Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Identitätsverbund Schweiz – IDV

February 2020

Technical Integration of the IDV Broker

From the perspective of the relying party

Purpose of this Document

The IDV broker is a technical realisation of the principles of identity federation. This document explores the various operation modes of the IDV broker and elaborates on the general conditions that RPs must meet in order to integrate with the broker.

Definitions

Identity Provider (IdP): A service that manages digital identities. Its main purpose is to verify a user's identity by applying authentication methods using, for instance, a password or a smartcard.

Authentication: The process of verifying a user's identity, often referred to as login or sign-in on a website. (An IdP's perspective on the same procedures is called *authentification*.)

User: A person who authenticates with an IdP.

Security ticket: A digital document issued by the IdP indicating that the user has been authenticated. The security ticket is comprised of at least a user identifier and may contain additional personal data.

Attribute: A persons particulars provided with the data in the security ticket.

Relying Party (RP): A service or resource that requires the user to be authenticated. The RP leaves the task of authentication to the IdP and relies on the information it finds in the security ticket.

Starting Point

Modern RPs are implemented such that they rely on the service of an IdP they sufficiently trust rather than performing the authentication procedures themselves. There is a trend in modern websites to let users pick their preferred IdP from a predefined list. The idea is to re-use the existing affiliation of the user with an IdP that's already available instead of forcing them to register with a new one.

This principle builds on the assumption that the RP enjoys a trust relationship with every IdP the user can chose from and that it is capable of interpreting and verifying the security ticket. Different IdPs would normally issue security tickets of a slightly different structure, such that RPs are forced to invest in technical and organisational measures to integrate each of them separately.

There is an easier way for RPs to accomplish the same goal: Let a mediating server take the role of a middleman between the RPs and IdPs, presenting itself to the RPs as the only IdP there is. The RPs would communicate to that single IdP which, in turn, acts as a placeholder for a potentially large number of IdPs. By doing so, it shields the RPs from the complexity of having to integrate with many IdPs on a one-to-one basis. This mediating IdP is called an *identity broker* (or *broker*, for short) while the conceptual principle is referred to as *identity federation*.

Conceptual Background

Generic Use Cases for Identity Federation

From the viewpoint of an RP, the identity broker is the same as any other IdP covering a number of typical use cases, in particular:

(1) *Authentication brokerage*. The broker would mediate a real IdP to the RP in order to have users authenticate with this one.

(2) *Identity brokerage*. Used mostly for the initial enrolment of a user with a service, making this use case rare in comparison to authentication. Apart from confirming authentication, the security ticket would often contain a person's particulars as attributes.

(3) *Attribute brokerage*. The broker mediates the appropriate IdP for attribute inquiries of which results may change over time, like "is over 18".

User Identification (nameID)

In order to unambiguously distinguish users, a user identifier, nameID, is assigned to each of them and provided to the RP in the security ticket. nameID comes in two flavours:

- *persistent*: Once nameID is assigned to the user, it would remain the same in every security tickets, no matter how long into the future.
- *transient*: nameID is assigned to the user temporally and would change with every other security ticket the IdP creates for the user.

Persistent is the usual way of user identification, as it allows an RP to recognise the user in subsequent login procedures and hence keep a history record and profile, e.g. a user account. Transient, on the other side, is a way to grant anonymity to the user. There is no way for the RP to correlate the possibly many login procedures of a user, preventing it from accumulating profile data.

Aggregation und Identity Linking

Usually, the broker would be connected to a number of different IdPs, some of which support authentication brokerage only, while others can be used for identification brokerage, i.e. they provide personal attributes of users in the security ticket.

Aggregation

Aggregation is the process of combining attributes from separate IdPs into one set.

