This kind of liberal eIdentity creation can even be automated giving rise to several privacy and security issues. It is apparent that a mechanism is needed that allows users to prove certain properties of their identities, without revealing more than it is necessary however.

If we compare the online-clothing show with the one in the street, we notice that for an online-shop it is beneficial to know the size of the customer, this size does not need to be verified information. Moreover the online-shop does not need additional personal information, although they might be part of users' personal information. While collecting all kinds of information about customers might be cumbersome for a shop on the street, it is common practice in online shops. Thus a privacy preserving technology is needed that provides data minimization and makes collecting hard similar hard than it is at the shop on the street (e.g. a customer that just browses the catalogue does not need to be recognizable).

2.2 Assumptions

In what follows, some basic assumptions are stated to provide the reader with a more thorough understanding of the Privacy ABCs deployment framework envisaged in this document. These assumptions are presented from a legal as well as a technical perspective.

2.2.1 Legal (L) Assumptions

2.2.1.1 L1 European data protection framework applicable

The first assumption is that the European legal data protection framework is applicable. Taking a viewpoint from the European Data Protection Directive 95/46/EC only, any other legal frameworks are out of scope for being analysed within the ABC4trust project. While throughout the project runtime, the national data protection law of the countries in which the pilots are conducted are taken into consideration, the Directive 95/46/EC sets the minimum level of protection of personal data in Europe. The challenge is to build a Privacy-ABC system which fulfils the requirements of the

Directive and still be flexible enough to allow adaptations to even more specific legal requirements on a national level. Nevertheless, when setting up such a Privacy-ABC system, a common precondition for the processing of personal data is always the existence of a valid legal support, such as a specific law or informed consent on the users' side. In this context, it is assumed that changes due to the upcoming Data Protection Regulation on European level will not waive this requirement.

2.2.1.2 L2 No material change in the definition of personal data

It is assumed that the definition of personal data as manifested in the Directive 95/46/EC will also be integrated into the upcoming Data Protection Regulation in such a way that all data linkable to a specific person will be considered as personal. This shall be assumed regardless of the data controller himself holding the necessary information to establish such a link. An example is the dynamic IP addressing for Service Providers due to the decision of the European Court of Justice of November 12th 2011, assessing IP addresses as personal data.

2.2.1.3 L3 Encrypted personal data remains personal data

While PETs (Privacy Enhancement Technologies) pseudonymisation, but to a degree that allows linking to the person under special circumstances, the personal data remains personal in the sense of the European Data Protection Directive 95/46/EC. Within the Privacy ABC building blocks described in this document, this is especially relevant for Inspection, which caters for pseudonymity, but not full anonymity. Thus, the processing of pseudonymous data must still fulfil specific requirements regarding data security and data protection. However, these requirements may be less strict than they are for data which are directly linkable to the individual or contains clear text information, making the setup of processing systems for pseudonymous data potentially easier.

2.2.1.4 L4 Data protection in the eIDAS regulation

It is assumed that the upcoming European Regulation on eIdentification, eAuthentication, eSignatures and Trusted Services (eIDAS)[3] will enforce privacy also on eIDs, fostering another application field for Privacy-ABCs.

2.2.1.5 L5 Data minimisation remains a key principle of the European data protection framework

It is assumed that the principle of purpose limitation and data minimization, in the sense of the Directive 95/46/EC, will remain a central requirement in the revision of the European legal data protection framework. Thus, Relying Parties (RPs) need to consider which data is absolutely necessary, for the purpose in hand, to process. In this respect, we also refer the reader to Chapter 4, where under subsection 4.2, the legal prerequisites that are specific for Service Providers are explained.

2.2.2 Technical (T) Assumptions

2.2.2.1 T1: Users do not reveal additional information

Individuals using Privacy-ABC technologies are assured that no attributes contained in their credentials will be revealed to a verifier, beyond those necessary to satisfy the policy of the verifier. However we assume that the users do not reveal, on their own, additional attributes using side channels. For example, we assume that users do not type their real name in an online chat room, if their intention is, really, to remain pseudonymous. Additional side channels, such as analysis of user

[3] See the draft of the European Commission and the corresponding legislation documents: http://ec.europa.eu/prelex/detail_dossier_real.cfm?CL=en&DosId=201689.

interactions (e.g. the user always visits three specific pages after checking the status of another) are also out of scope of the Privacy-ABC technology.

2.2.2.2 T2: Data minimization principle respected by the verifier

Privacy-ABCs are used in conjunction with presentation policies defined by the verifier. That is, the Privacy-ABCs are built to satisfy a specific policy, without revealing any information beyond the requested. Therefore, we assume that this policy requires only the absolutely minimum information to be revealed by the user, which is required by a service provider, in accordance with the data minimization principle.

2.2.2.3 T3: Transport Layer Security

Data that the service provider transfers to the users will sometimes include information that allows inferring attributes about the user (e.g. a message originating from a chat room only for boys). An attacker which is able to eavesdrop in the transaction between the user and service provider can learn these attributes. We assume that the service provider secures the connection between the user and the service provider, e.g. by means of TLS.

2.2.2.4 T4: Transport Layer Privacy

Providing unlinkability between two statements of the same user is an important property of the Privacy-ABC technology. A malicious service provider may attempt to link the two transactions, where the statements were made, by means of comparing the network address of the statements' origin, i.e. the IP addresses from which the two statements originated. We assume that users take care of transport layer privacy on their own using mechanisms and tools such as such as TOR [TOR].

2.2.2.5 T5: Cryptographic Strength holds

The Privacy-ABCs deploy RSA-based cryptography, blind signatures and zero knowledge proofs. For Privacy-ABCs to provide the properties required by the scenarios described in later sections of this document, it is assumed that no efficient attacks on these primitives are known.

2.2.2.6T6: Organization process matches technology

A system that, under certain circumstances, allows the lifting of the anonymity of a user, also requires a carefully designed process that supervises this process. The lifting of anonymity is called Inspection. The Privacy-ABC technology cannot, alone, guarantee the protection of the individual throughout the process of inspection. The design of the administrative and organizational processes has to match the technology and this can, also, be seen as an technical assumption.

3 A generic High Level Architecture and ABC4Trust Building Blocks

Ecosystems in the Internet rely on information exchange. A large portion of this exchange concerns personal information about users. Based on such personal information, important business and financial decisions (e.g. pop-up advertisement, e-mail offers etc.) are made.

Today's IDM-based Ecosystems usually give out information about users without giving full control to them over their own personal data. While different technologies were proposed to correct this situation (see. Chapter 1 of Deliverable D5.2 or D2.1), ABC4Trust proposes an approach based on strong cryptographic primitives that allow users to have full control over the disclosure of their personal information. The architectures of the adopted, by the project, cryptographic systems of IBM and Microsoft were analysed and a unifying architecture was created. Deliverable D2.1 describes this architecture from the perspective of the cryptographer. In this chapter, we will study the perspective of an application developer. Such a reader may be a privacy solutions architect, business owner or legislation regulator wishing to facilitate privacy Ecosystem development. Chapter 3 of D5.2 contains more material on the administrative and strategic aspects of such a development. However it fails to provide a thorough technological overview, a task which will be undertaken in this section.

3.1 Description of Actors

Just like any other identity management system, Privacy-ABCs systems typically have a number of mandatory actors as well as some optional ones, depending on the specific features of the employed Privacy-ABCs. The User is a mandatory actor that can be any entity that uses a Privacy ABC system. The Verifier (also called Relying Party or Service Provider) as well as the Issuer (also called Identity Service Provider) are, also, mandatory entities. However, some additional actor roles are foreseen for Privacy ABC systems, which are optional. In particular, these are the Revocation Service and the Inspector. Figure 1 shows the different actors, as they are later used in the description of the high level building blocks. The reader will recognize the actors as the domains in the high level building block description in Section 3.2.

We differentiate between *active* and *passive* actors. Active actors initiate a process by an action or wish for action. Passive actors react upon a request of an active actor, but don't initiate a chain of actions themselves. Users and Inspector are seen as active actors. Users, generally, wish to login to a verifier and, thus, they must obtain, beforehand, appropriate credentials that satisfy the verifier's policies. For users to be able to get credentials, an Issuer has to take some actions in advance: to define, verify credentials, as well as provide the user with an account to retrieve the credentials. For simplicity, we consider this setting up of user accounts as a process initiated by the user, wishing to login to a verifier. Thus we view only the user and the inspector as actors that may initiate a process (both are noted with light blue colour).

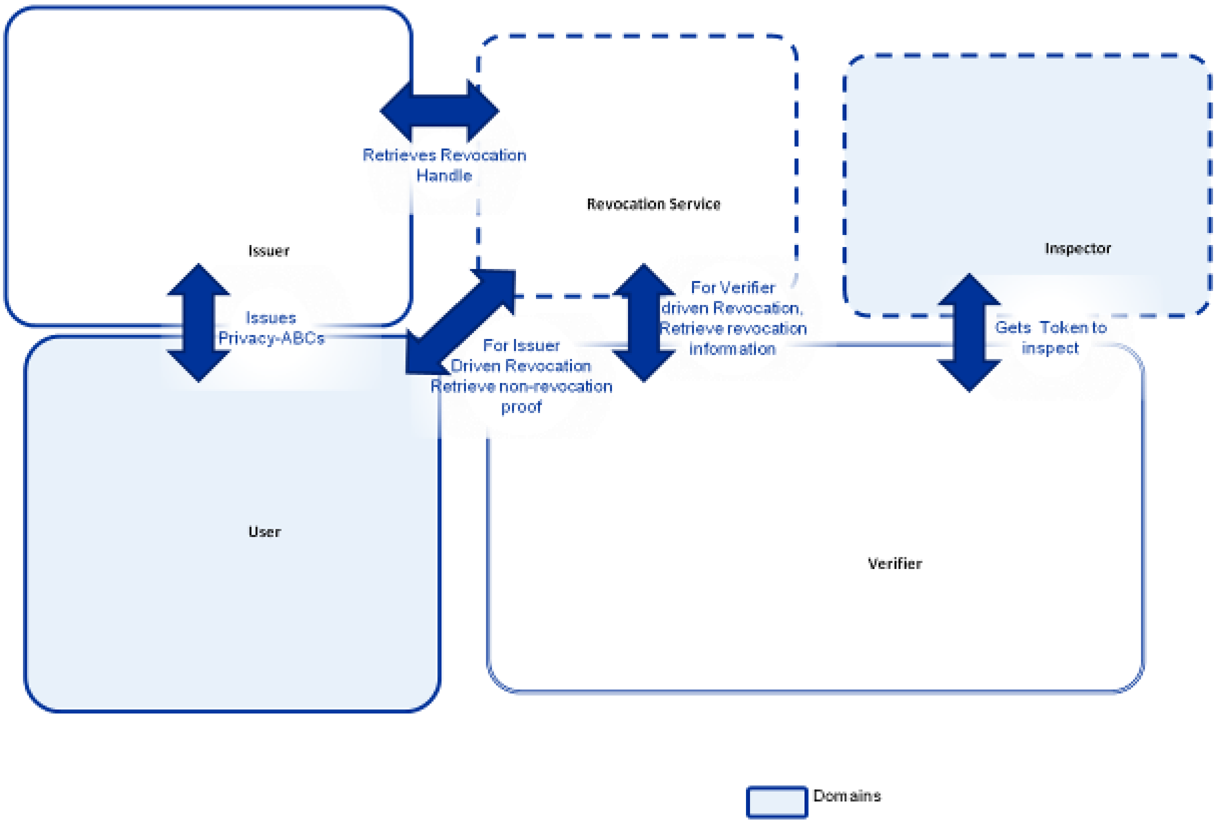

Retrieves Revocation Handle
Revocation Service
Issuer
Inspector
For Verifier driven Revocation, Retrieve revocation information
Issues Privacy-ABCs
For Issuer Driven Revocation Retrieve non-revocation proof
Gets Token to inspect
User
Verifier
Domains

Figure 1 Actors

In real-life scenarios, actors can be one or more entities, such as physical or legal parties taking over responsibilities for their particular tasks within a Privacy-ABCs system. The roles of such parties have been, already, described in Deliverable D5.1 ([BGL12], page 15 and following). However, that document presents a, somewhat, detailed overview of all relevant actors involved in the context of Privacy-ABCs. Therefore, a shorter discussion on these actor roles (both mandatory and optional) is given in the following table, taking into account considerations on building blocks and potential use cases. The table below is based on the work done within the context of Deliverable D5.1, with suitable adaptations.

ABC Roles	Definition	Commentary
Issuer	The Issuer generates and provides to the User credentials containing her Attributes.	On request, the Issuer generates a credential (or more) during the issuance protocol and provides it to the User. Depending on the use case, the credential information may be provided either by the User herself or the Issuer, if the Issuer already has this information in the attribute database. Ideally, the Issuer can provide the information he attests directly, being an authoritative source. In doing so, he should have the right to assign the relevant attribute to

ABC Roles	Definition	Commentary
		various entities. For instance, an Issuer may attest the studentship of a User, or may attest to the bar association the fact that a User is an advocate, or may attest to the trade register the status of a company. Finally, attributes may also be generated "jointly random"[4], which may be useful for specific uses or cryptographic processes.
User	The User is issued the credentials while interacting with the Issuer enabling her, later, to provide proofs of her attributes towards a Verifier.	The User acts in different roles. She receives credentials from the Issuer and provides a proof for certain requested attributes towards the verifier. In some cases, additional information needed for inspection is provided as well.
Verifier	The Verifier receives a presentation token from the User allowing him to verify that the User possesses certain attributes.	The Verifier usually provides some kind of access to restricted services that require the User to prove her eligibility to access them. This authentication step requires the user to either reveal or to prove possession of certain attributes values.
Inspector	The Inspector reveals the identity or other hidden attribute values of a User (i.e. performs a lifting anonymity process) upon a formal request. To this end, the Inspector has to examine the validity of the request the previously declared and agreed upon inspection legislation.	The Inspector is an optional entity in an ABC system. In general, the ultimate goal of using Privacy-ABC systems is to provide Users with the ability to act fully anonymously while using various services. By deploying an Inspector entity, full anonymity does not exist anymore. Therefore, a Privacy-ABC system involving an Inspector entity is to be considered as pseudonymous only, with corresponding consequences for legal data protection requirements. Therefore, the use of an Inspector building block should not be the default setting but should be based on a well-thought decision. While the alternative to Inspection is to authorize data controllers to store identifying information for all Users in case the need arises to trace a User (e.g. unpaid bill, upload of illegal content etc.) Inspection may offer a more privacy-preserving solution. Data controllers

[4] "jointly random" stands here for a Nonce, or token that cannot be predicted and changes, so that replay attacks become impossible.

ABC Roles	Definition	Commentary
		may reveal data about users without them knowing about it. Inspection makes the whole process transparent to Users since it is always clear why and when their anonymity is lifted. The Inspector should preferably be an entity independent from the Verifier. Still, in case the Verifier belongs to a multiparty entity (like one department inside a company), it may be possible that another, independent party inside this entity can function as an Inspector. An example could be a company's internal data protection officer being assigned Inspector responsibilities.
Revocation Service	A Revocation Service is responsible for revoking issued credentials, so that these credentials can no longer be used for generating a valid presentation token.	The Revocation Service is an optional component of an ABC system. In practice, the entity operating the Revocation Service may often be identical to the Issuer, who is likely to have the most accurate information about changing attributes. Also, in this case, Users have a well-known contact entity (i.e. their Issuer) to request the revocation of their credentials.

Table 1: Description of Actors

3.2 Description of Building Blocks within Actor Domains

Below we describe the building blocks of a Privacy-ABC technology. Our emphasis is on completeness and generality rather than focus on a specific scenario. Some components, such as the Issuer or the User ABC System, may be based on the same cryptographic primitives (or actual software modules).

