univention
be open.

ID Broker manual for service providers 1.0.0

Release 1.0.0

Univention GmbH

Oct 28, 2022

CONTENTS:

This documentation is intended for administrators and developers of *service providers* who want to connect a *service* to the *ID Broker* (page 3).

To follow this text, you need to understand the basic concepts of LDAP and UCS@school, be familiar with OAuth 2.0 and OpenID Connect (OIDC) and be able to integrate these concepts into your app.

This documentation doesn't cover the on-boarding of *school authorities*, who want to connect to the *ID Broker* (page 3). To find more information about these topics, visit the Univention documentation[1].

As a *service provider*, you can use the *ID Broker* (page 3) to use single sign-on (SSO) for end users between the *Identity Provider (IDP)* of connected *school authorities* and your service. Service specific pseudonyms are used to ensure that users activities can't be combined to build user profiles. Additional information, like class membership or school roles, can be accessed by the *Self-Disclosure-API* (page 11).

[1] https://docs.software-univention.de

CHAPTER

ONE

BIG PICTURE OF UNIVENTION ID BROKER

The Univention ID Broker eases the integration between identities of learners and teachers managed by *school authorities* or federal states and the various *service providers* for educational purposes with respect to the data protection regulations in Europe[2].

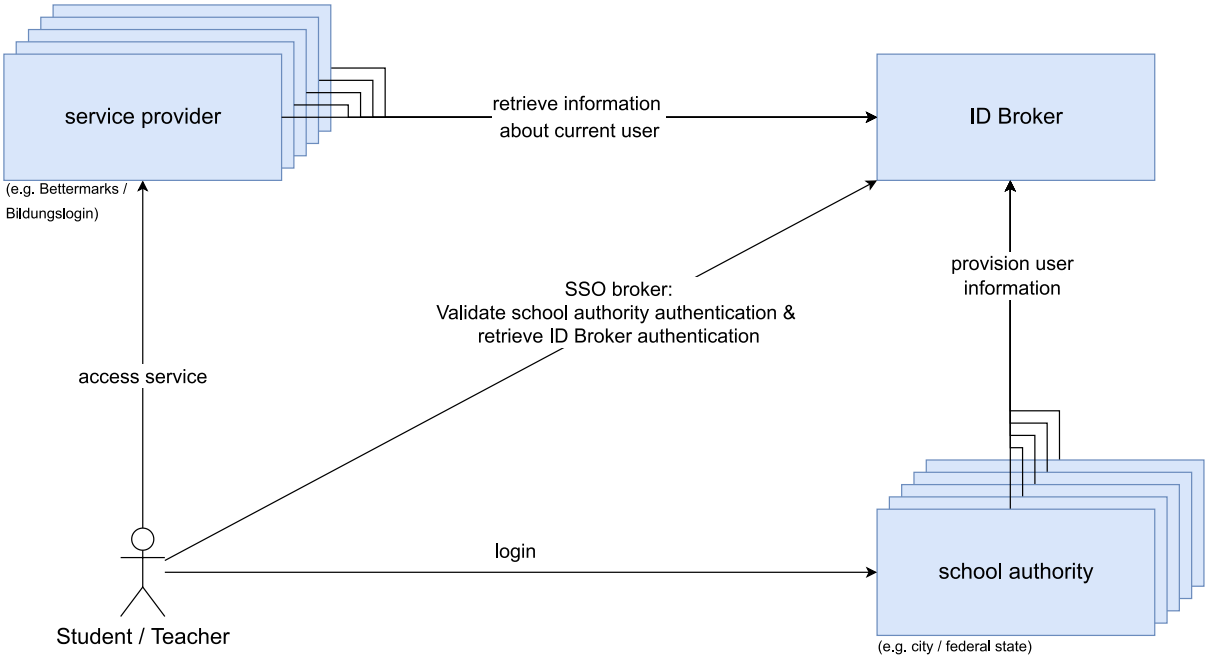

Figure 1.1: Overview of the involved components of the ID Broker and external Systems.

To reach this goal the ID Broker ensures the following:

- Single sign-on for end users between the *IDP* of a *school authority's IDP* and *service providers* (educational SaaS offerings).
- Only one configuration step to connect with the ID Broker both for IDPs and service provider. There is no need to configure each IDP with each service.
- User identification uses service specific pseudonyms instead of global identifiers. Service specific pseudonyms prevent user profiles based on combined user activities in the different services.
- To give end users a *complete* environment from scratch, *service providers* can retrieve information about the role and the courses of users.
- To ensure data protection, the ID Broker environment only stores the user's first name, last name, email address as well as the school class and school memberships.

