Account linking

or Identity Consolidation, Identity Linking

For end-users, this functionality is called Account linking, but technically it is Digital Identity Linking (but the user does not know what Digital Identity is, so it is much more natural for him to use the word account).

To allow two or more identities to be linked, the user must prove that all identities belong to him. We solve this by logging in. Thus, the identity linking application gradually prompts the user to log in with the identities he wants to connect.

When linking identities, the user demonstrates two or more identities that he wants to merge. An important aspect of identity when merging is whether the identity is registered in the IAM system (Perun). The only possible situation where it technically makes sense to combine identities is when the user has exactly one identity that is registered and all the others are unregistered in the IAM system.
End states of merging identities depending on the provided identities: (described on the merging of just two identities, the order of identities does not matter)

1. The connection is successful - 1. Identity is registered 2. Identity is not registered
2. Identities are already connected - 1st and 2nd identities are registered and these identities are already connected to the same user
3. Identities cannot be connected automatically - 1st and 2nd identities are registered but are connected to two different users, user support intervention is required
4. We have nothing to connect to identities - both identities are not registered in the system, so we have nothing to connect to, we require the user to have an identity that is registered in the system

Benefits of merging accounts - Why we actually need it

Change or loss of identity

One of the main advantages is obtaining new verified information from linked identities. Each newly added identity enriches a number of identity attributes. This is a great advantage, for example, when a user loses his identity or when he changes jobs. If a user loses his identity, he can easily log in with another one - and get into the services he is used to using. When changing jobs, moving from an academic institution to a commercial sphere or another academic institution, the user retains an account in the IAM system and access to services that were not dependent on the attributes of the identity that was revoked (attributes of this identity are changed or canceled in the IAM system). In the case of a user who is a student at two different universities, this user gains the advantage of both identities (for example, access to university buildings).

Identity provider failure

It is natural that the identity provider sometimes fails. For users, this means that they cannot log in with this identity provider. If it has other identities attached, then it is not a problem for the user to log in with another identity provider.

User experience

The user can log in with any of their connected identities without seeing a difference in functionality or availability.

Ways the user gets to connect identities

Combining identities is part of some pre-existing flow. Therefore, it is important to understand how the user actually gets to connect identities.

Through the registration system

One way a user enters the registration system is to access one of the services (for example, o365). The user successfully authenticates with IdP, however, the access management system stops the user at this point, explains to the user that he must register to access the services - and sends him to the registration system to register. The registration system also has the function of searching for similar users in the system to prevent users from registering twice. This is an undesirable condition. Therefore, the registration system will offer him identities that could be his. If the user remembers that it is one of his, the account linking begins.
Another way is that the user is sent an email with the application. It enters the registration system through a link. Even in this case, the registration system will ask him if one of the similar identities is not his.

Through the user profile

A user profile is an application for managing a user's personal settings. There is also an identity management section. The user has an overview of his already connected identities and also has the option to connect to a new identity.

Direct access through a link to the application

This is the case when a user has somehow obtained a link to an application designed to merge identities. A link to the account linking application can be sent to him by user support in case of various problems.

Real cases of using the application for linking accounts

1. The user accesses the service -> logs in -> the identity he logged in is not registered in the IAM system (Perun) -> we will send him to register -> the registration system will offer similar, already registered users -> the user remembers that he has already registered so he decides he wants to link accounts ->

    a. the user successfully links the accounts (logs in with the correct registered account)
    b. the user does not manage to link accounts - he does not remember what he actually registered -> writes an email to user support -> support sends him a URL with an application designed for a single purpose - Account linking
2. The user has a registered account A and at the same time independently has a registered account B - these two accounts are not connected. A colleague at the user's conference informs him that he could link these accounts (to make it more convenient) and sends him a link to the account linking application.

Technical issues

Proof of ownership of identity

To allow two or more identities to be linked, the user needs to prove that all identities belong to him. We solve this by logging in. Thus, the identity linking application gradually prompts the user to log in with the identities he wants to connect.

Impossibility to link identities from the same identity provider

When proving identity ownership, the user first logs in with the first identity. The user then logs in with his second identity. If the user tries to log in to the same identity provider during the second login, the Single Sing-on functionality works, which automatically logs the user in again with the already logged in identity and an unwanted action occurs when the user tries to connect the identity to the same identity. This is an issue that some IdPs do not yet address (Some IdPs already support new login enforcement).

Obsolescence of identities

The general problem with super identities is their obsolescence. The IAM system updates the identity information when the user logs in. Therefore, if the user does not log in with a certain identity for a long time, then this identity begins to become obsolete.

The problem of identity obsolescence is described in the following example. The user has their identity with the university's identity provider and Google. These identities are connected. In the past, the user normally logged in under their university identity, but for convenience, they have only recently logged in with their Google identity. There are two problems with this situation.

The first problem is that identity attributes are becoming obsolete. In the example described, it may happen that for the IAM system and the services connected to it, the user will be a student even though the study has already been completed. And the user will be a student until he logs in with a university identity. Only then is the attribute that says the user is a student updated. Until then, the user will be treated as a student for the services and this may affect the functionality of the services.