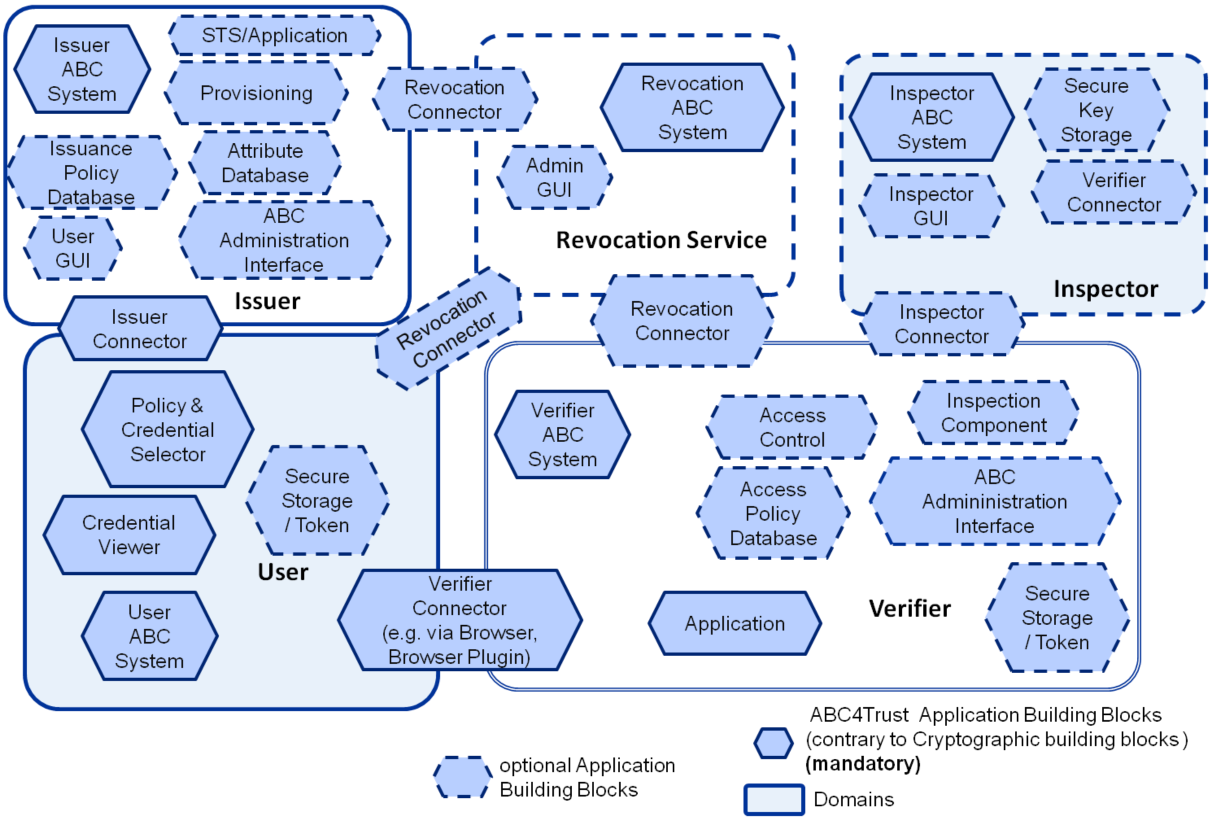


Figure 2 Building Blocks and Domains

The overall architecture (see Figure 2) can be divided into the different actors (or domains), described earlier. Each of the domains is explained below.

3.2.1 User Domain

The user domain is the one that is most widely seen, as it operates on the computer of the user in his home. Any application, such as web browser, cloud client, or standalone application that loads information from an ABC enabled online storage, is part of the user domain.

The building blocks in the user domain are the following:

User ABC System:

This building block includes the cryptographic computations to be done on the client side. Ideally, this building block is located on a secure computing platform (e.g. Smart Card, or Security Module). The involved cryptographic primitives go beyond simple symmetric encryption and, thus, most of the computation burden is places on computer and not smart cards, which do not have sufficient computational power.

Secure Key Storage:

This building block stores the keys. It can be on a secure computing platform, a removable or otherwise protected storage. A smart card is a good secure key storage for its high-resistance against attacks. The concern that a smart card might provide only limited storage capacity can be defeated by e.g. storing the decryption key of a much larger secure key stored on the client's computer.

Credential Viewer / Selector:

From a privacy perspective, the credential selector and viewer is the most important building block. It allows users to selectively provide credentials or choose not to provide them. It also makes the users aware of what credentials they have, or informs them that they have to contact an issuer in order to retrieve a missing (or revoked) credential. Finally, it allows users to perform some operations on their credentials, such as list, view or delete them using a credential manager. Non-ABC technologies lack the credential selector and, thus, it is not easy to provide the same level of information and control to the user. At the same time, the credential selector and viewer pose a user interface challenge, in order to be usable and understandable by users with no specialized technical knowledge.

Policy Viewer / Selector:

A website may allow access by people of variable status or rank. For example, a school website is accessed by pupils and their parents, teachers as well as school administrator staff. The same person may be allowed to access a system under different access policies and roles and, thus, the system can react differently (e.g. show an administration menu or not). The functionality of the policy viewer building block is to inform the user if he/she satisfies a specific access policy. The selector building block gives the ability to a user who satisfies a policy sent from either an Issuer or a Verifier to select alternatives (i.e. attribute sets) for satisfying the policy. In any case, if a user can not satisfy a policy the policy will not be shown to her/him without providing clearly stated information about which credentials/attribute values are needed to satisfy the policy.

Verifier Connector:

The verifier connector implements the functionality that handles the user interaction with the verifier. The User ABC System already implements the protocols required to retrieve a policy and provide proof of possession of the corresponding credentials plus the fact that they have not been revoked. However, the correct order of calling the corresponding API and providing the required parameters is left as an implementation decision. An external service-based API for each of the ABCE components is described in Chapter 5 of D2.1.

The ABC4Trust project publishes a software package that contains the source of the crypto libraries, the token and claim handlers and a summary documentation. This software package will be available on Github: https://github.com/p2abcengine/p2abcengine/wiki

A more in depth description of privacy ABCs and the process of presentation of a token can be found in Chapter 2.2 and 3.3 of deliverable [D2.1]

Revocation Connector:

The revocation connector contacts the revocation authority to retrieve Revocation Information according to a specific UID given by the Verifier, which can be used to check if the credential is revoked and, if not, provide a proof that it is not revoked. There could also be more than one revocation authorities.

The user does not need to reveal anything towards the revocation authority, but can obtain information whether the revocation handle, embedded by the issuer, is still valid.

Issuer Connector:

The issuer connector executes the protocol of credential issuance with the Issuer. If the underlying technology needs it (e.g. U-Prove) it also runs a protocol to retrieve additional tokens, which are used for the supported presentation proofs.

3.2.2 Issuer Domain

The issuer domain is the domain where bootstrapping occurs but it also functions as the link to the non-privacy protected user profile information. The issuer provides infrastructure services and requires components beyond the typical Privacy-ABC building blocks, for example to interface with other personal information domains (e.g. retrieving information that will be handed outside the Privacy-ABCs framework).

The building blocks in the issuer domain are the following:

STS/Application:

This building block implements the business logic of the issuer which includes, for instance, to what entities it can issue credentials. In a system that does not perform any verification but rather issues credentials to any entity requesting them, e.g. a queue ticket system, the application component has, almost, no functionality.

Issuer Connector:

This building block is mandatory, as it implements the network interface to the user client. It calls the external API (see Chapter 5.1 of [D2.1]) or wrapper functions of the ABC4Trust sample application. The issuer connector needs to be implemented with scaling behaviour and the surrounding network architecture (e.g. Firewall) in mind. The issuer connector needs also to be accessible (i.e. online), if the client is to receive fresh U-Prove tokens.

Attribute Database:

This building block subsumes all kinds of blocks that may be needed to retrieve information about the user which, then, is provided in form of ABC credentials. Also attributes generated on the fly are conceivable here such as, for example, in cases where a different IDM system issues tokens that are translated into ABC credentials after verifying their correctness (see scenario “Do not Track Relying Parties (prevented by the translation service)”.

Provisioning:

The most basic mechanism to fill the database is by provisioning the information, e.g. from the administration staff. But also one-time provisioning from other sources or continuous online-provisioning is possible here. Provisioning provides information to the attributes database.

Issuer ABC System:

This building block includes all ABC credential functionalities. It also implements the protocols of issuing credentials, as described in Section 3.4 of Deliverable D2.1. For more security critical applications, the issuer's private key should, optimally, be stored in a Hardware Security Module (HSM), although server-based installations are prone to using such secure credential storage.

Issuance Policy Database:

This building block controls the management of credentials. Section 3.4 of D2.1 provides more technical details on this issue. It needs to be carefully decided who is allowed to receive which credentials, which is formulated in the issuance policy (see. 4.5.1 of D2.1). Possible policies can include the requirement to have a certain other credential or to have revocation and/or inspection enabled.

Administration Interface:

The admin GUI allows administrators to change the entries in the database and revoke credentials. Administration also includes providing the Revocation Authority URL, Issuing Policies and Credential Specifications.

User GUI:

This building block is the front end to the users through which they can retrieve their credentials. Normally they will only visit the GUI during the bootstrapping phase but in case of credential changes by the administrator they may need to visit the user GUI, again, to download the new credentials. The user GUI does not need to be visited upon revocation or retrieval of new U-Prove Tokens (an underlying Privacy-ABC technology that relies on tokens) as this is done automatically using the Issuer Connector.

Revocation Connector:

The revocation connector at the issuer retrieves the revocation handles that will be embedded into the credentials.

3.2.3 Verifier Domain

The verifier domain is the domain where the actual business service is located. Access to the service or sub-services is governed by the Privacy-ABC technology, by means of verifying that users can satisfy a predefined policy.

The building blocks in the verifier domain are the following:

Application:

The Application is both the business logic and the GUI to the service provider. Certain resources (e.g. chat rooms) of the application might only be accessible upon providing Privacy-ABCs. The application might also include a database where information is stored. There is no need to store information about users, as they can only be identified if scope-exclusive pseudonyms are used. Review of the respective system would be necessary, in order to identify whether the data includes user identifying information that has been send outside of the official privacy ABC information channels (e.g. a pupil mentioning in a chat room his real name).

ABC Administration Interface:

This building block concerns both the service as such and the Privacy-ABC technology aspects, such as setting up the access policy for the (sub-) services. It is also possible, for example, that users administrate the applied access policy themselves, in case of user generated services/offerings (e.g. chat rooms).

Inspection Component:

The inspectable section of the credentials containing the identifying information is encrypted and cannot be used to identify the user in the normal run of business. However, under conditions that require the identification of a user which have been fixed previously and provided to the Users (inspection grounds), the identification can be done by the inspector. In this event, the inspection component needs to have an interface to the entity that may trigger an inspection, e.g. users reporting illegal content or the accounting department asking for the identity of a user not paying for a service. Then an administrator can provide the inspector with the necessary information to verify that the inspection grounds are met (e.g. the questioned chat log, the complaint of the accounting department, a court order etc.). Finally upon the verification of the inspection grounds, the inspection component needs a safe way to retrieve the inspectable token and provide it to the inspector for decryption. The inspection component should be implemented in a way so that no confusion between different inspection requests may arise even if the several cases of inspection are handled simultaneously.[5]

Access Control:

This is the component handling the access control to the application parts that can be accessed only upon proving certain properties, e.g. being over a certain age or member of a certain group.

Access Policy Database:

The access control system has to provide the verifier’s ABC System with a policy required to decide the access control question. In the access policy database the different access control policies are stored. Section 4.4 of D2.1 discusses the technical details of the access control policies such as, for example, the types of expressions that can be used. In systems where Privacy-ABCs are used for reasons other than access control, the Access Policy Database is just a plain Presentation Policy Database.

[5] For further details on how inspection may be deployed, see [ABDG12] where the inspection process has been implemented for a school communication network as part of the Swedish ABC4Trust pilot. Other grounds for inspection could be if the guest of a hotel does not show up and has to pay the deposit.

Revocation Connector:

This building block is interfacing with the revocation system. It is only necessary if issuer driven revocation is used (see D2.1 Section 3.6 and Section 4.3 for more details).

Verifier ABC System:

This building block contains the cryptographic algorithms and ABC credential functionality. It can be accessed directly by the external API (see Section 5.1 of D2.1) or using wrapper functions.

Verifier Connector:

The verifier connector is the building block that the user system connects to, in order to run the proof protocol, which includes supplying the Presentation Policy to the user and receiving the Presentation Token from the user.

Inspector Connector:

This is the building block that interfaces with the inspector. For inspection purposes, information of the inspectable tokens has to be retrieved. The interface can be a web based user interface, or a very simple command line based utility that allows, in the rare case of inspection, to retrieve the relevant information.

3.2.4 Revocation Authority Domain

The building block is essential to all credential based systems that want to invalidate credentials. However, use cases might also exist that do not require revocation (e.g. in our course evaluation pilot, the course attendance system never has the need to revoke that a student visited a specific lecture).

The building blocks in the revocation authority domain are the following:

Revocation ABC System:

This building block contains the cryptographic algorithms and ABC credentials functionality.

Admin GUI:

This is the interface that the administrator uses to in order to setup the revocation relationship, including the option to revoke credential handles. The revocation authority has no knowledge of which user or credential the credential handle belongs to. For this, the issuer application (or other component) needs to keep track of revocation handles and provide them upon a revocation request from authorized entities.

Revocation Connector (to Issuer):

This building block is used by the Issuer for retrieving revocation handles that will be embedded in the revocable credentials. There can be several revocation authorities, but each credential corresponds to, exactly, one revocation authority (revocation handle). Retrieval of a revocation handle is done on-the-fly during the issuance of credentials.

Revocation Connector (to User System or to Verifier):

This building block is used by the Verifier (or User System) for retrieving a proof that a respective credential has not been revoked. This is done by verifying a revocation handle, something that does not constitute a threat to privacy, because the revocation authority does not learn the identity of the requestor.

3.2.5 Inspector Domain

The inspector domain is optional and it is only needed in cases where the anonymous use of a system may create a situation that requires lifting the anonymity of the user. The users will see, on a single credential verification basis, whether at a later stage their anonymity can be lifted. Besides the technical building blocks, the inspector domain requires a clear organisational process adapted to the needs of the particular use case, to avoid compromising of user privacy.[6]

The building blocks in the inspector domain are the following:

Inspector ABC System:

This building block includes the cryptographic functionality to implement inspection. It deals with the underlying cryptographic primitives required for decryption of the verification tokens.

Secure Key Storage:

This building block guarantees that the inspector key, that is able to lift the anonymity of users, will not be possible to be cloned or misused. Organizational measures are needed, such as the inspector knowing the PIN to the inspector smart card, but only obtaining (physically) the card from the inspection decision board in case an inspection is initiated

Inspector Connector:

The verifier connector interfaces with the verifier to retrieve the necessary information for lifting the anonymity. The component could be network or file based.

Inspector GUI:

The GUI building block allows the inspector to select the token to be inspected and create a revision safe log entry about the event of inspection. The Inspector GUI has to be integrated into the organizational process of inspection.

[6] See also [ABDG12] where the inspection process has been implemented for a school communication network as part of the Swedish ABC4Trust pilot. To address the specific participation requirements of a school system an inspection board has been established. The board decides on the question whether the inspection grounds are met or whether less invasive means such as deleting an entry in the social network is sufficient. The board's decision is necessary before the inspector may decrypt the inspectable token.

3.3 Design, Setup and Interaction of the Systems

This section discusses considerations about general system architecture and setup. Design options and good design practices of designing a system for specific scenarios, such as those described in Chapter 3, will be discussed. The description stays on a high level, however, as technical details of the relevant implementations are provided in other deliverables (see, for instance, D2.1 and H4.1). Example XML files are also included in the appendix to this document for ease of reference.

3.3.1 Order of Generation of the Files

A scenario that involves different stakeholders instantiating the actors (e.g. Issuer, Verifier, User Service, etc.) will, most probably, wish to work independently of each other, in parallel. Independent development is, to a large extent, possible based on our example code and dummy implementations (available from the project). However, during the integration phase, imposing a specific order of the actions is beneficial while, sometimes, it is even mandatory.

1) Write Credential Specification

In any scenario the verifier is, ideally, the one who defines which credentials are needed but it is also conceivable that the issuer defines a set of credentials that are generic enough to suit the needs of all verifiers. In any case, the credential specification has to be written first. Considerations that are relevant to the credential specification include the following:

- Attributes and Attribute Types (including how the initial value per user is derived).
- Whether the credential is revocable.
- Size of resulting credential (in case of a size limited smart card is used).
- Whether a credential is key-bound, or can be transferred.

The credential specification is the data model of the Privacy-ABC System. It specifics issues such as scope-exclusive pseudonyms or which attributes are inspectable, issues that although they are not directly involved with writing the actual credential-specification file, they should nevertheless considered when developing the data model, as presentation policies may depend on them.

2) Generate Issuer Parameters

By running the ABCE with the issuance policy installed, the system will create new cryptographic keys, which are the issuer's parameters. These parameters are needed, later, during the setup process.

3) Generate Revocation Authority Parameters

If there are one or more revocation authorities, the corresponding cryptographic parameters need to be generated and provided to the issuer, verifier, as well as user system developers. This may be done directly by the revocation authority or the issuer, in which case it provides them along with the other issuer parameters to the involved entities.

4) Generate Inspector private/public keys.

If there are one or more inspectors, the cryptographic parameters need to be generated and provided to the issuer, the verifier, as well as the user system developers. This may be done directly by the inspector(s), or by giving them to the issuer, which provides them along with the other issuer parameters to the involved entities.

5) Generate Installer (with Credential Specifications, Issuer Parameters, revocation parameters and inspector public key)

This can be, most likely, done by the service provider (verifier), as the naming of credential attributes and options in the user system GUI are dependent on the scenario. In cases where a generic system is used by several issuers, the installer can be provided by a third party, such as public institution.

6) Write the Issuance Policy (+ install HTTPS Certificate)

The next setup step is with the issuer, who creates an issuance policy. There is, also, the unrelated but strongly suggested task of installing the HTTPS certificates for transport layer confidentiality.

7) Write the Presentation Policy (+ install HTTPS Certificate)

The presentation policy at the verifier can be written at the very end and it is even possible to have it semi-automatically generated on the fly. However, it is good to write a draft of the presentation policy during the first phase of the system setup, as it provides clarity with respect to which type of credentials are needed by the application.

3.3.2 User Domain

The user system is unique in that there are two different setup phases: (a) by the system architect who creates the software installer for the end user and (b) by the end user who installs the user system on the PC.