[2] https://en.wikipedia.org/wiki/General_Data_Protection_Regulation

The UCS@school components of the ID Broker, like the UCS@school Kelvin REST API, are built on top of UCS core components OpenLDAP, UDM and the UDM REST API. To learn more about UCS and it's components, see Univention documentation[3].

[3] https://docs.software-univention.de

CHAPTER

TWO

IN-DEPTH

In section *Authentication and user data retrieval* (page 5) we have a look at the steps, that users follow when they access a service registered at the ID Broker. The section *SSO Broker* (page 7) covers the role and access points of the SSO Broker.

2.1 Authentication and user data retrieval

One design goal of the ID Broker architecture is that the users of multiple *school authorities* can securely access the resources of multiple service providers, so that the *school authorities* and the *service providers* don't have to communicate with each other. Users only login at their *school authority*. The *service providers* don't store any user information.

When a user wants to access a resource of one of the *service providers*, they need to authenticate themselves. The *service provider* requires some data about the users to provide an individualized service.

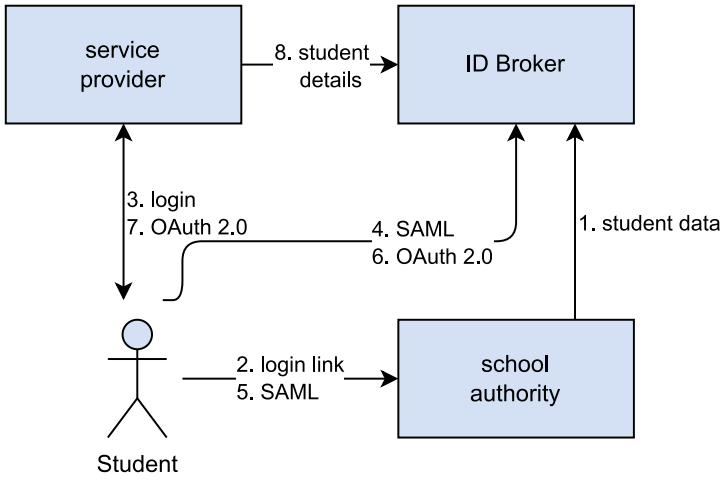

Figure 2.1: ID Broker - connections

Figure 2.1 above shows the connections between a student, the *service provider*, the *school authority* and the ID Broker. It redirects the user to the ID Broker, which in turn redirects the user to the *IDP* of its school authority.

The ID Broker verifies the signature of the *school authorities IDP* and gives a ticket to the user. The user passes that ticket to the *service provider*, which can now retrieve data about the user from the ID Broker.

The following Figure 2.2 covers the interactions between the components in more depth.

1. student data The *school authority* syncs student data to the ID Broker.

2. request service provider login at school authorities portal page The student clicks a link on the *school authorities* portal page, and is redirected to the *service provider*. This redirection makes the combination of SAML and OpenID Connect possible - the ID Broker must know which SAML backend needs to be used.

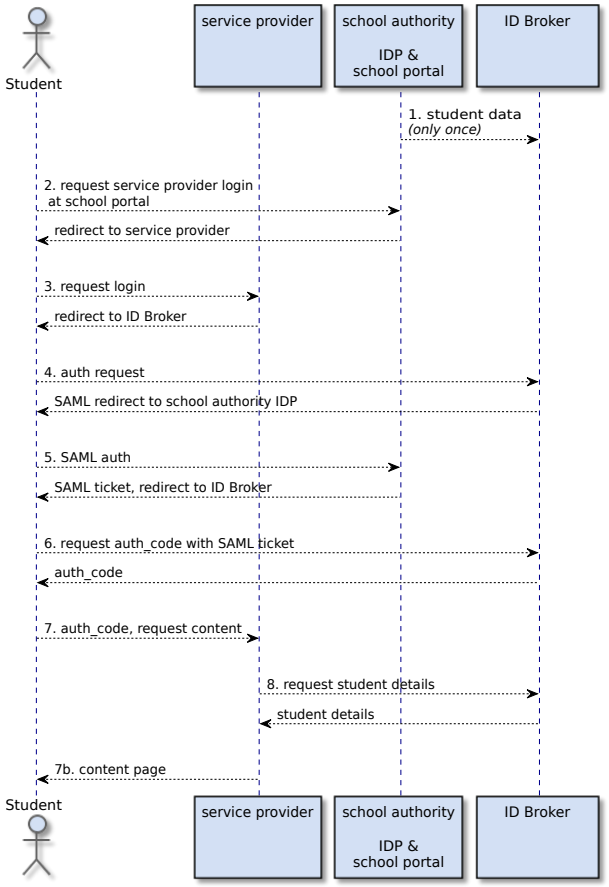

Figure 2.2: ID Broker sequence: authentication and user data retrieval sequence

3. request login The student requests a login at the *service provider's* page, and is redirected to the ID Broker.

