



Authentication methods

Planon Software Suite

Version: L120

© 1997 - 2025 Planon. All rights reserved.

Planon and the Planon logo are registered trademarks of Planon Software Development B.V. or its affiliates. All other product and company names mentioned herein are trademarks or registered trademarks of their respective companies. Planon Software Development B.V., its affiliates and/or licensors own the copyright to all Planon software and its associated data files and user manuals.

Although every effort has been made to ensure this document and the Planon software are accurate, complete and up to date at the time of writing, Planon Software Development B.V. does not accept liability for the consequences of any misinterpretations, errors or omissions.

A customer is authorized to use the Planon software and its associated data files and user manuals within the terms and conditions of the license agreement between customer and the respective legal Planon entity as soon as the respective Planon entity has received due payment for the software license.

Planon Software Development B.V. strictly prohibits the copying of its software, data files, user manuals and training material. However, customers are authorized to make a back-up copy of the original CD-ROMs supplied, which can then be used in the event of data loss or corruption.

No part of this document may be reproduced in any form for any purpose (including photocopying, copying onto microfilm, or storing in any medium by electronic means) without the prior written permission of Planon Software Development B.V. No copies of this document may be published, distributed, or made available to third parties, whether by paper, electronic or other means without Planon Software Development B.V.'s prior written permission.

About this Document

Intended Audience

This document is intended for *Planon Software Suite* users.

Contacting us

If you have any comments or questions regarding this document, please send them to: support@planonsoftware.com.

Document Conventions

Bold

Names of menus, options, tabs, fields and buttons are displayed in bold type.

Italic text

Application names are displayed in italics.

CAPITALS

Names of keys are displayed in upper case.

Special symbols



	Text preceded by this symbol references additional information or a tip.
	Text preceded by this symbol is intended to alert users about consequences if they carry out a particular action in Planon.

Table of Contents

Authentication.....	6
Authentication methods.....	6
OpenID Connect (OIDC).....	8
OIDC concepts.....	8
Browser clients.....	8
Mobile apps.....	9
System integration.....	9
Default configuration per client.....	10
Planon App.....	10
Planon Connect for Analytics.....	11
Planon Connect for AutoCAD.....	14
Planon ProCenter.....	15
Planon SDK.....	16
Configuration of Authentication.....	19
Planon Cloud.....	19
Environment Management Gadget.....	19
Configuration.....	27
On-premise.....	50
Single Sign-On.....	51
WAFFLE.....	52
How WAFFLE SSO authentication works.....	53
Configuring the web server.....	54
Use domain user for the web server service.....	55
Apache SPNEGO implementation.....	56
How SPNEGO SSO authentication works.....	57
Generating a key tab.....	58

Amending the Tanuki configuration file.....	59
Configuring web server.....	60
Planon Web Client configuration.....	61
Verifying the configuration.....	62
Enabling logging.....	63
Configuring browsers.....	64
Configuring Internet Explorer.....	65
Configuring Chrome.....	66
Configuring Firefox.....	67
Troubleshooting SPNEGO and WAFFLE.....	68
Tomcat Keycloak adapter.....	69
Keycloak json.....	70
Configuring the web server.....	71
Index.....	72