3.3.2.1 Perspective System Architect

The system architect needs to decide whether the user module is particular to the scenario in hand and, thus, a specific installer needs to be created (we will assume this case in our scenario). If the user system is generic, then only the scenario specific credentials and parameter files need to be added.

Prior to packaging the user system, the developer needs to obtain the following files:

- Credential specifications
- System parameters
- Issuer parameters
- Revocation authority parameters (assuming revocation is used)
- Inspector public key (assuming inspection is used)

A scenario specific installer has the advantage that the user interaction elements can be adapted to be more intuitive for that scenario scenario. For example, if a scenario allows the existence of several issuers and revocation authorities, then the user system may include a selector that allows the user to select a preferred issuer. Or, in some scenarios, a credential may be act as a proof of a user having participated in a number of events. Thus, deleting this credential might be called "reset event participation information". The system architect shall carefully develop not only the verifier's service web site, but also the user system, credential specifications (esp. alternative texts) and presentation policies (esp. the verbal description part).

In case of a generic user system used in combination with a smart card, special care has to be taken that different services do not attempt to run presentations (or issuance) simultaneously, as there is one state in the ABCE and only the last proof state can be stored.

3.3.2.2 User Perspective

Setting up the User System is relatively easy. One just needs the following:

- Access to a computer connected to the Internet

- A PC/SC compliant USB smart card reader
- A properly initialized smart card and its PIN/PUK values
- The appropriate User System installer file (installer.exe)

In order to activate the smart card reader, a user has to plug it in the USB port and allow the operating system to install the appropriate drivers. If the operating system does not provide the drivers, then the user has to download and manually install the appropriate drivers from the smart card reader's company website.

In our pilots, the user service is implemented using Java Runtime Environment. Hence, when the User System installer file (installer.exe) is executed on the user's computer, it first checks whether the Java Runtime Environment (JRE) and Microsoft's .Net runtime are present. If they are not, they are downloaded and installed upon the user's permission. Then the user-service (user ABCE) is installed and deployed locally on Jetty, a Java web service container. Finally, special browser plug-ins are installed on the user's web browser (e.g. support for Firefox or Internet Explorer). In case of a Linux operating system, the user has to install manually the Java Runtime Environment, the Mono development framework and the browser plug-in in order to deploy the user service. For user services implemented in other frameworks, similar steps apply.

Then there is the browser plug-in which provides a menu that offers to the user some useful functions such as browsing the credentials stored on the smart card, changing the smart card PIN, backing up the smart card or unlocking the smart card using the PUK number. Moreover, when the user interacts with an Issuer or a Verifier, the browser plug-in is responsible for communicating the messages between the user ABCE and the Issuer/Verifier ABCE. In case of such communication, the plug-in also invokes a user-interface called credentials selector. This interface informs the user about what kind of data is exchanged between her side and the Issuer/Verifier side. When a user is involved in a verification session, the interface informs her about the possible ways she can satisfy the Verifier's policy (e.g. use credential A or B or a combination of them) and allows her to select among them. Finally, if some user attributes can be revealed during a session, the user is notified of this fact and gives his consent to proceed with the proof or interrupts the process.

3.3.3 Verifier

For setting up a Verifier, a server with the preferred web technology of the service provider has to be installed. Additionally, the ABCE modules need Java (WebService) and .Net. There are, also, some mandatory system files, like the system parameters, the issuance parameters for each type of credential, the revocation authority parameters, as well as the credential specifications that need to be present at the web container's application directory.

The verifier ABCE provides two basic web services. The first service is responsible for communicating a presentation policy to the users who wish to access a resource or service. A presentation policy states which credentials a user should possess and what attributes he should reveal to the service out of them in order to obtain access to a resource. The second service is responsible for verifying the validity of a presentation token provided by a user, with respect to the presentation policy.

In a typical ABC presentation interaction, the user first requests access to a protected resource, upon which the verifier sends a presentation policy that describes which credentials the user should present to obtain access. The user agent module then checks whether it has the necessary credentials to satisfy the verifier's presentation policy and, if so, it generates a presentation token containing the appropriate cryptographic proofs.

Upon receiving the presentation token, the verifier checks that the cryptographic evidence is valid for the presented credentials and checks, additionaly, that the token satisfies the presentation policy. If both tests succeed, it grants access to the resource.

The presentation policy is an XML file communicated by the verifier to the user when asking access to the service. This XML file states which credentials a user should possess and what attributes he should reveal from them in order to grand him access. A presentation policy can also include place holders that will be filled by the verifier before providing it to the client. In cases where the presentation policies are not fixed beforehand (for instance, when a previously unknown combination of presentation policies is required), it can be beneficial to implement an XML policy generator. For example, in the Greek (Patras) pilot all XML policies are manually created, while in the Swedish (Söderhamn) pilot a combination of manually and automatically generated XML-files is used. The Swedish pilot had a requirement to allow end-users (school personnel, pupils and parents) to be able to create different Restricted Areas (similar to chat rooms) and define different access policies that apply for each Restricted Area. These access policies protect the Restricted Area from unauthorized access so that only users with credentials containing attribute values (e.g. credSchool.Gender = Boy) that comply with the access policy are allowed to enter a Restricted Area (for boys, in this example). As access policies protecting Restricted Areas from unauthorized access are not known in advance, they should be generated on the fly. The Swedish pilot makes use of both manually and automatically generated XML-files. Even the manually generated XML policies will contain a random nonce unique to the session along with application data with the details of the required policy.

In general an architect or developer interested in implementing a Privacy-ABC system needs to make decisions on how to generate XML policy files and what information they should contain to comply with the requirements of the intended implementation. In Appendix A an example presentation policy is being presented.

3.3.4 Issuer System

The issuer component is required for bootstrapping the scenario. First a user has to obtain credentials with attributes which later can be used to satisfy presentation policies of verifiers.

In the very first interaction of the user with a Privacy-ABCs application, it is not possible to employ Privacy-ABCs based technology. We have to have another means of identifying the users and, thus, provide them with the required credentials. It can be said that the first issuer a user contacts in the Privacy-ABC world often knows the user both in the non-privacy protected and Privacy-ABC world[7].

There can be more than one issuers, like the Tombola ticket issuing service in the second round of Patras pilot[8]. Such issuers rely on a Privacy-ABC attribute and thus do not need to be able to identify the user in a non-privacy protected way.

3.3.4.1 Bootstrapping

In order to create an issuer service, several processes have to be initiated:

1) Provisioning of the attributes database

The attributes database has to be filled and initialized. Often the real-world scenario knows already a list of users along with their attribute values. Or there may exist a (non-privacy protecting) system that can be contacted to aid the provisioning process. After the provisioning step is accomplished, it is good to check if the credential specifications can be satisfied by the attribute database.

2) Bootstrapping with ecosystem

In revocable credentials, a revocation handle must be embedded. To this end, the issuer has to create a connection to the revocation authority. Other connections are for inspection and for the distribution of

[7] Due to the unlinkability in the privacy protected world the Issuer cannot track the user.
[8] The upcoming Deliverable D7.3 will contain information about the Tombola service.

the cryptographic parameters to the user components. Along with the cryptographic parameters, legal contracts might be necessary to be signed. For example if the issuer gets special user attributes from the verifier, effectively outsourcing part of the attribute handling, a contract between the issuer and the verifier is required.

3) Issue initial identity to users

At this step, an initial authentication process of the user has to be performed. In one of pilots (Patras pilot) we employed a mechanism based one time passwords that were unique to each user account and given to users in sealed envelopes along with their smart cards. After this authentication step, the users log in the system using only Privacy-ABCs technology. Other means of issuing the initial credentials to the users are, also, conceivable.

3.3.4.1 Issuance Policies

The issuer has to create an issuance policy. This policy is similar to the presentation policy. The overall system performance can be improved by having dependencies between issuance of different credentials and attributes. For instance, a pupil can only obtain a class attribute if she is member of the school (i.e. the pupil has the school credential). If such a dependency does not exist, then both attributes need to be revocable, otherwise for only one of them, provided during issuance and presentation, this dependency is checked.

3.3.4.2 Performance Requirements

In designing an issuer, scalability is a mandatory requirement, because in most scenarios there will be a large demand for issuing attributes at the start of a use scenario (e.g. when a new school semester starts) and only little demand later on. The issuer is a service that can be benefitted from the scaling advantages of cloud computing, provided the protection of the attribute database can be ensured by the cloud operator.

3.3.5 Revocation Service

Theoretically, a variety of revocation mechanisms can be implemented. The choice of the revocation mechanism can have several impacts on the application, such as support for different features and overhead on other entities.

In our pilots, we have chosen a revocation mechanism based on cryptographic accumulators [CaLy02]. This enables immediate revocation (timeliness) while it is, also, scalable. However, this mechanism imposes some burden in the Users and the Verifiers.
Just like other entities, the Revocation Authority also needs an initialization procedure before it can be deployed and integrated in the architecture, along with the other entities. During this process, this entity generates its public parameters along with the corresponding private ones. The parameters of the Revocation Authority are static (i.e. not changing over time) and the public parameters are made available to the other entities (Users and Verifiers).

3.3.5.1 Interaction with other entities

If revocable, each credential contains a separate attribute, known as the revocation handle, which is injected into the credential. This attribute is used for revoking the corresponding credential, when it this is necessary. This attribute must never be revealed to the Verifier.

In our implementation, the Issuer contacts the Revocation Authority during the issuance, asking for a revocation handle. The Revocation Authority supplies the handle, which is both stored at the accumulated (and published) value at the Revocation Authority. It is, then, inserted in the newly issued credential from the Issuer.

After a successful issuance protocol, Users receive the non-revocation evidence together with the credential. This value can be kept up to date using the public revocation information to successfully prove that a credential has not been revoked. The non-revocation evidence must not and will not be disclosed to the verifier in a presentation protocol. For details please check D2.1.

The Revocation Authority maintains a single public value, which *accumulates* the revoked revocation handles. This value is dynamic and changes after each revocation ("deletion" of a revocation handle). Therefore, it should always be available to both Verifiers and Users.

During the presentation, the Verifier informs the User of the version of the revocation information they should, both, use. It also sends a link to the publicly available information from the Revocation Authority, from which this information item should be updated, if necessary. Therefore, the User must prove during the presentation not only that she fulfills the presentation policy in terms of possession of certain credentials, but also that these credentials are still valid, according to the version of the revocation information the Verifier accepts. This is directly translated into computational cost, but also communication overhead for the User depending, also, on the need to update her non-revocation information (must be online). It must be noted that when updating their non-revocation evidence, Users do not reveal their identity to the Revocation Authority (nor to any other entity), but rather fetch the latest updates and insert them locally into their non-revocation evidence. If many revocation operators have been performed since the last time the User has updated her "*witness*", this update process may take more time as a new update must be done for each revoked handle individually.

3.3.5.2 Availability issues

As we have seen, the Revocation Authority is involved in many processes of the other entities: it interacts with the Issuer during the issuance process while it also interacts with both Verifiers and Users at later stages, for presentation proofs. In case a large number of users access the system, the task of disseminating the latest public revocation-related information (the accumulator) can be delegated to a separate entity (server), instead of obtaining it directly from the Revocation Authority. Moreover, it can also be split among more entities, in order to provide more availability, depending on the other application requirements. As we see, the role of this entity is quite important. Therefore, service level agreements should be in place and the system designer should consider strategies to deal with the possibility of the revocation service being unavailable at some times.

3.3.5.3 Choice of the revocation mechanism

The choice of the revocation mechanism should be made after carefully reviewing the requirements of the application. In case the application does not require revocation immediately after deployment, but can allow a period of time during which the credential can be accepted as valid, short-lived credentials can also be used as an alternative. In this way, the validity period of the credential is encoded as an additional attribute and when it expires a new credential may be issued. However, this then shifts the burden to the Issuer. This alternative has not been developed for our pilots.

Alternatively, the revocation based on accumulators can also be customized to enable a short period of time during which the User does not need to update the non-revocation information, which has a positive impact on the efficiency of the presentation, as contact with the Revocation Authority may be avoided (temporarily). This period of time can be decided depending on the application requirements on other scenarios, i.e. it can be on a 24h basis.

3.3.6 Inspector

While the technical setup of an inspector is straightforward and only involves the generation of the cryptographic parameters, which need to be distributed with the User System, the organizational and legal setups require special care.

Transparency and legal requirements demand a proper previous documentation (3.3.6.1) while all XML specific modelling and necessary files need to be, suitably, set up (3.3.6.2)

3.3.6.1 Legal and transparency considerations for inspection use cases

Inspection allows the avoidance of personal data collection where such a collection is not necessary in the normal course of business. Yet, in case the data becomes necessary, as documented in previously defined and publicly available documents, identity data can be retrieved. This can occur, for instance, in cases of breach of contractual obligations by the User (such as non-payment of service use).

In general, such a measure enhances data protection as the majority of Users are not identified by the Verifier. Rather, the Verifier only learns the attributes necessary for the normal course of business such as, for instance, age verification.

But while inspection is an advantage for a general privacy-friendly system operation, it does not render compliance with data protection laws obsolete. The User is still identifiable, and thus all information stored in relation to such a presentation token is still, considered, personal data. Therefore a solution allowing fully anonymous operation (e.g. realised with Privacy-ABCs without enabling the inspection feature) is a first choice from a privacy perspective but using Privacy-ABCs with inspection enabled is still highly preferable in comparison to the current practice of unconditionally identifying all users if the need arises.

Generally, the Verifier is the responsible entity (data controller) for the processing of the obtained personal data. In cases where the data revealed in the presentation token does not allow the identification of the user, such as, for instance, when only age verification is necessary, the Verifier does not learn the identity of the user. However, this does not change the legal status of the Verifier as a data controller. The identifying information is still contained within the inspection part of the presentation token and the information is obtainable with aid from the inspector.

To comply with the data protection legislation and other legal requirements a data controller needs some inspection specific documentation and the adaptation of existing documents prior to starting processing personal data. In this respect, dee the summary below:

- Inspection grounds: A document which defines the cases and conditions under the Inspector may disclose the personal data in a presentation token. This document must contain the reasons for which the Verifier considers revealing the User's identity necessary. It should further mention generic reasons beyond the control of the Verifier, such as requests by law enforcement entities, e.g. based on a court order.
- Privacy policy, which informs the User that personal information may exposed and linked to her identity under predefined conditions. The policy should include or link to the inspection grounds document.
- Likewise, a user consent form must mention that the identity may be revealed under the conditions laid down in the inspection grounds document.
- The inspection process must be defined beforehand and, clearly, described in a document defining the steps to be taken in the event that inspection is required. Questions to address, in this context, may include the following: (a) who checks that the conditions in the inspection grounds are fulfilled? (b) how does the Inspector obtain the necessary information to verify this? (c) how is the encrypted part of the presentation token obtained by the inspector for decrypting?[9] In short, this document should explain all the necessary steps in a sufficient detail level to enable external examination of the inspection process itself. It, also, serves as internal information for the Verifier, the Inspector and third parties having legal interests such as data protection authorities or law enforcement agencies.
- Whenever possible, the relation between the Verifier and the Inspector should be governed by a contract including the inspection grounds as an integral part of the document. Generally, the contract should be on the appropriate detail level as required by a data controller contract.

[9] This, essentially, asks for a description of the inspector connector building block.

- In legislation contexts where a notification of processing of personal data to the national Data Protection Authority is required by law, a short description should be given on the involved processes with references to the previously listed documents.
- For all documents and product descriptions: Do not mention that the User is really "anonymous" since inspection only grants "pseudonymisation", in the legal terminology. Where necessary, consider a wording such as "The User does not get identified towards [Name of Relying Party] in the normal course of business."

In addition to these documents, several additional issues should be addressed before operation to prevent unpleasant consequences and to avoid later needs for adaptations of the system that may prove to be difficult to implement:

- Availability: Can the planned inspection process handle a larger number of inspection cases than expected? How timely and reliable must the inspection result be available?
- Is it ensured that the inspector can, both, keep his keys securely stored and also avoid losing access to them?
- How may a change in the Inspector entity be accomplished, e.g. due to changes in employees, sickness or other reasons for unavailability of the inspector?

Depending on the audience addressed, all documents intended for User information may, in addition to the full legal text, be accompanied also by a high level summary, which is easier to understand and describes the essential parts in simple language omitting the legal details.[10] Likewise, the internal documents may be in a format appropriate for User entity, e.g. using a flowchart to describe the inspection process.

3.3.6.2 Technical Setup for inspector

In our setup, the inspector domain does not include service-like components, such as the verifier or issuer, which need to be always online and provide an API or a user interface. The inspector has to generate a key pair and make it available to the other domains, so that they can use it to make attributes inspectable.

An attribute becomes inspectable if the presentation policy requires it to be so and, hence, the presentation policy viewer is important (see Section 3.2.1). The presentations of an inspectable and non-inspectable component do not differ in complexity.

The complexities in the technical setup of inspection lie, mostly, in the management of inspection requests and the provision of inspection tokens to the inspector. There are numerous ways to accomplish these tasks and which is the ideal on depends on the setup of the verifier. Thus, the inspector and the verifier have to, mutually, agree on the required interfaces. Sometimes a web-interface will allow the inspector to retrieve the information. Also, if the Privacy-ABC technology is deployed along an email-service, the email will be copied and transferred to the inspector, or maybe for a payment service, providing an XML file is the best method of sending the inspectable content to the inspector.

