Differences between SURFconext and SURFsecureID

This page describes the differences between SURFconext and SURFsecureID that are relevant for a SAML service provider (SP) that is migrating from SURFconext to the SURFsecureID gateway, or that is using both simultaneously.

Note that this is only relevant if you connect to the SURFsecureID gateway directly because you need advanced features. Normally, SP's that want to use SURFsecureID can just interfacce with SURFconext.

Architecture

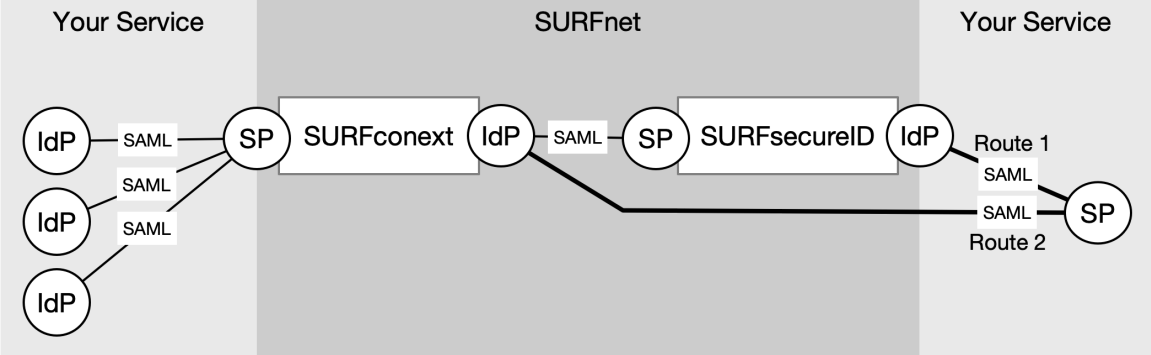
SURFconext (engine.surfconext.nl) and SURFsecureID (sa-gw.surfconext.nl) are both SAML proxies. The image below shows how SURFconext and SURFsecureID relate to each other.

Your SP connects to the IdP side of the proxy. An authentication to SURFsecureID (route 1) will pass though SURFconext. This means that SURFconext functionality like consent, authorization, attribute aggregation functions just like a direct authentication to SURFconext (route 2).

A service that uses SURFsecureID directly (route 1 in the image below) thus can continue to use SURFconext for authentication (route 2 in the image below). It is your SP that, for each authentication, decides which route it wants to use. The SP will receive the same attributes and user identifiers regardless of the route it takes. To get strong authentication your SP must authenticate to SURFsecureID.

Security advice

If your SP trusts multiple IdPs (e.g. SURFconext IdP and the SURFsecureID IdP), your SP must always verify from which IdP (the `Issuer`) it received the `SAMLResponse`. If your SAML library supports the IdP initiated flow (a.k.a unsolicited assertions), your SP could receive, and accept as valid, a `SAMLResponse` from any trusted IdP at any time.



SURFsecureID and OpenID Connect

Please note that SURFsecureID does not yet support OpenID Connect whereas SURFconext does. If you must use OpenID Connect, using an OpenID Connect - SAML proxy (like SaToSa) may be an option.

Metadata

Because the SURFsecureID proxy is a separate SAML IdP from the normal SURFconext, it has different SAML 2.0 metadata. The EntityID, SAML signing certificate and Single Sign On Location are all different from the normal SURFconext.

This means that some metadata related features are unavailable when using the SURFsecureID gateway:

- IdP selection through metadata (transparent metadata, Dutch scoping)

Attributes

The attributes you receive from SURFsecureID come from the SURFconext. SURFsecureID requires that your SP receives the eduPersonTargetedID (EPTI) attribute. If this attribute is not present, the authentication will fail at the SURFsecureID gateway.

> ⚠ **eduPersonTargetedID (EPTI) is required**
>
> Ensure that you receive the eduPersonTargetedID (EPTI) attribute when migrating your SP from SURFconext to SURFsecureID. This attribute is required for SURFsecureID. You can ask support@surfconext.nl to enable this attribute for your SP.

Authentication Request

AssertionConsumerServiceIndex

Selecting a binding using the `AssertionConsumerServiceIndex` attribute is not supported.

ForceAuthn

Single Sign on for the second factor is not provided: for each authentication request with a LoA > 1 the user must always authenticate using his second factor.

Setting `ForceAuthn` to true may force a re-authentication of the user at the institutional IdP for the first factor, but this cannot be guaranteed.

No SessionIndex in the AuthnStatement

The SURFconext Strong Authentication gateway does not provide a `SessionIndex` in the SAML Response returned to the SP. More information (see at line 1107).

IdP initiated login

SURFsecureID does not support IdP initiated login (IdP-geïnitieerde login).

Because your SP can still use the SURFconext, you could first authenticate the user there and then authenticate the user at SURFsecureID. Because the user has SSO at the IdP, and SURFconext remembers the selected IdP the user will not get a WAYF (IdP selection screen) and does not need to reauthenticate at the IdP.

Scoping using IDPList

Selecting IdP(s) by adding a `IDPList` in the `Scoping` element in a `AuthnRequest` is not supported.

Because your SP can still use the normal SURFconext, you could first authenticate the user there and then authenticate the user at SURFsecureID. Because the user has SSO at the IdP, and SURFconext remembers the selected IdP the user wil not get a WAYF (IdP selection screen) and does not need to reauthenticate at the IdP.

AuthenticationContext in the Assertion

After successful authentication the SURFsecureID gateway will report the level of assurance in an `AuthnContextClassRef` element in an `AuthenticationContext` in the SAML `Assertion` sent to the SP.

Non-production environments use different identifiers!

Authentication failure

When authentication fails, it is generally because the user:

1. cancels authentication during verification of the second factor or
2. does not have a suitable second factor identification

The SURFsecureID gateway will send a SAML Response to the SP about the failure. The SP should be ready to handle the response. The response contains non-success `StatusCodes`:

1. First level `StatusCode` `urn:oasis:names:tc:SAML:2.0:status:Responder`
   with second level StatusCode `urn:oasis:names:tc:SAML:2.0:status:AuthnFailed`: user canceled the authentication.
   or
2. First level `StatusCode` `urn:oasis:names:tc:SAML:2.0:status:Requester`
   with second level `StatusCode` `urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext`: authentication cannot be performed at the requested LoA.

More info:

- SAML Response with user cancelling authentication
- SAML Response with error StatusCode

Supported SAML bindings

The SAML bindings supported are limited to those in the Interoperable SAML 2.0 Web Browser SSO Deployment Profile. This means that:

- The SP must send SAML Authentication Requests using `urn:oasis:names:tc:SAML:2.0:bindings:`**`HTTP-Redirect`**.
  `urn:oasis:names:tc:SAML:2.0:bindings:`**`HTTP-POST`** is not supported.
- The SP must be able to receive SAML Responses using the `urn:oasis:names:tc:SAML:2.0:bindings:`**`HTTP-POST`** binding.