Digital Security: Cybersecurity, Privacy and Trust

DS-02-2014 Access Control

European Commission

Horizon 2020
European Union funding
for Research & Innovation

From Real-world Identities to Privacy-preserving and Attribute-based CREDentials for Device-centric Access Control

WP4 – Identity Consolidation and Real-to-Online Identity Mapping

Deliverable D4.3 "Online identity and profile management"

Editor(s):	EXUS
Author(s):	Vasileios Sarris (EXUS)
	Vangelis Bagiatis (UPCOM)
	Savvas Zannetou, Harris Partaourides, Antonis Papasavva (CUT)
Dissemination Level:	Public
Nature:	Report
Version:	1.1

ReCRED Project Profile

Contract Number	653417
Acronym	ReCRED
Title	From Real-world Identities to Privacy-preserving and Attribute-based CREDentials for Device-centric Access Control
Start Date	May 1st, 2015
Duration	36 Months

Partners

University of Piraeus	University of Piraeus research center	Greece
Telefónica Investigación y Desarrollo	Telefonica Investigacion Y Desarrollo Sa	Spain
verizon	Verizon Nederland B.V.	The Netherlands
certSIGN	Certsign SA	Romania
wedia	Wedia Limited	Greece
EXUS	EXUS Software Ltd	U.K.
upcom Bringing business and IT together	Upcom Bvba (sme)	Belgium
THE PRODUCTIZERS	De Productizers B.V.	The Netherlands
Cyprus University of Technology	Cyprus University of Technology	Cyprus

	Universidad Carlos III de Madrid	Spain
cnit	Consorzio Nazionale Interuniversitario per le Telecomunicazioni	Italy
BAKER & McKENZIE	Studio Professionale Associato a Baker & Mckenzie	Italy

Document History

AUTHORS

Vasileios Sarris (EXUS)
Vangelis Bagiatis (UPCOM)
Savvas Zannetou (CUT)

VERSIONS

Version	Date	Author	Remarks
0.10	2017-06-20	EXUS	Template and TOC creation
0.2	2017-07-13	EXUS	Initial addition of the Identity management module
0.3	2017-07-17	UPCOM	Consent management mobile, Identity Profile Management mobile applications contribution
0.4	2017-07-21	CUT	Consent management backend contribution
0.5	2017-07-21	EXUS	Merge and refinement of content
0.6	2017-07-24	CUT	First review comments
0.7	2017-07-27	EXUS	Addressed first review comments
1.0	2017-07-31	EXUS	Final adjustments
1.1	2018-04-30	CUT	Revised Consent Management Module

Executive Summary

This document describes the design, implementation and functionalities of the Online Identity and Profile Management module of the Identity Consolidator Platform.

The Profile management module allows users to manage their identity attributes that are maintained across various online services, transfer attributes among Identity Providers, get de-anonymization risk indications and allow users to specify their consent policies that define when and how identity attributes can be revealed and/or transferred among entities of the ReCRED ecosystem.

Table of Contents

Executive Summary 4
List of Figures 7
1 Introduction 9
2 General Overview of Identity Management Module 9
2.1 Identity Profile Management module 11
2.1.1 Trust Modes 12
2.1.2 Communication with Identity Providers 12
2.1.3 Transfer of Identity Attributes between ID Providers 13
2.1.4 Partial Verifiable Profiles 17
2.2 Consent Management module 18
3 Identity Profile Management 19
3.1 Identity Profile Management web application 19
3.1.1 Identity Profile Management web application operations overview 19
3.2 Identity Profile Management mobile application 31
3.2.1 Identity Providers 32
3.2.2 Physical Documents 35
3.2.3 Service Providers 35
3.2.4 Partially Verifiable Profiles 37
3.3 Risk Management 40
3.3.1 De-Anonymization Risk Assessment 41
4 Consent Management 44
4.1 Consent Management Back-End 45
4.1.1 High Level Operations ..... **Error! Bookmark not defined.**
4.1.2 Create Policy ..... **Error! Bookmark not defined.**
4.1.3 View Policy ..... **Error! Bookmark not defined.**
4.1.4 Delete Policy ..... **Error! Bookmark not defined.**
4.1.5 Evaluation Requests ..... **Error! Bookmark not defined.**
4.1.6 Policy Recommendation ..... **Error! Bookmark not defined.**
4.2 Consent Management mobile application 59
4.2.1 Create new Consent Policy 59
4.2.2 View and Manage Consent Policies 60
4.3 Consent Management Web Interface 61
4.3.1 Create Consent Policies ..... **Error! Bookmark not defined.**

4.3.2 View and Delete Consent Policies ..... **Error! Bookmark not defined.**
4.3.3 Consent Policies Recommendations ..... **Error! Bookmark not defined.**
5 Conclusion 61
6 References 67

List of Figures

Figure 1: Generic architecture regarding Identity and profile management 12
Figure 2: OpenID Connect extension for transferring attributes 16
Figure 3: Consent Management Architecture 18
Figure 4: Identity Profile Management web application main page 23
Figure 5: Identity Profile Management web application. Identity Providers users' identity attributes 23
Figure 6: Edit the value of a user’s identity attributes acquired from his Identity Providers 24
Figure 7: Edit dialog for identity attributes acquired from an Identity Provider 24
Figure 8: Edit attributes @ Facebook 25
Figure 9: Identity Profile Management web application. Edit identity attribute values or transfer identity attributes among IdPs functionalities 25
Figure 10: Identity Profile Management web application. Transfer identity attributes among IdPs functionalities, log in to provide consent 26
Figure 11: Identity Profile Management web application. Transfer identity attributes among IdPs functionalities ready 27
Figure 12: Identity Profile Management web application. User's financial information page 28
Figure 13: User's names information 28
Figure 14: User's acquired names details and verification information 28
Figure 15: User Locations information 29
Figure 16: URLs information 29
Figure 17: Uploaded Media Information 30
Figure 18: Physical Ids information 30
Figure 19: User’s email information 30
Figure 20: User emails’ information 31
Figure 21: User’s addresses information 31
Figure 22: Mobile App Landing Page 32
Figure 23: List of Identity Attributes 33
Figure 24: Identity Providers per Attribute 33
Figure 25: List of Identity Providers 34
Figure 26: Attributes per Identity Provider 34
Figure 27: Documents from Physical Acquisition 35
Figure 28: Attributes for Specific Document 35
Figure 29: List of Identity Attributes 36
Figure 30: Service Providers per Attribute 36
Figure 31: List of Service Providers 37
Figure 32: Attributes per Service Provider 37
Figure 33: List of PVPs 38
Figure 34: Share a PVP 38
Figure 35 Create new PVP (select attributes) 39
Figure 36 Create new PVP (select documents) 39
Figure 37 Create new PVP (select name/context /expiration) 40
Figure 38 Identity Provider field risks 41
Figure 39 Identity Provider field risks 42

Figure 40 Financial Information Risks 43
Figure 41 Create new Consent Policy 60
Figure 42:List of Consent Policies 60
Figure 43: View policy details 61
Figure 44: Delete a Consent Policy 61
Figure 45: Create a policy web interface ..... **Error! Bookmark not defined.**
Figure 46: View and Delete Consent Policies Interface ..... **Error! Bookmark not defined.**
Figure 47: Policy Recommendation Web Interface ..... **Error! Bookmark not defined.**

1 Introduction

The Identity Management module of the Identity Consolidator is an important component of the Identity Consolidator. It plays significant role in the management and representation of the identity attributes of the ReCRED users, while offering services to the whole identity consolidator for enabling functionalities such as transferring attributes amongst identity providers and risk calculation.

The Identity Management module is subdivided in two sub-modules, the identity management and the consent management. The Identity Management module enable the users to manage their identity attributes through a user-friendly web and mobile interface. Furthermore, it presents to the users’ identity attributes that are known or maintained by Identity Providers and Service Providers, as well as associated risks of de-anonymization for identity attributes that are not explicitly revealed. Also, it allows users to create partial verifiable profiles, which include a user-defined subset of his identity attributes, in order to share with other entities. This enables end-users to easily prove parts of his identity attributes with other entities that only need to click on a URL. Finally, the identity management module allows end-users to define consent policies regarding the reveal and/or transfer of identity attributes between Online Services of the ReCRED ecosystem. This is achieved through a user-friendly mobile and a web interface.

The remainder of this document is structured as follows; In Section 2 we provide a general overview of the identity management module and its subcomponents. In Section 3 we provide a detailed description of the Identity Profile Management module as well as its respective Web and Mobile application. Section 4 provides a detailed specification of the Consent Management Module. Specifically, we present the documentation for the supported back-end operations as well as the mobile application. Finally, in Section 0 we conclude this document.

2 General Overview of Identity Management Module

The Identity Management module is common framework that serves as a standard for the definition and representation of user identity attributes within a given online service. Identity Management module is subdivided in two sub-modules, the identity management and the consent management.

In general, it provides an identity matrix containing the different type and range of identifiers, and unique identity attributes a user can have. A representative use case is **reputation** in a certain online platform such as eBay, which we view as an attribute of that user. This identity attribute is currently used only by eBay and its users. Apart from that, this module offers a protocol to transfer identity attributes between ID providers and at the same time guarantees the security in the transfer of such sensitive information. Also, it gives users the option to create partial verifiable profiles, which consist of selected identity attributes of a user, to be presented to verifiers depending on the context and the access control requirements. Furthermore, it allows users to define their consent for the management of their various identity attributes.

The Identity Management module is implemented as mobile and web applications alongside with a backend service for both.

In summary the Identity Management module offers the following set of operations that are grouped according to their users.

End-users:

- View and manage identity attributes: The user is presented with a list of all the identity attributes supported by the IDC. For each attribute, the user can see the filled-in value (if not blank) and whether it has been verified or not. The user can also update the value of an attribute (or fill-in a blank attribute) which may or may not trigger a verification process. In that case, an update request is sent to all the IDPs that maintain the updated identity attributes, so that the user data remains synchronized across different IDPs. The user can also delete the value of an identity attribute and a delete request is sent to all the IDPs that maintain the deleted attribute.
- View identity attributes shared with Online services: The user can select an identity attribute and see which online services have access to that attribute. Alternatively, the user can select an online service and see which identity attributes are shared with that service. In both cases, risk calculation is executed, so that the user can see the probability with which an online service can infer the value for an attribute, even if it hasn't been explicitly shared with it.
- View and manage identity attributes maintained by ID Providers: The user can select an identity attribute and see a list with the IDPs that maintain that attribute, as well as the value of the attribute on each IDP. Alternatively, the user can select an IDP and see which identity attributes are maintained by that IDP and with which values. In the latter case, the user can select one or more identity attributes, in order to execute the following actions:
    - transfer the values of those identity attributes from the selected IDP to the ID Consolidator.
    - transfer the values of those identity attributes from the selected IDP to other IDPs also taking into account any specific rules defined by the selected IDP).
    - delete the values of those identity attributes from the selected IDP.
- Create partial verifiable profiles: The user can select a subset of identity attributes and create a partially verifiable profile with those attributes. The IDC creates and presents a short URL that links to the created profile.
- List partial verifiable profiles: The user is presented with a list of all the partially verifiable profiles that he/she has created.
- View and manage partial verifiable profiles: The user can view extensive details regarding a selected verifiable profile (included attributes, short URL, creation date, etc.) and can preview the public verifiable profile. The user can also manage (add / remove) the identity attributes that are associated with the selected verifiable profile or delete the profile altogether.
- Define their consent for their various identity attributes. These are implemented as consent policies and are evaluated when a request is made to transfer attributes or when there is a request to issue a cryptographic credential.

ID Provider Administrators:

Manage /define consent rules for identity attributes transfer: The IDP Administrator can define specific rules for the transfer of attributes maintained by the IDP to other IDPs. E.g. a bank administrator can create a rule so that the customers’ bank accounts and loan data can be transferred only to other verified banks.

2.1 Identity Profile Management module

This module provides the user’s interface that allows the user to know and manage what each identity provider and verifier knows about them. It also enables the user to determine the risk of identity providers and verifiers inferring information about them that they didn’t explicitly reveal to them. This information can leak by statistically analyzing correlations between identities attributes, thus the risk will be calculated by using similar techniques. Additionally, it allows a user to transfer identity attributes from one identity provider to another with respect to the policies defined in identity consent management module. It also allows the user to invoke the online ID acquisition module to transfer ID attributes from the ID providers to the IDC. Furthermore, the user has the ability to delete an identity attribute from an IDP. Lastly, this module provides the required functionality to the user to create and manage partial verifiable profiles.

The Identity Profile Management module comprises the following:

- A backend that provides a REST API including methods covering all the management functionality. The backend communicates with the Identity Repository over the Storage API. Users are authenticated using the ReCRED Authentication module
- A web frontend that offers a UI for ReCRED users that wish to manage their Identity Profile over a web connection
- A mobile application that offers ReCRED uses the opportunity to manage their Identity Profile using their mobile device.

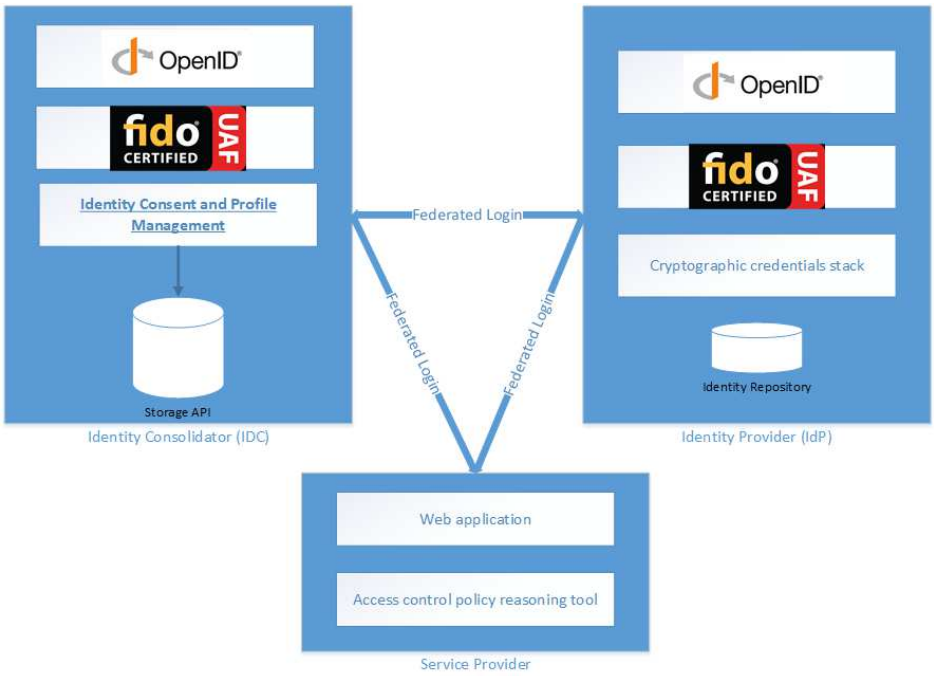

Figure 1: Generic architecture regarding Identity and profile management

2.1.1 Trust Modes

2.1.1.1 Fully Trusted

A user may choose to fully trust the Identity Consolidator. This means that all user attributes are stored along with their values in the Identity Repository. In this case whenever the user edits any of the stored attributes, Identity Management module stores all changes in the Identity Repository and contacts all the associated Identity Providers in order to have them update their local instances of the user’s attributes. In order to ensure consistency between the Identity Repository and the involved Identity Providers, the ID Management module will need to confirm the update and mark (in the Identity Repository) associated ID Provider records as “update confirmed” or not.

2.1.1.2 Not Trusted

Users may select to store only the associations between user attributes and ID Providers. In this case, the ID Management module will contact each of the associated ID Providers with the updated attribute values, and confirm that the update has been successful. It will also mark associated ID Provider records as “update confirmed” or not.

Note: Confirmation is the simple process of retrieving the user’s attributes from the Identity Provider and comparing them against the updated values.

2.1.2 Communication with Identity Providers

ID Providers leverage Federated Login protocols (i.e., OpenID Connect) for the communication and exchange of data with the Identity Management module. To do so, the IDC needs to maintain the base URLs to it’s Identity Repository. In a typical scenario, the ID Management module will get the base URL of each ID Provider from the Identity Repository, and will then have to get permission from the user to access each of the ID Providers. Then it will attempt an OpenID Connect/OAuth

communication with the ID Providers in order to retrieve and/or update identity attributes for the requesting user.

2.1.3 Transfer of Identity Attributes between ID Providers

Transfer of attributes between ID Providers will be performed in the following way:

- The user specifies the attribute(s) and destination(s) ID Provider(s).
- The ID Management module retrieves from the Identity Repository the source ID Providers that maintain the attributes.
- It retrieves the requested attribute(s) from the source ID Provider(s) using the OpenID Connect specification.
- It transfers the retrieved attribute(s) to the destination ID Provider(s) by leveraging a modified extension of the OpenID Connect specification.
- It may display a “please confirm overwrite” prompt for destination ID Providers that indicate they already have an instance of this attribute for the user’s identity.
- It may then send a “overwrite confirmed” message to the ID Provider according to how the user answers the prompt.
- It confirms (i.e. retrieves and compares against the original) the stored attributes for each destination ID Provider.
- Finally, it stores in the Identity Repository an association between the retrieved attributes and the destination ID Providers.
- (Optional) In the case of Trusted Operation, the ID Management module will also store the retrieved attributes in the Identity Repository

Below is an outline of the REST API call that the ID Management module implements for Identity Attribute Transfer

Method POST

URL http://.../API/attribute/transfer

Content

```
{
      “attributeId”:”a”,
      “sourceProviderId”:”x”,
      “destinationProviders”:
      [
            {“ProviderId”:”y”},
            {“ProviderId”:”z”},
            …
      ]
```

```
}
```

Response

```
{
      “attributeId”:”a”,
      “sourceProviderId”:“x”,
      “destinationProviders”:
      [
            {“ProviderId”:”y”, “verified”:true},
            {“ProviderId”:”z”, “verified”:false},
            …
      ]
}
```

The following method transfers most important attributes from Identity Provider x to Identity Providers y and z assuming that this can be done so (the attributes exist as options in the target, the authorization has been given etc)

Method POST

URL http://…/ /transfer/i-d-provider-fields/{id1}/{id2}

Content

```
{
 Long path parameter    Id1, Long path parameter Id2
}
```

Response Status 200

In order to enable the transfer of attributes between identity providers the operation must be supported by the targeted Identity Provider. For this purpose since the conventional OpenID Connect doesn’t support write of attributes from applications but only read, we proposed an extension to the protocol so that the operation of transfer can be done so.

The use case can be described with the following scenario. Initially the user notifies the Service Provider (Source IdP) about the identity attributes that he/she wants to transfer to the Identity Provider (Destination IdP). After that the Service Provider requests from the user to authenticate with the Identity Provider (Destination IdP) in order to prove the possession of his account in the Identity Provider. Upon successful authentication, the Service Provider requests from the user

authorization for write access to the Identity Provider in order to write the values of the requested identity attributes to the ID Repository of the Identity Provider (Destination IdP).

When an entity (SP) wants to update an attribute from another entity (IDP), he first needs to obtain an access token. The access token should contain information about which attributes the SP can modify for which user. To request consent to change attributes, the SP should launch an OpenID connect request to the IDP.