Example: IdP A provides family name and first name as personal attributes of a user, while IdP B is a citizen registry that contains the city of origin. Let's assume the RP asks for authentication of a user along with the first name, last name and city of origin as required attributes. There are two ways to combine (aggregate) the two sources:

a) *RP-managed*: The RP issues two separate requests to the broker, each of which is relayed to the IdP capable of providing the appropriate response. The first request goes to IdP A in order to obtain the family name and first name. The second request goes to IdP B for the city of origin. The RP would then combine the two responses into one by itself.

b) *Broker-managed*: The RP issues a combined authentication and attribute request to the broker. The broker would split the request into two, one for IdP A and another one for IdP B, combine their respective responses and return the final result to the RP.

Identity Linking

In order to combine attributes from distinct IdPs, the broker would need to know whether the user is in fact registered with each of the IdPs involved, and if so, what their nameID is with each of them. This is the only way the broker can possibly split the RPs request it has received to the appropriate IdPs and specify the user with each of them. The principle saying that the broker is aware of every user's nameID with every IdPs it is connected to is called *identity linking*.

Without it, the RP would need to manage the aggregation by itself. In order to accomplish this, it would have to be aware of a user's nameID with every single IdPs and ask for authentication and attributes from the broker in separate requests. This would require the IdPs and RPs to closely interact off-line and exchange personal data of users, which may prove difficult in the light of privacy and data protection. Or, as an alternative, the RP could try to actively involve the user in an endeavour to build up a correspondence table between IdPs, their users and nameID. This would be a very tricky and error-prone procedure, far from user-friendly, and probably very difficult to explain to the average user.