In the case where the inspection is performed upon user request, there also has to be a way for the user to inform the system which of the (privacy protected) presentation tokens need to be inspected. Then this token, along with the reasoning and accompanying legal documentation, has to be provided to the inspector. The documentation might require the verifier to support additional technology in order to

[10] As it has been suggested by the Article 29 Working party towards more readable privacy policies – see [Art29WP100]. However, the full text versions are still necessary for reference purposes and control by the corresponding authorities.

(a) protect the privacy of non-involved parties, and (b) provide an adequate evidence context for an informed decision by the inspector.

Additional attention should be paid to cases where several inspection requests are handled simultaneously so as no confusion arises (i.e. to avoid opening or mapping the wrong token).

Finally, when all inspection requirements have been satisfied, the inspector uses his secret key to open the inspectable presentation token and act upon the revealed information. This action entails either informing the Verifier and other authorities or acting autonomously. However, the inspector actions must to be defined in the inspection process documentation (see the the previous section on legal considerations).

4 Some Typical Privacy-ABC Scenarios

Prior to discussing typical ABC Scenarios, a high level view of user activities in the Internet will be provided in order to understand better the need of privacy protection.

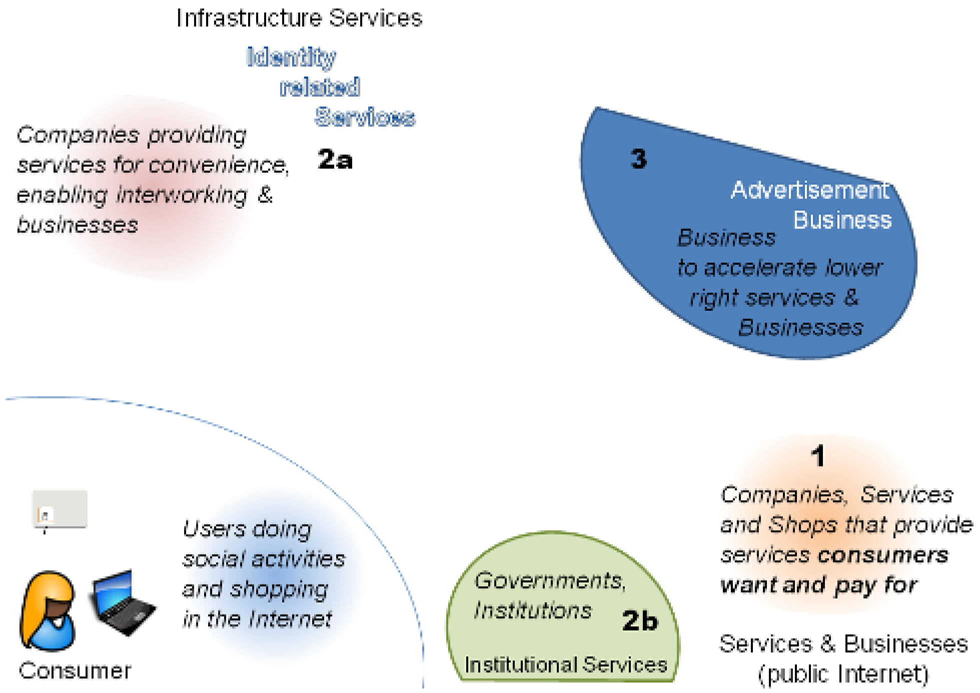

Infrastructure Services
Identity related Services
Companies providing services for convenience, enabling interworking & businesses
2a
3
Advertisement Business
Business to accelerate lower right services & Businesses
Users doing social activities and shopping in the Internet
Consumer
Governments, Institutions
2b
Institutional Services
1
Companies, Services and Shops that provide services consumers want and pay for
Services & Businesses (public Internet)

Figure 3 High-level View of User Activities on the Internet

As shown in Figure 3, typical users (lower left-hand side) visit a company that provides some service (1). This service can be immaterial, such as accessing a social network account, obtaining better Web search results, or participating in discussion groups. Or it can result in some material return to the user, such as print pictures, online shopping, etc. These user interactions drive the economic system and business processes.

Besides providing quality goods or services, the employment of proper marketing and advertisement (3) is an important element in conducting business. Two companies may provide a similar product but the company with better marketing and customization to people's needs will, ultimately, get the financial benefits. Thus, knowing users' habits and charactiristics becomes an imporant business element.

Knowledge of the users implies his identification or, simply, knowing some attributes of his identity. Examples of attributes that a business operator may wish to know about the user community include ownership of a car, family status, recent visits to certain types of electronic services etc. Government institutions (2b) and infrastructure service providers (2a) as well as companies that provide services (1) are the ones that are interested in collecting (and, ultimately, collect) information about the users. The stronger the business interest and the harder the competition, the more interesting it becomes to profile the users.

While infrastructure services (2) may be able to compete among themselves offering stronger privacy protection of their users, the companies and services (1) have little intrinsic interest to not profile the user because they want to maximize their profits through this profiling. Thus, it becomes important that the user cannot be easily recognized as a specific returning customer, or even the same customer, through a collaboration of various Internet sites he has visited in the past. This anonymization, and privacy protection, can be achieved, elegantly, by Privacy-ABC technology.

Compared to existing standards, such as OAuth, SAML, OpenID, etc. Privacy-ABCs have built-in mechanisms that allows users to select which information they whish or need to disclose towards a service, keeping all remaining information private. At the same time, standardization of privacy protecting protocols is slowed down due to the low number of well established underlying privacy technologies. Privacy-ABCs based technology can improve on this issue, as well.

In what follows, we outline some scenarios along with their utilization of specific building blocks and domains. The interested reader will find examples of well-known, publicly discussed scenarios, such as enhancing eIDs with privacy protection mechanisms. Scenarios that facilitate the evolution of business ecosystem, such as banks entering the IDM business domain, There are, also, two scenarios that require more technical ecosystem effort and, thus, they can be considered as applicable in the future.

Since the different scenarios highlight different privacy protection goals, it will be possible for the reader to compare our scenarios with his and, thus, deduce which of the optional high level building blocks are necessary to his application. It should be noted that the protection goals of one of our scenarios could also be included or achieved in the others. However, we restricted each scenario to one protection goal for clarity of exposition.

4.1 Scenario: eIDs

4.1.1 Introduction to the Use Case

Electronic Identity (eID) smart cards are rapidly emerging in Europe and are gaining, gradually, wide user acceptance. As an authentication token and personal data container, an eID card is a gateway to personal information. This, however, entails certain risks to the privacy of the citizen, through the unwanted disclosure of personal information and its subsequent misuse. As the information in official eID documents is verified by an entity trusted by most market participants this, in addition, adds a new quality to the data in comparison to ID information provided by the Users themselves. It is good for the accuracy of the data but also deprives Users from acting under self-chosen pseudonyms [Zwin11]. These privacy risks could become even more prominent in the future, if citizens would be using their eIDs not only for e-government services, but also in e-commerce for shopping online, booking rooms at hotels, renting cards online, managing bank accounts, etc.

Several European countries have taken extra care to protect their citizens against these risks [ENISA09]. A notable example is the German eID card. The German eID card provides a set of features to protect the user's privacy. Before gaining access to a German eID, the Service Providers must perform a checking procedure done by the German federal authority and prove that the personal data requested is necessary for the requested service[11] (for details on the process please refer to [Zwin11]). This serves the basis of obtaining a digital certificate, which is also used to identify the Service Provider and display the purpose of the processing to the user. This establishes asecure identification process at the Service Provider's side first.[12] Also, the possibility to have the User reveal only parts of the attributes, so that the User has full control of his personal data, is another important

[11] See § 21 German Personalausweisgesetz (German law on eIDs), online: http://www.gesetze-im-internet.de/pauswg/BJNR134610009.html

[12] See § 18 Sec. 4 German Personalausweisgesetz.

requirement. Moreover, citizens must consent to every attempt, by service providers, to access their personal data. On-card verification supports use cases such as anonymous age verification and proof of place of living as well as selective disclosure of attributes. Finally, service-specific pseudonyms allow a secure re-identification of users while being unlinkable across different services they have used in the past.

The European Commission published a proposal for a “Regulation of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market” (herein: eIDAS).[13] This proposal aims at removing existing barriers to the digital development in Europe by providing the legal basis for a wider acceptance of electronic identification and authentication means, as mandated by the Digital Agenda.[14]

Achieving cross border interoperability is, also, an important goal. However, the Regulation should not be stated in a way that it, essentially, prevents privacy preserving solutions from being applicable by Member States due to cross-border legal and technological differences among the EU countries. The privacy legislation experts within the ABC4Trust project have performed an analysis addressing these obstacles in detail also prosing solutions for the lawmakers. In [ZwiScha13] the authors discuss the legal prerequisites that must be met in order to deploy Privacy-ABCs directly as a part of the officially issued eIDs, which would be our first option. However, at the moment, neither the EU Member States nor the market are sufficiently advanced to adopt such a solution while even the strongly privacy-preserving German eID framework would require a major update of its technology to directly support Privacy-ABCs. For this use case we, therefore, would rather address a solution that allows a combination of Privacy-ABCs with existing national eID schemes, where Privacy-ABCs can act as intermediate solution. If this approach is adopted by EU countries, then it will be easier for all Member States to introduce eID schemes with direct support of Privacy-ABCs.

4.1.2 Issues to solve

Despite the steps outlined above to protect user’s privacy in eID scenarios, several security and privacy concerns still remain. The problem originates from the fact that authentication schemes follow passive authentication protocols with bearer tokens. Bearer tokens (security tokens) containing user's claims are delivered by the eID server to the service provider without user intervention. Unless each relying party operates its own eID server, which is a resource intensive task, this model is subject to several threats, as elaborated in [BKPR12]:

- The eID server knows all user transactions. Even though the eID server does not necessarily need to know where the user is authenticated and which service she is requesting, this knowledge is passed, by design, to the eID server in the current eID solutions. More specifically, the eID server is involved each time a user authenticates herself to a service provider using her eID and, thus, it is able to keep track of the user actions. This enables the eID server to trace and link all communications and transactions of each user (user profiling).
- The eID server knows all the customers of a service provider. Reversing the above threat, the involvement of the eID server in every user authentication constitutes a threat for the service providers’ business secrets as well, since the eID server learns who are the customers using a specific service. Especially if the eID server is operated by a private company, it might be a

[13] For the proposal text and other related legislative documents see: http://ec.europa.eu/prelex/detail_dossier_real.cfm?CL=en&DosId=201689.

[14] Key Action 16 reads: “Propose a Council and Parliament Decision requesting Member States to ensure mutual recognition of e-identification and e-authentication across the EU based on online 'authentication services”, in A Digital Agenda for Europe, COM (2010) 245final, online: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:52010DC0245:EN:NOT .

competition threat, if it can learn all the customers of another company (i.e. the service provider).

- User impersonation. Since the user does not perform an active role in the information exchange between the eID server and the service provider, there is a high security risk of user impersonation by insider attackers at the eID server or outsider intruders who can gain access to the eID server's resources. An eID server under the control of an attacker (either insider or outsider) has the ability to impersonate every user at applications using eIDs for authentication. For example, insiders can copy or alter users' credentials and, thus, steal their identities. In general, in a federated eId environment, the insiders or outsiders who acquire a user's credentials can impersonate the user and get access to the assets at different services belonging to the federated domain.
- Availability. The eID server becomes a business critical component (single point of failure) as it is needed for every transaction the user performs with the service applications. Denial of Service attacks towards the eID server will impact all applications using the service. Attacking this component may have a huge economic impact because the attack can, then, spread over many different services.

All of the above problems become even more critical when there are only a few eID servers operating instead of a fully scalable, distributed model.

Meanwhile, the requirement that the eID providers must not be able to track the behaviour of eID holders is becoming more prominent. In the evaluation assessment of the recent proposal of a Regulation "on electronic identification and trusted services for electronic transactions in the internal market" it is stated that a solution to this tracking problem should be aligned with the current on-going revision of the Data Protection Directive and include specifically privacy-by-design rules. In the next section we discuss specifically how the above threats can be addressed within the privacy-by-design model.

On the legal side, now, there is a major issue to solve as well. The initial version of the German law on eIDs ("Personalausweisgesetz"[15] = PAuswG) strictly prohibited that a relying party professionally (German "geschäftsmäßig") transfers the obtained information. The objective of this rule was the prevention of, e.g., address brokers from obtaining and selling personal data without their owners' consent and control.[16] The law was amended in 2013 and now allows the transfer of the obtained information to previously defined third parties.[17] However, it is still not allowed, by the statements in this law, to have an identity broker obtaining personal data since ID brokers ("geschäftsmäßig") transfer the data by profession.

Progress can be made if we allow yet another amendment to the German law stating that specific ID brokers are allowed. Then, these must either transfer the data only on behalf and under control of the user to third parties or issue credentials to the user based on the obtained personal data.

4.1.3 Advantages of a Privacy-ABCs solution

Privacy-ABCs have a significant potential to enhance existing eID smart-card based privacy solutions. Their integration with existing eId infrastructures is perfectly realizable today, although slight

[15] http://www.gesetze-im-internet.de/pauswg/BJNR134610009.html

[16] See § 21 Para. 2 Nr. 2 German PAuswG.

[17] The amendment adding § 21 Para. 2 Nr. 2a German PAuswG as ratified in the Gesetz zur Förderung der elektronischen Verwaltung sowie zur Änderung weiterer Vorschriften (EVerwFG) as in force since August 1st. 2013.

modifications of some of their elements may be required. For example, enhancing German eID servers with the capability of issuing Privacy-ABCs would make the eID server act as a Privacy-ABC issuer.

Deploying the Privacy-ABC issuer applications can leverage Privacy-ABC Tokens providing unlinkability and anonymity to the users.

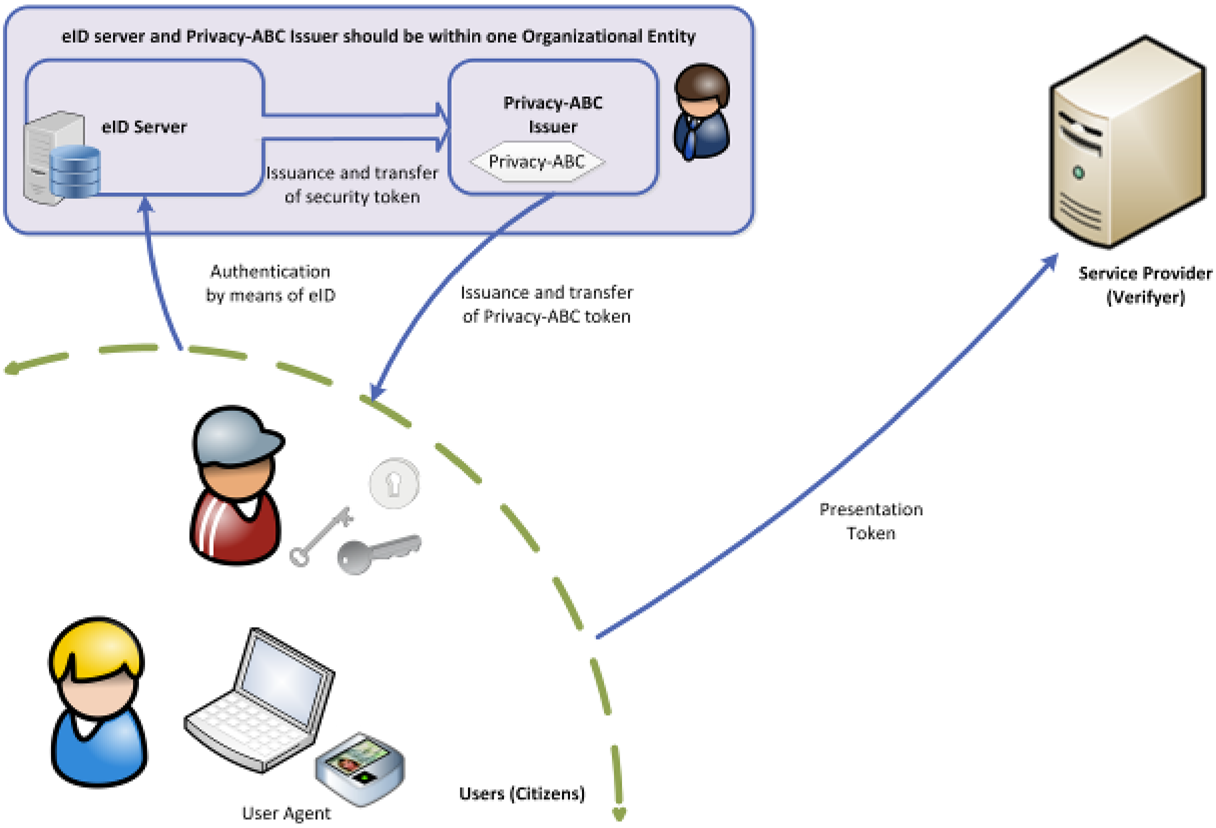

eID server and Privacy-ABC Issuer should be within one Organizational Entity
eID Server
Privacy-ABC Issuer
Privacy-ABC
Issuance and transfer of security token
Authentication by means of eID
Issuance and transfer of Privacy-ABC token
Service Provider (Verifyer)
Presentation Token
Users (Citizens)
User Agent

Figure 4 Overview of an eID scenario using Privacy-ABCs

In more detail, Figure 4 shows the entities involved in such a case. In particular, the combined eID/Privacy-ABC Issuer has the capability of issuing Privacy-ABCs. With this combination we achieve an interesting solution of high assurance on the identity of individuals through their eIDs and full anonymity when using a service. User anonymity is possible since the presentation token cannot be linked to the true identity of the user. This identity was established during her authentication at the eID server but, afterwards, Privacy-ABC tokens were used to transfer the information. Privacy-ABCs ensure the unlinkability between the issuance of the token and its usage through the presentation proof. For completeness, we would like to point out that in [Bjones10, BKPR12] a different architecture is proposed. The architecture is close to Scenario 4.3 as the eID server is treated as a separate entity from the Privacy-ABC issuer. However, the fast time to market of this architecture is achieved at the risk of the Privacy-ABC issuer altering the eID attributes during transformations.