For example:

http://.../oauth2/authorize?response_type=code&client_id=openidclient&realm=%2F&scope=address%20email%20phone%20openid%20profile%20change_phone%20recredid%20change_language&redirect_uri=http%3A%2F%REDIRECT_URL_ENDPOINT &state=af0ifjsldkj&acr_values=2%204%201

Using oauth2 custom "scopes", the SP indicates what information he wants to change.

Currently configured scopes are:

change_phone

change_email

change_language

change_phone will allow a transfer request for attribute PhoneNumber.

change_email will allow a transfer request for attribute Email.

change_language will allow a transfer request for attribute Language.

The addition of more scopes is just a part of the current IDPs configuration.

When the IDP receives an OpenID Connect request, he will first ask the user to authenticate. Next, the IDP should validate whether consent can be granted for the request (e.g. there is no blacklist policy). In some cases, the IDP will ask the user explicit consent. If consent is arranged, the SP will be able to get an access token.

When a service provider now wants to update an attribute, he adds the access token to the headers so that the update can be done so.

This request is now sent to an authorization layer. The authorization layer will perform some checks. It reaches out to the IDP to check if the token is still valid; to which userID the token belongs, and what scopes are included. The service provider will only be able to change attributes of a user if the right user identifier is used, and if only attributes are changed that are included in the token.

Below, Figure 2 summarizes the current OpenID Connect extension proposed:

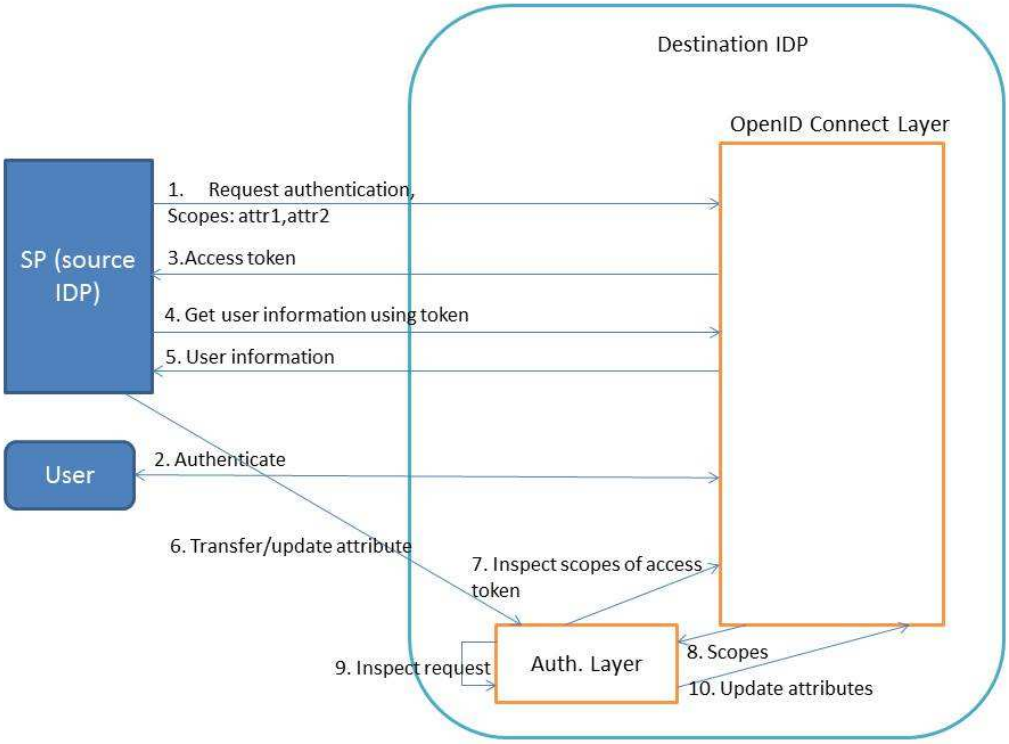

Figure 2: OpenID Connect extension for transferring attributes

2.1.4 Partial Verifiable Profiles

Users can group some of their attributes in profiles so that verifiers may be offered only a selection of verified attributes instead of the entire collection of the user’s stored identity. For example, a user may choose to only prove that he/she is a citizen of the European Union, or that a User is a registered professional with the corresponding national Professional Association.

The ID Management module implements a REST API to facilitate the creation and management of profiles. This will in the backend call the physical identity acquisition REST API for the handling of the PVPs.

Create:

Method POST

URL http://.../API/profile/user/1

Content

```
{
      “profile_name”: ”My new partial profile”,
            “attributeID”:”x”,
      “attributeID”:”y”

}
```

Response

```
{
      "user_id": 2,
            "created_profile_id": 6,
            "result": "success"
}
```

Retrieve:

Method GET

URL http://.../API/profile/1234

Response

```
{
      “id”:1234,
      “name”: ”My new partial profile”,
      “attributes”:
```

```
        [
            {“attributeID”:”x”}
            {“attributeID”:”y”}
            …
        ]
    }
```

Delete:

Method	DELETE
URL	http://.../API/profile/1234

2.2 Consent Management module

The Consent Management module allows users, Identity Providers (IDPs) and the Identity Consolidator (IDC) to define their consent for their various identity attributes. These are implemented as consent policies and are evaluated when a request is made to transfer attributes or when there is a request to issue a cryptographic credential.

The policies are divided in whitelist to give consent and blacklist to deny consent and they can involve the attribute name, level of assurance (LOA), confidence score (CS) as well as IDPs and SPs name and LOA. In the case of issuing credentials, we also involve the type of protocol.

There are two major components in the Consent Management Module the Back-End and the Front-End. At the Back-end all the consent management functionalities are implemented in a REST API format that will be used by the Front-End. The Front-End is further divided into four parts. Three web Front-Ends for the user, the IDP and the IDC respectively and a mobile app for the user.

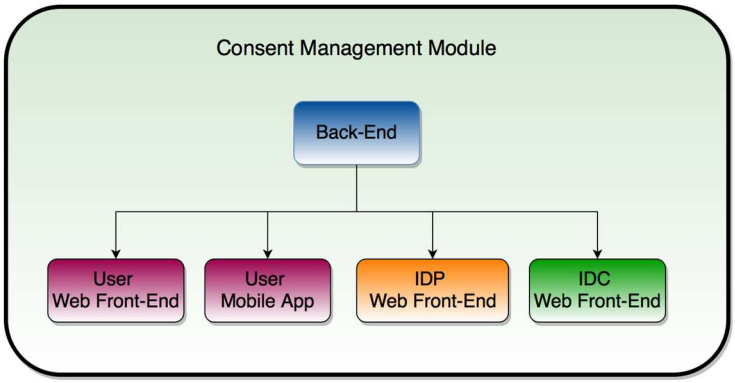

Figure 3: Consent Management Architecture

The Consent Management mobile front-end is an Android mobile app that allows the users to define their consent for their various identity attributes, by defining policies regarding the Identity Providers and Service Providers to which their attributes should be revealed.

The mobile app communicates with the Consent Management back-end, allowing the end-users to access the following functionality:

- Create new consent policies, which whitelist or blacklist specific identity attributes (or groups of attributes) from Identity Providers and/or Service Providers
- View the consent policies that he has created
- Modify or delete consent policies that he has already created

3 Identity Profile Management

3.1 Identity Profile Management web application

3.1.1 Identity Profile Management web application operations overview

The Identity Profile Management module contains a web and mobile application, through which the users can view and manage their own identity data, as well as the identity attributes that each identity provider maintains for them or any online service knows. Therefore, the application contains two main interfaces:

1. **Identity Data Management**: The user can view all current data that are stored in the Identity Repository. The user is also able to centrally update the value of some identity attributes, as long as this is allowed by the user-defined policy or the ID provider-defined policy for these attributes. For example, the user cannot change their name or age (especially if these have already been verified) but they can change their current job or address details. An update on some attributes may or may not trigger the initiation of an identity acquisition and verification process (e.g. a new proof of address). After the data is updated (and verified, if necessary), all the online services that maintain the updated identity attributes are notified and the values they hold are automatically updated as well.
2. **Shared Identity Attributes**: The user can view all of the identity attributes that are shared with various online services. This can be achieved in two alternative ways:
   a) The user selects an identity attribute and all the online services that maintain that attribute are fetched and displayed.
   b) The user selects an online service and all the identity attributes maintained by that service are fetched and displayed.

The application uses the Storage API in order to store and retrieve data from the Identity Repository. More specifically, the following methods are used:

- A GET method to retrieve the user's identity data from the Identity Repository.
- A PUT method to update the user's identity data.
- A GET method to search for identity providers that maintain data for a given identity attribute.
- A GET method to search for identity attributes that are maintained by a given identity provider.
- DELETE methods to:
    - delete an attribute from the Identity Repository but let the associated ID Providers keep it (at least the source one).
    - delete an attribute from an ID Provider but keep it in the Identity Repository.
    - refresh the value of an identity attribute from the source ID Provider.
    - toggle the trust mode of an identity attribute i.e., retrieve and store the value from the ID Provider in Full Trust and delete the value of the attribute from the Identity Repository in No Trust mode.

Example 1: Get the identity attributes for the user with id=1

Method	`GET`
URL	`http://.../API/identity/user/1`

Response	`{`
	`"id":1`
	`"lastName":"Doe"`
	`"firstName":"John"`
	`"birthdate":"01-01-1980"`
	`…`
	`}`

Example 2: Update the address details of the user with id=1

Method	`PUT`
URL	`http://.../API/identity/user/1`

Payload	`{`
	`"address":"10, Gloucester Road"`
	`"city":"London"`
	`"country":"UK"`
	`…`
	`}`

Example 3: Get the online services that maintain the user's street address attribute

Method	`GET`

URL	http://.../API/identity/serviceProvider?f=attribute=address

Response	[
	{
	“id”:1
	“serviceName”:”Service1”
	“URL”:”http://...”
	…
	}
	{
	“id”:2
	“serviceName”:”Service2”
	“URL”:”http://...”
	…
	}
	…
	]

Example 4: Get the identity attributes that are maintained by the online service with id=1

Methodf	GET
URL	http://.../API/identity/attribute?f=providerId=1

Response	[
	{
	“attributeId”:1
	“attributeName”:”last name”
	…
	}
	{
	“attributeId”:2
	“attributeName”:”first name”
	…
	}
	
	…
	]

Example 5: Delete the identity attributes specified in the request. The ID Management module will also contact all associated ID Providers and request that they also delete the specified attributes.

Method	DELETE
URL	http://.../API/identity/attribute

Content

```
[
        {
                “attributeId”:1
        },
        {
                “attributeId”:2
        },
        …
]
```

3.1.1.1 Identity Profile Management web application front-end

The functionality offered is:

- The user can view all current data that are stored in the Identity Repository.
- The user is also able to centrally update the value of some identity attributes, as long as this is allowed by the ID provider-defined policy for these attributes.
- Transfer identity attributes amongst Identity providers that enable such a procedure.
- Modify an attribute and nullify it if possible.
- Allow users to check which of their attributes are present in which Identity Providers.
- View de-anonymization risks with regards to identity attributes which are unknown to the Identity Providers

The UI has been redesigned with material design of Google [7].

3.1.1.1.1 Main menu

The main menu of the Identity Profile Management module allows an authenticated user to have access to almost all the identity information that is relevant to him regarding Identity providers’ data. The mapping is based on the Storage API and allows the user to view and/or edit the following information:

- Identity Providers acquired information (ID Provider Fields)
- Identity Providers acquired information risks data (ID Provider fields risk)
- Financial Information (Financial Info)
- Financial Information risks (Financial Info risks)
- User declared names (User name)
- User captured location information (User Location)
- URL data
- Uploaded Picture information (Media Item)
- Physical Identity documents information (Physical ID)
- User Phone numbers
- User Email Addresses
- User Addresses information

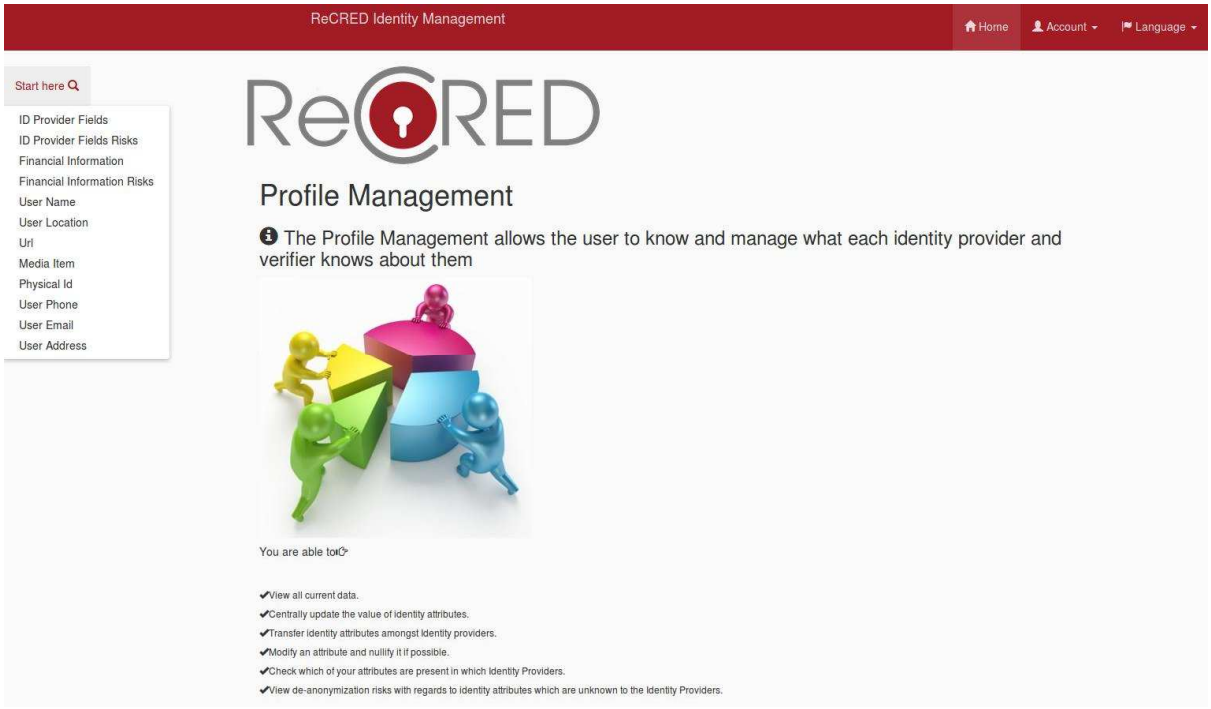

Figure 4: Identity Profile Management web application main page

In general, the user has the ability to do the same for most of the identity attributes categories above. For the identity attributes that are available and maintained only by the Identity Providers and we just store the reference to those Identity Providers are presented with a gray square surrounding them while all the other identity attributes without a gray square are maintained by the Identity Consolidator.

3.1.1.1.2 Users’ acquired identity attributes from Identity Providers

In the Identity Provider fields menu the user can view his identity attributes defined in all the Identity Providers. The Identity Management web application retrieves all the required data from the Identity Repository through the Storage API. Currently the administrator of the Identity Profile Management module has access and is presented with all the available identity attributes for all users.

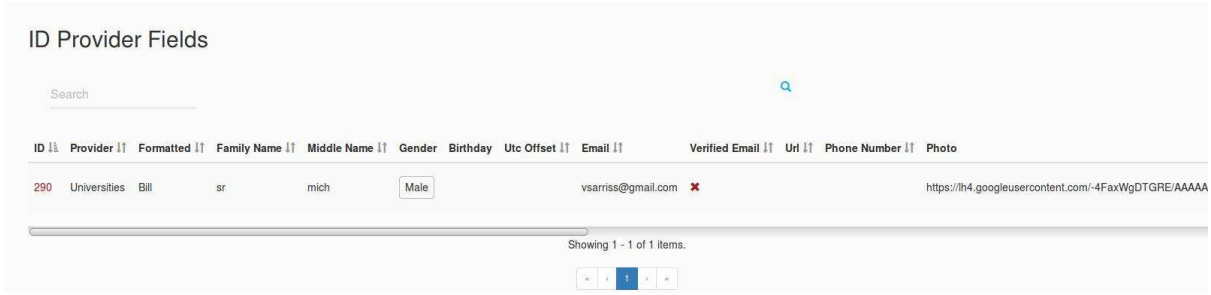

Figure 5: Identity Profile Management web application. Identity Providers users' identity attributes

By pressing the edit button a user can edit any of the available attributes as shown below.

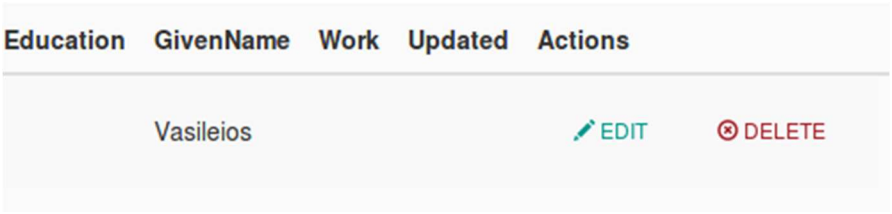

Figure 6: Edit the value of a user’s identity attributes acquired from his Identity Providers

When a user chooses to edit the identity attributes for one of his Identity Providers a popup dialog menu opens and there the user can edit/check all the attributes as shown in figure below. If for example the user changes the name, the attribute value is updated in the Storage API using the appropriate REST API calls at the backend, when the user presses the save button in the popup window.