The second problem is that our identity as such is beginning to become obsolete. The identity provider may decide to revoke the identity. The IAM system does not know that the identity does not exist.

The solution to this problem is to consider the validity of attributes and identities as such only for a certain period of time. For different services, the time may be different depending on the type of service provided. After the time expires, the attribute or identity is invalidated and the user is prompted to log in with a specific identity. This solution has an adverse effect on user perception.

Design of the linking process

We need to authenticate users at least twice. The first authentication takes place when the user accesses one of the applications from which the consolidation process can begin. So that we do not have to implement a second (and potentially n-additional) authentication in all applications where we want to implement account linking, it makes sense to create a new application whose purpose will be to authenticate users and perform the identity linking itself. We named this application Linker.

The problem that needed to be solved was how to make the user log in a second time. At present, various systems have begun to implement user logins to modal window services. Modal windows overlay current content instead of taking users to another page. It's less worrying for users than being dragged to a new page, which he finds distracting. In addition, the user can easily return to where he started if the merging process cannot be completed (for example, if the IdP fails). The Linker application is thus opened as a modal window from applications in which the user can get to link identities.

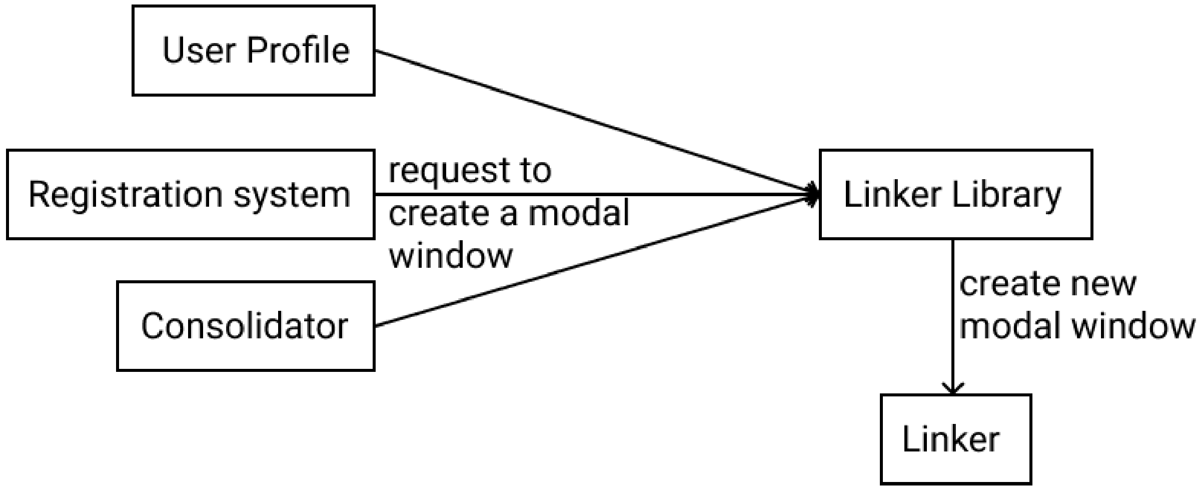
Therefore, the Linker is a modal window application that performs the linking of identities. The user logs in to this application, the Linker calls the backend method for the linking and responds to the result of the identity linking. The result of the identity linking is also passed to the application from which the Linker was opened.

In order for applications not to have to implement the Linker separately, the Linker Library component is also implemented, which will open the Linker window at the request of the application.

It was necessary to create another application designed for user support needs. In case the user fails to combine the identities in the user profile or registration system applications, he can write to the user for help. User Support - as the Linker application was designed - cannot send a link to the Linker application (the Linker application expects the user to log in to the application that opens it.). That is why the Consolidator application was created.

The consolidator is a stand-alone application that aims to log a user in and put them in the situation they are in (and what their next steps may be). The consolidator is one of the applications that requests to open the Linker modal window and shows the result of the identity linking operation after closing the Linker window.

Linker and Consolidator are technically two different web applications because it gives us more readable source code and the ability to split functionality into multiple components. However, the main reason is the stability of the login in the Consolidator application. The user remains logged in with the same identity to the Consolidator application. If it were one application, the login would change, which could have a negative impact on user perception.