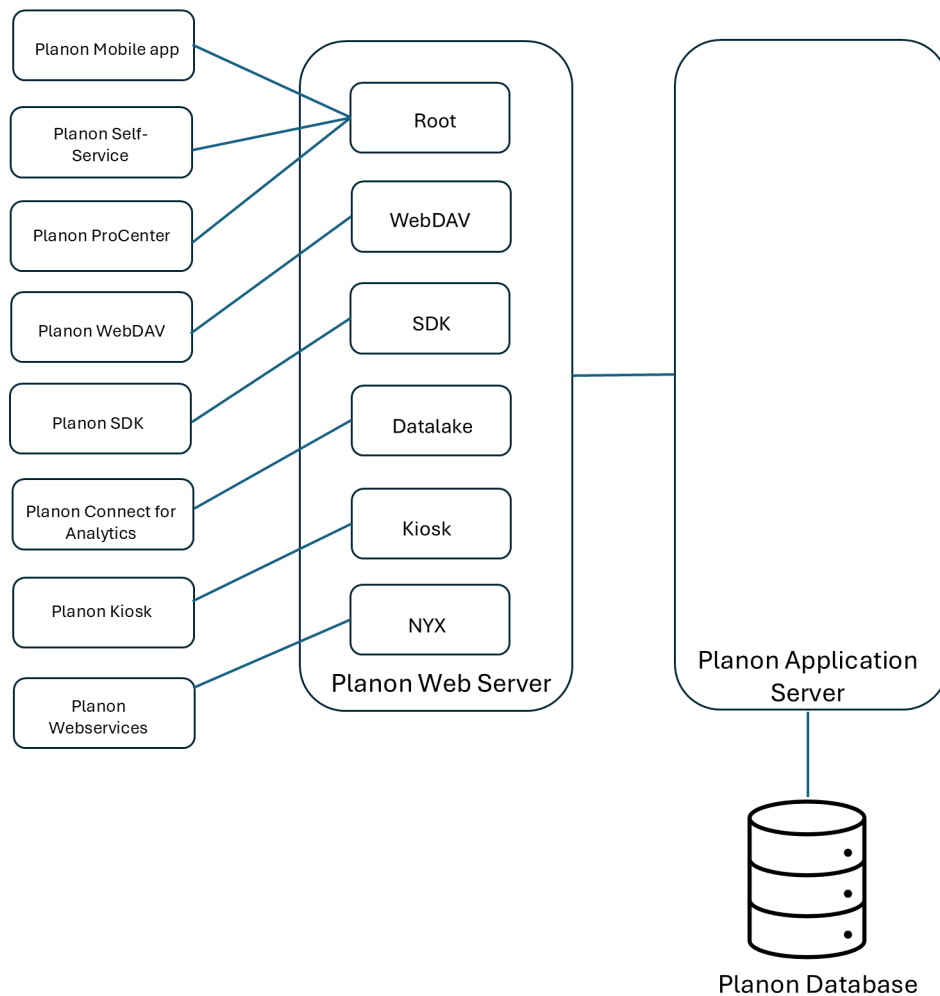
Authentication

This document explains the way Authentication is set up for the Planon Universe solution. It describes the technical aspects of the components involved.

Authentication methods

Introduction

Planon offers multiple technical clients in its solution, as shown in the following diagram:



Supported authentication methods

The following table shows the available authentication methods per client:

V = default enabled

X = not supported

O = optional

Client	Form authentication	Planon Access Keys	Basic authentication	Waffle	SPNEGO	Keycloak
Planon ProCenter	V	X	X	O	O	O
Planon Self-Service	V	V	X	O	O	O
Planon WebDAV	X	X	V	X	X	O
Planon SDK	V	V	X	X	X	O
Planon app	X	X	X	X	X	V
Planon AppSuite	V	V	X	X	X	O
Planon Connect for Analytics	X	X	V	X	X	O
Planon Kiosk	V	V	X	X	X	X
Planon SOAP Webservices*	X	X	X	X	X	X

*Planon SOAP Webservices authentication is part of the interface.

For more information about how to configure the various authentication methods for the available clients, see [Authentication \(Deployment Overview\)](#).

OpenID Connect

Planon is introducing a more future-proof authentication method for clients. For information on the introduction of and migration to the new **OpenID Connect** authentication, see [OpenID Connect](#).

OpenID Connect (OIDC)

Planon is migrating to a fully **OpenID Connect** based authentication. This section introduces the concept of **OpenID Connect**, also referred to as **OIDC**, within the Planon Universe solution. It also explains specific terminology and the technical details about authentication for the various Planon clients.

OIDC concepts

Planon Universe introduces **Keycloak** as part of the Planon Universe Suite.

The essence of **OpenID Connect** is that it sends a token to the application with every request. These tokens are generated at the **Keycloak** service. The way a token is obtained depends on the client's technology.

The newly introduced **Keycloak** service becomes the *identity broker* that forms the authentication layer for all Planon related components and services.



For Planon Cloud customers this solution is already available via the [Environment management gadget](#). For on-premise customers this solution will be introduced in the near future.

Options

- You can connect the **Keycloak** service to the Planon back-end to obtain a seamless transition for customers using the current form authentication in Planon and to store all user credentials in the Planon database.
- You can connect the **Keycloak** service to an external **Identity provider** to obtain a single-sign-on experience for end users. There are various protocols available to connect the external **Identity provider** to Keycloak, but Planon recommends OpenID Connect.

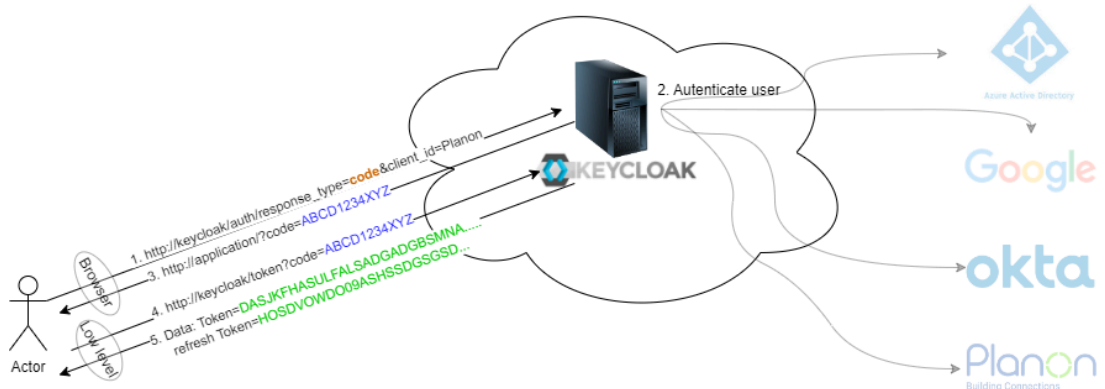
Browser clients

In browser applications, users are redirected to the **Keycloak** service when they visit the Planon Web application without being authenticated. The typical process is as follows:

- When users successfully authenticate to the source configured in the **Keycloak** service, they will receive an *authorization code*. This authorization code can be exchanged for an *access token*. The access token is a token with a short lifespan, usually 5 to 15 minutes.
- Together with the access token a *refresh token* is retrieved. If the access token has expired, a new set of tokens can be retrieved by exchanging the refresh token to the **Keycloak** service.
- The refresh token is a longer-lived token, usually 8 hours from the first time the token set was generated.
- Both the access token and the refresh token are stored in the *web server session*.

Mobile apps

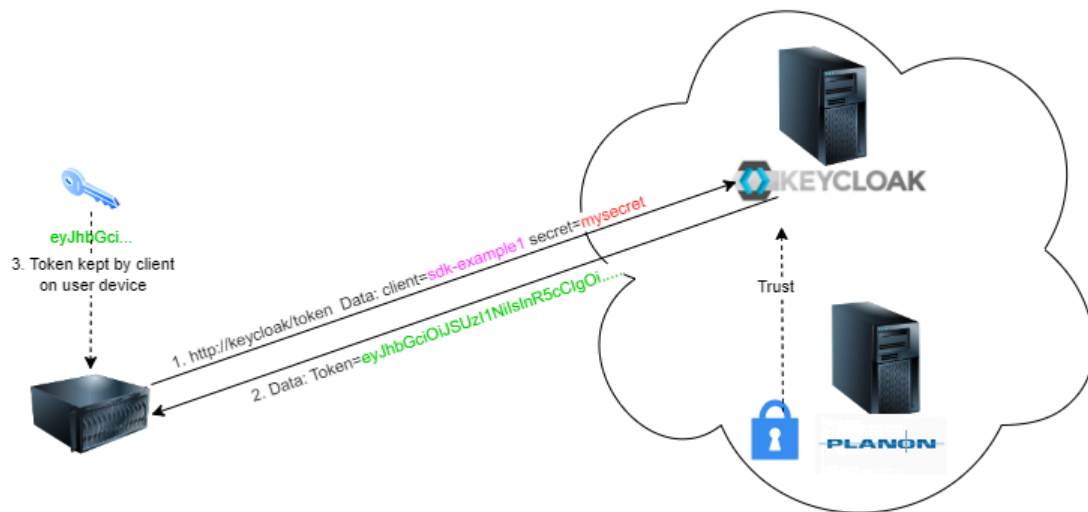
For mobile apps an *offline token* is generated which is a long-lived token. The default lifespan is 30 days. For an offline refresh token the default is 180 days. These authentication flows are called the **Authorization code flow**. By default, Planon uses **PKCE** (Proof key for code exchange) in its clients and also strongly recommends PKCE usage for additional clients, to enhance security with an additional layer. The following image provides a schematic representation:



System integration

Planon SDK is recommended for automated interfacing systems towards a Planon connection. The authentication to SDK is also token-based. To retrieve a token, a unique client is created in **Keycloak**. This client is generated with a **ClientID** and **client secret**. An access token can be retrieved from this client by sending the *ClientID* and *client secret* to **Keycloak**.

This flow is called the **Client credentials flow**.



Default configuration per client


In the following chapters you can find information about the default **OpenID Connect** configuration per Planon client:

- Mobile apps
- Connect for analytics
- Planon Connect for AutoCAD
- Planon ProCenter
- Planon SDK

Planon App

Planon Mobile needs the authorization code with a public client and **Proof key for code exchange** (PKCE) flow and will use *offline tokens*.

To use Planon Mobile with OIDC you must configure your cloud environment via the **Environment management gadget** on the **SSO** tab and enable **OpenID Connect** for the Mobile solution.