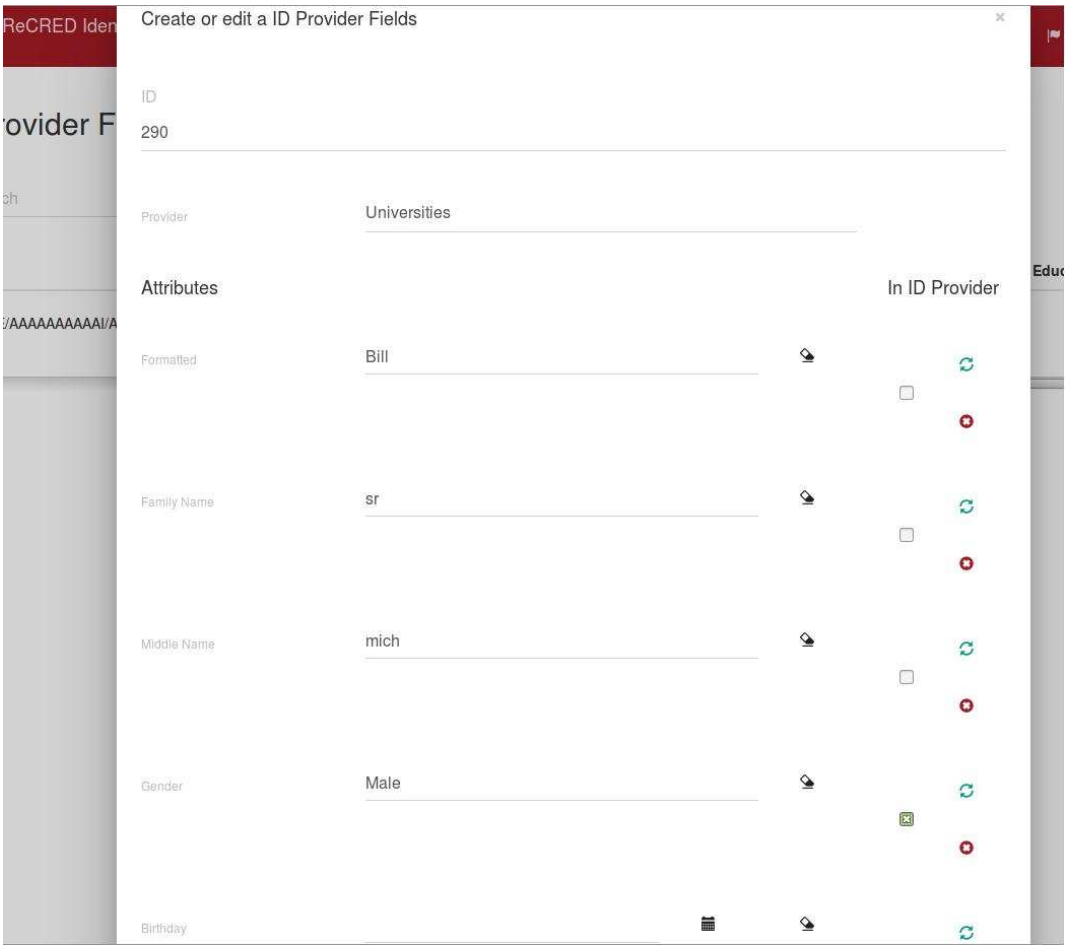

Figure 7: Edit dialog for identity attributes acquired from an Identity Provider

For the editing of the user’s attributes, the information apart from being changed in ReCRED, it has to be updated in the provider itself. Therefore upon the save of the edited information, the user is prompted to visit the settings page of the identity provider for the change of his information as far as this is permitted. For example for the Facebook use case the user will be redirected to the following:

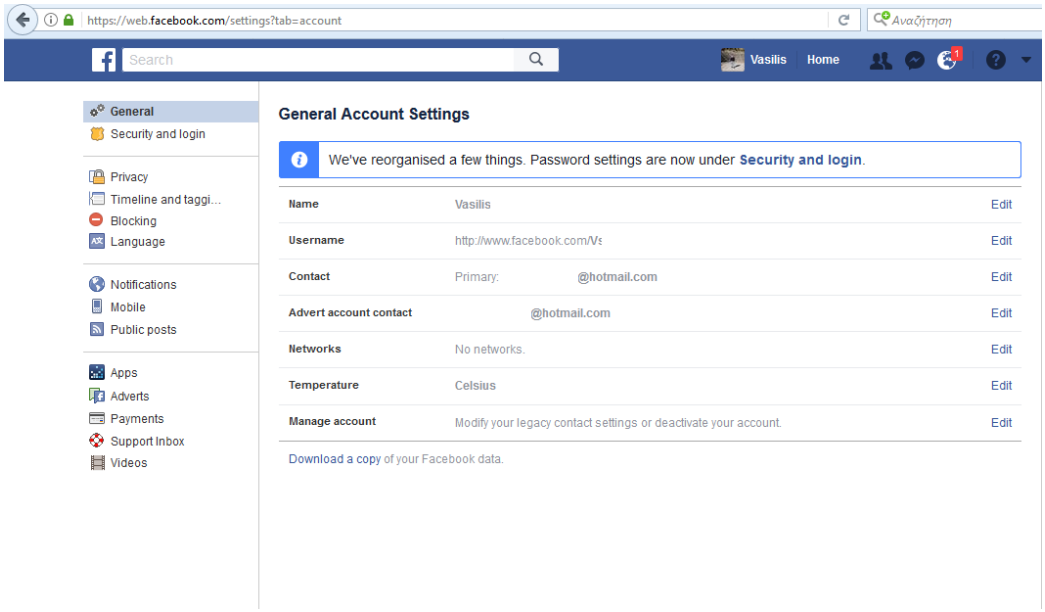

Figure 8: Edit attributes @ Facebook

The transfer button below allows the user to transfer one or more identity attribute from one Identity Provider to another. This functionality currently utilizes the OpenAM extensions described in **Error! Reference source not found.** as a showcase.

Figure 9: Identity Profile Management web application. Edit identity attribute values or transfer identity attributes among IdPs functionalities

Based on the typical OpenID Connect flow, for the description of this functionality we assume that the user wants to transfer on of his identity attributes from IDP_A to the IDP_B. In a typical OpenID

Connect scenario the OpenID Connect Provider is IDP_A that holds the identity attribute that the user wants to transfer and the Service Provider is IDP_B that will receive the transferred identity attribute. In order the identity attribute to be transferred to IDP_B (Service Provider) the user has to authenticate with IDP_A (OpenID Connect Provider) and provide his consent. When he has successfully authenticated with IDP_A and provided his consent the IDP_B receives the identity attribute and stores it.

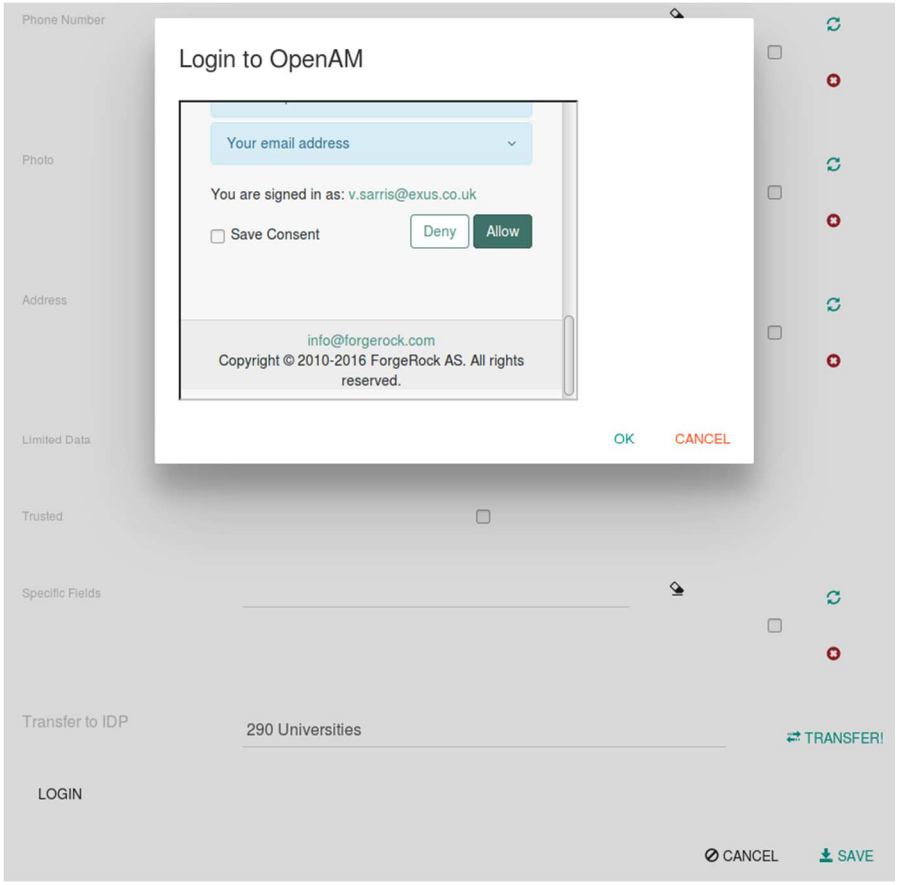

Figure 10: Identity Profile Management web application. Transfer identity attributes among IdPs functionalities, log in to provide consent

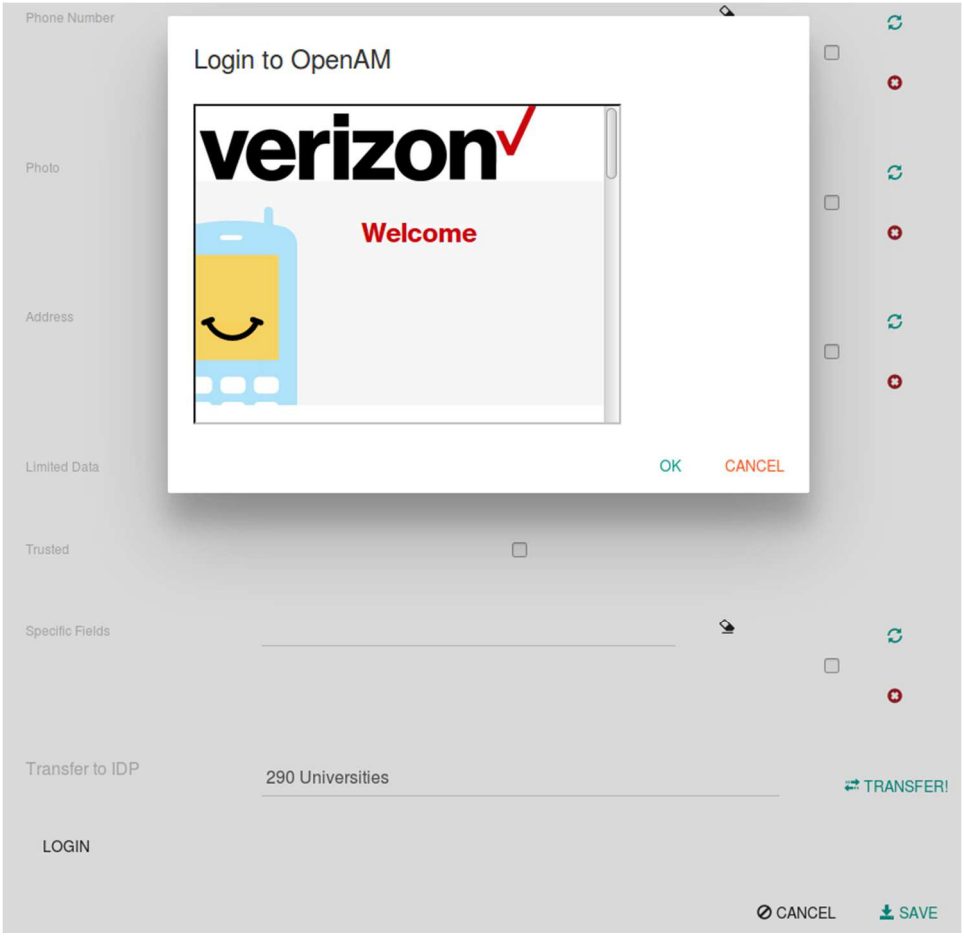

Figure 11: Identity Profile Management web application. Transfer identity attributes among IdPs functionalities ready

In the case where the user wants to delete or update the value an identity attribute from an Identity Provider then he first has to authenticate against this ID Provider (OpenID Connect Provider) and then he has to provide his consent for the deletion or update of the identity attribute. In this scenario the Service Provider that request the deletion or update of the identity attribute is the Identity Consolidator (Identity Profile Management web application).

3.1.1.1.3 User’s financial information

The Financial Info page of the Identity Profile management web application presents to the user all his acquired financial information (e.g., his Income, Monthly Loan etc.). The figure below is an example of this page.

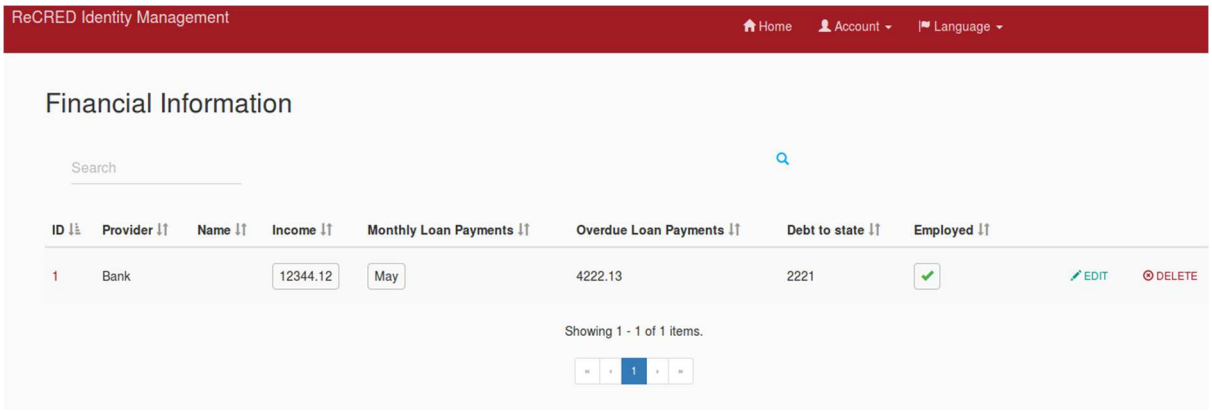

Figure 12: Identity Profile Management web application. User's financial information page

3.1.1.1.4 User’s declared names information

The figure below presents to the user the information of the different names acquired through Online Identity Acquisition from his multiple online accounts and the information of the different names acquired through Physical Identity Acquisition from his multiple identity documents. We allow the storage of multiple names because the declared user’s name in his multiple online accounts and real-world identities may differ.

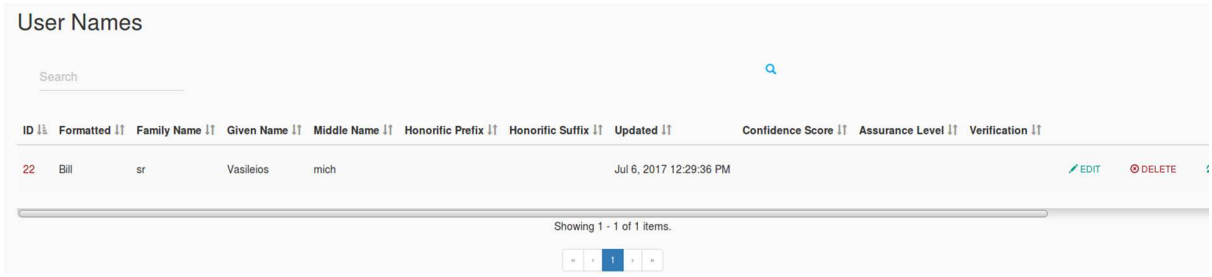

Figure 13: User's names information

Furthermore, verification information is also displayed for all the names that have been acquired through the Physical Identity Acquisition module as shown in the figure below.

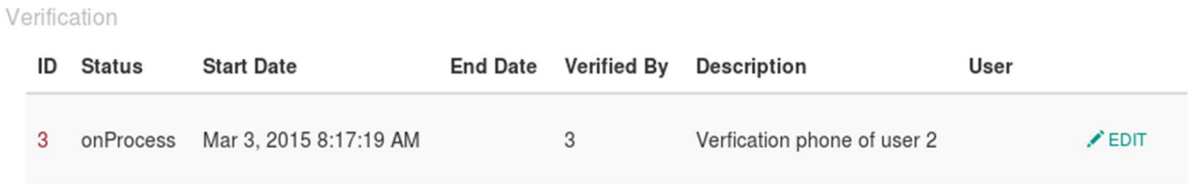

Figure 14: User's acquired names details and verification information

3.1.1.1.5 User’s acquired Locations information

The Identity Profile management module also offers a page that presents to the user all the locations acquired through the location tracking functionality of the Physical Identity Acquisition module for address verification.

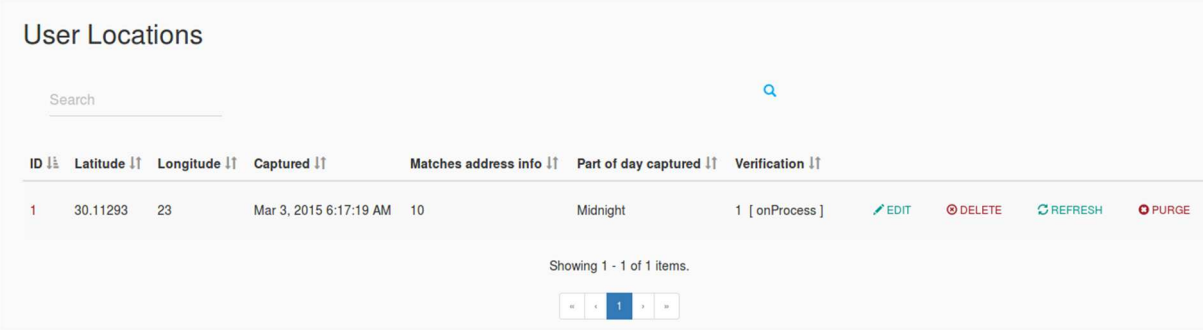

Figure 15: User Locations information

3.1.1.1.6 URLs information

The Identity Management module also offers a page the presents to the user the information of his declared websites etc.

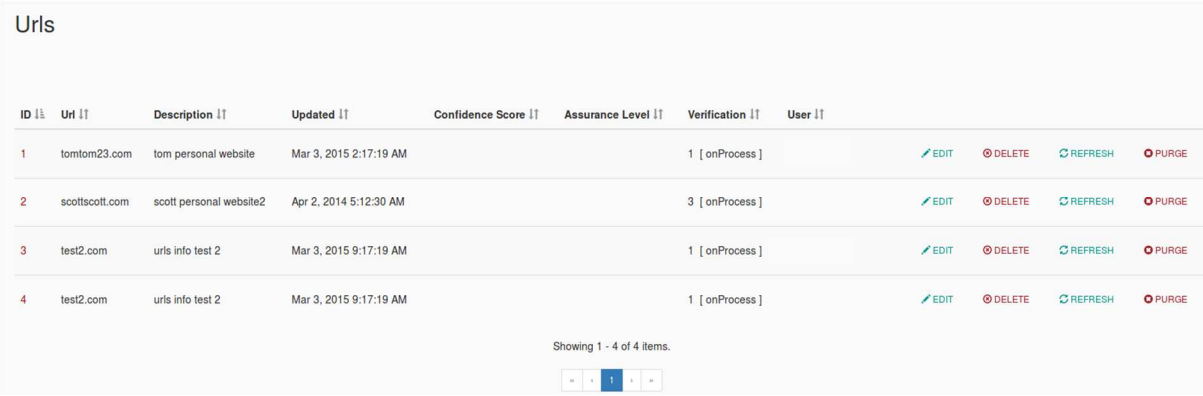

Figure 16: URLs information

3.1.1.1.7 Uploaded Media items information

This page shows to the user the information of the media items (pictures, videos, etc.) that are maintained by the Identity Consolidator. Most of the stored media items are acquired by the Physical Identity Acquisition module for the verification of the users’ physical identity documents.

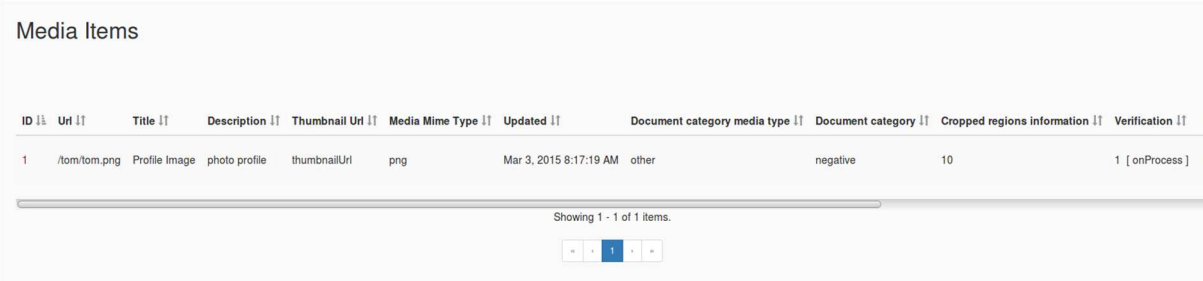

Figure 17: Uploaded Media Information

3.1.1.1.8 Physical Identity Documents information

The Identity Management module allows the users to view and manage the information of the physical identity documents that he has declared and verified through the Physical Identity Acquisition module. Figure 49 below is an example of this page that offers this functionality.