4. auth request The ID Broker doesn't login the student and instead redirects the student to the *school authority*, which has an *IDP* (SAML) provider.

5. SAML auth The actual SAML authentication of the user happens. The student receives a SAML ticket and is redirected to the ID Broker.

6. request `auth_code` Using the SAML ticket, the user requests an `auth_code` from the ID Broker. The user is redirected to the *service provider*.

7. `auth_code`, request content The user passes the `auth_code` to the *service provider* while asking for the content. The service provider exchanges the `auth_code` for an `access_token` and an `id_token` (this step is left out of the diagram for clarity reasons). The `id_token` contains the pseudonyms for the requested data, as well as, for the requesting user.

8. request student details Using the `access_token` and the pseudonyms inside the `id_token`, the *service provider* can now request pseudonymized user data from the ID Broker.

7b. content page This is the continuation of step 7 - the student receives the requested content from the *service provider*.

2.2 SSO Broker

In this section you learn about the architecture with focus on the SSO Broker and the single sign-on part of the ID Broker. The software Keycloak provides the functionality for the SSO Broker.

The main job of the SSO Broker component is to handle multiple-tenant authentication, using pseudonyms. This involves the student doing the login and passing authentication tokens back and forth.

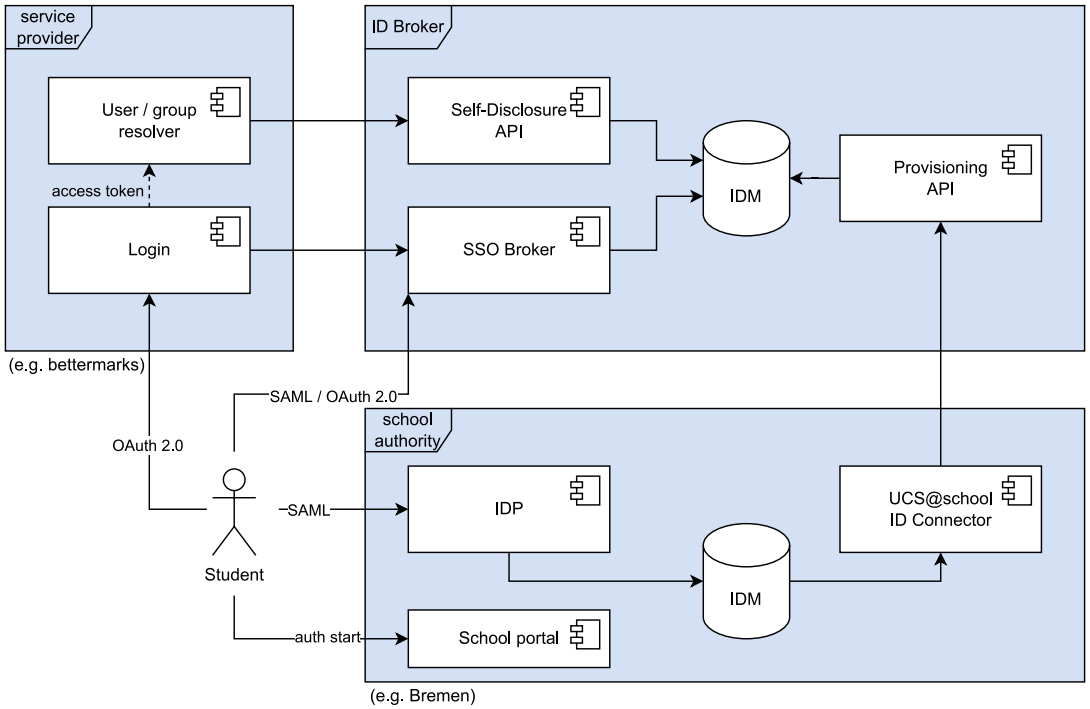

Figure 2.3: SSO Broker communications

The *SSO Broker* participates in the following communications:

- The school portal redirects the student to the SSO Broker upon first login. This first step is part of the OpenID Connect (OIDC) flow. The SSO Broker redirects the student to the *school authority's IDP* for SAML authen-

tication. The student authenticates with a real user identifier. The student returns the SAML ticket to the SSO Broker, which they received in the authentication step.

- The ID Broker IDM system provides a service provider specific pseudonym for the SSO Broker. The pseudonym also includes other user data from the *school authority*. The student receives an `auth_code` that is valid for the *service provider* specific pseudonym. The student sends the `auth_code` to the *service provider*.
- The *service provider* exchanges the `auth_code` for both an `access_token` and an `id_token` at the SSO Broker. The *service provider* processes the `id_token` that contains the pseudonym. It uses the `access_token` to request more data about the student through the pseudonym at the *Self-Disclosure API*.