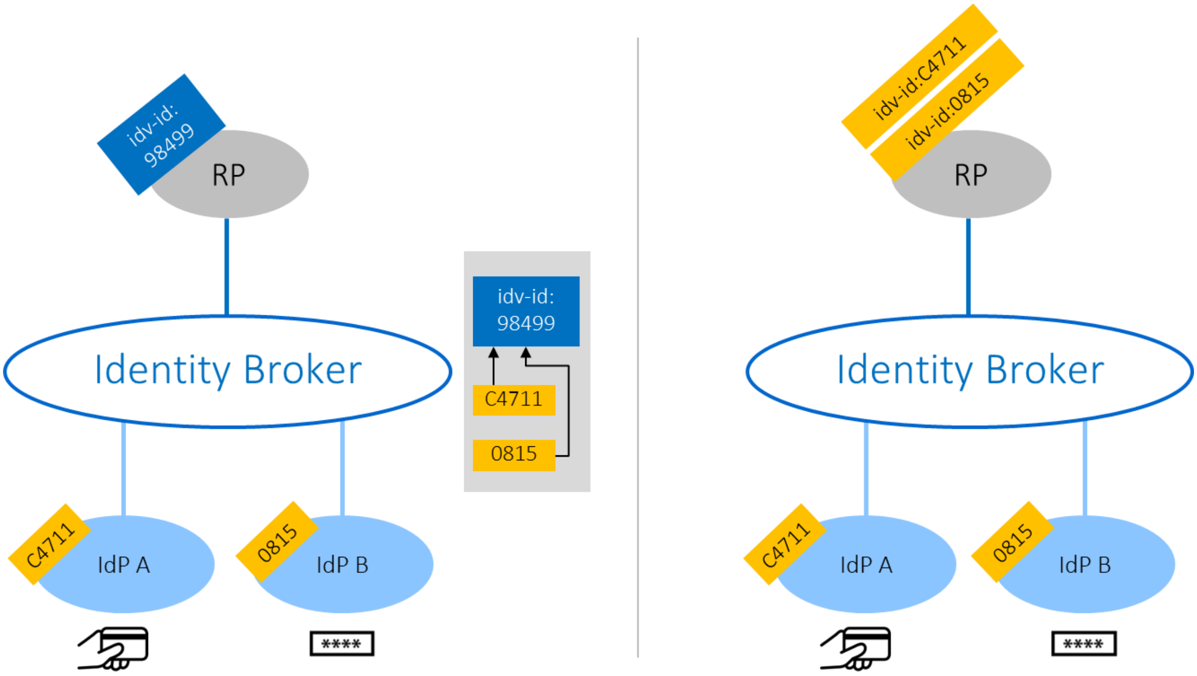

Illustration: With (left) and without (right) identity linking

Attribute Linking and Harmonisation

In general, IdPs use their own vocabulary for naming attributes. Take the attribute for a first name, for instance. IdP A may call this "name" while IdP B would use "firstName". If the user choses IdP B and the RP asks for the first name attribute in the authentication request, the broker would need to translate this into "firstName" as the required attribute to IdP B, as this is what it understands. As a consequence, the broker has to be aware of the attribute vocabulary of each IdP and then transform the RPs request accordingly. This feature is called attribute linking.

Attribute harmonisation means to use a common vocabulary across all IdPs, making attribute linking obsolete from the start. For example, the first name attribute may be traded by everyone as "name". Putting attribute harmonisation in place requires organisational unification, an option likely to be well suited for the IdPs targeted at a specific application domain or industry.

Aggregation Procedures

There are several ways to aggregate attributes from different IdPs.

(1) Request goes to a single IdP: The identity broker is a discovery service, so it knows which IdP can provide the requested attributes. In this case, aggregation becomes obsolete, as all the attributes come from one IdP only.

(2) Request goes to many IdPs using harmonised nameID: This option may work out for groups of IdPs that belong to a specific application domain, governed by the same rules, possibly using nameID in a unified way, like social security number AHVN13, for instance.

(3) Request goes to many IdPs without harmonised nameID: The most complex procedure, as it requires both identity linking and attribute linking.

Proxying

According to eCH-0168, *relaying* is a technique in which the broker forwards the original security ticket from an IdP without changing its content, so nameID and attribute names are provided "as is".

With *proxying*, on the other hand, the broker uses the IdP's original ticket to compose a security ticket of its own and hand it over to the RP claiming to be the originator.

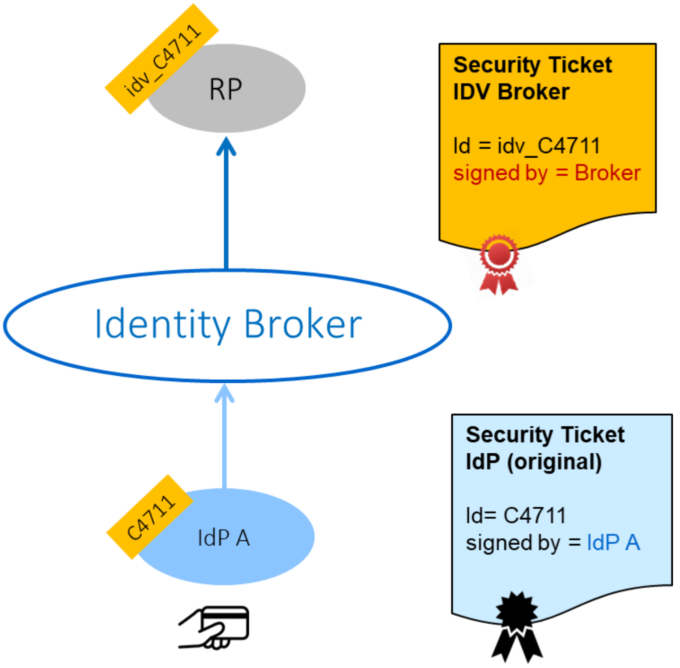

Illustration: Proxying

IDV Broker

The IDV Broker is an identity broker implementing the principles of identity federation. The IDV Broker connects to a number of IdPs and RPs using a standard interface.

The IDV Broker does not authenticate users, but rather delegate this task to one of the affiliated IdPs (A or B in the illustration below). As soon as the user tries to log in at the RP, the IDV Broker intercepts the process and provides a list of IdPs from which the user is asked to choose. The user would then pick his or her favourite IdP and authenticate with it.