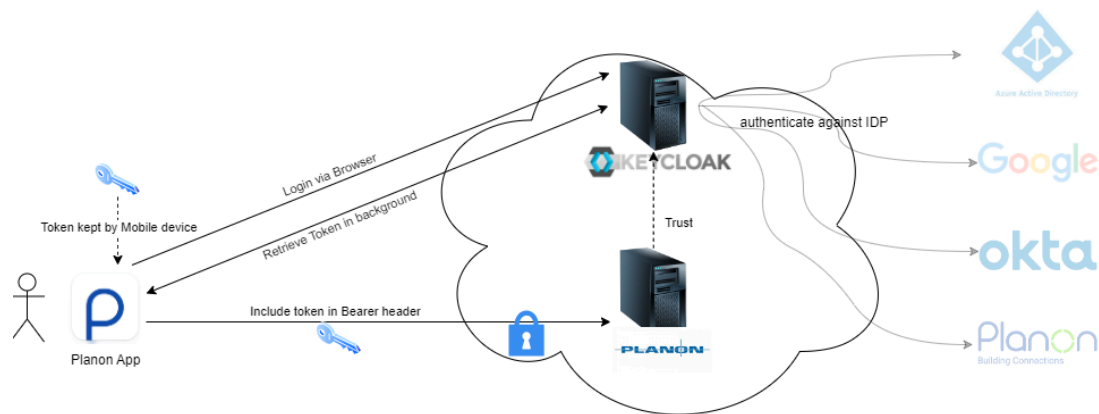
 Default Keycloak configuration is present in your environment. Make sure the **Offline session times** are always longer than one hour! Shorter session times might result in unexpected behavior.

Setting	Value
Client type	openid-connect
Client ID	planon-mobile-app

Setting	Value
Client authentication	Off
Authentication flow	Standard flow
Root URL	
Home URL	https://live.planon.app
Valid redirect URIs	https://live.planon.app/signin
Web origins	https://live-planon-app https://live-app planon://live-app
Access Token Lifespan	Expires after 15 minutes
Client Token Idle	Inherits from realm setting
Client Token Max	Inherits from realm setting
Client Offline Token Idle	Expires after 30 days
Client Offline Token Max	Expires after 180 days
Proof Key for Code Exchange Code Challenge Method	S256
Authentication, Required action: * Welcome the user	Enabled

* On-premise environments require a Keycloak Plugin.

Technical information - mobile apps



Planon Connect for Analytics

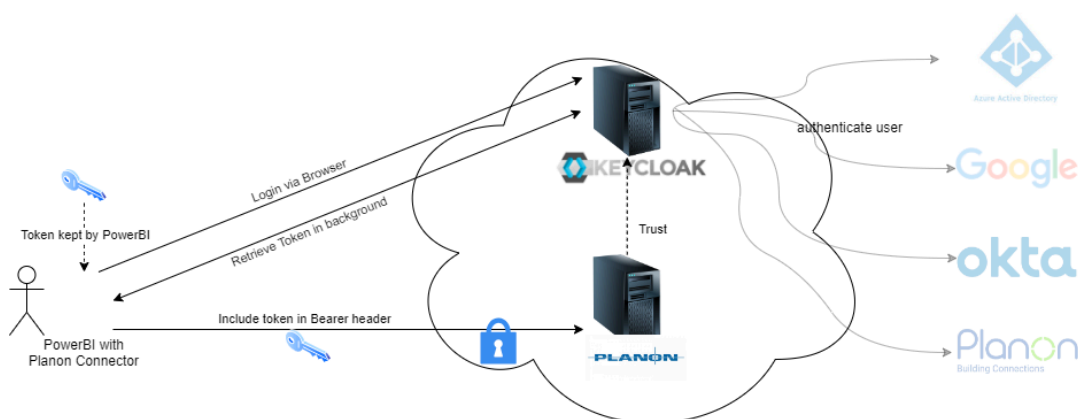
Connect for Analytics supports the authorization code flow with a **public client** and **Proof key for code exchange (PKCE)**. This way users will authenticate against the configured **Identity provider** or user provider.

To use Connect for Analytics with OIDC you must configure your cloud environment via the **Environment management gadget** on the **SSO** tab and enable **OpenID Connect** for the Planon Connect for Analytics solution. Additional **Keycloak** configuration is needed. You must add a public client with the following settings:

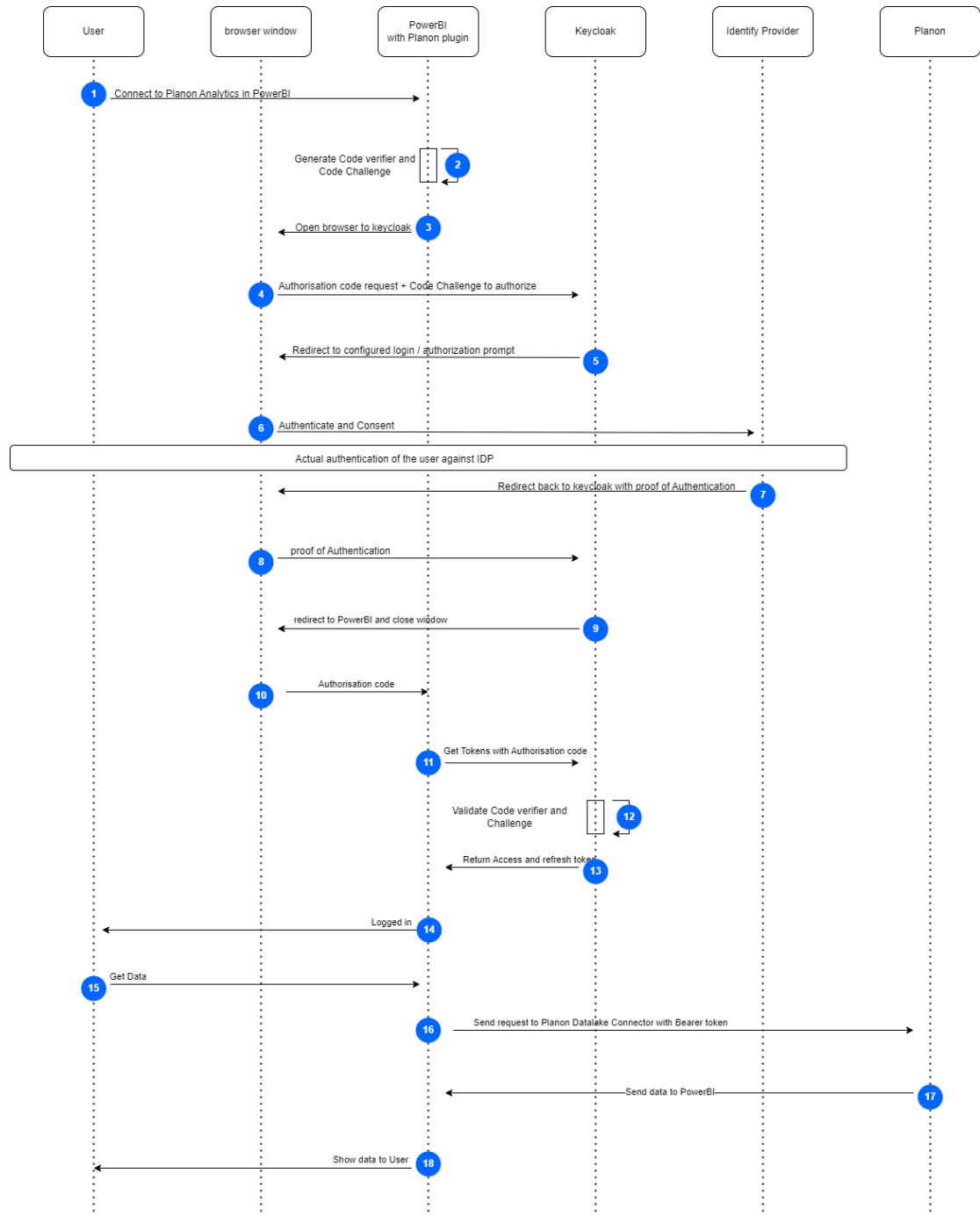
Setting	Value
Client type	openid-connect
Client ID	powerbi (all smaller cases)
Client authentication	Off
Authentication flow	Standard flow
Root URL	https://yourcloudenvironmenturl/datalake
Valid redirect URIs	https://oauth.powerbi.com/views/oauthredirect.html
Proof Key for Code Exchange Code Challenge Method	plain

Technical information - Planon Connect for Analytics

Using **OpenID Connect** as authentication protocol for **Planon Connect for Analytics** gives users access to the solution via authentication against the configured Identity Provider via Keycloak.



This will result in the following flow:



1. The user clicks **Sign in when Get Data** via Planon Connector .
2. Planon Connector generates a random code verifier and code challenge.
3. Planon Connector opens a browser window.
4. Planon Connector redirects the user to the Keycloak Authorization server along with the code challenge and gives PowerBI call-back URL with the request.

5. Keycloak sends a 'redirect' to the configured IDP.
6. The user opens the IDP and logs in.
7. User returns from the IDP as 'authenticated'.
8. There is a response from the browser to Keycloak that the user is logged in.
9. The user is directed to PowerBI.
10. An authorization code is sent from the browser to **Planon Connector** and the browser is closed.
11. Planon Connector sends an authorization code to Keycloak.
12. The code verifier and code challenge are verified.
13. Planon Connector retrieves an access token and refresh token.
14. The user sees that he/she is logged in.
15. The user clicks **Connect**.
16. When the request is sent, the access token is sent as **Bearer token** to the Planon Datalake.
17. The data is sent to PowerBI.
18. Data is shown to the user.

Planon Connect for AutoCAD

Planon Connect for AutoCAD needs the authorization code with a public client and **Proof key for code exchange** (PKCE) flow and will use *offline tokens*.

To use Planon Connect for AutoCAD with OIDC you must configure your cloud environment via the **Environment management gadget** on the **SSO** tab and enable **OpenID Connect** for the SDK solution.