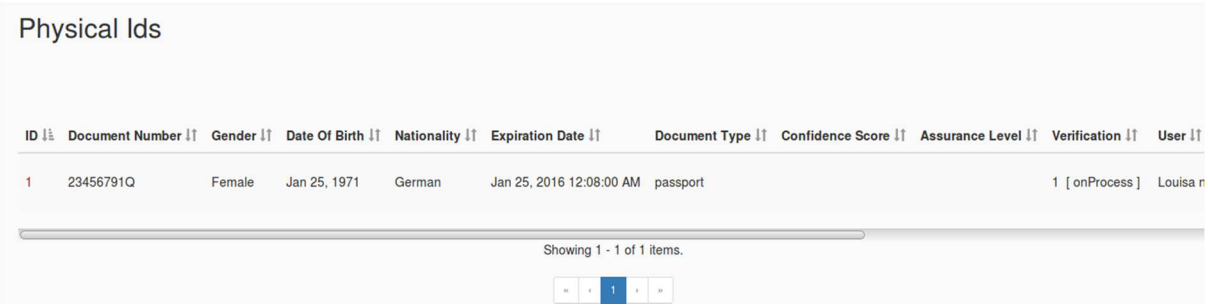

Figure 18: Physical Ids information

3.1.1.1.9 User Phones information

Additionally, the Identity Management module offers a page that presents to the user the information of the phone numbers that the Identity Consolidator has stored about him.

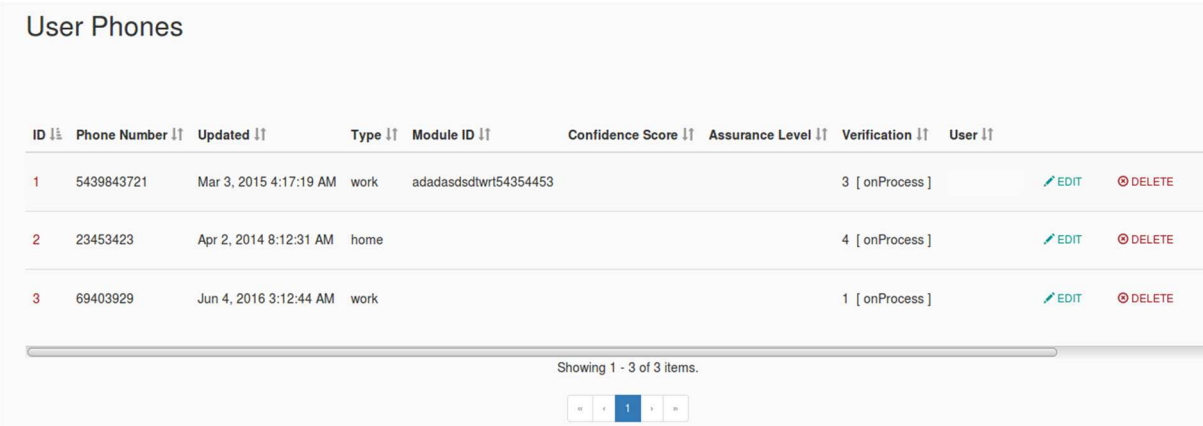

Figure 19: User’s email information

3.1.1.1.10 User’s email addresses information

This page presents to the user the email addresses the Identity Consolidator acquired about him through Online Identity and Physical Identity Acquisition modules.

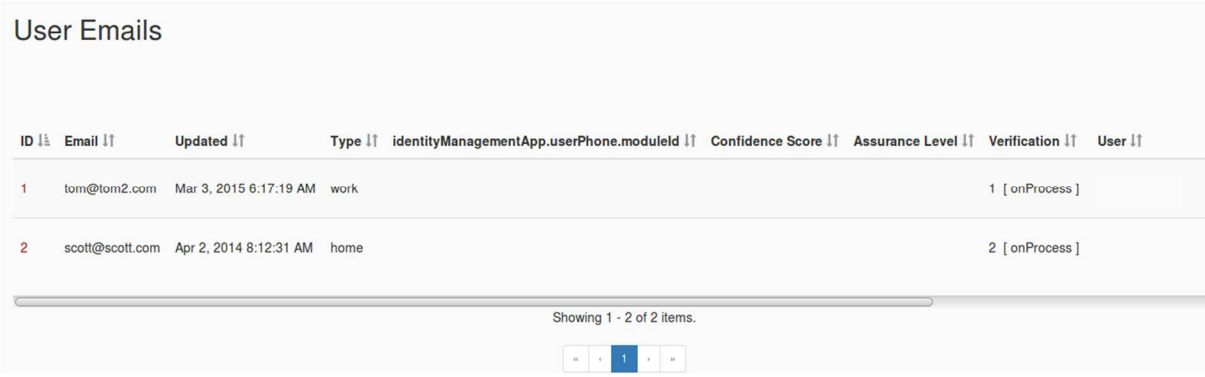

Figure 20: User emails’ information

3.1.1.1.11 User’s Addresses information

The last functionality that the Identity Profile Management module offers to the users in terms of viewing their identity attributes is a page that presents the information of their addresses information acquired and verified through the Physical Identity Acquisition module. Additionally, this may include addresses acquired from the various online accounts of a user.

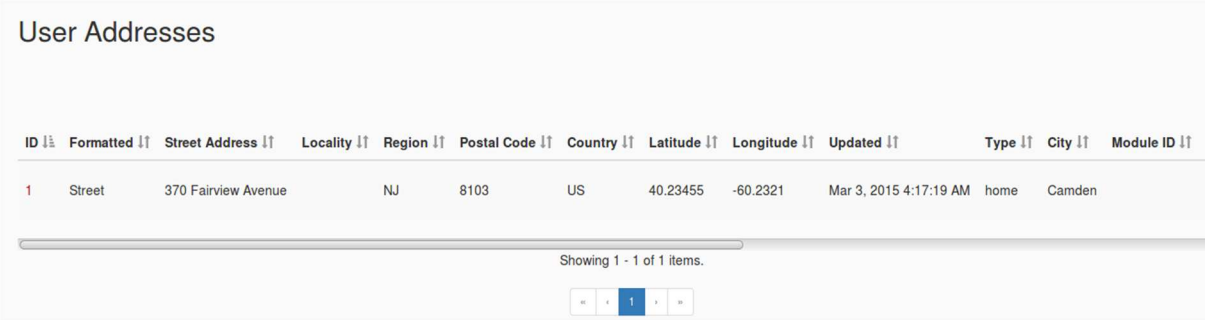

Figure 21: User’s addresses information

3.2 Identity Profile Management mobile application

The Identity Profile Management mobile front-end is essentially an Android mobile app, aiming to provide the same functionality as the web front-end through a mobile device. The app communicates with the Identity Profile back-end, in order to allow the user to view and manage his online and physical identities. More specifically, through the Identity Profile Management mobile app, the users can have access to the following functionality:

- View their online and physical identities, as they are acquired by the online and physical acquisition modules respectively,
- View which identity attributes are maintained by which Identity Providers
- Transfer identity attribute values among different Identity Providers or from an Identity Provider to the ID Consolidator
- Delete an identity attribute value from an Identity Provider that maintains it
- View which identity attributes have been revealed to Service Providers

- View de-anonymization risk indicators regarding unrevealed identity attributes to Identity Providers and Service Providers
- Create and manage partially verifiable profiles

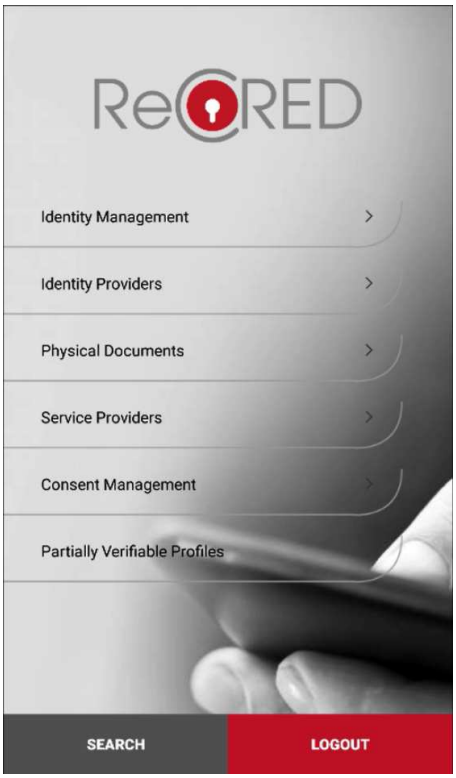

Figure 22: Mobile App Landing Page

3.2.1 Identity Providers

Through the “Identity Providers” option, the user can see which of his identity attributes are maintained by which Identity Providers. These Identity Providers are connected with the user’s account through the Online Identity Acquisition Module. This functionality is offered to the user through two alternative approaches.

1st Approach: View Identity Providers by Attribute

The user can see a list with all his identity attributes (Figure 23). After selecting a specific attribute, a new list is displayed, with all the Identity Providers that the user has connected, and what each Identity Provider knows or can infer regarding the selected attribute (Figure 24).

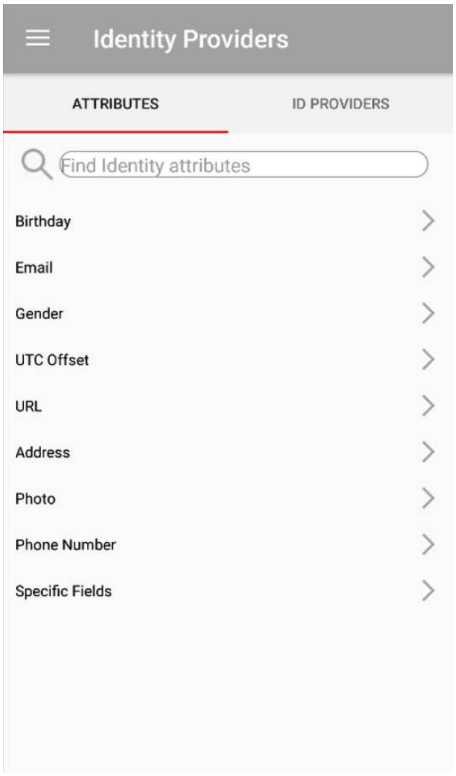

Figure 23: List of Identity Attributes

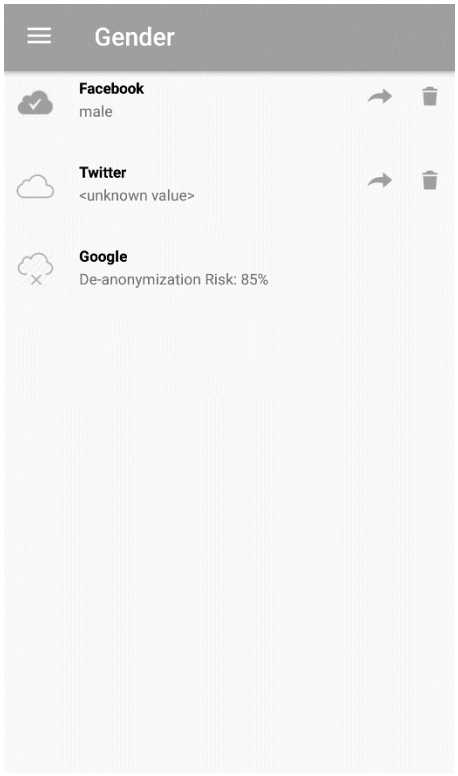

Figure 24: Identity Providers per Attribute

2nd Approach: View Attributes by Identity Provider

The user can see a list with all his Identity Providers, and for each Identity Provider the number of the attributes it maintains is also displayed (Figure 25). After selecting a specific Identity Provider, a new list is displayed, with all the identity attributes and what the selected Identity Provider knows about each attribute (Figure 26).

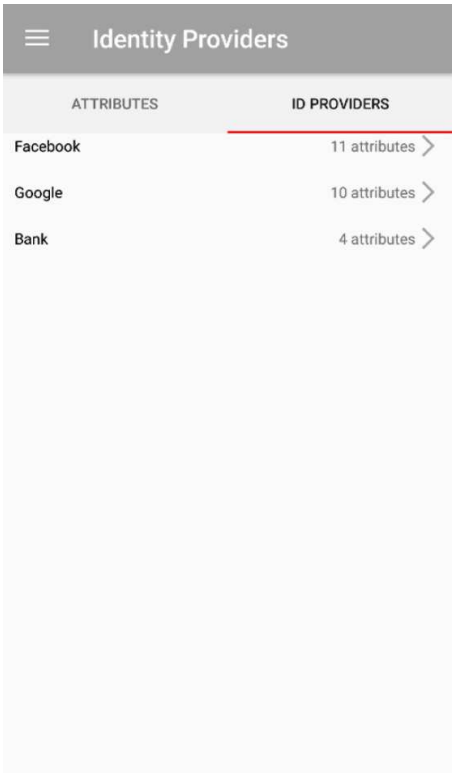

Figure 25: List of Identity Providers

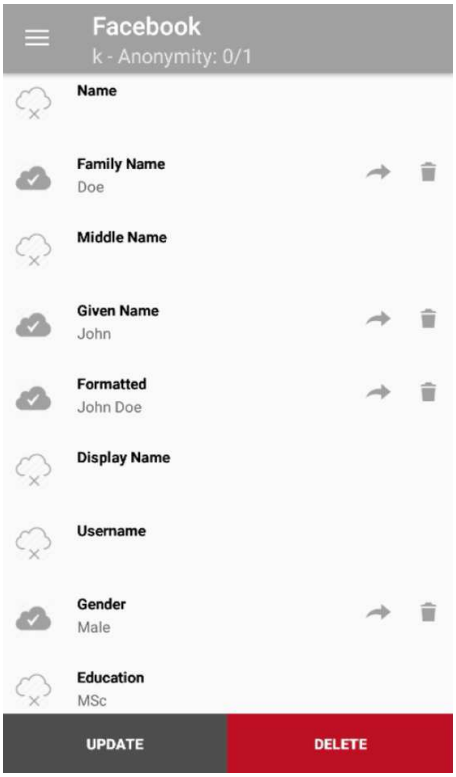

Figure 26: Attributes per Identity Provider

In both approaches, three different cases are identified, regarding what an Identity Provider and the Identity Consolidator know about an attribute:

- Both the Identity Consolidator and the Identity Provider know the value of a given attribute. This is depicted by an icon next to the Identity Provider, and the value of the attribute is also displayed.
- The Identity Consolidator knows that the Identity Provider maintains a given attribute, but it does not know its actual value (mainly because the user has chosen not to trust the Identity Consolidator). This is depicted by an icon next to the Identity Provider, but the actual value of the attribute is not displayed (since it is unknown).
- The Identity Provider does not maintain a given attribute, which is depicted by an icon next to the Identity Provider. In that case, the de-anonymization risk is also displayed, as long as the selected attribute is not unique (e.g. email address or telephone number).

We also support an extension to the OpenID Connect protocol, so that a user can authorize a service provider to request from an identity provider to update and/or delete his attributes. In that case, the user will be able to also transfer attributes from one Identity Provider to other Identity Providers, overriding any existing values maintained by the target Identity Providers, while respecting, at the same time, the user’s policies defined through the Consent Management module. The user will also be able to delete an attribute value from an Identity Provider that maintains that attribute.

3.2.2 Physical Documents

Through the “Physical Documents” option, the user can see his identity attributes that have been transferred to the Identity Consolidator through the Physical Identity Acquisition module. At first, the user can see all the official documents that she has used for physical acquisition, such as a passport or a driving license (Figure 27). After selecting a specific document, the values of the respective attributes are displayed, along with the confidence score and the Level of Assurance (LoA) for that specific document type (Figure 28).

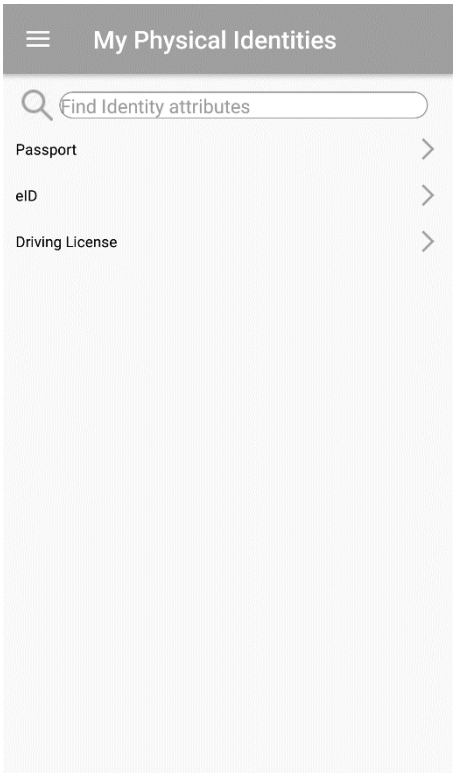

Figure 27: Documents from Physical Acquisition

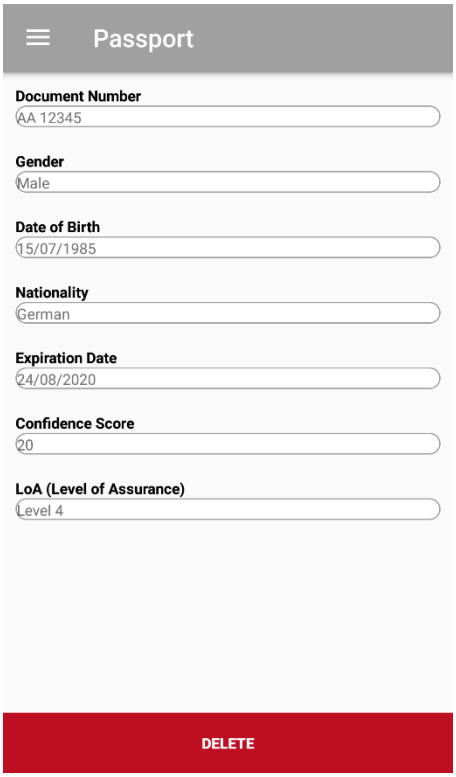

Figure 28: Attributes for Specific Document

3.2.3 Service Providers

Through the “Service Providers” option, the user can see which Service Providers know which aspects of his identity. This functionality is offered to the user through two alternative approaches.

1st Approach: View Service Providers by Attribute

The user can see a list with all his identity attributes (Figure 29). After selecting a specific attribute, a new list is displayed, with all the Service Providers to whom that attribute has been revealed (Figure 30). If an attribute has not been explicitly revealed to a Service Provider, the de-anonymization risk is displayed, as long as the selected attribute is not unique (e.g. email address or telephone number).