Figure 5 shows the building blocks and components related to this eID use case. Compared to the generic Figure 2, the Issuer application encompasses the eID server. We suggest utilizing a validity attribute within the Privacy-ABC tokens, instead of revocation, to facilitate timely propagation of changes in the attributes or the eIDs revocation information.

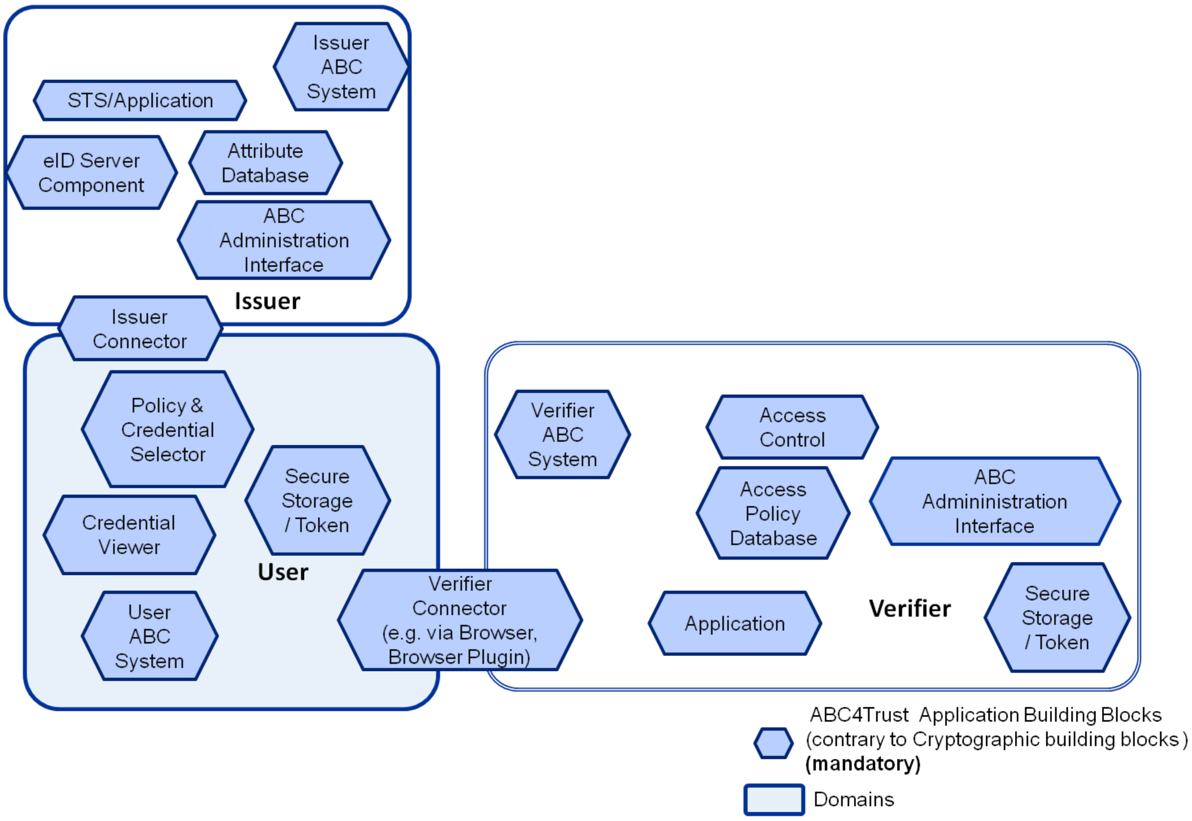


Figure 5 Building Blocks of the eID use case

Table 2 gives more details on the components.

Domain	Building Blocks of domain	Comment
User Domain	• User ABC System • Credential Viewer/Selector • Policy Viewer/Selector • Secure Key Storage • Issuer Connector • Verifier Connector	The user domain is a standard deployment.
Issuer Domain	• Issuer ABC System • STS Application • Issuer Connector • ABC Administration Interfaces	In this special case the issuer has an actual application logic that parses traditional eID tokens and issues Privacy-ABCs, which are then used by the user for the presentation.
Revocation Service	• ./.	Revocation checks will only happen during the authentication with the eID server. For this reason it is advisable to reduce the lifetime of the Privacy-ABC tokens.
Verifier Application	• Verifier ABC System	The verifier application is a standard ABC

	• User GUI • Application • XML Policy Generator • Database • Admin GUI	component.
Inspector	• ./.	Inspection is not needed in this scenario.

Table 2: Building blocks of the eID scenario

4.2 Scenario: Anonymous Participation in Decisions and Polls

4.2.1 Introduction of the Use Case

In times of low participation in election procedures and widespread political disinterest, an increase of people in governmental decision processes can be considered of an utmost value in itself. This includes opinion polling and decision making processes on any aspect of people's daily lives [ZwiJen13].

At the same time, the wide availability of inexpensive networked devices, such as smartphones, TV-sets or video games, bring a large part of the earth population online. A natural consequence of this is the idea to deploy electronic communication means and devices to trigger an increase of people's participation in decision making processes. The use of such means and devices may also result, as an added benefit, in the inclusion of people who are otherwise hindered to participate in these processes such as handicapped people as well as people who cannot afford to travel to the places where decision making processes are held. Other reasons that hinder participation include professional or family obligations as well as avoidance of the burden to travel long distances within a limited time period. All these deterrents could be removed if it was possible to hold decision making processes online.

However, the deployment of Privacy-ABCs for eVoting processes is not recommended. While some European Union Member States already have online elections, others are highly reluctant to deploy electronic processes for general elections. For example, the German Constitutional Court has set very high requirements regarding the transparency of the voting and counting processes as well as the verifiability of the results by independent observers.[18] These demands can barely be met by most e-voting machines and processes existing today. As political elections are the fundamental cornerstone of any democracy, the German law decision should be respected and taken as an opportunity to trigger further research in the eVoting with respect to transparency. While Privacy-ABCs may remedy some shortcomings of existing eVoting processes, it is not the aim of the ABC4Trust project to propose or develop a legally compliant eVoting mechanism.

Privacy-ABCs might prove more effective to approach European citizens for more democratic participation on a less ambitious level than national elections with non-critical decisions on particular topics relevant, for instance, to a municipality, an organization or a club. Polls and decision making that call for people's opinions towards issues set forth by a board of otherwise formally elected

[18] Bundesverfassungsgericht judgement of March 3rd 2009, 2 BvC 3/07, 2 BvC 4/07, See in particular reasons para. 105 et seq., German text of the judgement: http://www.bundesverfassungsgericht.de/entscheidungen/cs20090303_2bvc000307.html. English press release, available online: http://www.bundesverfassungsgericht.de/pressemitteilungen/bvg09-019en.html. The court ruled the use of Nedap voting machines unconstitutional due to the lack of publicity of the voting process (the publicity principle refers to the necessary transparency that allows observers to verify the correctness of the procedure).

representatives (e.g. a ministry) are already in place for influencing legislation with petitions or initiatives.[19]

Existing processes tend to require authentication of participants with their full identifying information. As the Users/Citizens do not know the details of operation of the polling system, it is not possible for them to know whether their particular opinion or vote is linked to their identity or whether some mechanism for anonymity is in operation. Even worse than the action of collecting identifying information from citizens, some systems can even publish the names of supporters of specific opinions, unless the voters actively opt for a pseudonym.[20]

4.2.2 Issues to solve

Making decisions online and participating in polls requires trust in the underlying mechanisms that they preserve the participants' anonymity and vote confidentiality. Moreover it is also necessary to ensure the equality of voters by preventing people from voting more than once. Some existing systems solve this latter problem by keeping clear text lists containing voter identifying information. This poses, however, the risk of linking people to vote. This is especially crucial where the mere act of participation can, potentially, reveal information on the opinion of the user. This can occur, for instance, in petitions where the only action allowed to participants is to support a single fact (the petition subject). In this case, every one listed as a participant is, automatically, known as a supporter of the petition question.

Also, existing solutions that allow anonymous participation in voting procedures often cannot verify that a voter has a particular attribute such as, for example, being member of a club or the management board of a company. This implies that the verification of this attribute can only be effected through the revelation of the full identity of the voter.

Another problem of anonymity systems in general – also for eParticipation systems – is the unknown size of the, so called, anonymity set. It is known that whether or not a person is identifiable depends, to a large extend, on whether the person is indistinguishable within a set containing other individuals. This set is called the anonymity set ([PfiHan10], p. 9). In the relevant literature, there is the concept of k-anonymity where k is the number of entities that share the same attributes as the examined entity (see e.g. [Sweeney02], p. 9; [EmaDan08], p. 628).[21] The larger the set is, the less likely it becomes that a particular user can be identified even if additional information is obtained and linked to the existing data sets. E-participation solutions, thus, must avoid storing information that could allow re-identification and links to other databases, e.g. time-stamps, birthdates, or ZIP-codes that could be used in connection with information stored in service or in log files of other data controllers to identify a participant.

[19] For instance, in the context of the European Citizens' Initiative' which is based on Article 11, Paragraph 4, citizens may propose opinions on issues of concern to the European Commission if they see a legal act of the Union as necessary. For such an initiative, at least one million signatories are required. Privacy-ABCs would allow collecting signatures anonymously.

[20] See, for instance, the privacy policy of the petition system of the German Bundestag, online: https://epetitionen.bundestag.de/epet/service.$$$.rubrik.datenschutz.html

[21] The concept of k-anonymity has been further refined in scientific literature with the concepts of l-diversity and t-closeness.

4.2.3 Advantages of Privacy-ABCs solutions in eParticipation

The special privacy needs of participation processes lead to a series of requirements which, we believe, can be fulfilled with Privacy-ABCs better than other mechanisms.

In this context, the participation in petitions, polls or surveys – whether they are organized by private entities or governmental agencies – should be possible even for emotionally debated topics such as abortion, same-sex marriage, or governmental measures infringing upon civil rights. This is an important issue since such controversial discussions in society have the potential of hindering citizens' will for participation in decision making. There is a variety of reasons for this such as, for instance, fear of potential identification by political opponents or negative consequences in life stemming from discrimination. Thus, an anonymity preserving eParticipation method is necessary to address such concerns and eliminate negative attitudes towards participation. As unlinkability is a key feature of Privacy-ABCs, these should be considered as a solution for those challenges for eParticipation systems.

While anonymity is necessary, it must still be ensured that Users may not participate more often than they are entitled to. Here, the scope exclusive pseudonym feature of Privacy-ABCs may be used. Within this scope, e.g. each separate poll or process, it is possible to see if a user accesses the service several times, thereby effectively preventing multiple votes.

Being able to participate from any place and not, necessarily, from a protected voting booth where anonymity and secrecy of the votes can be imposed, implies the risk of interfering with the participant's decision either through coercion or vote buying. As a countermeasure, the participant can change her vote as many times as she likes, until the end of the voting period with only the last vote taken into account. Here, the scope exclusive pseudonym feature of Privacy-ABCs may be used, allowing overwriting previous votes. Alternatively, Privacy-ABCs may be set up as one-show credentials, allowing a single or more previously specified number of uses (votes in our case) in an anonymous and unlinkable way.

Finally, to the issue of having a large anonymity set, Privacy-ABCs can offer solutions as well. The possible strict limitation to the necessary types of data, irrespective of other potentially linkable information that may be contained in typical source of credentials, allows reducing the revealed information and, consequently, enlarging the anonymity set. The possibility to provide proof of attributes such as, for instance, a proof that the user belongs to a certain age range instead of revealing the exact birthdate further supports the formation of large anonymity sets. For eParticipation instances where re-identification may be necessary, e.g. to modify votes, Privacy-ABCs require information that is only known to the user and can link the new action to the previous one. Thus, also in these cases linking of actions does not become easier for an attacker. In an ideal case, the anonymity set includes all persons eligible to participate irrespective of place of living, sex, birthdate, or other attributes typically contained in authentication tokens. Also, previous participation can neither be verified nor denied without the specific secret known only to the user. The size of the anonymity set then depends on the number of eligible persons – a size usually known before participation. With this feature, Privacy-ABCs also enhance the transparency due to the ability to estimate the anonymity set's size.

In summary these requirements may be seen as a protective wall preventing the organisation from identifying users or establishing a connection between users and their particular opinions stated in the poll. The eParticipation use case could take advantage of the Privacy-ABCs feature of anonymous authentication of attributes and unlinkability of the shown tokens to verify that a participant is eligible for participation.[22]

For a schematic view of an eParticipation system comprising the necessary features for secure and anonymous participation, see Figure 4 below. The dotted line forms a protective wall making it impossible for the participation system to learn the identity of the Users or even link a particular vote to a specific User. Whenever such linking is necessary, only the User has the required secret information that allows a valid re-authentication towards the system. Depending on the use case, the Verifier may also act as an Issuer, e.g. municipalities issuing state eIDs used for eParticipation on a local level, companies issuing eIDs for their employees which may, also, be used for participation in other decision bodies within the company.

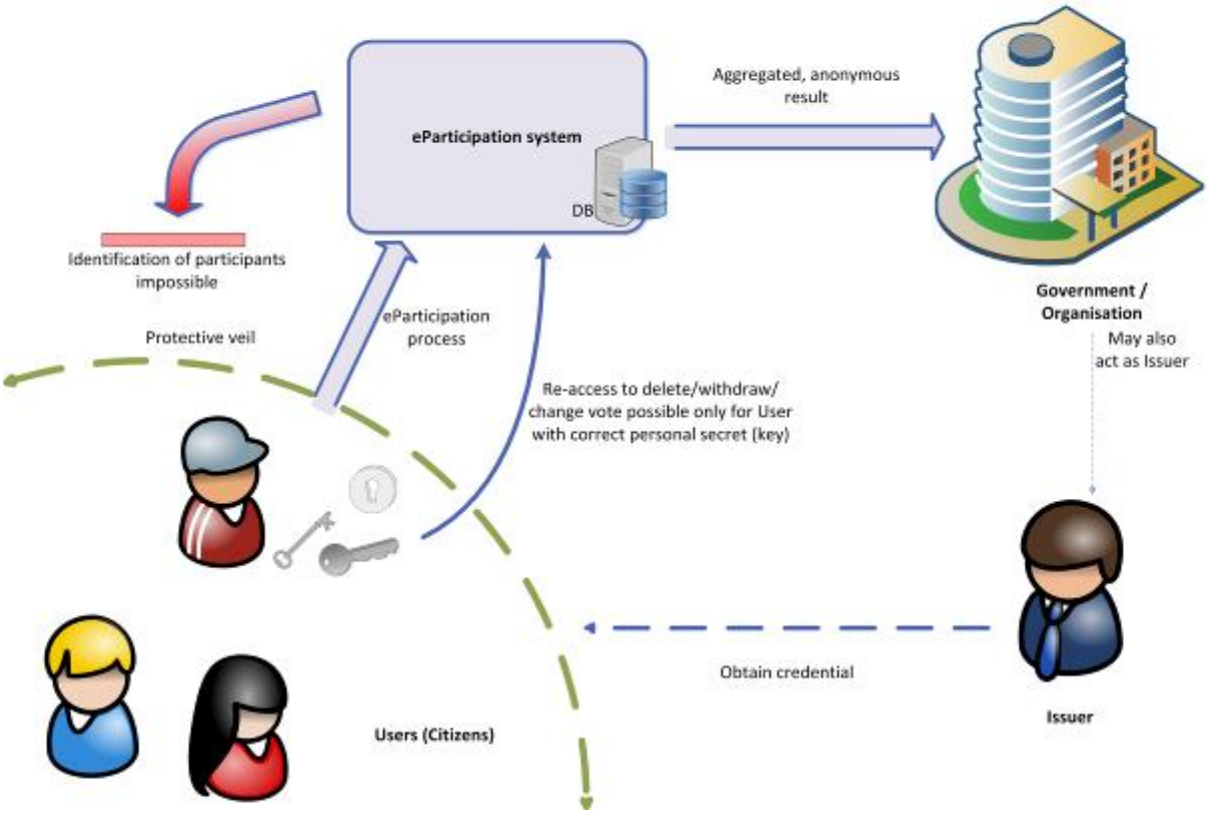

ePartiticipation system
DB
Aggregated, anonymous result
Identification of participants impossible
Government / Organisation
May also act as Issuer
ePartiticipation process
Protective veil
Re-access to delete/withdraw/ change vote possible only for User with correct personal secret (key)
Obtain credential
Issuer
Users (Citizens)

Figure 6 Overview eParticipation use case

The eParticipation scenario requires the following domains, with the corresponding building blocks:

Domain	Building Blocks of domain	Comment
User	• User ABC System • Credential Viewer/Selector	A standard User domain. A secure key storage is optional. For polls

[22] Alternatively the feature of anonymous one-show credentials which allow identifying the same credential which has been used more often than the allowed number of times. This has been proposed for eCoins under Identity Mixer but is currently not supported by the ABC4Trust architecture.