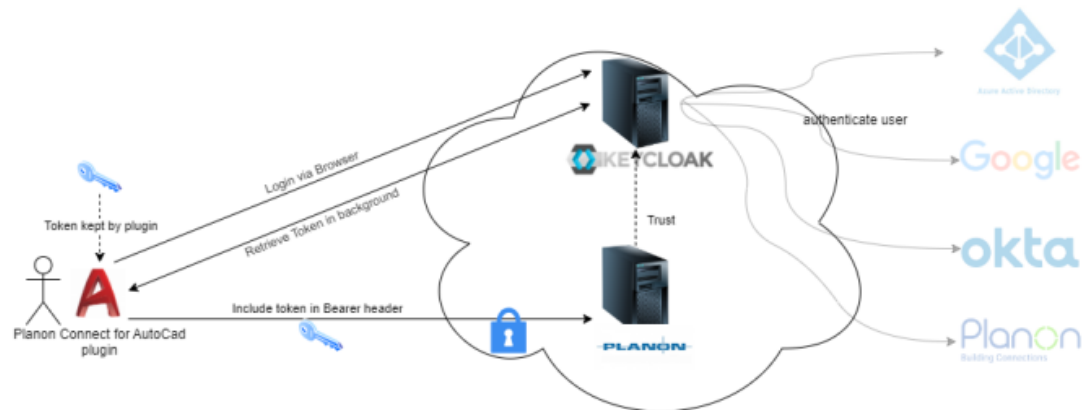
Default Keycloak configuration is present in your environment. Make sure the **Offline session times** are always longer than one hour! Shorter session times might result in unexpected behavior.

Client type	openid-connect
Client ID	PC4A
Name	Planon Connect for AutoCAD
Client authentication	Off
Authentication flow	Standard flow
Root URL	
Home URL	
Valid redirect URIs	pc4a://oidc_auth_callback

Web origins	
Front channel logout	Off
Backchannel logout session required	Off
Access Token Lifespan	Expires after 15 minutes
Client Token Idle	Inherits from realm setting
Client Token Max	Inherits from realm setting
Client Offline Token Idle	Expires after 30 days
Client Offline Token Max	Expires after 180 days
Proof Key for Code Exchange Code Challenge Method	S256

Technical information

Using OpenID Connect as authentication protocol for Planon Connect for Analytics gives users access to the solution via authentication against the configured Identity Provider via Keycloak.



Planon ProCenter

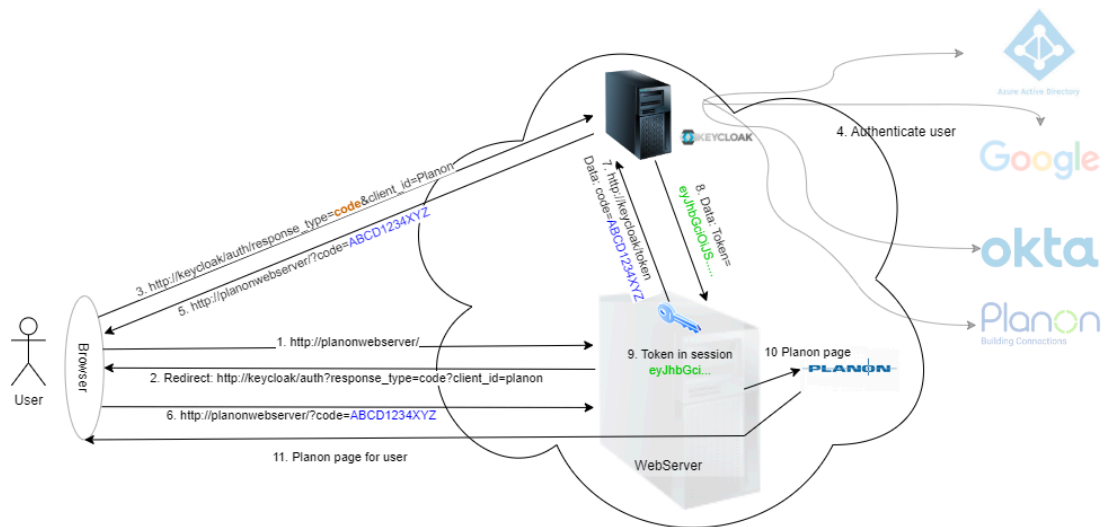
Planon ProCenter consists of a Web application and Planon Self-Service. Both solutions are configured via the same authentication method.

To use ProCenter with OIDC you must configure your Cloud environment via the **Environment Management Gadget**, on the **SSO** tab and enable **Single sign-on**.

The default **Keycloak** configuration is present in your environment.

Setting	Value
Client type	openid-connect
Client ID	Planon
Root URL	
Home URL	https://yourcloudenvironmenturl
Valid redirect URIs	https://yourcloudenvironmenturl/*
Web origins	https://yourcloudenvironmenturl
Admin URL	https://yourcloudenvironmenturl/webclient
Client authentication	On
Authentication flow	Standard flow
Proof Key for Code Exchange Code Challenge Method	Choose your preference and match with interfacing system

Technical information - ProCenter



Planon SDK

SDK supports both the authorization code with a public client and **Proof Key for code exchange** (PKCE) flow, as well as a client credentials flow.