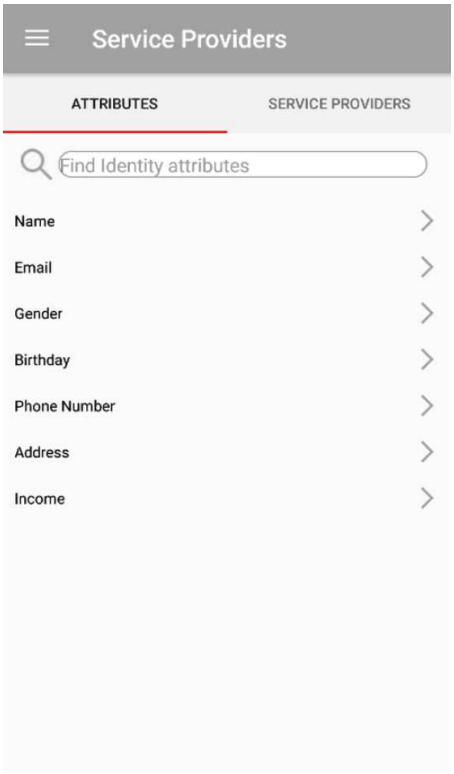

Figure 29: List of Identity Attributes

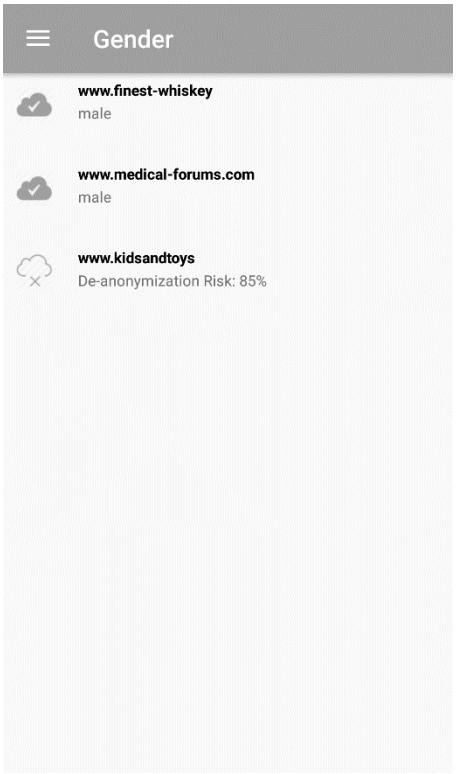

Figure 30: Service Providers per Attribute

2nd Approach: View Attributes by Service Provider

The user can see a list of Service Providers, and for each Service Provider the number of the attribute values that it knows is also displayed (Figure 31). After selecting a specific Service Provider, a new list is displayed, with all the identity attributes and what the selected Service Provider knows or can infer about each attribute (Figure 32).

Figure 31: List of Service Providers

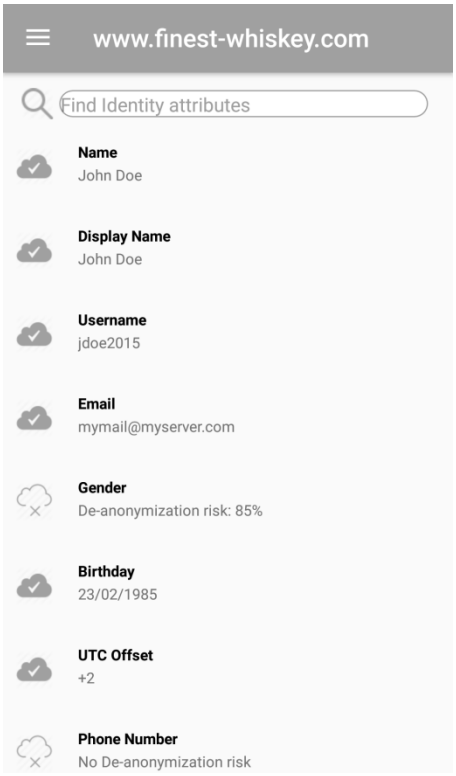

Figure 32: Attributes per Service Provider

In both approaches, if the Service Provider knows the value of a given attribute, an icon appears next to the Service Provider, and the value of the attribute is also displayed. Otherwise, an icon appears, along with the de-anonymization risk.

3.2.4 Partially Verifiable Profiles

Through the “Partially Verifiable Profiles” option, the user can select a number of identity attributes and create Partially Verifiable Profiles (PVP) that will contain these attributes. These PVPs can then be presented to verifiers, depending on the context and the access control requirements.

At first, the user can see all the PVPs that he has created, along with the attributes they contain and the date they were last updated (Figure 33).

In order to share a particular PVP, the user can tap on the icon, in which case a sharing dialog appears, showing the attributes included in the selected PVP, along with the actual URL that the user can use in order to share his PVP (Figure 34).

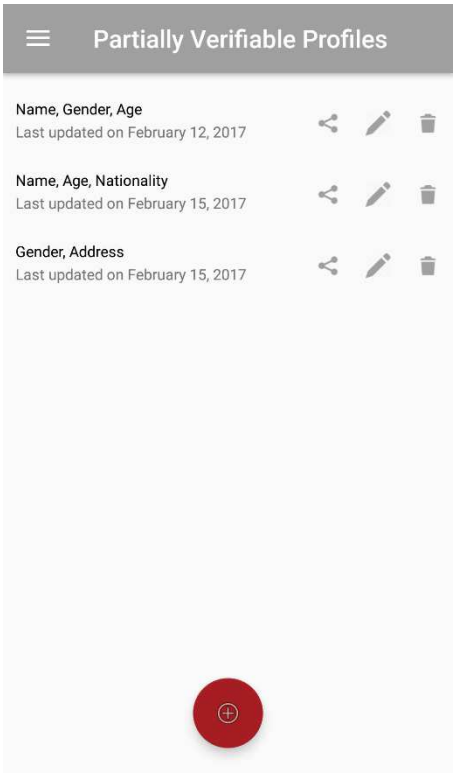

Figure 33: List of PVPs

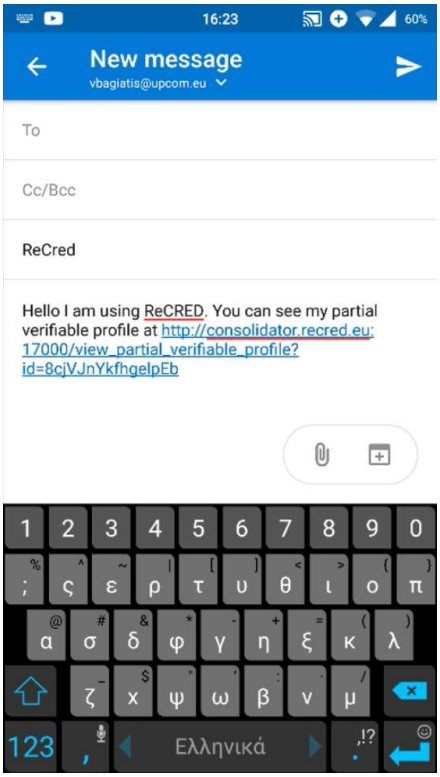

Figure 34: Share a PVP

The user can also tap on the icon in order to modify any PVP or tap on the icon in order to delete it.

In order to create a new PVP, the user must select his identity attributes that will be included in the new PVP (**Error! Reference source not found.**).

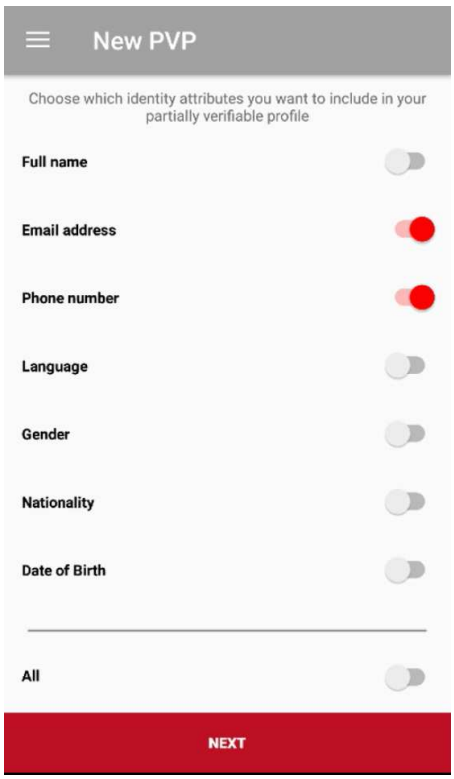

Figure 35 Create new PVP (select attributes)

After that, the user can choose to include her Identity Card and/or Front Face Photo in the new PVP (**Error! Reference source not found.**).

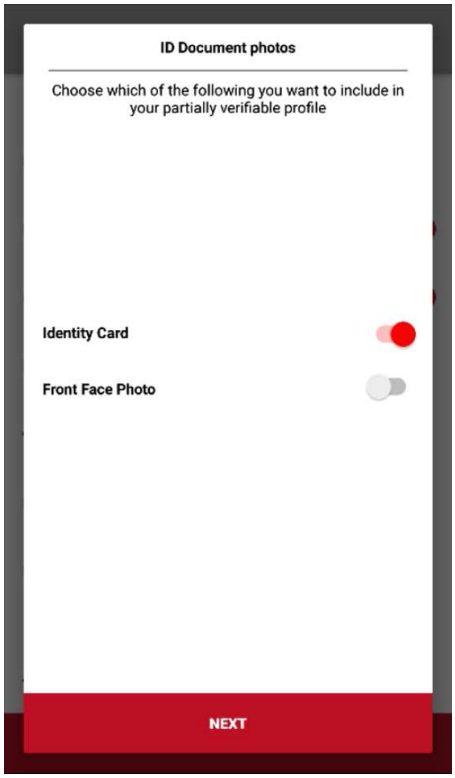

Figure 36 Create new PVP (select documents)

In the final screen (

3.3 Risk Management

The ID Management module will display to the user a risk figure indicating the possibility that an ID Provider or a Service Provider may infer the values of unknown user attributes based on the known user attributes that the ID Provider maintains for this user. The risk indicator is separate for each unknown attribute and ID Provider permutation.

Note: Risk can be calculated only for attributes with known values i.e. attributes whose values are stored in the Identity Repository. For attribute values stored only in ID Providers it is impossible to determine their distribution and hence cannot calculate a risk factor.

The calculation of the risks is based on the discrete uniform distribution [9] and the formula is derived from the probability definition. The ReCRED users belong to several identity providers. In addition we don’t have all the users of all the identity providers registered in ReCRED IDC. So we need to make the following assumptions and adoption of a well-defined model for the risks calculations.

We assume that the ReCRED users who belong to a specific identity provider have the same statistical characteristics as all the users of the identity provider. We therefore consider the count of the ReCRED users to be enough so that this property exists. Conclusively the statistical distribution assumed that better explains our model is the discrete uniform distribution. By definition then a finite number of values, namely attributes in our example are equally likely to be observed taking into account the count of users who have the same other known attributes.

Then, the risk number is the percentage of users with the same values for the known attributes and same known value for the unknown attribute over the total number of users with the same values for the known attributes only.

$$Risk = 100\frac{sameUnknownValuesRecordCount}{knownValuesRecordCount}\ \%$$

3.3.1 De-Anonymization Risk Assessment

Besides identity attributes view and management, the Identity Profile Management module offers the de-anonymization risk assessment functionality. This functionality has been integrated with the Identity Profile Management web application in order to provide a user friendly and more coherent User Experience.

In general, de-anonymization risk assessment functionality helps a user to protect his privacy with respect to confidence in unauthorized identity attribute value inference. De-anonymization risk is calculated for identity attributes related to Identity Providers, Service Providers, and for all the identity attributes that compose the financial information of a user. The de-anonymization risks that are taken into account involve risk of identity attribute value inference, and risk of user identification without his authorization.

Confidence in unauthorized identity attribute value inference is essentially the risk that an Identity Provider or Service Providers can infer the value of another identity attribute of a user that he has

not shared with them based on the identity attribute values that the user shared with them and based on the distribution of the identity attributes across the web. In statistical terms it is an indication of where the user fits within the user population as this is segmented based on the revealed identity attribute values and the unrevealed identity attributes that the risk is calculated for.

The current implementation of this functionality addresses the risk of identity attribute value Inference for Identity Providers and for the financial information of the users. Implementation of risk calculation for the Service Providers and of risk calculation for user identification is still in progress.

Figures below show an example of the page that allows the users to view the risk calculation for Identity Providers.

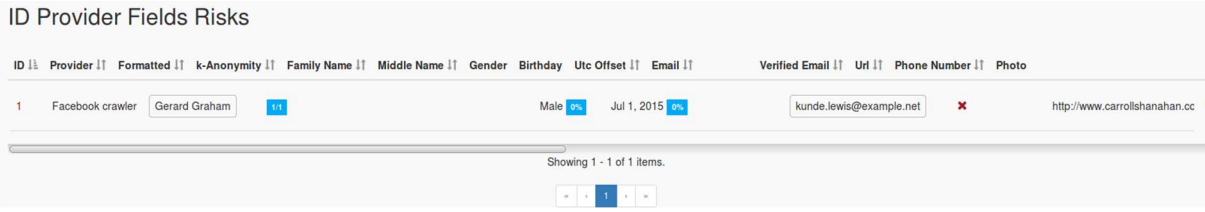

Figure 38 Identity Provider field risks

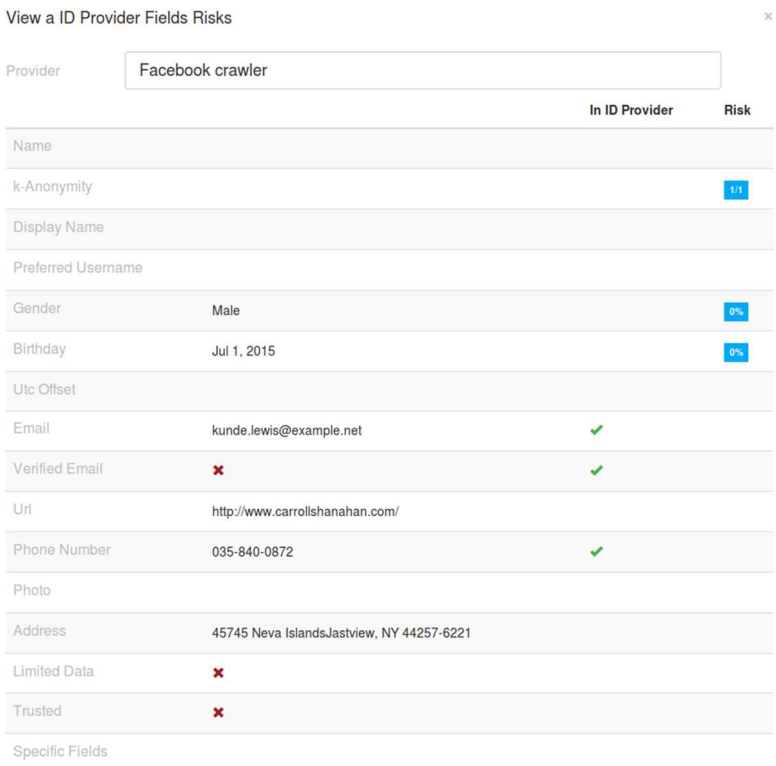

Figure 39 Identity Provider field risks

Each one of the identity attributes where de-anonymization is applicable is associated with a confidence percentage that represents the possibility that an Identity Provider can infer the value of this attribute.

As shown in the figure, a detailed view of the de-anonymization risk calculation for Identity Providers displays the confidence percentage next to each identity attribute that is applicable. A star ‘*’ symbol is used to indicate which attributes have been used for risk calculation, i.e. the *known* attribute values.

Calculation is based on the assumption that all the Identity Providers have acquired the distribution of identity attributes across the web from external means and this is distribution is at least the same as the one that we have in the Identity Repository of the Identity Consolidator platform.

For example, assume that the ID Provider knows that out of 100 employed users, with income between 20000 and 30000, 30 have Overdue Loan Payments between 1000 and 2000. So for an employed user, with income between 20000 and 30000, who has not revealed to the ID Provider that he has Overdue Loan Payments amount of 1500, the ID Provider can estimate that there is a 30% possibility to have Overdue Loan Payments amount between 1000 and 2000. Therefore, the de-anonymization risk for attribute Overdue Loan Payments amount of this user is 30%.

As mentioned before, de-anonymization risk assessment is also calculated for all the identity attributes related to the financial information of the user stored in the Identity Repository. An example of the risk calculation for those attributes is shown in figures below.

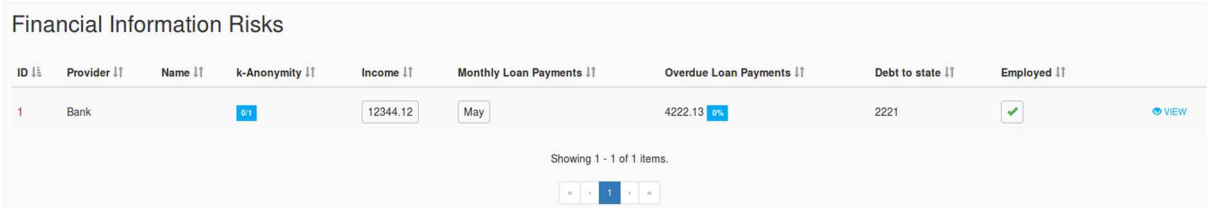

Figure 40 Financial Information Risks

The de-anonymization risk calculation functionality for ID Providers and Service providers is separated in two different categories depending on the method that the user is using to prove the hold of the identity attribute and access the Service.

These categories are the following:

Vanilla OpenID Connect: The first is when a user uses the vanilla OpenID Connect to prove an identity attribute to a Service Provider. With OpenID Connect we have no untraceability and any Service Provider or Identity Provider can track the user across the web. In this case we want to protect the user against Service Providers and Identity Providers that has a unique identifier for each user and slowly by retrieving the identity attributes of each user they are in place to build a complete profile for each user. As a result, anywhere that a user goes to the web the Identity Providers that authenticates him are in place to trace him. The only thing we can do here is to prevent Identity Providers and Service Providers to build the complete profile of a user.

In order to achieve this, the Identity Profile Management module needs to be aware of the identity attributes shared with what Service Providers either from the Identity Consolidator or from an external Identity Provider. Such information will be provided by OpenAM, which keeps logs each time a user shares an identity attribute with a Service Provider.

In general, the only anonymity that we can provide with vanilla OpenID Connect is that we can produce the confidence probability whether a Service Provider can infer the value of an attribute, which the user has not revealed to this Service Provider. So as soon as this functionality is implemented the user should be able anytime to go to the Profile Management module and see that will happen if he revealed a specific identity attribute or a combination of identity attributes. We should call this Privacy with respect to Confidence in unauthorized attribute value inference.

Idemix and U-Prove: The second case is when a user uses P-ABAC (Idemix and U-Prove) to prove that he is an identity attribute and access a Service Provider. Using P-ABAC we have untraceability and unlinkability. With Idemix we have the Verifying Identity Providers who run an Idemix stack. In this case the user runs idemix with the verifying Identity Provider and with an idemix credential he proves to this Identity Provider one of his identity attributes or a combination of identity attributes (e.g., his age) and then the Identity Provider assures the Service Provider that the user is the holder of a credential that proves his age.

With Idemix we have two concepts of anonymity (untraceability and unlinkability - untraceable and unlinkable credentials). This is because when the same user shows a combination of three attributes to an Identity Provider and Service Provider at time A, if the same user go to another Service Provider and prove the same three attributes at time B there is no mechanisms that can infer or assure that I am the same person as the one at time A. This means that I am untraceable and my credentials cannot be linked. This also means that no Identity Provider or Service Provider can built a complete profile for me.

Based on the aforementioned, in the case of Idemix the risk assessment is different and we need to calculate the risk per session. Additionally, with Idemix the calculation of the risk for de-anonymization does not depend on the credentials that a user shared before with Service Providers. Also, it does not depend on the policies that a Service Provider may have for access control but on the specific credential (identity attribute) or combination of credentials that a user is about the reveal to a Service Provider.

When we are talking about risk assessment with Idemix we need to know whether a given attribute can be inferred by a Service Provider and what is the confidence probability based on the combination of the credentials that a user is about to share with this Service Provider. For this we assume that all the Service Providers and Identity Providers have acquired the distribution of identity attributes and the distribution of the combinations of identity attributes (the frequency this combination of attributes occurs in the complete population) from external means.