	• Policy Viewer/Selector • (Secure Key Storage) • Issuer Connector • Verifier Connector	and opinion gathering in clubs or smaller organisations storing credentials on the User's hard disk could be sufficient.
Issuer, optional	• Issuer ABC System • Provisioning • XML Policy • User GUI • Issuer Connector • Database • Admin GUI • Revocation Connector	A standard issuer domain, not necessarily part of the organisation in which the participation process takes place, e.g. where existing eIDs issued by another entity may be used. Whenever Users authenticate themselves through a role, e.g. membership in a decision making group, the issuer is a necessary entity to include.
Revocation Service, optional	• Revocation ABC System • Admin GUI • Revocation Connector	Whether revocation is necessary should be carefully examined. As most participation processes have a fixed deadline, revocation might be replaced with short expiration times for issued credentials. For the overall process, this may be a better compromise than risking the organisation excluding users, known to have an undesired effect on participation, by revoking their credentials.
Verifier	• Verifier ABC System • Voting Application with User GUI • Revocation Connector • XML Policy Generator • Verifier Connector	The verifier subsystem in the participation system needs to check that the User is entitled to participate and whether she already did so. If the User has already participated, it must be possible to recognize this fact so the polling application can either deny repeated access or allow previous input may be overwritten with the new input.
Inspector	• ./.	To ensure anonymity and untraceability of the participation, inspection must not be enabled unless the specific use case specifically requires for a possibility to identify participants. An example would be mandatory transparency guidelines for bodies of public entities demanding the revelation of an individual's voting behaviour after a certain time period.

Table 3: Building Blocks for the eParticipation scenario.

The modification of the previously introduced generic overview of the building blocks (Figure 2) with the use case specific considerations results in Figure 7 below. As inspection is not necessary, all related building blocks are excluded. Whether revocation is useful or necessary depends on the specific use case and, in particular, on the associated risks in case of an abuse of credentials and the duration of the credentials' validity:

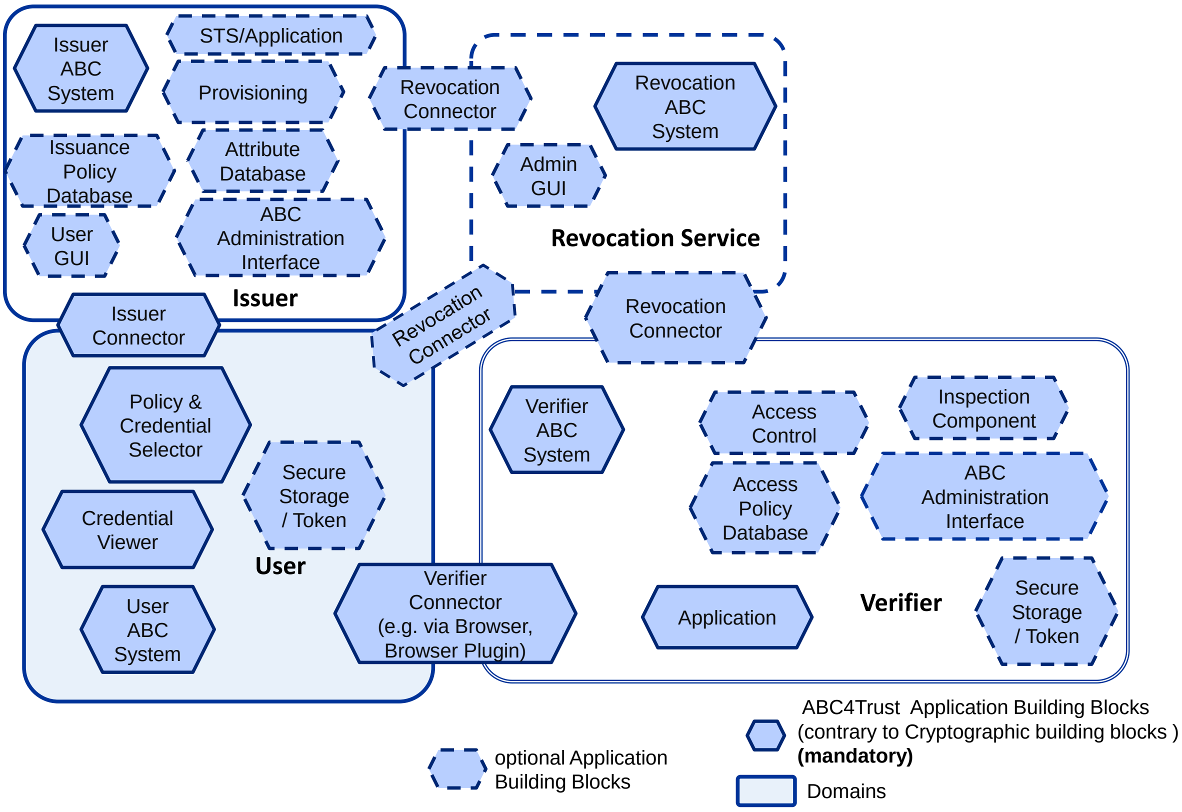


Figure 7 Building blocks for eParticipation use case

4.2.4 Summary of key points

The electronic eParticipation use case points to possibilities of how Privacy-ABCs may be deployed for enhancing participation in eParticipation systems. It demostrtates requirements that need to be fulfilled, in particular with respect to anonymous and pseudonymous but yet properly authenticated participation. In comparison to, e.g., existing national eIDs with privacy-preserving features, Privacy-ABCs offer the advantage of verification of other attributes not typically contained in these eIDs such as memberships and profession. Such information may be obtained by other attribute providers that act as Issuers for these attributes. However, while Privacy-ABCs can provide anonymity, other necessary features of eParticipation systems such as easily understandable realisation of the voting process, full transparency of the counting procedure and verifiability of the result must be handled by the eParticipation system itself, outside the privacy-ABCs framework.

4.3 Scenario: Bank as Identity Service Provider

4.3.1 Introduction of the Use Case

Financial institutions are normally trusted sources of information about their customers since it is crucial that this information is accurate and up-to-date. Therefore, the idea of having financial institutions (e.g. banks) as Identity Service Providers is being, actively, discussed the last decade[23]. The implementation of such a scenario, gives the opportunity to service providers to rely on the information provided by the financial institutions and delegate the authentication process to them. Figure 8 demonstrates the scenario of accessing a Job Search Portal that relies on the user's bank for identity information:

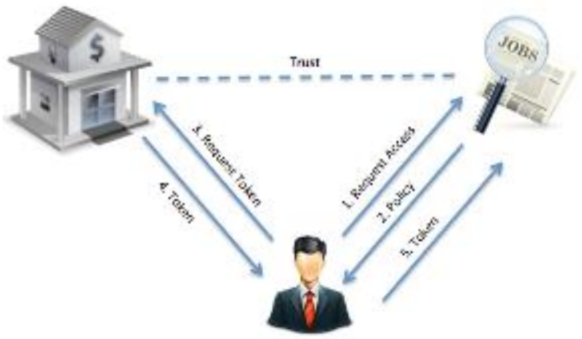


Figure 8 Bank as Identity Service Provider without Privacy-ABCs

4.3.2 Issues to Solve

A very important factor for financial institutions in evaluating their customers' credibility and ability to meet their financial obligations is their job status. Now, imagine Bob loses his job and goes to the job search portal to look for appropriate job positions. The portal requires Bob to login via his bank using a typical federated identity infrastructure allowing the portal to acquire proofs about Bob's identity. The bank learns about the contact with the job portal. Due to this transaction, the bank might suspect that something has happened with Bob's career and he is now looking for a new job. Therefore, this extra information can negatively impact Bob's credibility assessment for his next loan application with the bank.

4.3.3 Advantages of a Privacy-ABC solution

Deploying Privacy-ABCs can easily resolve this issue since the two phases of Issuance and Presentation of the credentials are unlinkable. As a result, the Identity Service Provider would not learn where the user shows his authentication tokens and which services he is accessing. At the same

[23] E.g. Austria and Sweden

time, the Relying Party makes sure that it is receiving authentic claims issued by the corresponding Issuer. In our scenario, as it is shown in Figure 9, Bod can obtain Privacy-ABCs from his bank and later use them to authenticate himself towards the job search portal. In this case, Bob’s bank will not be involved in the later phase and, therefore, will not learn about the fact that Bob is looking for a new job.

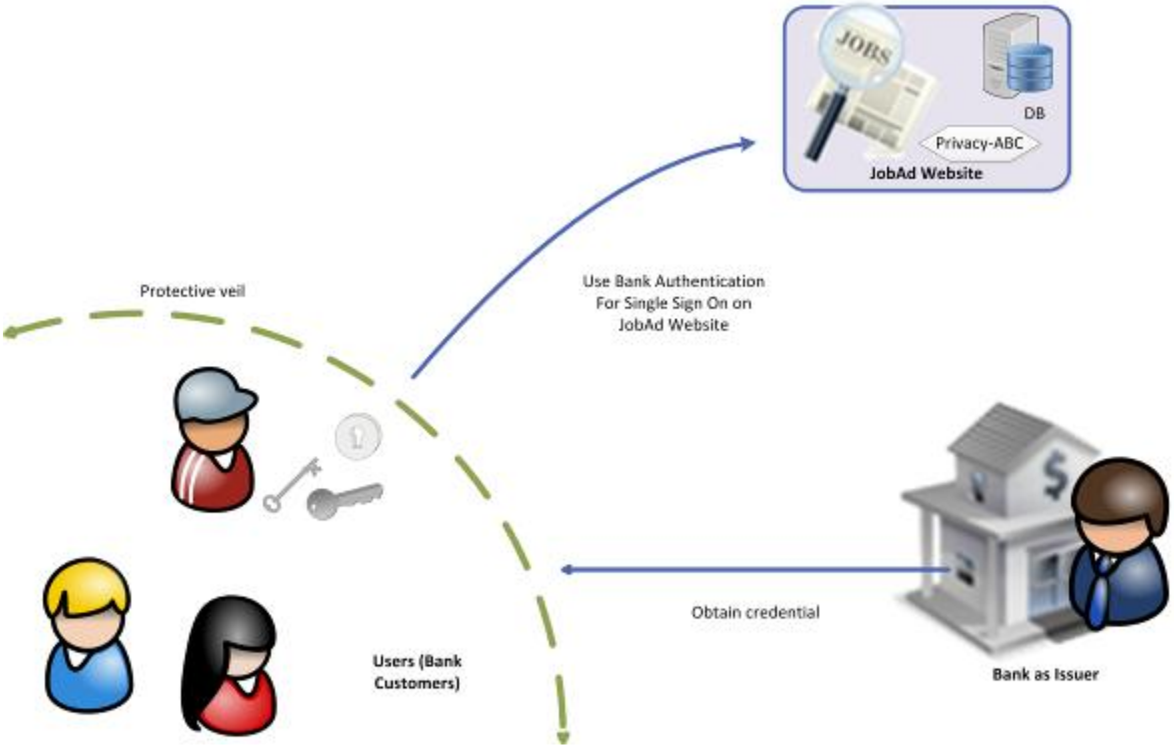


Figure 9 Bank as Identity Service Provider using Privacy-ABCs

The Bank as Identity Service Provider is a straightforward authentication example without any special subcases. Users obtain identity credentials from the bank and use them to authenticate themselves to the verifiers. The credentials can become invalid (revoked), but inspection is not needed.

The Bank as Identity Service Provider scenario requires the following domains with the corresponding building blocks:

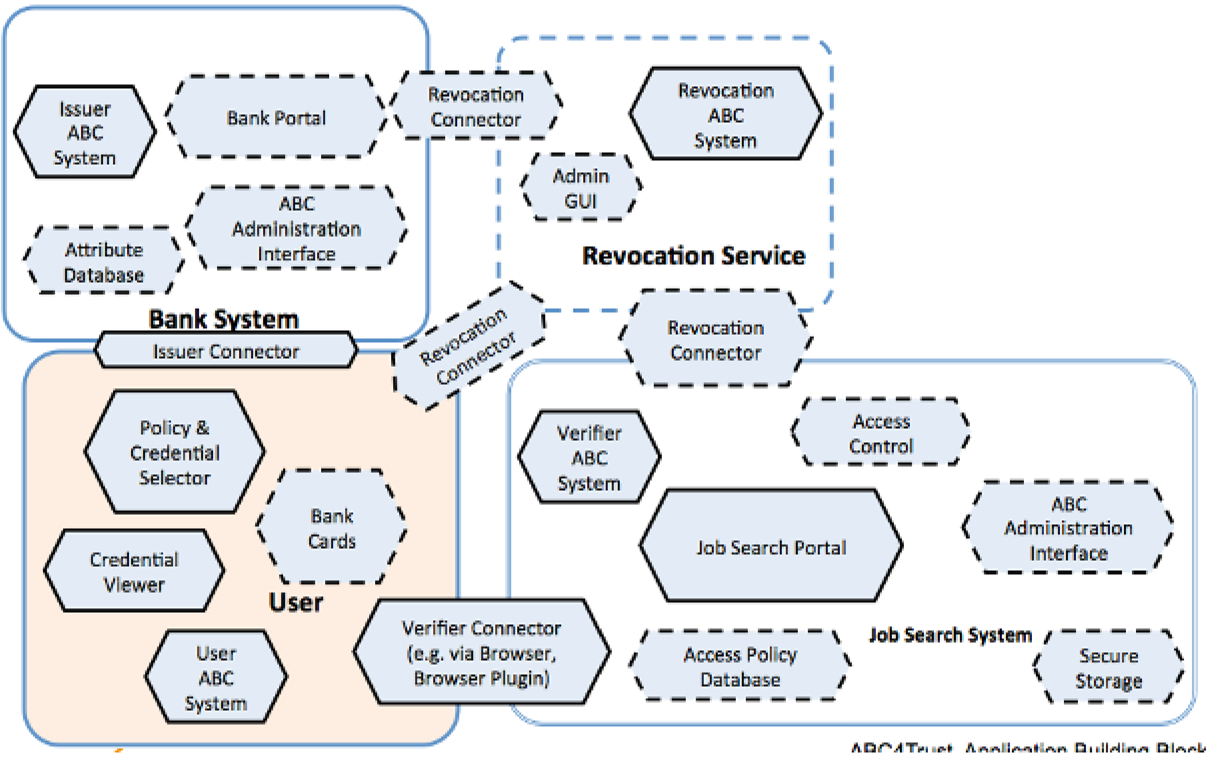


Figure 10 Building Blocks Bank as Identity Provider

Comparing this figure with the general high-level architecture model provided earlier in this document, one can observe that the role of Inspector is completely removed from this architecture since it is not needed in this scenario. Therefore, all the corresponding components (e.g. Inspector Connector) are removed. With regard to the Issuer box, we assume that the Bank Portal could already offer the functionality of the Provisioning and User GUI components existing in the general model. Furthermore, the Bank does not need to act as Secure Token Service provided and, thus, we do not need the STS component. Last, but not least, in our scenario every customer of the bank could receive credentials that she can use to authenticate to the other service providers and there is no prerequisite for that. As a result, the Issuance Policy Database can be omitted. On the Verifier's side, the Job Search Portal represents the Application.

Domain	Building Blocks of domain	Comment
User	• User ABC System • Credential Viewer/Selector • Policy Viewer/Selector • Bank Cards (Secure Key Storage) • Issuer Connector • Verifier Connector • Revocation Connector	The user domain is a standard deployment. The ATM card of the users could be made more advanced to support Privacy-ABCs cryptographic operations.
Bank, as Issuer	• Issuer ABC System • Bank Portal • Attribute Database • ABC Administration	Normally banks hold a database of certified attributes accessible by their customer portal application. The whole system can be patched by Privacy-ABC modules to enable privacy preserving authentication.

	Interface • Revocation Connector • Issuer Connector	
Revocation Service, optional	• Revocation ABC System • Admin GUI • Revocation Connector	It is possible that the bank customers lose their cards. Therefore revocation mechanisms are necessary.
Job Search System, Verifier	• Verifier ABC System • Revocation Connector • Job Search Portal • Revocation Connector • Secure Storage • Access Policy Database • ABC Administrative Interface • Access Control	The Job Search Portal offers the full functionality of the domain application and its access control system is equipped with a set of access policies, which Privacy-ABCs modules should consider as the basis in their verification process.
Inspector	• ./.	Inspection is not needed in this scenario.

Table 4: Building Blocks for do not track users.