This procedure assumes that the user is registered with at least one of the IdPs, and that he or she can authenticate with it. If this is not the case, the user has to enrol and then authenticate with the RP separately.

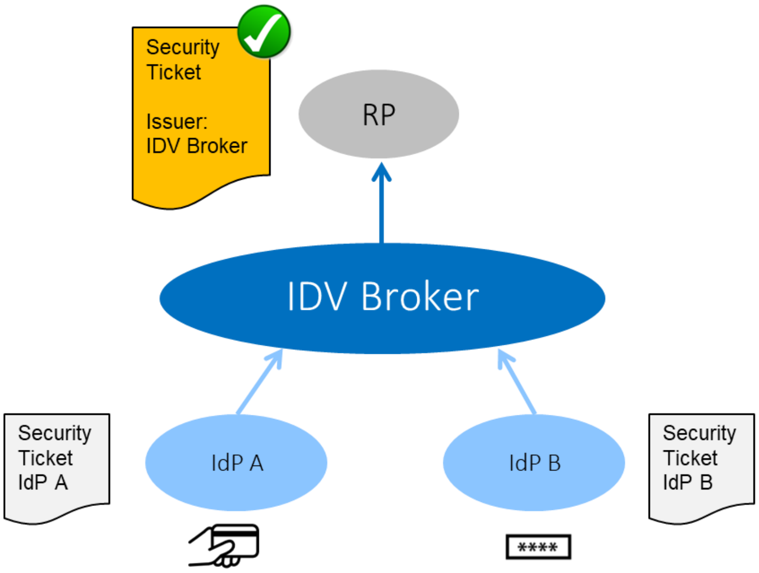

Illustration: Identity federation with the IDV Broker

IDV Broker modes of operation

The IDV Broker supports *proxying* according to eCH-0168 with some extensions, in particular:

- The IDV Broker allows to authenticate and obtain attributes from one of the affiliated IdPs. See aggregation procedure (1) on page 3.
- The IDV Broker uses the IdPs security ticket and composes a new one on its own behalf.
- The IDV Broker can be configured to use persistent or transient nameID. Persistent nameID allows the RP to recognise the same user in subsequent sessions over time, whereas transient is an effective means to implement anonymity of the user.
- You can never rule out the possibility that two different persons are given the same nameID in two separate IdPs. However, the IDV Broker ensures uniqueness of nameID across all IdPs for a specific user by providing its own nameID in the security ticket.
- The IDV Broker implements attribute harmonisation by defining a common attribute vocabulary and performing behind-the-scenes name transformation for each IdP. The internal attribute names of an IdP are no longer relevant. For example, the RP would always obtain the attribute "firstName" for a person's first name, irrespective of the actual attribute name used by the IdP that holds the information.
- The RP fully trusts the IDV Broker and does not require checking the security ticket of the IdP that carried out the authentication.

And, as a special feature:

- The IDV Broker can be configured such that the original security ticket of the IdP is attached to the broker's security ticket in redundancy. Doing so, the RP has a way to independently verify the origin and authenticity.