For the computation of risk for de-anonymization in the case of Idemix we need to make sure, for each session separately, that when a user reveals some identity attributes (credentials) to the verifying Identity Provider or the Service Provider they do not learn any other of the user's identity attributes that he does not want to reveal to them. In general, we want whenever a user goes to an Identity Provider to inform him for example –given the distribution of identity attributes - that this combination of identity attributes that you are going to reveal are also appear 10 times across the population and if you reveal them, the Service Provider has the knowledge to guess the values of some more of your identity attributes. This computation should be offered by the Identity Profile

Management module and the Idemix client that runs on the verifying Identity Provider should invoke the Identity Management module and display the calculated risk that the given Identity Provider or Service Provider can infer other identity attributes of the user.

), the user can select a context and a name for the new PvP. She can also select an expiration date and time and/or create a one-time PvP (which can be sent to one single person to view one single time).

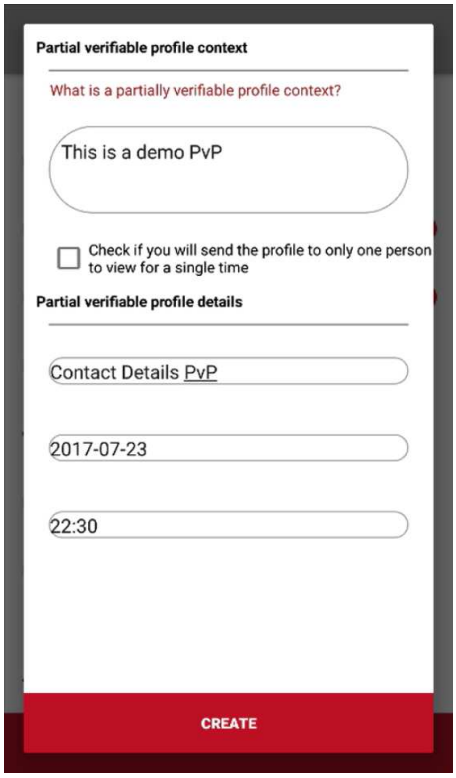

Figure 37 Create new PVP (select name/context /expiration)

3.4 Risk Management

The ID Management module will display to the user a risk figure indicating the possibility that an ID Provider or a Service Provider may infer the values of unknown user attributes based on the known user attributes that the ID Provider maintains for this user. The risk indicator is separate for each unknown attribute and ID Provider permutation.

Note: Risk can be calculated only for attributes with known values i.e. attributes whose values are stored in the Identity Repository. For attribute values stored only in ID Providers it is impossible to determine their distribution and hence cannot calculate a risk factor.

The calculation of the risks is based on the discrete uniform distribution [9] and the formula is derived from the probability definition. The ReCRED users belong to several identity providers. In addition we don’t have all the users of all the identity providers registered in ReCRED IDC. So we need to make the following assumptions and adoption of a well-defined model for the risks calculations.

We assume that the ReCRED users who belong to a specific identity provider have the same statistical characteristics as all the users of the identity provider. We therefore consider the count of the ReCRED users to be enough so that this property exists. Conclusively the statistical distribution assumed that better explains our model is the discrete uniform distribution. By definition then a finite number of values, namely attributes in our example are equally likely to be observed taking into account the count of users who have the same other known attributes.

Then, the risk number is the percentage of users with the same values for the known attributes and same known value for the unknown attribute over the total number of users with the same values for the known attributes only.

$$Risk = 100\frac{sameUnknownValuesRecordCount}{knownValuesRecordCount}\ \%$$

3.4.1 De-Anonymization Risk Assessment

Besides identity attributes view and management, the Identity Profile Management module offers the de-anonymization risk assessment functionality. This functionality has been integrated with the Identity Profile Management web application in order to provide a user friendly and more coherent User Experience.

In general, de-anonymization risk assessment functionality helps a user to protect his privacy with respect to confidence in unauthorized identity attribute value inference. De-anonymization risk is calculated for identity attributes related to Identity Providers, Service Providers, and for all the identity attributes that compose the financial information of a user. The de-anonymization risks that are taken into account involve risk of identity attribute value inference, and risk of user identification without his authorization.

Confidence in unauthorized identity attribute value inference is essentially the risk that an Identity Provider or Service Providers can infer the value of another identity attribute of a user that he has not shared with them based on the identity attribute values that the user shared with them and based on the distribution of the identity attributes across the web. In statistical terms it is an indication of where the user fits within the user population as this is segmented based on the revealed identity attribute values and the unrevealed identity attributes that the risk is calculated for.

The current implementation of this functionality addresses the risk of identity attribute value Inference for Identity Providers and for the financial information of the users. Implementation of risk calculation for the Service Providers and of risk calculation for user identification is still in progress.

Figures below show an example of the page that allows the users to view the risk calculation for Identity Providers.

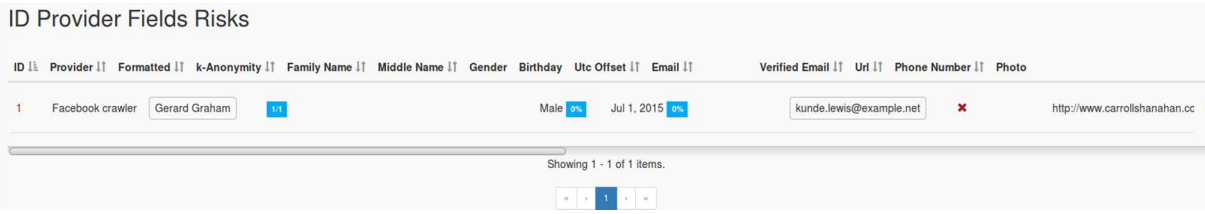

Figure 38 Identity Provider field risks

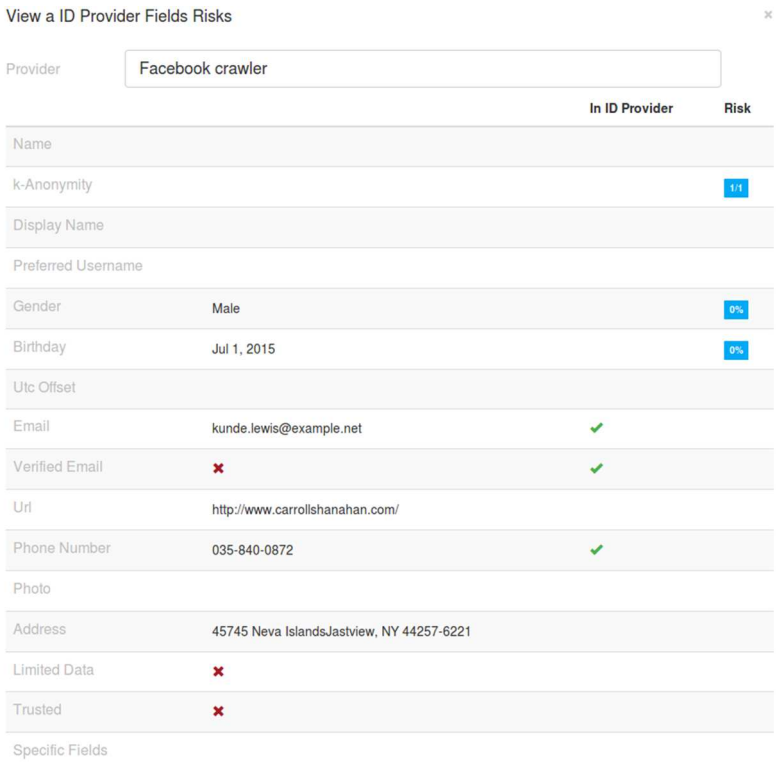

Figure 39 Identity Provider field risks

Each one of the identity attributes where de-anonymization is applicable is associated with a confidence percentage that represents the possibility that an Identity Provider can infer the value of this attribute.

As shown in the figure, a detailed view of the de-anonymization risk calculation for Identity Providers displays the confidence percentage next to each identity attribute that is applicable. A star ‘*’ symbol is used to indicate which attributes have been used for risk calculation, i.e. the *known* attribute values.

Calculation is based on the assumption that all the Identity Providers have acquired the distribution of identity attributes across the web from external means and this is distribution is at least the same as the one that we have in the Identity Repository of the Identity Consolidator platform.

For example, assume that the ID Provider knows that out of 100 employed users, with income between 20000 and 30000, 30 have Overdue Loan Payments between 1000 and 2000. So for an

employed user, with income between 20000 and 30000, who has not revealed to the ID Provider that he has Overdue Loan Payments amount of 1500, the ID Provider can estimate that there is a 30% possibility to have Overdue Loan Payments amount between 1000 and 2000. Therefore, the de-anonymization risk for attribute Overdue Loan Payments amount of this user is 30%.

As mentioned before, de-anonymization risk assessment is also calculated for all the identity attributes related to the financial information of the user stored in the Identity Repository. An example of the risk calculation for those attributes is shown in figures below.

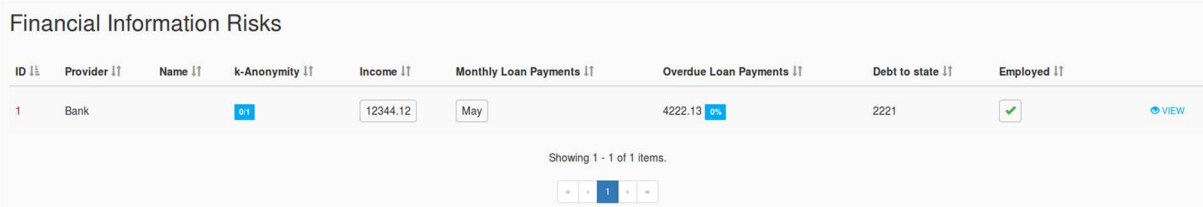

Figure 40 Financial Information Risks

The de-anonymization risk calculation functionality for ID Providers and Service providers is separated in two different categories depending on the method that the user is using to prove the hold of the identity attribute and access the Service.

These categories are the following:

Vanilla OpenID Connect: The first is when a user uses the vanilla OpenID Connect to prove an identity attribute to a Service Provider. With OpenID Connect we have no untraceability and any Service Provider or Identity Provider can track the user across the web. In this case we want to protect the user against Service Providers and Identity Providers that has a unique identifier for each user and slowly by retrieving the identity attributes of each user they are in place to build a complete profile for each user. As a result, anywhere that a user goes to the web the Identity Providers that authenticates him are in place to trace him. The only thing we can do here is to prevent Identity Providers and Service Providers to build the complete profile of a user.

In order to achieve this, the Identity Profile Management module needs to be aware of the identity attributes shared with what Service Providers either from the Identity Consolidator or from an external Identity Provider. Such information will be provided by OpenAM, which keeps logs each time a user shares an identity attribute with a Service Provider.

In general, the only anonymity that we can provide with vanilla OpenID Connect is that we can produce the confidence probability whether a Service Provider can infer the value of an attribute, which the user has not revealed to this Service Provider. So as soon as this functionality is implemented the user should be able anytime to go to the Profile Management module and see that will happen if he revealed a specific identity attribute or a combination of identity attributes. We should call this Privacy with respect to Confidence in unauthorized attribute value inference.

Idemix and U-Prove: The second case is when a user uses P-ABAC (Idemix and U-Prove) to prove that he is an identity attribute and access a Service Provider. Using P-ABAC we have untraceability and unlinkability. With Idemix we have the Verifying Identity Providers who run an Idemix stack. In

this case the user runs idemix with the verifying Identity Provider and with an idemix credential he proves to this Identity Provider one of his identity attributes or a combination of identity attributes (e.g., his age) and then the Identity Provider assures the Service Provider that the user is the holder of a credential that proves his age.

With Idemix we have two concepts of anonymity (untraceability and unlinkability - untraceable and unlinkable credentials). This is because when the same user shows a combination of three attributes to an Identity Provider and Service Provider at time A, if the same user go to another Service Provider and prove the same three attributes at time B there is no mechanisms that can infer or assure that I am the same person as the one at time A. This means that I am untraceable and my credentials cannot be linked. This also means that no Identity Provider or Service Provider can built a complete profile for me.

Based on the aforementioned, in the case of Idemix the risk assessment is different and we need to calculate the risk per session. Additionally, with Idemix the calculation of the risk for de-anonymization does not depend on the credentials that a user shared before with Service Providers. Also, it does not depend on the policies that a Service Provider may have for access control but on the specific credential (identity attribute) or combination of credentials that a user is about the reveal to a Service Provider.

When we are talking about risk assessment with Idemix we need to know whether a given attribute can be inferred by a Service Provider and what is the confidence probability based on the combination of the credentials that a user is about to share with this Service Provider. For this we assume that all the Service Providers and Identity Providers have acquired the distribution of identity attributes and the distribution of the combinations of identity attributes (the frequency this combination of attributes occurs in the complete population) from external means.

For the computation of risk for de-anonymization in the case of Idemix we need to make sure, for each session separately, that when a user reveals some identity attributes (credentials) to the verifying Identity Provider or the Service Provider they do not learn any other of the user’s identity attributes that he does not want to reveal to them. In general, we want whenever a user goes to an Identity Provider to inform him for example –given the distribution of identity attributes - that this combination of identity attributes that you are going to reveal are also appear 10 times across the population and if you reveal them, the Service Provider has the knowledge to guess the values of some more of your identity attributes. This computation should be offered by the Identity Profile Management module and the Idemix client that runs on the verifying Identity Provider should invoke the Identity Management module and display the calculated risk that the given Identity Provider or Service Provider can infer other identity attributes of the user.

4 Consent Management

The Consent Management Module is a tool to properly define and utilize consent policies on user attributes. These are needed to be evaluated when request for transfer is made. In order to achieve that, the tool is divided into three components. The Back-end that manages policies and requests, a

mobile front-end and a web front-end for the user to easily define his/her consent policies of its attributes.

The policies are stored in xml format under the XACML protocol and also at the database for easier management. When a request is made to transfer an attribute from on idp (source) to another (destination), a check is performed in the back-end to evaluate the request. The evaluation checks, a) if the idp (source) gives its consent to transfer the attribute, and b) if the user allows the transfer also. Only, if it passes both the checks the transfer is valid for execution.

The module also provides policy recommendations to the user. The functionality makes recommendations based on the collective amount of active policies of all users and the request logs for transfer requests. These are utilized to train a Markov Logic Network (MLN). The MLN is a machine learning model that infers the weights of importance of policies. Thus when trained the recommended policies (the ones with high valued weights) are essentially the most probable ones. Furthermore, the recommendations are improved as more users are on the platform by retraining the network when enough data is present.

4.1 Consent Management Back-End

The Back-End includes a collection of REST API functions that can be grouped into to three categories.

- **User Policy REST operations:** It allows the users to manage policies that define: (i) which identity attributes can be revealed to specific Service Providers and (ii) which identity attributes can be transferred between specific identity providers. For example, the user may define that it does not wish to reveal his/her address to Identity Providers with an Identity Assurance Level (IAL) below a given threshold.
- **Identity Providers REST operations:** It allows Identity Providers to manage policies that define whether specific attributes can be revealed to Relying Parties or whether identity attributes can be transferred among specific Identity Providers or whether the Identity Provider can issue cryptographic credentials that involve specific identity attributes. For example, policies can be in the following form: (i) be able to transfer attribute A to IDP or SP or IDC and (ii) be able to Issue credentials for attribute A with protocol Z. The former is used from IDPs who do not wish certain attributes, or attributes with specific IAL or Authenticator Assurance Level (AAL), to be revealed to certain unauthorized parties or other entities that allow authentication below a certain IAL. For example, the Social Security Administration (ID provider) provides the social security number that should be revealed only to SPs that have high authentication assurances, such as banks. The latter form is used, when an IDP decides that it does not want the attributes of its users to be proven using Idemix/U-Prove and that they should be proven through the IDP via OAuth instead (so that the IDP always knows where these credentials have been shown). It also allows the IDC to manage consent policies similarly to the Identity Providers.
- **Evaluation REST operations**: Based on the consent policies defined by the users, IDPs and IDC we evaluate requests to transfer attributes or issue credentials. When a request is made we draw the relevant consent policies from the database and create the up to date XACML

files in order to evaluate the request. This is the Policy Decision Point under the XACML terminology.

Below we describe in more details the supported REST operations of the Consent Management Module.

4.1.1 High Level Operations

The high level operations that the current version of the Access Control Reasoning Tool for Consent Management supports can be divided to the following categories:

1. User Policy REST operations
    a. Create a Blacklist user policy for an identity provider
    b. Create a Blacklist user policy for a service provider
    c. Create a Whitelist user policy for an identity provider
    d. Create a Whitelist user policy for a service provider
    e. View a Blacklist user policy
    f. View a Whitelist user policy
    g. Delete a Blacklist user policy
    h. Delete a Whitelist user policy
    i. View all users Blacklist policies
    j. View all users Whitelist policies

2. Identity Providers REST operations
    a. Create a Blacklist identity provider policy for another identity provider
    b. Create a Blacklist identity provider policy for a service provider
    c. Create a Blacklist identity provider policy for issuing credentials
    d. Create a Whitelist identity provider policy for another identity provider
    e. Create a Whitelist identity provider policy for a service provider
    f. Create a Whitelist identity provider policy for issuing credentials
    g. View a Blacklist identity provider policy
    h. View a Whitelist identity provider policy
    i. Delete a Blacklist identity provider policy
    j. Delete a Whitelist identity provider policy
    k. View all identity providers Blacklist policies
    l. View all identity providers Whitelist policies
3. Evaluation REST operations
    a. Evaluate request to transfer attributes
    b. Evaluate request to issue credentials

4.1.2 Create Policy

Description: This REST operation is used to create a Policy. For example, user with id 3 wants to deny transfer of attribute surname to identity provider facebook.