Used technologies

SaToSa - access management system
Perun - IdM system
Communication protocol - OIDC

Backend method *consolidate*

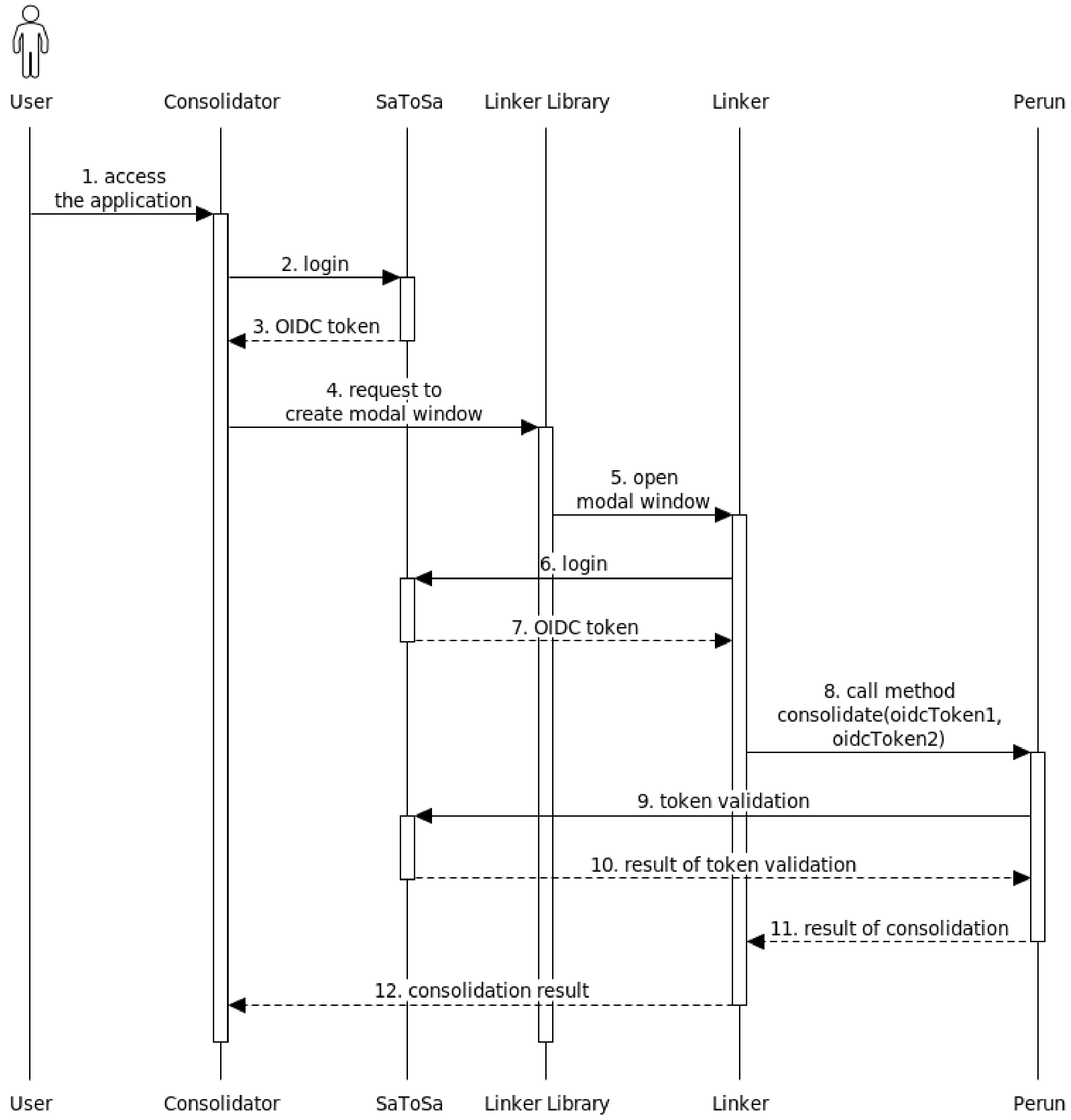
The *consolidate* method is a new method in Perun. The parameter of this method is the OIDC access token. The second access token, which is needed for consolidation, is part of the request header. The return type of this method is *void*. If it fails, it throws an exception.

The method is designed to connect only two identities, but by calling it repeatedly we can connect as many identities as needed.

There is an *ExtSource* object in Perun that is a source of identities. This object contains the attributes name IdP and type. There is also a *UserExtSource* object, which represents an identity in Perun. This object contains information such as identity source, identity attributes, login (Unique, long-lasting, unchangeable and never-reused identity identifier, in most cases this login is different from the login the user logs in.) Identity, User ID to which he belongs, or the time of the last user login with this identity.

The method tries to combine the identity that called this method with the identity that we get from the access token. In order to do this, it must obtain the following information about both identities: identity provider name, login identity, information on whether the user is registered. For the logged-in identity, this information is hidden in the *PerunPrincipal* object. For the identity obtained from the access token, we must obtain this information by querying the UserInfo Endpoint of the access system with the given OIDC access token. The access system will provide information about the name of the source of the identity and the login of the identity. Using these two pieces of information, we can search for users in the Perun system, if one exists.

Flow consolidation process


User
Consolidator
SaToSa
Linker Library
Linker
Perun
1. access the application
2. login
3. OIDC token
4. request to create modal window
5. open modal window
6. login
7. OIDC token
8. call method consolidate(oidcToken1, oidcToken2)
9. token validation
10. result of token validation
11. result of consolidation
12. consolidation result
User
Consolidator
SaToSa
Linker Library
Linker
Perun