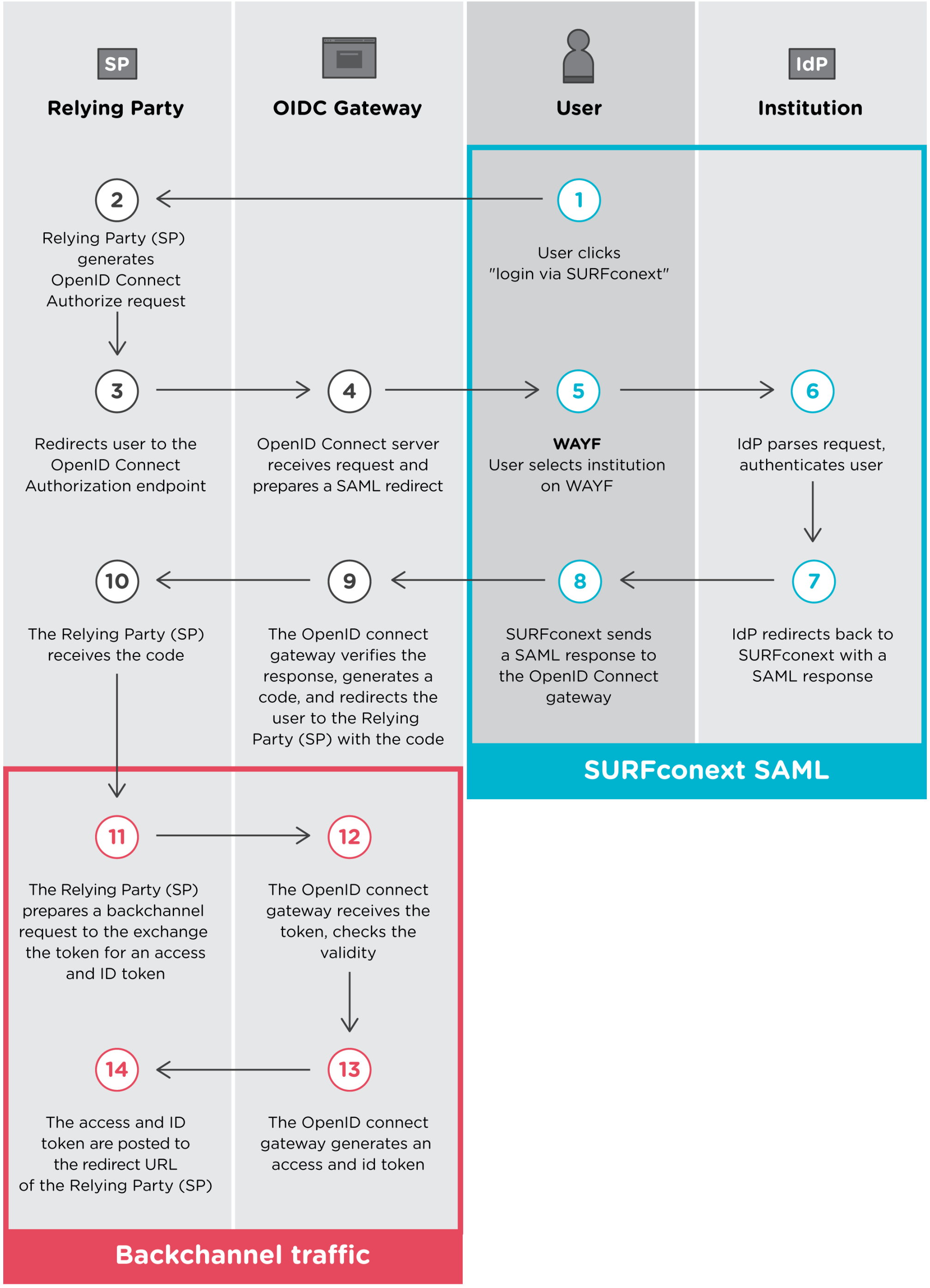
OpenID Connect authentication flow

With OpenID Connect the trust is established out-of-band. This means that a network or channel separate from the primary network supplies the Service Provider with a username and a secret. All the necessary technical information such as endpoints, supported algorithms and supported claims can be found at the `.well-kown` endpoint: https://connect.surfconext.nl/.well-known/openid-configuration

SURFconext connects the SP and the IdP based on specific rules. Note that SURFconext itself does not authenticate users: this is done by the connected Identity Providers. This authentication flow in OpenID Connect is depicted below. Let's dive into this.