The SSO Broker is available:

- for OIDC at `https://`*`FQDN`*`/auth/realms/`*`SERVICE PROVIDER ID`*`/protocol/openid-connect`
- for SAML at `https://`*`FQDN`*`/auth/realms/`*`SERVICE PROVIDER ID`*`/broker/saml`

CHAPTER

THREE

ONBOARDING PROCESS

To use your application as a *service provider* with the ID Broker, you must provide a service that is configured for single sign-on with the ID Broker. The ID Broker must first register your service.

You have to contact the operator of the ID Broker at service-provider-admin@univention-id-broker.com to start the registration process. Your registration request has to include at least one redirect URL where the user's browser is directed to after successful authentication through OIDC.

The ID Broker operator will provide you with a *client ID* and *client secret* that the service can use to authenticate itself to the ID Broker.

3.1 Service and app requirements

Your app must support the OpenID Connect (OIDC) authentication protocol, more specifically the Authorization Code Flow[4]. OIDC is an identity layer built on top of the **OAuth 2.0 protocol**[5].

In addition, the service provider app must use the **Proof Key for Code Exchange (PKCE) protocol**[6]. Regarding OIDC and OAuth 2.0, see OAuth 2.0 Security Best Current Practice[7].

3.2 Connect to ID Broker

The ID broker publishes its OIDC metadata under the following well-known URIs:

- For the staging environment: https://sso-broker.staging.univention-id-broker.com/auth/realms/ID-Broker/.well-known/openid-configuration
- For the production environment: https://sso-broker.production.univention-id-broker.com/auth/realms/ID-Broker/.well-known/openid-configuration

The URI returns a JSON list of OIDC endpoints, supported scopes and claims, public keys used to sign the tokens, and other details. The service provider application can use this information to create a request to the ID Broker (OIDC server) and validate the `access_token`.

[4] https://openid.net/specs/openid-connect-core-1_0.html#CodeFlowAuth
[5] https://datatracker.ietf.org/doc/html/rfc6749.html
[6] https://datatracker.ietf.org/doc/html/rfc7636.html
[7] https://datatracker.ietf.org/doc/html/draft-ietf-oauth-security-topics

3.3 Information required by the operator

When you register your service, you need to hand in the following information to the operator:

- The service name.
- When the Self-disclosure API is used: The IP addresses of the service, so that the operator can configure an appropriate firewall rule for the Self-disclosure API.

CHAPTER

FOUR

SELF-DISCLOSURE API

If your service needs information for or about the requesting user, you can fetch user data like school roles and class memberships by calling the *Self-disclosure API*. This is illustrated in step 8 (*students details*) of *Authentication and user data retrieval* (page 5).

The user, who logs in with their browser, sends an `auth_code`. It enables the API to fetch the requested data. The student or teacher in return receive this `auth_code` from the IDP of their school authority.

The *Swagger User Interface (Swagger UI)* is accessible at

- Staging environment: https://self-disclosure.staging.univention-id-broker.com/ucsschool/apis/docs
- Production environment: https://self-disclosure.production.univention-id-broker.com/ucsschool/apis/docs

The IP of your service will be added to the allow-list (whitelist) and you will be able to call the API from your application after you have registered it as described in *Onboarding process* (page 9).

For more information about the API, see the ID Broker manual for the Self-disclosure API[8].

[8] https://docs.software-univention.de/idbroker-self-disclosure-api-manual/index.html#introduction

CHAPTER

FIVE

GLOSSARY

Identity Provider (IDP) Instance that provides information to authenticate and authorize identities. In case of ID Broker scenarios this is typically a SAML or OpenID Connect IDP hosted by a *School Authority*.

Provisioning API REST API of the ID Broker. *School authorities* use the API to send pseudonyms and a limited set of meta information on users and groups to the ID Broker.

School Authority In context of this document, the term *school authority* subsumes various institutions which serve one or several schools with IT infrastructure. The school authority is the data source for all students and teachers of an environment. The ID Broker will receive a minimal subset of this data, see *Big Picture of Univention ID Broker* (page 3). This can be a single school, a school authority with several schools, or an environment hosting services for a federal state. The environments are hosting a UCS@school domain.

Service In the context of this document a *service* is an application, which uses single sign-on with the ID Broker and provides a service for students and teachers. For example a learning platform, that offers books.

Service Provider (SP) Instance that provides a *service*.

Self-disclosure API REST API of the ID Broker which allows retrieval of meta information of an authorized user. It focuses on the role of the user and the assigned learning groups.

Swagger User Interface (Swagger UI) Allows visual interaction with the API's resources without the need for an external implementation. You find the URLs to the respective Swagger UI in the appropriate sections.

INDEX