Operation: POST
http://consolidator.recred.eu/idc_consent_management/create_policies/<creator_type>/<list_type>

Request:

```
POST
http://consolidator.recred.eu/idc_consent_management/create_policies/<creator_type>/<list_type>
```

```
Accept: application/json

Authorization: Bearer <access_token>
```

Description of Elements in Request URI:

Element	Description	Valid Value
creator_type	The creator of the policy	[user, idp]
list_type	The type of the policy to be created	[blacklist, whitelist]

Following are 2 examples including all creator type options available on the request url

CREATE A USER POLICY

Request:

```
POST http://consolidator.recred.eu/idc_consent_management/create_policies/user/<list_type>

Accept: application/json

Authorization: Bearer <access_token>
```

Description of Elements in Request URI:

Element	Description	Valid Value
list_type	The type of the policy to be created	[blacklist, whitelist]

Request Body:

```
{
          "user_id":3,
          "idp_a":"twitter",
          "attr_name":"surname",
          "AAL_attr":"2",
          "AAL_attr_func":"less-than-or-equal",
          "IAL_attr":"1",
          "IAL_attr_func":"less-than-or-equal",
          "attr_cs":"1",
          "attr_cs_func":"greater-than-or-equal",
          "idp_b":"facebook",
          "AAL_idp_b":"2",
          "AAL_idp_b_func":"less-than-or-equal",
          "IAL_idp_b":"2",
```

```
        "IAL_idp_b_func":"less-than-or-equal",
        "exp_date":"15-2-18"  }
```

Response Body:

```
200
Content-Type: application/json
{
  "Success": {
    "Created": "Policy",
    "creator_type": "user",
    "list_type": "blacklist"  }}
```

Description of Elements in Request Body

Element	Description	Required	Valid Value
user_id	User’s unique Identifier	Yes	String
idp_a	The identity provider that is the source of the attribute	Yes	String
attr_name	The name of the attribute	*	String
AAL_attr	The Authenticator Assurance Level of the attribute must have	*	[1,2,3]
IAL_attr	The Identity Assurance Level of the attribute must have	*	[1,2,3]
AAL_attr_func	The function to execute on the level of Authenticator Assurance of the attribute. Requires AAL_attr attribute. If AAL_attr_func not specified default function is equal.	No	[greater-than-or-equal, less-than-or-equal]
IAL_attr_func	The function to execute on the level of Identity Assurance of the attribute. Requires IAL_attr attribute. If IAL_attr_func not specified default function is equal.	No	[greater-than-or-equal, less-than-or-equal]
attr_cs	The confidence score that the attribute must have	*	[0-100]
attr_cs_func	The function to execute on the confidence score of the attribute. Requires attr_cs attribute. If attr_cs_func not specified default function is equal.	No	[greater-than-or-equal, less-than-or-equal]
idp_b	The identity provider that is the destination of the attribute	**	[greater-than-or-equal, less-than-or-equal]
AAL_idp_b	The Authenticator Assurance Level of the identity provider must have	**	[1,2,3]
IAL_idp_b	The Identity Assurance Level of the identity provider must have	**	[1,2,3]
AAL_idp_b_func	The function to execute on the level of Authenticator Assurance of the provider. Requires AAL_idp_b attribute. If AAL_idp_b_func not	No	[greater-than-or-equal, less-than-or-equal]

	specified default function is equal.		
IAL_idp_b_func	The function to execute on the level of Identity Assurance of the provider. Requires IAL_idp_b attribute. If IAL_idp_b_func not specified default function is equal.	No	[greater-than-or-equal, less-than-or-equal]
sp	The service provider that is the destination of the attribute	**	String
exp_date	The day the policy expires	No	Date (YYYY-MM-DD)

* At least one

** At least one of [idp_b, AAL_attr, AAL_attr] or [sp]

CREATE AN IDP POLICY

Request:

```
POST http://consolidator.recred.eu/idc_consent_management/create_policies/idp/<list_type>
Accept: application/json
Authorization: Bearer <access_token>
```

Description of Elements in Request URI:

Element	Description	Valid Value
list_type	The type of the policy to be created	[blacklist, whitelist]

Request Body:

```
{
         "idp_a":"twitter",
         "attr_name":"surname",
         "AAL_attr":"2",
         "AAL_attr_func":"less-than-or-equal",
         "IAL_attr":"1",
         "IAL_attr_func":"less-than-or-equal",
         "attr_cs":"1",
         "attr_cs_func":"greater-than-or-equal",
         "idp_b":"facebook",
         "AAL_idp_b":"2",
         "AAL_idp_b_func":"less-than-or-equal",
         "IAL_idp_b":"2",
         "IAL_idp_b_func":"less-than-or-equal"
         "exp_date":"15-2-18"  }
```

Response Body:

```
200
Content-Type: application/json

{
  "Success": {
    "Created": "Policy",
```

```
    "creator_type": "idp",
    "list_type": "blacklist"}
}
```

Description of Elements in Request Body

Element	Description	Required	Valid Value
idp_a	The identity provider that is the source of the attribute	Yes	String
attr_name	The name of the attribute	*	String
AAL_attr	The Authenticator Assurance Level of the attribute must have	*	[1,2,3]
IAL_attr	The Identity Assurance Level of the attribute must have	*	[1,2,3]
AAL_attr_func	The function to execute on the level of Authenticator Assurance of the attribute. Requires AAL_attr attribute. If AAL_attr_func not specified default function is equal.	No	[greater-than-or-equal, less-than-or-equal]
IAL_attr_func	The function to execute on the level of Identity Assurance of the attribute. Requires IAL_attr attribute. If IAL_attr_func not specified default function is equal.	No	[greater-than-or-equal, less-than-or-equal]
attr_cs	The confidence score that the attribute must have	*	[0-100]
attr_cs_func	The function to execute on the confidence score of the attribute. Requires attr_cs attribute. If attr_cs_func not specified default function is equal.	No	[greater-than-or-equal, less-than-or-equal]
idp_b	The identity provider that is the destination of the attribute	**	[greater-than-or-equal, less-than-or-equal]
AAL_idp_b	The Authenticator Assurance Level of the identity provider must have	**	[1,2,3]
IAL_idp_b	The Identity Assurance Level of the identity provider must have	**	[1,2,3]
AAL_idp_b_func	The function to execute on the level of Authenticator Assurance of the provider. Requires AAL_idp_b attribute. If AAL_idp_b_func not specified default function is equal.	No	[greater-than-or-equal, less-than-or-equal]
IAL_idp_b_func	The function to execute on the level of Identity Assurance of the provider. Requires IAL_idp_b attribute. If IAL_idp_b_func not specified default function is equal.	No	[greater-than-or-equal, less-than-or-equal]
sp	The service provider that is the destination of the attribute	**	String

protocol	The protocol in which to issue credentials	**	
exp_date	The day the policy expires	No	Date (YYYY-MM-DD)

* At least one
** At least one of [idp_b, AAL_attr, IAL_attr] or [sp] or [protocol]

4.1.3 View Policy

Description: This REST operation is used to view Policies. For example, I want to view all the blacklist policies of user 5.

Operation: GET

http://consolidator.recred.eu/idc_consent_management/show_policies/<creator_type>/<list_type>/<creator_id>

Request:

```
GET
http://consolidator.recred.eu/idc_consent_management/show_policies/<creator_type>/<list_type>/<creator_id>
Accept: application/json
Authorization: Bearer <access_token>
```

Description of Elements in Request URI:

Element	Description	Valid Value
creator_type	The creator of the policy	[user, idp]
list_type	The type of the policy to be created	[blacklist, whitelist]
creator_id	The creator id. If creator_id is not specified it returns the complete list of policies	String

Following are 2 examples including all creator type options available on the request url

SHOW USER POLICIES

Request:

```
GET
http://consolidator.recred.eu/idc_consent_management/show_policies/user/<list_type>/<creator_id>
Accept: application/json
Authorization: Bearer <access_token>
```

Description of Elements in Request URI:

Element	Description	Valid Value
list_type	The type of the policy to be created	[blacklist, whitelist]
creator_id	The creator id. If creator_id is not specified it returns the complete list of policies	String

Response Body:

```
200
Content-Type: application/json
[
  {
    "attr_name": "surname",
    "idp_a": "twitter",
    "idp_b": "facebook",
    "user_id": "3",
    "users_blacklist_id": 10  },
  {
    "attr_cs": "1",
    "attr_cs_func": "greater-than-or-equal",
    "AAL_attr":"2",
    "AAL_attr_func":"less-than-or-equal",
    "IAL_attr":"1",
    "IAL_attr_func":"less-than-or-equal",
    "attr_name": "surname",
    "exp_date": "15-2-2018",
    "idp_a": "twitter",
    "idp_b": "facebook",
    "AAL_idp_b":"2",
    "AAL_idp_b_func":"less-than-or-equal",
    "IAL_idp_b":"2",
    "IAL_idp_b_func":"less-than-or-equal"
    "user_id": "3",
    "users_blacklist_id": 11  }]
```

Description of Elements in Response Body

Element	Description	Valid Value
users_blacklist_id	The blacklist unique identifier for the users	Integer
users_whitelist_id	The whitelist unique identifier for the users	Integer
user_id	User’s unique Identifier	String
idp_a	The identity provider that is the source of the attribute	String
attr_name	The name of the attribute	String
AAL_attr	The Authenticator Assurance Level of the attribute must have	[1, 2, 3]
IAL_attr	The Identity Assurance Level of the attribute must have	[1, 2, 3]
AAL_attr_func	The function to execute on the level of Authenticator Assurance of the attribute. Requires AAL_attr attribute. If AAL_attr_func	[greater-than-or-equal, less-than-or-equal]

	not specified default function is equal.	
IAL_attr_func	The function to execute on the level of Identity Assurance of the attribute. Requires IAL_attr attribute. If IAL_attr_func not specified default function is equal.	[greater-than-or-equal, less-than-or-equal]
attr_cs	The confidence score that the attribute must have	[0-100]
attr_cs_func	The function to execute on the confidence score of the attribute. Requires attr_cs attribute. If attr_cs_func not specified default function is equal.	[greater-than-or-equal, less-than-or-equal]
idp_b	The identity provider that is the destination of the attribute	String
AAL_idp_b	The Authenticator Assurance Level of the identity provider must have	[1, 2, 3]
IAL_idp_b	The Identity Assurance Level of the identity provider must have	[1, 2, 3]
AAL_idp_b_func	The function to execute on the level of Authenticator Assurance of the provider. Requires AAL_idp_b attribute. If AAL_idp_b_func not specified default function is equal.	[greater-than-or-equal, less-than-or-equal]
IAL_idp_b_func	The function to execute on the level of Identity Assurance of the provider. Requires IAL_idp_b attribute. If IAL_idp_b_func not specified default function is equal.	[greater-than-or-equal, less-than-or-equal]
sp	The service provider that is the destination of the attribute	String
exp_date	The day the policy expires	Date (YYYY-MM-DD)

SHOW IDP POLICIES

Request:

```
GET
http://consolidator.recred.eu/idc_consent_management/show_policies/idp/<list_type>/<creator_id>
Accept: application/json
Authorization: Bearer <access_token>
```

Description of Elements in Request URI:

Element	Description	Valid Value
list_type	The type of the policy to be created	[blacklist, whitelist]
creator_id	The creator id. If creator_id is not specified it returns the complete list of policies	String

Response Body:

```
200
Content-Type: application/json

[
  {
    "attr_name": "surname",
    "idp_a": "twitter",
    "idp_b": "facebook",
    "idps_blacklist_id": 10  },
  {
    "attr_cs": "1",
    "attr_cs_func": "greater-than-or-equal",
    "AAL_attr":"2",
    "AAL_attr_func":"less-than-or-equal",
    "IAL_attr":"1",
    "IAL_attr_func":"less-than-or-equal",
    "attr_name": "surname",
    "exp_date": "15-2-2018",
    "idp_a": "twitter",
    "idp_b": "facebook",
    "AAL_idp_b":"2",
    "AAL_idp_b_func":"less-than-or-equal",
    "IAL_idp_b":"2",
    "IAL_idp_b_func":"less-than-or-equal"
    "idps_blacklist_id": 11  }]
```

Description of Elements in Response Body

Element	Description	Valid Value
idps_blacklist_id	The blacklist unique identifier for the identity providers	Integer
idps_whitelist_id	The whitelist unique identifier for the identity providers	Integer
idp_a	The identity provider that is the source of the attribute	String
attr_name	The name of the attribute	String
AAL_attr	The Authenticator Assurance Level of the attribute must have	[1, 2, 3]
IAL_attr	The Identity Assurance Level of the attribute must have	[1, 2, 3]
AAL_attr_func	The function to execute on the level of Authenticator Assurance of the attribute. Requires AAL_attr attribute. If AAL_attr_func not specified default function is equal.	[greater-than-or-equal, less-than-or-equal]
IAL_attr_func	The function to execute on the level of Identity Assurance of the attribute. Requires IAL_attr attribute. If IAL_attr_func not specified default function is equal.	[greater-than-or-equal, less-than-or-equal]
attr_cs	The confidence score that the attribute must have	[0-100]
attr_cs_func	The function to execute on the confidence score of the attribute. Requires attr_cs attribute. If attr_cs_func not specified	[greater-than-or-equal, less-than-or-equal]

	default function is equal.	
idp_b	The identity provider that is the destination of the attribute	String
AAL_idp_b	The Authenticator Assurance Level of the identity provider must have	[1, 2, 3]
IAL_idp_b	The Identity Assurance Level of the identity provider must have	[1, 2, 3]
AAL_idp_b_func	The function to execute on the level of Authenticator Assurance of the provider. Requires AAL_idp_b attribute. If AAL_idp_b_func not specified default function is equal.	[greater-than-or-equal, less-than-or-equal]
IAL_idp_b_func	The function to execute on the level of Identity Assurance of the provider. Requires IAL_idp_b attribute. If IAL_idp_b_func not specified default function is equal.	[greater-than-or-equal, less-than-or-equal]
sp	The service provider that is the destination of the attribute	String
protocol	The protocol in which to issue credentials	String
exp_date	The day the policy expires	Date (YYYY-MM-DD)

4.1.4 Delete Policy

Description: This REST operation is used to delete Policies. For example, I want to delete the blacklist policy with id 14.

Operation: DELETE

http://consolidator.recred.eu/idc_consent_management/delete_policies/<creator_type>/<list_type>/<delete_id>

Request:

```
DELETE
http://consolidator.recred.eu/idc_consent_management/delete_policies/<creator_type>/<list_type>/<delete_id>
Accept: application/json
Authorization: Bearer <access_token>
```

Response Body:

```
200
Content-Type: application/json

{
  "Success": {
    "Deleted": "11",
    "creator_type": "user",
    "list_type": "blacklist"  }}
```

Description of Elements in Request URI:

Element	Description	Valid Value

creator_type	The creator of the policy	[user, idp]
list_type	The type of the policy to be created	[blacklist, whitelist]
delete_id	The unique id of the policy. It can be the value of [users_blacklist_id, users_whiteklist_id, idps_blacklist_id, idps_whitelist_id]	Integer

4.1.5 Evaluation Requests

Description: This REST operation is used to evaluate
1) The request to transfer attributes based on user’s and identity provider’s Policies
2) The request to issue credentials based on identity provider’s policies
Operation: POST http://consolidator.recred.eu/idc_consent_management/requests/<action_type>
Request:

```
POST http://consolidator.recred.eu/idc_consent_management/requests/<action_type>
Accept: application/json
Authorization: Bearer <access_token>
```

Description of Elements in Request URI:

Element	Description	Valid Value
action_type	The action to be evaluated	[transfer, issue]

Following are 2 examples including all action type options available on the request url

TRANSFER REQUEST

Request:

```
POST http://consolidator.recred.eu/idc_consent_management/requests/transfer
Accept: application/json
Authorization: Bearer <access_token>
```

Request Body:

```
{
          "user_id":3,
          "idp_a":"twitter",
          "attr_name":"surname",
          "AAL_attr":"2",
         "IAL_attr":"1",
          "attr_name_cs":"80.4",
          "idp_b":"facebook",
```

```
          "AAL_idp_b":"2",
          "IAL_idp_b":"2" }
```

Response Body:

```
200
Content-Type: application/json

{
  "idps_blacklist": "NOT_APPLICABLE",
  "idps_whitelist": "APPLICABLE",
  "users_blacklist": "NOT_APPLICABLE",
  "users_whitelist": "APPLICABLE" }
```

Description of Elements in Request Body

Element	Description	Required	Valid Value
user_id	User’s unique Identifier	Yes	String
idp_a	The identity provider that is the source of the attribute	Yes	String
attr_name	The name of the attribute	Yes	String
AAL_attr	The level of Authenticator Assurance that the attribute (attr_name) of the user (user_id) has. If this attribute is not given it takes the lowest value (1)	No	[1, 2, 3]
IAL_attr	The level of Identity Assurance that the attribute (attr_name) of the user (user_id) has. If this attribute is not given it takes the lowest value (1)	No	[1, 2, 3]
attr_name_cs	The confidence score that the attribute (attr_name) of the user (user_id) has. If this attribute is not given it takes the lowest value (0)	No	[0-100]
idp_b	The identity provider that is the destination of the attribute	*	String
AAL_idp_b	The level of Authenticator Assurance that the identity provider (idp_b) has. If this attribute is not given it takes the lowest value (1)	No	[1, 2, 3]
IAL_idp_b	The level of Identity Assurance that the identity provider (idp_b) has. If this attribute is not given it takes the lowest value (1)	No	[1, 2, 3]
sp	The service provider that is the destination of the attribute	*	String

* At least one [idp_b, sp]

Description of Elements in Response Body

Element	Description	Valid Value

users_blacklist	Returns the evaluation of the user’s blacklist policies. Returns APPLICABLE if the user denies the request	[APPLICABLE, NOT-APPLICABLE]
users_whitelist	Returns the evaluation of the user’s whitelist policies. Returns APPLICABLE if the user permits the request	[APPLICABLE, NOT-APPLICABLE]
idps_blacklist	Returns the evaluation of the identity provider’s blacklist policies. Returns APPLICABLE if the identity provider denies the request	[APPLICABLE, NOT-APPLICABLE]
idps_whitelist	Returns the evaluation of the identity provider’s whitelist policies. Returns APPLICABLE if the identity provider permit the request	[APPLICABLE, NOT-APPLICABLE]

CREDENTIAL REQUEST

Request:

```
POST http://consolidator.recred.eu/idc_consent_management/requests/issue
Accept: application/json
Authorization: Bearer <access_token>
```

Request Body:

```
{
          "idp_a":"twitter",
          "attr_name":"surname",
          "AAL_attr":"2",
          "IAL_attr":"1",
          "attr_name_cs":"80.4",
          "protocol":"uprove"  }
```

Response Body:

```
200
Content-Type: application/json

{
  "idps_blacklist": "NOT_APPLICABLE",
  "idps_whitelist": "APPLICABLE"  }
```

Description of Elements in Request Body

Element	Description	Required	Valid Value
idp_a	The identity provider that is the source of the attribute	Yes	String
attr_name	The name of the attribute	Yes	String
AAL_attr	The level of Authenticator Assurance that the attribute (attr_name) of the user (user_id) has. If this attribute is not given it takes the lowest value (1)	No	[1, 2, 3]

IAL_attr	The level of Identity Assurance that the attribute (attr_name) of the user (user_id) has. If this attribute is not given it takes the lowest value (1)	No	[1, 2, 3]
attr_name_cs	The confidence score that the attribute (attr_name) of the user (user_id) has. If this attribute is not given it takes the lowest value (0)	No	[0-100]
protocol	The protocol in which to issue credentials	Yes	String

Description of Elements in Response Body

Element	Description	Valid Value
idps_blacklist	Returns the evaluation of the identity provider's blacklist policies. Returns APPLICABLE if the identity provider denies the request	[APPLICABLE, NOT-APPLICABLE]
idps_whitelist	Returns the evaluation of the identity provider's whitelist policies. Returns APPLICABLE if the identity provider permit the request	[APPLICABLE, NOT-APPLICABLE]

4.2 Consent Management mobile application

The Consent Management mobile front-end is an Android mobile app that allows the users to define their consent for their various identity attributes, by defining policies regarding the Identity Providers and Service Providers to which their attributes should be revealed.