With the use of Privacy-ABCs, which prevent the Issuer from learn about the habits of the user, the privacy problem can be addressed. The overall architecture requires the same number of involved parties as a typical federated ID solution. This use case illustrates that the Privacy-ABC solution may be very elegant and efficient. While the architecture allows a broad class of functionalities, these are not necessary for this use case, which is then reduced to containing a minimum number of necessary ABC-specific building blocks.

4.3.1 Conclusion of the use case

With the help of Privacy-ABCs, institutions that possess authentic sources of personal information can play the role of the Identity Service Provider without jeopardising the users' privacy and raising theis concerns of being profiled or negatively regarded by the institutions from their service usage habits.

4.4 Scenario: Do not Track Relying Parties (prevented by translation service)

4.4.1 Introduction to Use Case

In today's Internet Ecosystem of services built on top of other services, such as a picture sharing applications or online gaming, third party services sometimes rely on the authentication of the main web service (e.g. Facebook) while being integrated into the look and feel of the Users' Facebook page. For users, this convenient single-sign-on has advantages, but a small company's business might be threatened.

Today, the wish to customize the service's web sites according to the visiting user's needs often requires an optimization of the site's accessibility and usability. This includes the modus operandi of login to the services offered to the customer. To enable a smooth and simplified user experience, it is therefore desirable for any service provider to offer the possibility of using just one or at least few login interfaces to obtain access to various services. Therefore, in the current ecosystem of digital services and goods, the offer of an easily implemented single-sign-on function through market-leading and popular service providers (like Facebook, Google, etc.) is often taken up. However, such a monopoly of log-in-access may pose certain competition, privacy-compliance, related issues which require attention. In the presentation of this use case, we will explore how Privacy ABC's can support the goal of a user-friendly login-function while still preserving the privacy of the user in a satisfactory way.

4.4.2 Issues to Solve

Let's assume, in the scenario, that a small company depends on a bigger company providing a single-sign-on functionality. The small company may or may not pay a fee for this authentication service. If the service is innovative, the small company will, most probably, have many users. The service may be easy to provide, but the big company does not see see the innovation. Thus, the small company can make a good business at, virtually, no additional cost. With current technologies, the big company will be able to notice how well the small company's business works, by seeing the number of authentications it performs for the small company. It might even be possible to infer the stickiness[24] of the small company's website by the number of repeated authentications or attributes shown. Moreover, the involvement of the larger company poses a threat to the users of the smaller company's services, because their attributes are revealed, resulting in a revelation of personal information. Such a revelation may encompass not only the user's identifying information, but may also uncover details about the services requested by the user. All of these aforementioned issues results in a fear of the small company, which is dependent on the big company, that the big company may use the obtained information in some way, eventually becoming a competitor. Finally, the wider issue of reduced innovation due to an uncertain business environment, also, exists. Since many innovations build on top of each other, the issue to solve in this scenario is an important one.

4.4.3 Advantages of a Privacy-ABC solution

The Privacy-ABC technology with its unlinkability can solve the issues described above. However, this has a certain precondition: in a basic setup phase, the big company would need to adopt the Privacy-ABCs technology. But such a basic setup may not be taken, for at least two reasons: a) additional implementation costs will be incurred on the big company, with no added value, and b) the big company will no longer be able to process and analyse the obtained information routed through its own systems, for its own purposes, Therefore, it is to be assumed that these reservations of the big company pose a major obstacle to the implementation of the Privacy-ABCs technology, despite its obvious benefits for the privacy of the users as well as for the smaller company.

Hence, this scenario will be vieaed with a slightly more complicated setup, in a more realistic business background. Instead of the big company implementing the service, a third party will implement a kind of "translation service". This party will be an entity sitting as an intermediary between the smaller and the larger company. As such, this entity can implement Privacy-ABCs system and translate user information into privacy-preserving credentials which are, then, presented to the larger company.

This new business model of the translation service is to provide privacy for the users as well as business-related confidentiality for the smaller service provider. The translation service will most likely charge a fee in return for the service and, therefore, a useful economic model is required for it. The trust model, with respect to correctness of identity proofs, dictates that the translation service is trusted by the small company as well as the big company (e.g. both are trusted not to generate fake credentials). Still, additional safeguards could eventually be needed to enable the confirmation of this trust, because this third party will become an additional entity learning the user's personal information. For the big company, the translation service will appear like a successful small company. But further evaluation of information, especially aimed at the access to user's personal data as well as an

[24] Stickiness is the time a user spends on a website or page, the higher the stickiness the more valuable a page for advertisement is.

assessment which of the small companies behind the translation service is the most successful, will be effectively hindered.

However, this model depends on the big company accepting the translation service as a Relying Party. Since this is not, initially, a desirable action for the big company, the question remains of how such an interaction model can be established. Eventually, further legislative progress, especially in the fields of data protection and competition law, might be required to hinder the exploitation of market power and monopoly position by big companies. But despite these difficulties, the establishment of such an intermediate attribute translation service might be an opportunity to introduce Privacy-ABCs into this market and, in the process, enhance the privacy of users while at the same time protecting the interests of small and medium enterprises (SME's) offering digital services and goods.

Figure 11 shows the full communication structure for both the big company's traditional IDM and the small company, being protected by the translation service. The social networking service, normally providing the information directly to the small service provider, now interacts with the translation service. The small service provider interacts, and trusts, the translation service which protects their privacy towards the social networking service.

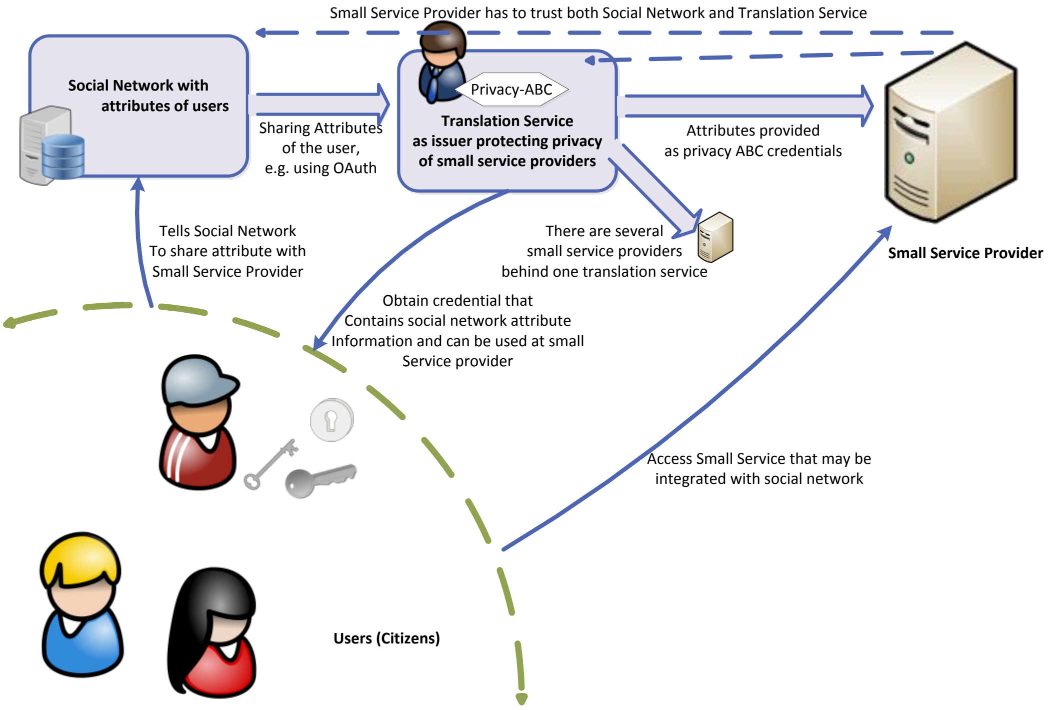

Small Service Provider has to trust both Social Network and Translation Service
Social Network with attributes of users
Sharing Attributes of the user, e.g. using OAuth
Privacy-ABC
Translation Service as issuer protecting privacy of small service providers
Attributes provided as privacy ABC credentials
Small Service Provider
Tells Social Network To share attribute with Small Service Provider
There are several small service providers behind one translation service
Obtain credential that Contains social network attribute Information and can be used at small Service provider
Access Small Service that may be integrated with social network
Users (Citizens)

Figure 11 Scenario Do not Track Relying Parties

Finally it is worth observing that, although being in the centre, the translation service cannot disrupt the privacy of the users, as they act as issuers for Privacy-ABCs and such issuers cannot profile their users.

From the perspective of implementation the do-not-track-the-relying-party scenario may appear more complicated, as there are two parties that know about the users, but in its core it is a basic attribute issuing use case. The third party provider is a special form of issuer, that does not have an own attribute database, but translates the information that it receives from others. Figure 12 shows this relation by incorporating the social networking side as an extra domain on the left of the Issuer. It is worth noting that a current enhancement for the German eID [BKPR12, Bjones10] utilizes this approach to avoid organisational integration with the eID servers. The drawback of this approach has been discussed in the eID scenario, in Section 4.1.

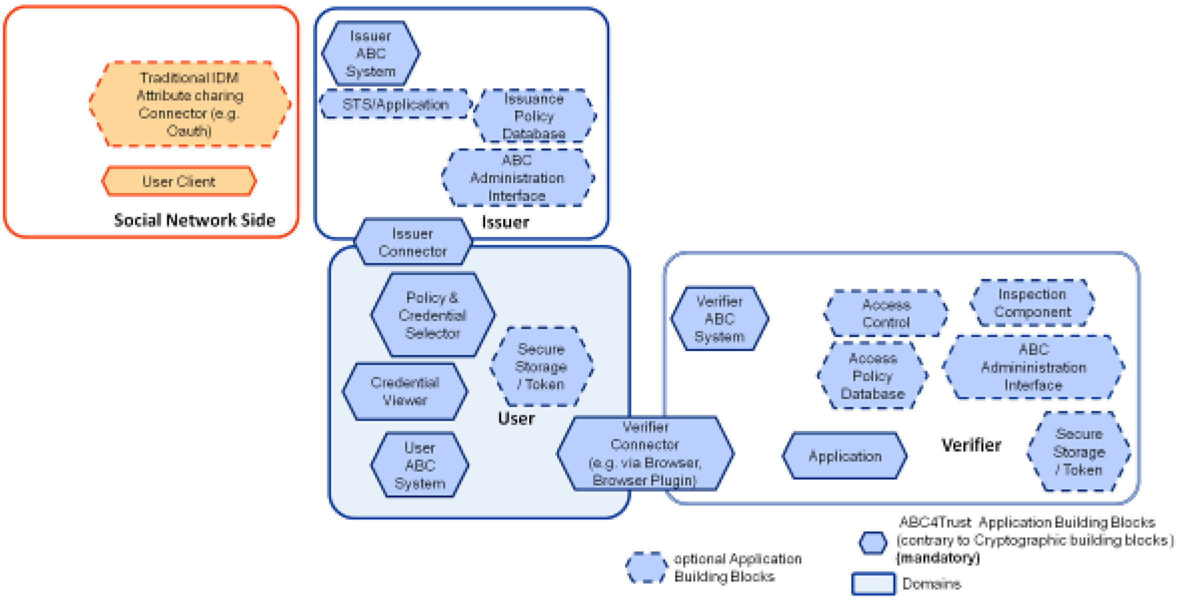

Traditional IDM Attribute charing Connector (e.g. Oauth)
User Client
Social Network Side
Issuer ABC System
STS/Application
Issuance Policy Database
ABC Administration Interface
Issuer
Issuer Connector
Policy & Credential Selector
Secure Storage / Token
Credential Viewer
User
User ABC System
Verifier Connector (e.g. via Browser, Browser Plugin)
Verifier ABC System
Access Control
Inspection Component
Access Policy Database
ABC Administration Interface
Application
Verifier
Secure Storage / Token
ABC4Trust Application Building Blocks (contrary to Cryptographic building blocks) (mandatory)
optional Application Building Blocks
Domains

Figure 12 Building Blocks of Relying Party Privacy Protection by Third Party

The building blocks in Figure 12 differ from the full set of building blocks, as shown earlier in Figure 2, in that neither inspection nor revocation are necessary. The translation service will want to set the validity of the credentials to a value that matches the validity given by the social network (original attribute provider). The original attribute provider is included in this diagram, to highlight that is it is connected to the issuer, rather than the verifier or user directly.

The scenario requires the following building blocks of the domains:

Domain	Building Blocks of domain	Comment
User Domain	• User ABC System • Credential Viewer/Selector • Policy Viewer/Selector • Secure Key Storage • Issuer Connector • Verifier Connector	The user domain is a standard deployment.
Issuer Domain	• Issuer ABC System • STS Application • Issuer Connector • ABC Administration Interfaces	In this special case the issuer has an actual application logic (the STS application) that on the one hand parses the traditional IDM attributes and issues respective credentials. There is no need for provisioning or an attribute database.
Revocation Service	•	Revocation service is not applicable.
Verifier Application	• Verifier ABC System • User GUI • Application • XML Policy Generator • Database • Admin GUI	The verifier application is a standard ABC component. If the verifier requires revocation, a second issuer that knows whether a user account is still valid is necessary.

Inspector	- ./.	Inspection is not applicable in this scenario.

Table 5: Relying Party Privacy Protection by Third Party

4.4.4 Summary of key points

The internet businesses often consist of a big business attracting many users on their website and smaller companies that provide additional or specialised services. The big company will be an identity and attribute provider, but it is also is in a position to know which service is most successful. A very competitive big company might even choose to provide similar services as the small company in an effort to increase revenue.

To prevent the analysis of the success of the small companies (relying parties) the Privacy-ABCs technology will be introduced. A simple scenario might have the big company to become an issuer. In cases where the big company is not willing to do that, it is also possible that a third party provides the translation and privacy protection service itself.

5 Conclusion

This amendment presented the high level building blocks from the perspective of a system architect or regulator. Various scenario assumptions were presented, such as the need for encryption and data minimization or that privacy of the verifier should also be protected. In the scenario sections we presented and discussed the following issues:

- Unlinkeability of users across different service providers (Verifiers)
- Minimization of revealed data, so that the status of a user cannot influence her eParticipation
- Being able to ensure users are able to do what they are entitled to, but not more
- Attribute provider (Issuer) not being able to track users and, especially, the services they use
- Attribute provider (Issuer) not being able to estimate e.g. the popularity of services, by tracking the amount of users or their interaction with the service

Privacy-ABCs provide a good privacy protection framework and can enhance existing Identity Provider Systems, such as eIDs, Social Network based identities or basic access control schemes. The deployment of Privacy-ABCs requires several roles in the Ecosystem to be present:

- Privacy-ABCs Issuer, that provides Privacy-ABCs to the users, which contain their attributes
- User, the entity wishing to use a service
- Verifiers, the service providers, that rely on the issued Privacy-ABCs

Optional roles follow below:

- Revocation Authority, that can invalidate attributes, e.g. for users that leave a company, or whose driving license is withdrawn
- Inspector, in cases where the user can interact with a system in a malicious way, or an emergency happens, the identity or single attributes might be revealed.[25]

The different actors implement the high level building blocks of their respective domain. They do so independently, but require exchanging information during the integration of the system. For example the issuer parameters and credential specification have to be known by all domains as soon as possible. Credentials specifications need to enable the required presentation policies. And, depending on the scenario, also revocation authority and inspector parameters need to be generated and distributed.

It can be said, that the easy creation of the Ecosystem with its actors is facilitated by the Privacy-ABCs components provided by the ABC4Trust project to the public domain (e.g. at the Github repository). The Privacy-ABCs take over the credential handling and cryptographic computations. For example, a verifier can provide the Privacy-ABCs engine with the presentation policy and connection to the user wanting to gain access and ther result of the interaction is whether access can be granted. The user domain, on the other hand, includes a policy viewer and a credential selector that facilitates meeting the required data-minimization requirements.

There are also elements such as inspection that require, besides the technical components, proper processes and legal contracts in place. Privacy-ABCs provide the tool set and mechanisms to inform

[25] The user has to be informed of the possibility of inspection on a per inspectable-attribute basis. This has to be done using the presentation policy.

the user, while the administrative and legal requirements have to be identified for each scenario. The presented scenarios do not provide guidance on the balance between privacy and protection in cases of emergencies. This cannot be considered a drawback of the technology, but highlights the difficulty of the involved issues – no generic solution is possible for inspection, for instance.