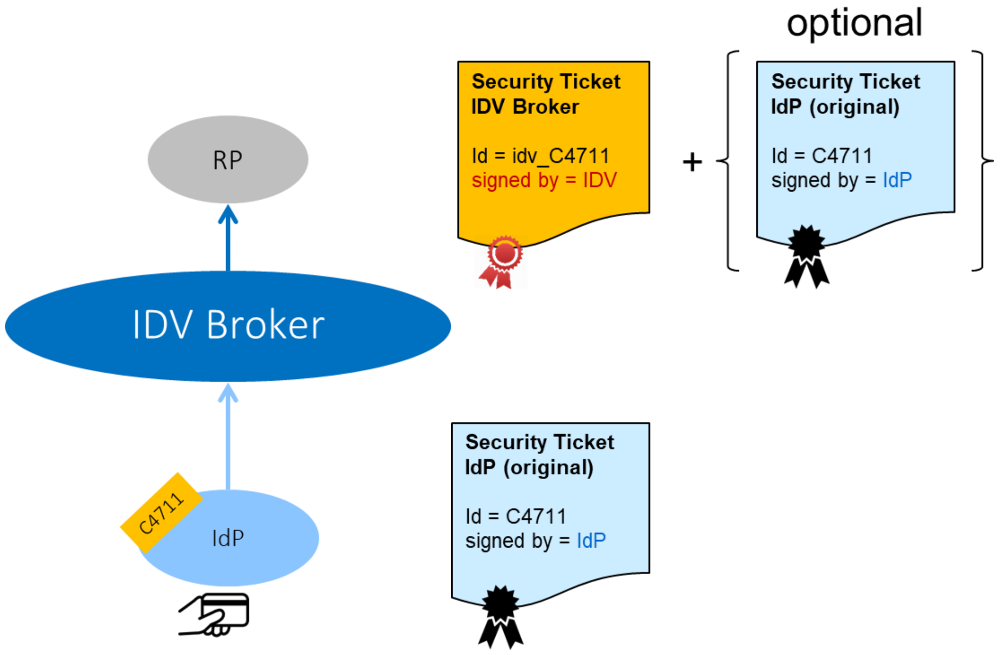

Illustration: Proxying mode along with the original security ticket

While in the *proxying mode*, the IDV Broker is aware of the relationship between the user and the IdP or RP, and it can see the attributes in plain. This is why the RPs must have full trust in the IDV Broker and the security tickets it issues, as the origin of the security ticket cannot be verified. Proxying is the IDV Broker's default mode and RPs fully rely on it.

Having the IdP's original security ticket attached as an option, RPs no longer depend on the IDV Broker alone. However, with this feature turned on, the RP obtains additional information about the relationship between the user and the IdPs, a fact that may rise privacy concerns.

Switching between IdPs

Take the following scenario: The user authenticates with the RP many times, using IdP A at one time and IdP B at another. As a consequence, the user's nameID toggles between C4711 and 0815.

If we wanted to give the RP a way to detect that C4711 and 0815 are in fact the same person, identity linking would have to be implemented. Without identity linking, the RP will not easily find out that C4711 and 0815 are distinct nameID values for the very same person.

In order to avoid poor usability and privacy concerns, identity linking was not implemented in the IDV Broker. The good side is that there are no user accounts with the IDV Broker and users don't have to manage preferences and settings. To the user, the IDV Broker is an injected search screen that helps them find the appropriate IdP. Apart from this, the IDV Broker operates in the background and users are not bothered by its existence.

RPs are free to provide their own identity linking to the users. They would have to get the user involved into some adjustment procedure, an endeavour that may turn out to be a challenge in terms of usability, security and data protection.

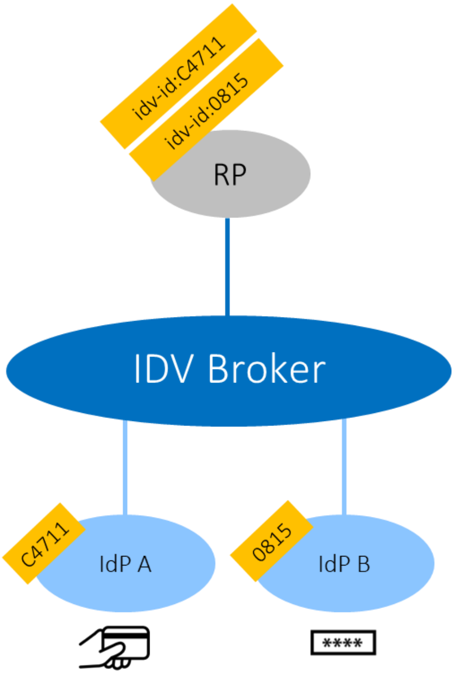

Illustration: Switching between IdPs

Anonymous IdP

Anonymization of the IdP means to disguise the origin of the security ticket, a feature for which you may have good reasons. By applying anonymization, RPs are prevented from finding out which IdP the user had picked, as this may turn out to be a sensitive personal information. For example, you may want to enforce that a user authenticating at a law enforcement IdP shall not be identifiable as a police officer at the RP. This is why the originating IdP is kept hidden in the broker's security ticket (see illustration below).