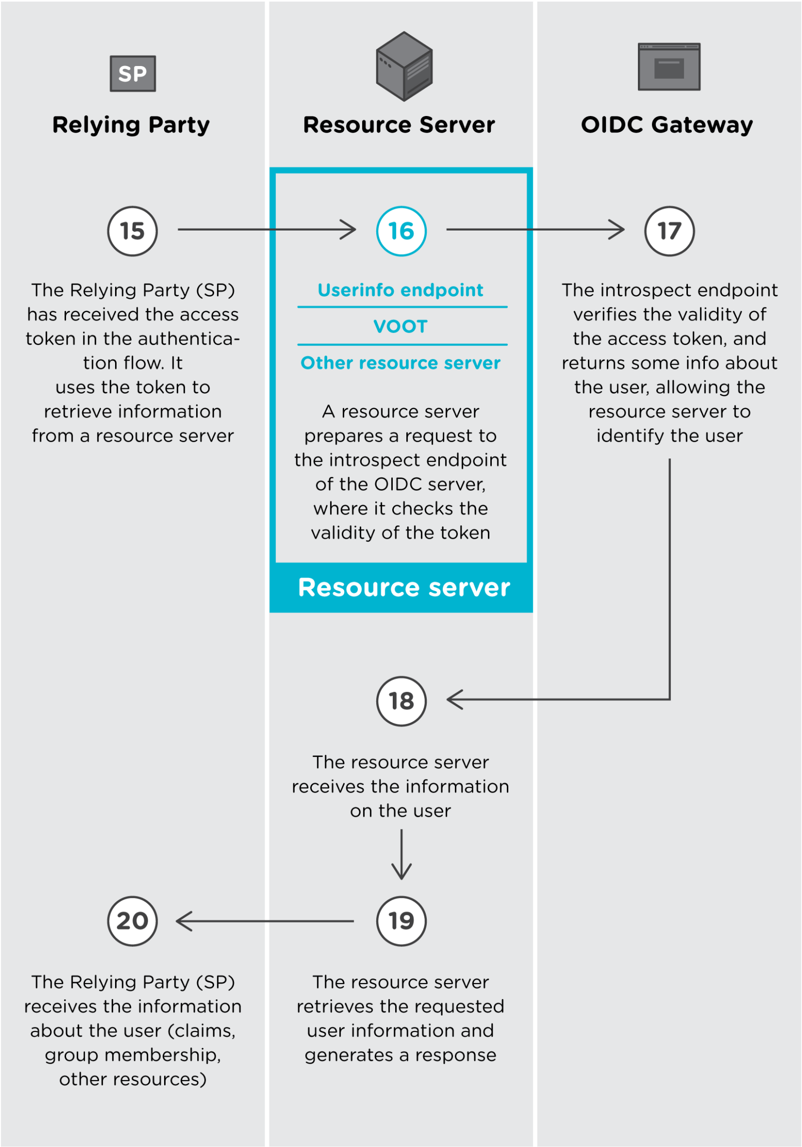


SP
Relying Party
OIDC Gateway
User
IdP
Institution
1
User clicks "login via SURFconext"
2
Relying Party (SP) generates OpenID Connect Authorize request
3
Redirects user to the OpenID Connect Authorization endpoint
4
OpenID Connect server receives request and prepares a SAML redirect
5
WAYF
User selects institution on WAYF
6
IdP parses request, authenticates user
7
IdP redirects back to SURFconext with a SAML response
8
SURFconext sends a SAML response to the OpenID Connect gateway
9
The OpenID connect gateway verifies the response, generates a code, and redirects the user to the Relying Party (SP) with the code
10
The Relying Party (SP) receives the code
SURFconext SAML
11
The Relying Party (SP) prepares a backchannel request to the exchange the token for an access and ID token
12
The OpenID connect gateway receives the token, checks the validity
13
The OpenID connect gateway generates an access and id token
14
The access and ID token are posted to the redirect URL of the Relying Party (SP)
Backchannel traffic

OpenID Connect authentication process in steps

The schematic is a representation of the login flow of the SURFconext OpenID Connect proxy.

1. A User accesses a Service Provider (Relying Party) and clicks "login via SURFconext"
2. The Relying Party (SP) generates an OpenID Connect Authorize request and
3. Redirects user to the OpenID Connect Authorization endpoint
4. OpenID Connect server receives request and prepares a SAML redirect to authenticate the user
5. In order to determine where to send the user for authentication, SURFconext shows the user a "Where Are You From?" (WAYF) page with all Identity Providers that have access to the service. The user chooses the institution that is his Identity Provider.
6. The IdP authenticates the user, usually by asking to enter his credentials. After validating,
7. the IdP generates a SAML response and redirects the user back to SURFconext with a response message, saying that the user is authenticated. The message also contains the attributes from the user.
8. SURFconext validates the response message and if OK makes some alterations, e.g. rewriting the user's identifier and adding or modifying attributes and sends the SAML response to the OpenID Connect gateway. According to the attribute release policy applied, SURFconext determines the attributes that are allowed through to the Service Provider.
9. The OpenID connect gateway verifies the response, generates a code, and redirects the user to the Relying Party (SP) with the code
10. The Relying Party (SP) receives the authorization code
11. The Relying Party (SP) prepares a backchannel request to exchange the token for an access and ID token
12. The OpenID connect gateway receives the token and checks the validity
13. The OpenID connect gateway generates an access and ID token
14. The access and ID token are posted to the redirect URL of the Client (SP)


SP
Relying Party
Resource Server
OIDC Gateway
15
The Relying Party (SP) has received the access token in the authentica-tion flow. It uses the token to retrieve information from a resource server
16
Userinfo endpoint
VOOT
Other resource server
A resource server prepares a request to the introspect endpoint of the OIDC server, where it checks the validity of the token
Resource server
17
The introspect endpoint verifies the validity of the access token, and returns some info about the user, allowing the resource server to identify the user
18
The resource server receives the information on the user
19
The resource server retrieves the requested user information and generates a response
20
The Relying Party (SP) receives the information about the user (claims, group membership, other resources)

The Relying Party (SP) now has 2 tokens; an ID token which contains a pseudonymous identifier and an access token that can be used to retrieve information from resource servers *on behalf of the user*. The picture above sketches this process.

15. The Relying Party (SP) has received the access token in the authentication flow. It uses the token to retrieve information from a resource server. The OpenID Connect server itself has a special resource server, called the *userinfo endpoint*. This endpoint can supply user claims (attributes).
16. The resource server receives the access token, and prepares a request to the introspect endpoint of the OpenID Connect server, where it checks the validity of the token
17. The introspect endpoint verifies the validity of the access token, and returns some info about the user, allowing the resource server to identify the user
18. The resource server receives the information on the user
19. The resource server retrieves the requested user information and generates a response
20. The Relying Party (SP) receives the information about the user (claims, group membership, other resources)

To get a feeling of how things work in reality, you can play around with our playground: https://oidc-playground.surfconext.nl/.

Navigate