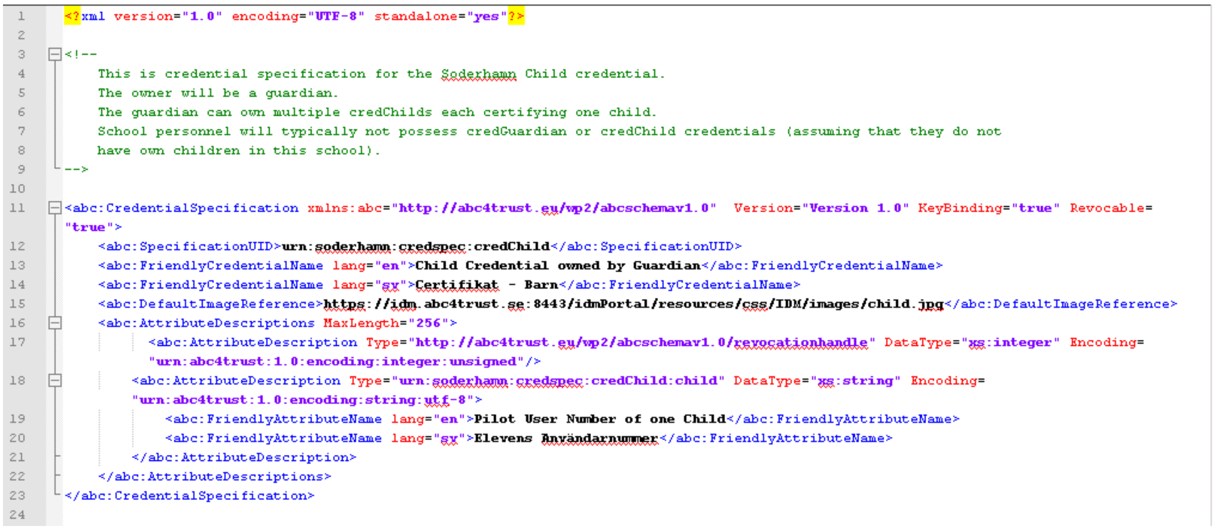
An attentive reader might have noticed that the scenarios discussed in Chapter 4 have different protection goals and, in each case, a single Privacy-ABC issuer. Today's traditional IDM system have several identity providers and the Privacy-ABCs technology can do the same. However, we deliberately left out issues that are connected to the user domain components being generic (i.e. provided by an independent party) and not matched to the specific verifier scenario. This research was also not so much a goal of the ABC4Trust research as it was for an earlier project, Primelife [FHZNHKGPH11], which focused on the usability aspects extensively. ABC4Trust's aim is to enable the actors in the Ecosystem to implement, integrate and operate the Privacy-ABCs system with ease and independently of the underlying the cryptographic primitives. This can be considered achievable by the high level building blocks presented in this document without the need for talking about specifics of U-Prove, Idemix or other Privacy-ABC technologies.

A Examples of required XML files

This appendix shows examples of policies, as needed by the different scenarios.

1) Credential Specification

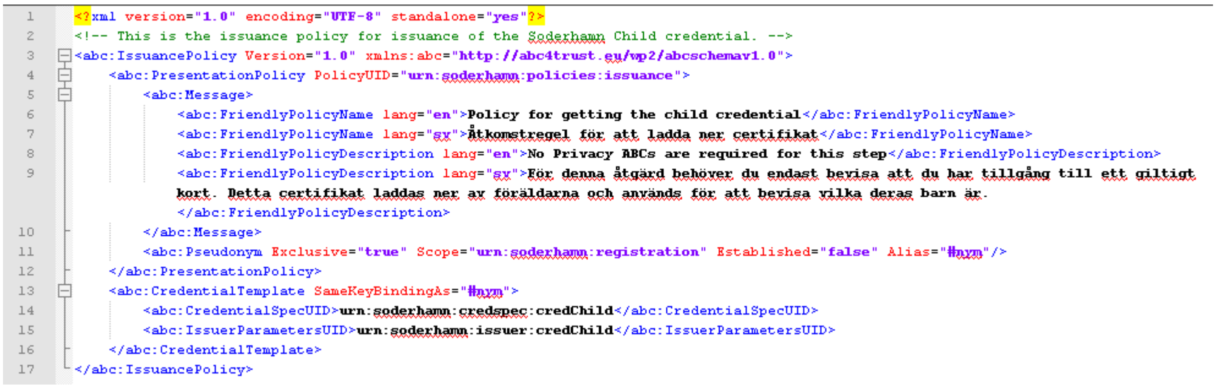
The Credential specification requires the human readable values that are presented to the users and a data type. There are different data types, a description can be found at D2.1 [CKLN11]



```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<!--
    This is credential specification for the Soderhamn Child credential.
    The owner will be a guardian.
    The guardian can own multiple credChilds each certifying one child.
    School personnel will typically not possess credGuardian or credChild credentials (assuming that they do not
    have own children in this school).
-->

<abc:CredentialSpecification xmlns:abc="http://abc4trust.eu/wp2/abcschemav1.0"  Version="Version 1.0" KeyBinding="true" Revocable=
"true">
    <abc:SpecificationUID>urn:soderhamn:credspec:credChild</abc:SpecificationUID>
    <abc:FriendlyCredentialName lang="en">Child Credential owned by Guardian</abc:FriendlyCredentialName>
    <abc:FriendlyCredentialName lang="sv">Certifikat - Barn</abc:FriendlyCredentialName>
    <abc:DefaultImageReference>https://idm.abc4trust.se:8443/idmPortal/resources/css/IDM/images/child.jpg</abc:DefaultImageReference>
    <abc:AttributeDescriptions MaxLength="256">
          <abc:AttributeDescription Type="http://abc4trust.eu/wp2/abcschemav1.0/revocationhandle" DataType="xs:integer" Encoding=
          "urn:abc4trust:1.0:encoding:integer:unsigned"/>
        <abc:AttributeDescription Type="urn:soderhamn:credspec:credChild:child" DataType="xs:string" Encoding=
        "urn:abc4trust:1.0:encoding:string:utf-8">
            <abc:FriendlyAttributeName lang="en">Pilot User Number of one Child</abc:FriendlyAttributeName>
            <abc:FriendlyAttributeName lang="sv">Elevens Användarnummer</abc:FriendlyAttributeName>
        </abc:AttributeDescription>
    </abc:AttributeDescriptions>
</abc:CredentialSpecification>

```

2) Issuance Policy

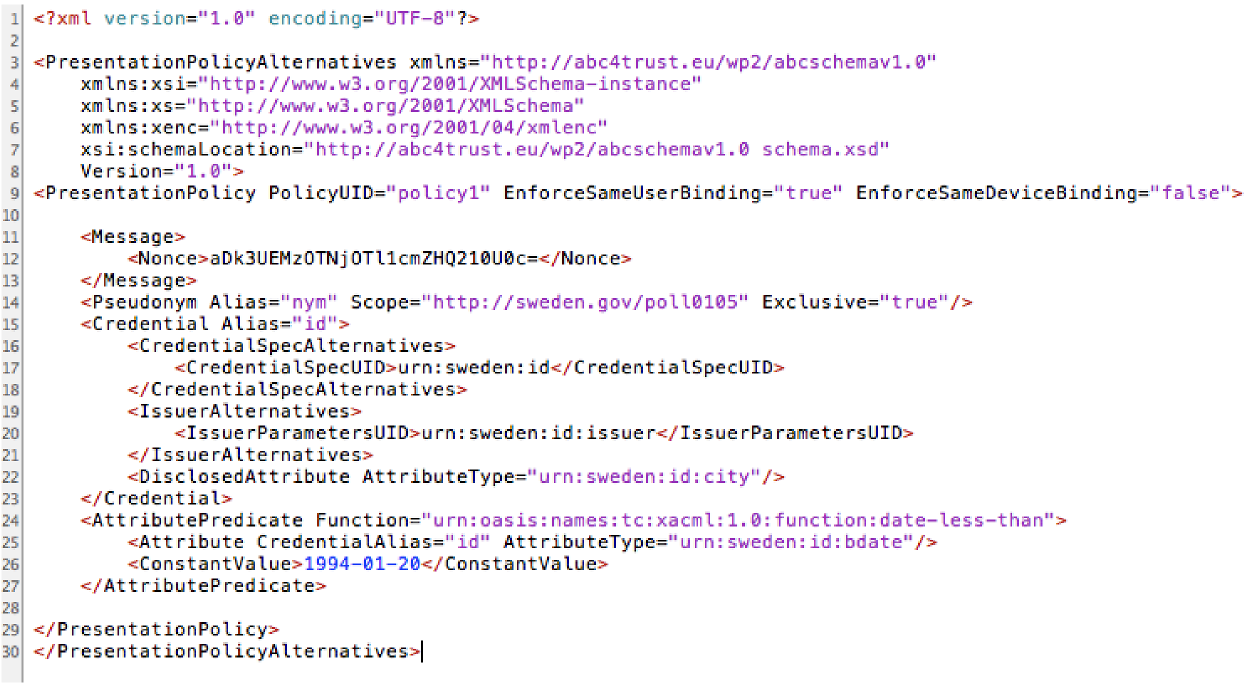
The Issuance Policy links the credential specification to the credential parameter. In the example below no requirements are needed for a user to retrieve this credential. D2.1 [CKLN11] has more technical details.



```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- This is the issuance policy for issuance of the Soderhamn Child credential. -->
<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 for getting the child credential</abc:FriendlyPolicyName>
            <abc:FriendlyPolicyName lang="sv">Åtkomstregel för att ladda ner certifikat</abc:FriendlyPolicyName>
            <abc:FriendlyPolicyDescription lang="en">No Privacy ABCs are required for this step</abc:FriendlyPolicyDescription>
            <abc:FriendlyPolicyDescription lang="sv">För denna åtgärd behöver du endast bevisa att du har tillgång till ett giltigt
            kort. Detta certifikat laddas ner av föräldarna och används för att bevisa vilka deras barn är.
            </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:credChild</abc:CredentialSpecUID>
        <abc:IssuerParametersUID>urn:soderhamn:issuer:credChild</abc:IssuerParametersUID>
    </abc:CredentialTemplate>
</abc:IssuancePolicy>
```

3) Presentation Policy

The presentation policy shown below is the policy like it is send to the user client. On the hard disk of the issuer, there might be a template stored, that is automatically completed (or combined) upon the interaction with the user. For example the nonce should not be hardcoded. D2.1 [CKLN11] describes presentation policy format.



```xml
<?xml version="1.0" encoding="UTF-8"?>

<PresentationPolicyAlternatives xmlns="http://abc4trust.eu/wp2/abcschemav1.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc"
    xsi:schemaLocation="http://abc4trust.eu/wp2/abcschemav1.0 schema.xsd"
    Version="1.0">
<PresentationPolicy PolicyUID="policy1" EnforceSameUserBinding="true" EnforceSameDeviceBinding="false">

    <Message>
        <Nonce>aDk3UEMzOTNjOTl1cmZHQ210U0c=</Nonce>
    </Message>
    <Pseudonym Alias="nym" Scope="http://sweden.gov/poll0105" Exclusive="true"/>
    <Credential Alias="id">
        <CredentialSpecAlternatives>
            <CredentialSpecUID>urn:sweden:id</CredentialSpecUID>
        </CredentialSpecAlternatives>
        <IssuerAlternatives>
            <IssuerParametersUID>urn:sweden:id:issuer</IssuerParametersUID>
        </IssuerAlternatives>
        <DisclosedAttribute AttributeType="urn:sweden:id:city"/>
    </Credential>
    <AttributePredicate Function="urn:oasis:names:tc:xacml:1.0:function:date-less-than">
        <Attribute CredentialAlias="id" AttributeType="urn:sweden:id:bdate"/>
        <ConstantValue>1994-01-20</ConstantValue>
    </AttributePredicate>

</PresentationPolicy>
</PresentationPolicyAlternatives>
```

6 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

Within Restricted Areas, in particular in Chats and Discussion boards, Users are represented by a self-chosen nickname, their alias. Each alias can be chosen only once. The alias will be bound to the User credential while preserving unlinkability allowing the User to reclaim the alias for subsequent visits.

Certified pseudonym

A verifiable pseudonym based on a 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.

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) 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 Verifier, 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.

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 the IdM application and IdM Portal to grant access to those Users that satisfy the access policy for a given restricted area. The Issuer that this Verifier trusts is the school administration office – which is the only issuer within the pilot.

7 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
GUI	Graphical User Interface
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure (HTTP secured by TLS or SSL)
ID	Identifier
Idemix	IBM Identity Mixer
IdM	Identity Manager
ISP	Internet Service Provider
PC	Personal Computer
RA	Restricted Area
RP	Relying Party
SC	Smart Card
SSL	Secure Sockets Layer
STS	Secure Token Service
TTP	Trusted Third Party
TLS	

	Transport Layer Security
XML	eXtensible Markup Language

8 Bibliography

[Art29WP100] Article 29 Working Party, “Opinion on More Harmonised Information Provisions, 25 November 2004”, Working Paper 100, 2004, online:
http://ec.europa.eu/justice/policies/privacy/workinggroup/wpdocs

[ABDG12] (D6.2) Jörg Abendroth, Souheil Bcheri, Kasper Damgaard, Hamza Ghani, Jesus Luna, Gert Læssøe Mikkelsen, Maxim Moneta, Monika Orski, NeerajSuri, Harald Zwingelberg, "D6.2 Necessary hardware and software package for the school pilot deployment", ABC4Trust project deliverable, Frankfurt, 2012, online:
https://abc4trust.eu/download/ABC4Trust-D6.2.Hard-and-Software-Package-for-School-Pilot.pdf

[ALPRSSSZ12] (D7.1) Joerg Abendroth, Vasiliki Liagkou, Apostolis Pyrgelis, Christoforos Raptopoulos, Ahmad Sabouri, Eva Schlehahn, Yannis Stamatiou,Harald Zwingelberg “D7.1 Application Description for students”, online:
https://abc4trust.eu/download/ABC4Trust-D7.1-Application-Description-Students.pdf

[BGL12] (D5.1) Souheil Bcheri, Norbert Götze, Vasiliki Liagkou, Apostolis Pyrgelis, Christoforos Raptopoulos, Yannis Stamatiou, Katalin Storf, Peder Wängmark, Harald Zwingelberg, ABC4Trust Deliverable “D5.1 Scenario Definition for both Pilots”, Frankfurt 2012, online: https://abc4trust.eu/index.php/pub

[BGOZ12] (D6.1) Souheil Bcheri, Norbert Goetze, Monika Orski, Harald Zwingelberg “D6.1 Application Description for the school deployment”, online:
https://abc4trust.eu/download/ABC4Trust-D6.1-Application-Description-School.pdf

[Bjones10] R. Bjones, "eParticipation Scenario Reference Guide," Microsoft, Tech. Rep., October 2010.

[BKPR12] R. Bjones, I. Krontiris, P. Paillier, K. Rannenberg, "Integrating Anonymous Credentials with eIDs for Privacy-respecting Online Authentication", Annual Privacy Forum 2012, 10-11 October 2012, Limassol, Cyprus.

[CaLy02]
Jan Camenisch and Anna Lysyanskaya. “Dynamic accumulators and application to efficient revocation of anonymous credentials.” CRYPTO 2002, volume 2442 of Lecture Notes in Computer Science, pages 61-76. Springer, 2002

[CKLN11] (D2.1) Jan Camenisch, Ioannis Krontiris, Anja Lehmann, Gregory Neven, Christian Paquin, Kai Rannenberg, Harald Zwingelberg, "D2.1 Architecture for Attribute-based Credential Technologies – Version 1", ABC4Trust project deliverable, Frankfurt, 2011, online:
https://abc4trust.eu/index.php/pub/107-d21architecturev1

[DGGL12] (D7.2) Kasper Damgaard, Hamza Ghani, Norbert Goetze, Anja Lehmann, Vasiliki Liagkou, Jesus Luna, Gert Læssøe Mikkelsen, Apostolos Pyrgelis, Yannis Stamatiou, "D7.2 Necessary hardware and software package for the student pilot deployment", ABC4Trust project deliverable, Frankfurt, 2012, online:
https://abc4trust.eu/download/ABC4Trust-D7.2.Hard-and-Software-Package-for-Student-Pilot.pdf

[ENISA09] Ingo Naumann, Giles Hogben, et al, „Privacy Features of European eID Card Specifications“, ENISA Position Paper, online:
http://www.enisa.europa.eu/activities/identity-and-trust/trust-services/eid-cards-en.

[FHZNHKGPH11]
Simone Fischer-Hübner, Harald Zwingelberg, Gregory Neven, Leif-Erik Holtz, Ulrich König, Staffan Gustavsson, Tobias Pulls, Hans Hedbom "D4.3.2 - UI Prototypes: Policy Administration and Presentation - Version 2", online:
http://primelife.ercim.eu/images/stories/deliverables/d4.3.2-policy_administration_and_presentation_ui_prototypes_v2-public.pdf

[ISO29101] ISO/IEC standard "Information technology – Security techniques -- Privacy architecture framework", to be published, online:
http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=45124

[Zwin11] Harald Zwingelberg, "Necessary Processing of Personal Data: The Need-to-Know Principle and Processing Data from the New German Identity Card", in Simone Fischer-Hübner, Penny Duquenoy, Marit Hansen, Ronald Leenes, Ge Zhang, Privacy and Identity Management for Life - 6th IFIP WG 9.2, 9.6/11.7, 11.4, 11.6/PrimeLife International Summer School, Helsingborg, Sweden, August 2-6, 2010, Revised Selected Papers, 2011, online:
http://link.springer.com/chapter/10.1007%2F978-3-642-20769-3_13.“

[ZwiJen13] Zwingelberg/Jensen chapter. TBD Add reference to FutureID Vision document once published on FutureID website (final text exists)

[ZwiSch99] Harald Zwingelberg, Jan Schallaböck, "The Proposal for a Regulation on Electronic Identification and Trust Services under a Privacy and ABC4Trust Perspective", ABC4Trust project heartbeat, forthcoming, to be published online:
https://abc4trust.eu/index.php/pub