It depends on the type of integration required, which grant type is preferred. For system-to-system integration, typically the client credentials grant is recommended. For an

integration that requires (end-)user interaction, it is recommended to make use of the authorization code flow.

To use SDK with OIDC please configure your cloud environment via the **Environment Management Gadget** on the **SSO** tab and enable **OpenID Connect** for the SDK solution.

Additional Keycloak configuration is needed. Please add a public client with the settings as described below to use authorization code flow:

Authorization code flow

Setting	Value
Client type	openid-connect
Client ID	"replace by a self-chosen name"
Client authentication	Off
Authentication flow	Standard flow
Root URL	https://yourcloudenvironmenturl/sdk
Valid redirect URIs	"url of the interface calling the sdk interface"
Proof Key for Code Exchange Code Challenge Method	Choose your preference and match with interfacing system (plain or S256)

For system-to-system authentication, the following template can be used.

Client credentials

Setting	Value
Client type	openid-connect
Client ID	"replace by a self-chosen name"
Client authentication	On
Authentication flow	Service accounts role
Root URL	https://yourcloudenvironmenturl/sdk
Valid redirect URIs	"url of the interface calling the sdk interface"
Proof Key for Code Exchange Code Challenge Method	Choose your preference and match with interfacing system (plain or S256)

When using the client credentials flow, a *service account user* must be present in Planon.

Example

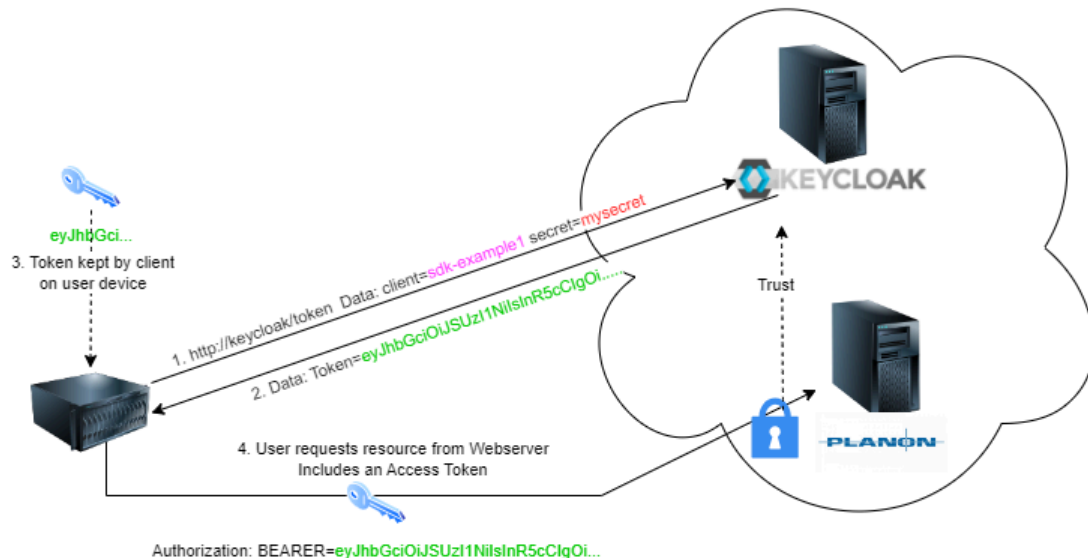
If the client name is *sdk-example1*, a user with account name *service-account-sdk-example1* must be present and active within the Planon application.

To get access to the SDK service via OpenID Connect, take the following steps:

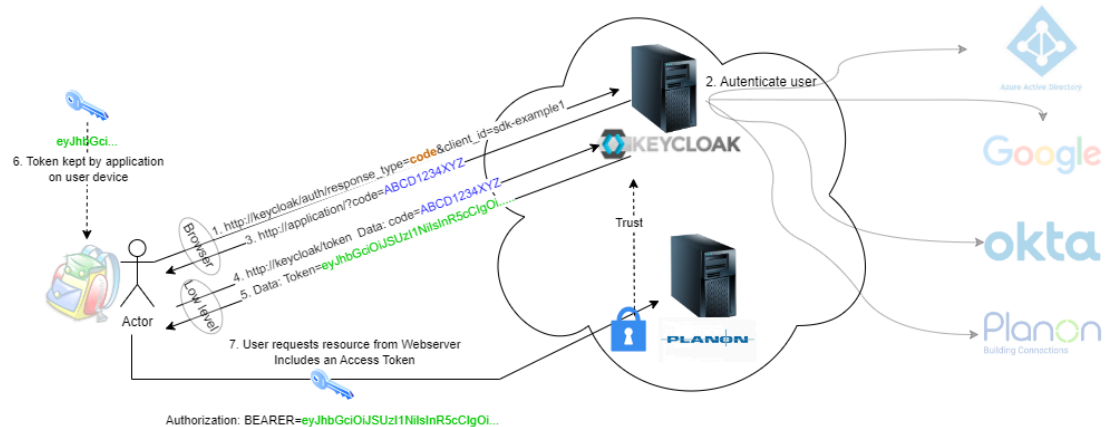
1. Retrieve an access token at the **Authentication service** via the client created in the installation steps.
2. Send the access token as a **Bearer token** to the Planon SDK service.