The mobile app communicates with the Consent Management back-end, allowing the end-users to access the following functionality:

- Create new consent policies, which whitelist or blacklist specific identity attributes (or groups of attributes) from Identity Providers and/or Service Providers
- View the consent policies that he has created
- Modify or delete consent policies that he has already created

4.2.1 Create new Consent Policy

The user can create new consent policies by defining the following details (Figure 40**Error! Reference source not found.**):

- **Source Identity Providers**: This is the Identity Provider that maintains the attribute(s) for which the new consent policy will be created.
- **Type of Policy**: The user can select between the following options.
  - Blacklisting Policy, meaning that the selected attributes will always be blocked from the selected IdPs / SPs
  - Whitelisting Policy, meaning that the selected attributes will always be revealed to the selected IdPs / SPs

- **Selected Attribute(s)**: The user can select the attribute(s) for which the new consent policies will be created. There are three alternative options.
  - Select a specific attribute,
  - Select many attributes, according to their LoA,
  - Select many attributes, according to their Confidence Score
- **Identity / Service Provider(s)**: The user can select the target IdP(s) or SP(s) of the new consent policies. There are four alternative options.
  - Select a specific Identity Provider,
  - Select many Identity Providers, according to their LoA,
  - Select a specific Service Provider,
  - Select many Service Providers, according to their LoA,

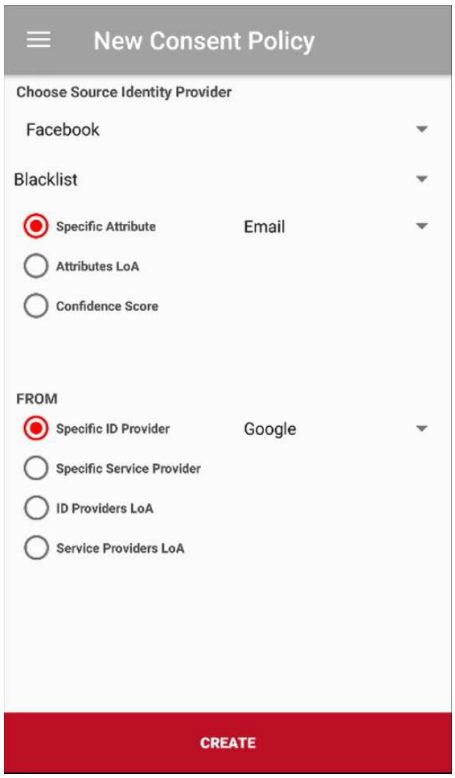

Figure 41 Create new Consent Policy

4.2.2 View and Manage Consent Policies

The user can see a list with all the consent policies that she has created. These attributes are grouped under whitelisting policies and blacklisting policies.

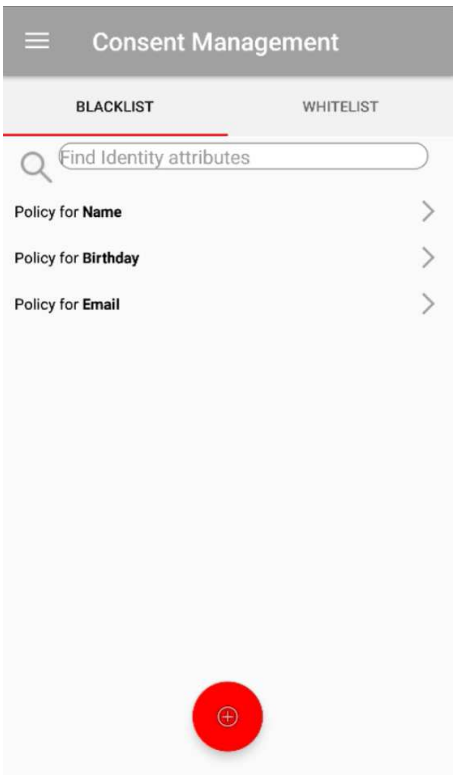

Figure 42:List of Consent Policies

Finally, the user can see more details regarding a consent policy by tapping on it (Figure 42**Error! Reference source not found.**). He can also long-tap on a policy, in order to delete it (Figure 43).

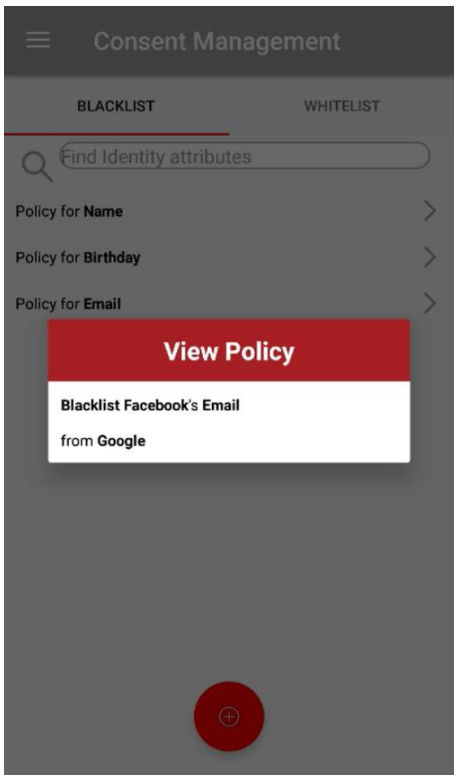

Figure 43: View policy details

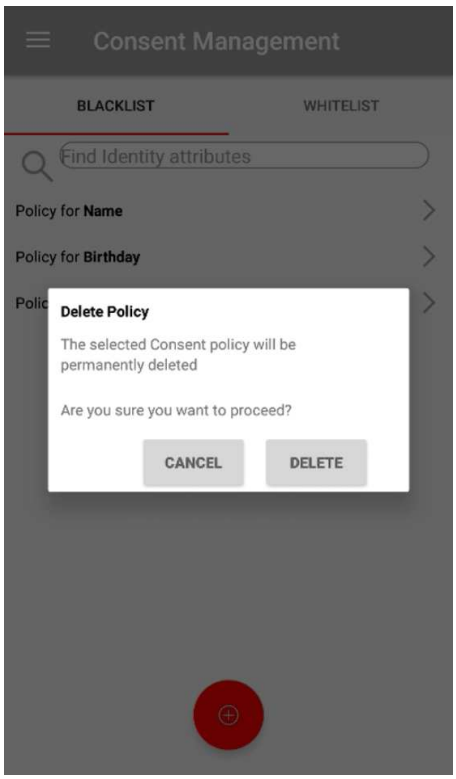

Figure 44: Delete a Consent Policy

4.3 Consent Management Web Interface

In this section we will describe and demonstrate the Web interface of the Consent Management module. At the main page of the Identity Consolidator, the user can navigate to the Consent Management Module by clicking the dedicated icon.

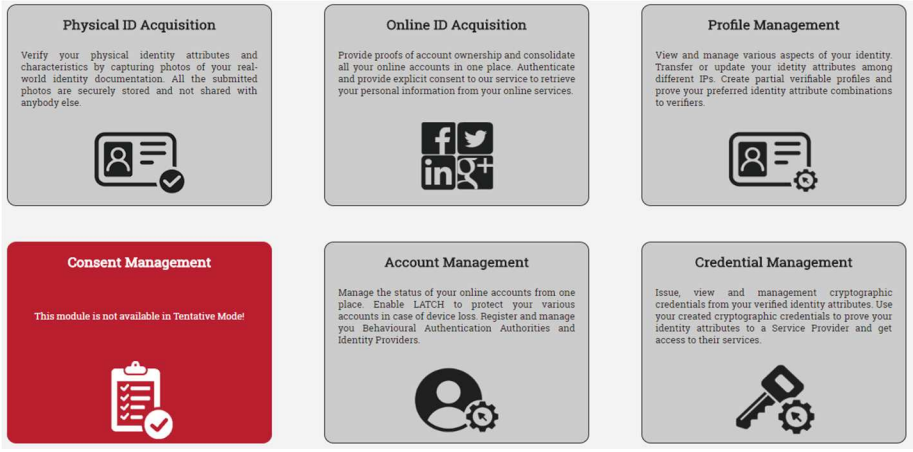

Figure 45. Consent Management Selection in IDC

4.3.1 Create Policy

In the Create Policies tab of the web interface, the user can define their consent policies. The policies are stored in xml under the XACML protocol and at the database for easier management. Specifically, a user is able to select the type of the policy (permit or deny), the source and destination IdP as well as the involved identity attributes. Essentially this functionality enables the user to set his preferences for the identity attributes transfer among Identity Providers.

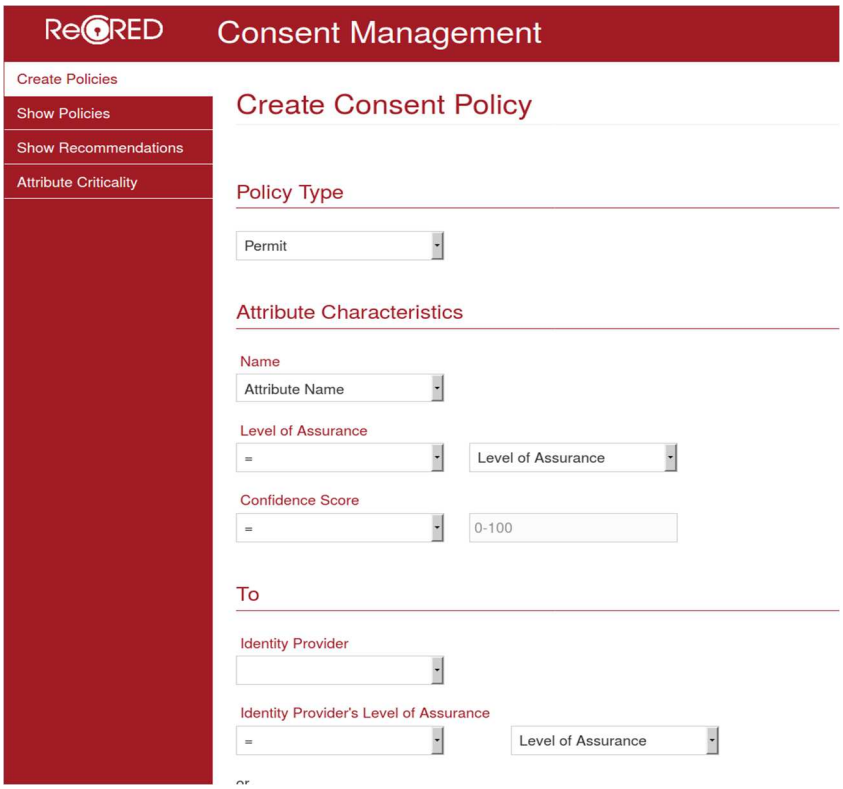

Figure 46. CMM Policy Creation

4.3.2 View and Delete Consent Policies

After creating some consent policies, the user is able to view the already defined consent policies as well as deleting them. The figure below demonstrates the web interface for this functionality. Each consent policy can be deleted by clicking on the delete button.

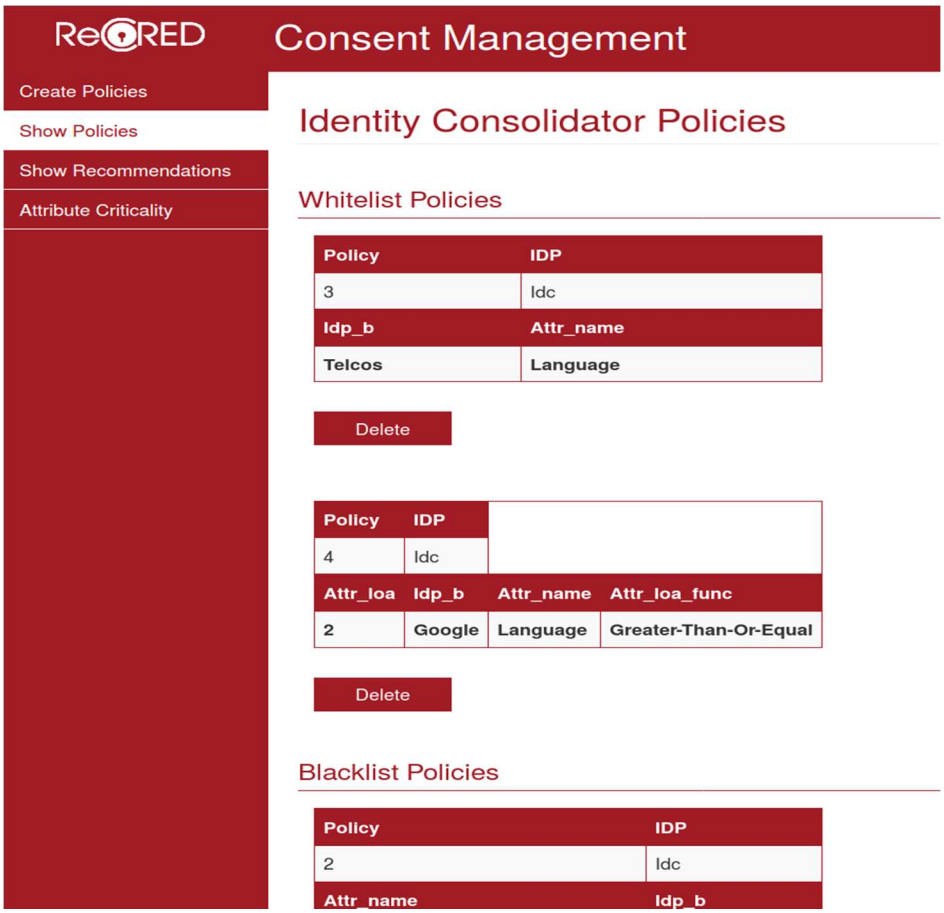

Figure 47. CMM Policy View/Delete

4.3.3.Consent Policies Recommendations

As the consent policies increase we want to assist the user in managing existing policies and recommend new ones. Rule based recommendations are simple and easy to derive in small scale systems. However, when the system gets more complex it becomes more difficult to find rules that will provide recommendations. Machine learning algorithms are used to derive the dependencies between information in an automate manner, can assist in deriving policy recommendations. In that regard, we utilize a machine learning model to provide policy recommendations to the user.

The model is based on a Markov Logic Network where each policy is assigned a weight of importance based on the active policies of all the users and the (transfer) requests logs. By assigning an importance weight on each policy we can provide recommendations to the user if the weight passes a specific threshold. This essentially means that as more users are active on the system the more accurate the recommendations are. Is worth mentioning that the threshold of each attribute is defined by the IDC's administrator. To limit the computational requirements of training the model, we resort to an offline process. That is, the consent policy recommendations and their associate weights are computed in weekly intervals or when enough requests are made that render the re-training essential.

Markov Logic Networks (MLN) combine Markov Networks with first-order logic; in our case the consent policies. In more detail, Markov Networks are undirected probabilistic graphical models that represent a joint probability distribution over a set of random variables; in our case the attributes. To soften the logic MLN associate a weight with each policy. Also, for the training procedure we require a finite set of constraints (domain) thus we transform the presence of any continuous attributes to discrete ones. This properly defines the domain and we train the model by maximizing the likelihood of the model given the data; in our case the request logs and the active policies. For recommendations, we search the whole domain space for other high probability policies and present them as recommendations.

At the Show Recommendations tab we present to the user consent recommendations based on the Markov Logic Network machine learning algorithm.

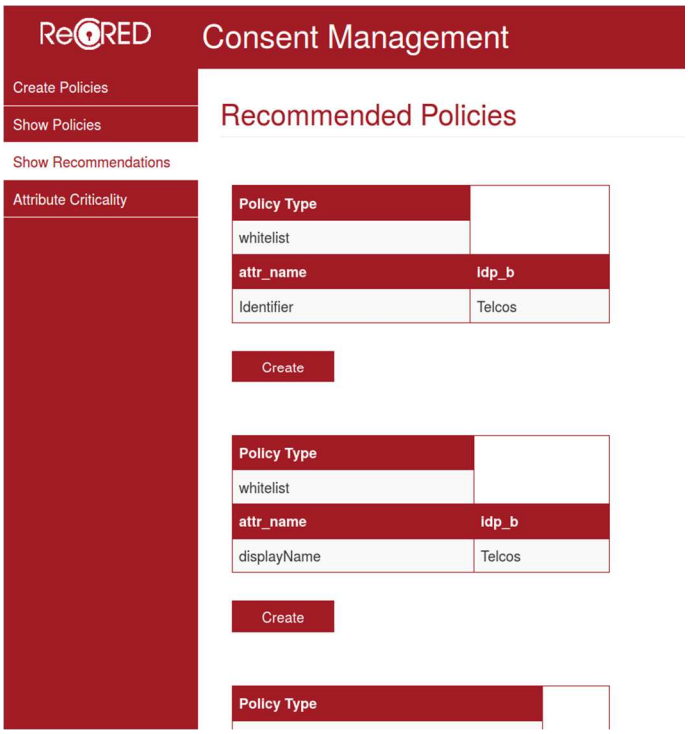

Figure 48. CMM Policy Recommendation

4.3.4 Attribute Criticality

The identity consolidator administrator (only) can define the attribute importance. This will limit the policy recommendations for the critical resources. In essence, this increases the threshold on the policy weight (derived by the MLN) in order to be recommended.

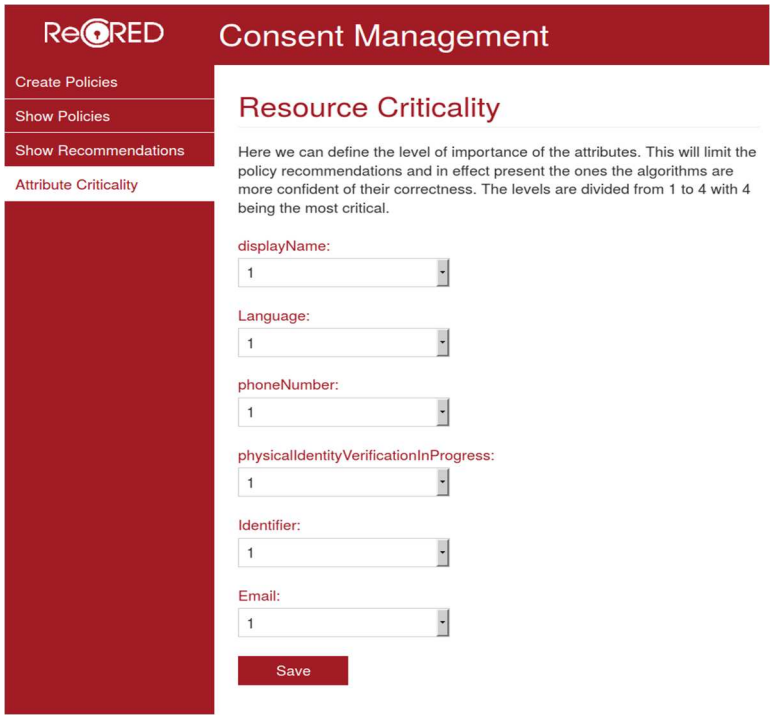

Figure 49. CMM Resource Criticality

5 Conclusion

The Identity and Profile Management is one of the major components of the ReCRED platform. This deliverable, “Online identity and profile management” describes the architecture, design and provides the implementation overview and details of the Identity and Profile Management.

The description and use cases accompanied by screenshots for all the versions of the applications for the Identity and profile management, namely the web and mobile ones are provided. A high description of the major APIs such as the transfer attributes, risks calculations and consent management APIs are included in this deliverable. Finally in this deliverable a detailed reference to the communication with other components relevant to the ReCRED architecture and internal ones is made.

6 References

[1] *OpenID Connect*; http://openid.net/connect/
[2] *The OAuth 1.0 Protocol*; https://tools.ietf.org/html/rfc5849
[3] *The OAuth 2.0 Authorization Framework*; https://tools.ietf.org/html/rfc6749
[4] *OpenAM*; https://forgerock.org/openam/
[5] *LATCH*; https://www.elevenpaths.com/technology/latch/index.html
[6] *OData*; http://www.odata.org/documentation/odata-version-2-0/uri-conventions/
[7] *Material UI*; www.material-ui.com
[8] *Apache Olingo*; http://olingo.apache.org
[9] *Discrete uniform distribution*; https://en.wikipedia.org/wiki/Discrete_uniform_distribution