The IDV Broker can be configured to just do that. RPs have to be prepared to receive security tickets that are not suited for double-checking the original security ticket, nor identify the original IdP. With anonymization, RPs have to fully trust the IDV Broker.

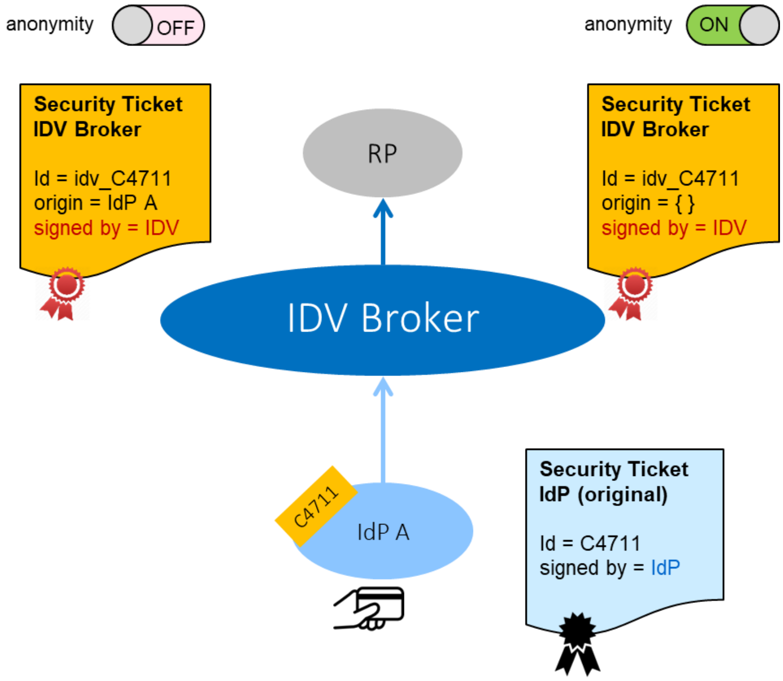

Illustration: Anonymous IdP as a configuration option

Conclusions

RPs using the IDV Broker have to take into account the following.

1. The IDV Broker mediates user identities from the affiliated IdPs and provides a security ticket to the RP containing a unique nameID of the user which is not the same nameID that the authenticating IdP has assigned to the user. It is the IDV Broker's own nameID.
2. The IDV Broker is designed such that RPs can fully trust it, if they want. Validating the security ticket is to verify that the IDV Broker is the issuer. This way, RPs have to plug in to only one IdP interface, that of the IDV Broker, to reach out to a possibly large number of IdPs.
3. The IDV Broker can support persistent and transient nameID, depending on the configuration. Persistent nameID can only be done if the IdPs provide nameID in a persistent manner too. Transient nameID can always be supported.
4. Provided that an IdP used persistent nameID, the nameID in the IDV Broker's security ticket is going to be persistent, too.
5. The IDV Broker does not implement identity linking. The nameID provided by the IDV Broker is a different one for each IdP the user has used.
6. The IDV Broker can be configured to attach the security ticket of the original IdP along with its own security ticket. This way, RPs can check the origin and the integrity independently and no longer have to rely on the IDV Broker alone.
7. The IDV Broker can disguise the IdP from the eyes of the RP, e.g. the IdP with which the user has authenticated is hidden.
8. The IDV Broker does not support attribute aggregation from different IdPs. If the RP requires attributes from different IdPs, it would have to execute a series of individual requests with the IdPs in question and combine the responses into the final outcome.