Technical information - SDK

Client credentials flow



Authorization code flow



Troubleshooting

Error	Description
401 Unauthorized	Either no access token or an already expired access token has been sent to Planon SDK service.
500 Internal error	The user account does not exist or is not active in the Planon application.

Configuration of Authentication

The Planon application is available for both Cloud customers and On-premise customers.

Because of their inherent implementation differences, the configuration of authentication also differs:

- [Cloud customers](#)
- [On-premise customers](#)

Planon Cloud

This section describes the information relevant to setting up authentication in Planon Cloud.

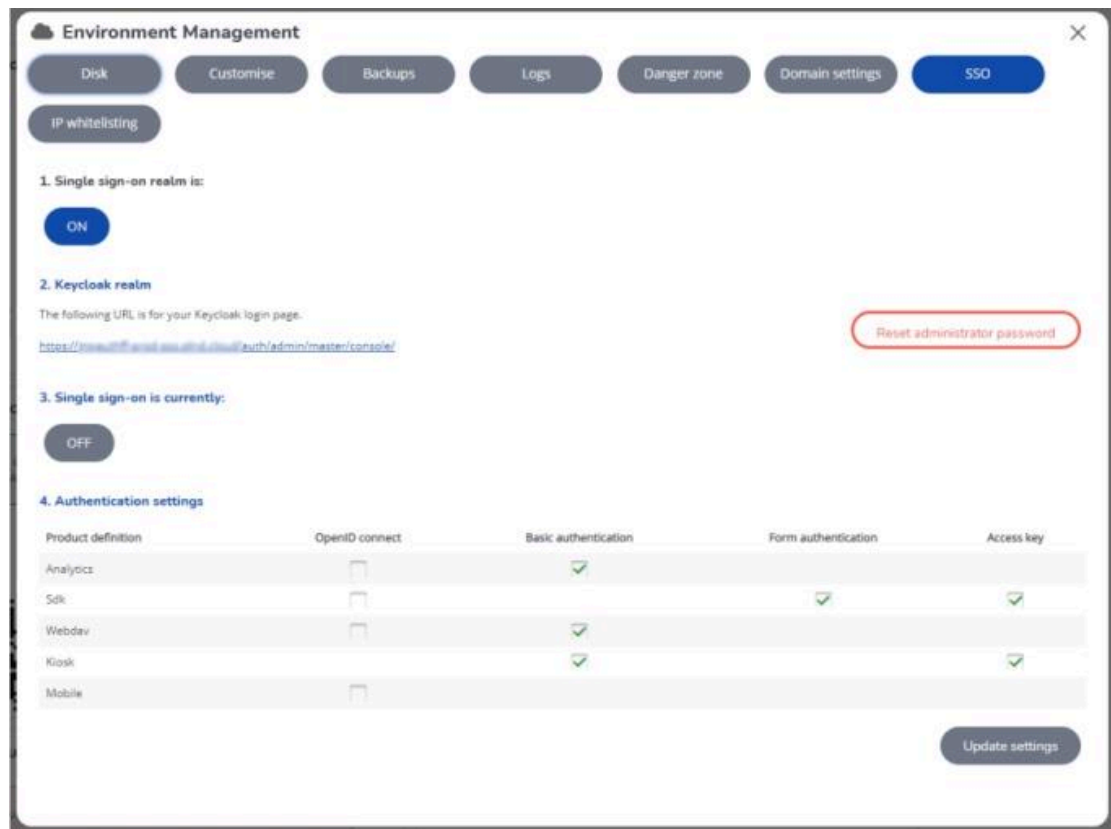
Environment Management Gadget

Authentication for Planon Cloud is managed in the [Environment Management gadget](#).

On the **SSO** tab, you can make various settings relevant to authentication as explained in the following topics.

SSO

When you are enabling SSO and click the tab for the first time, only a button is displayed indicating that the single-sign-on realm is **Off**.



i A realm is used to manage a set of users, credentials, roles, and groups. A user belongs to and logs into a realm. Realms are isolated from one another and can only manage and authenticate the users that they control.

Consequently, you first need to create a realm.

1. Click the **Off** button.
A warning message appears asking you to confirm enabling SSO.
2. Click **Enable SSO realm**.
The log in credentials appear (user name & password).
3. Store these credentials safely for later use. When this is done, select the check box and click **Continue**.

i If you misplace the credentials, you can reset the admin password here.

Your Keycloak environment is created, the login URL is displayed. You have finished creating the realm. Now, you can enable SSO.

4. In **3. Single-sign on is currently**, click the **Off** button to enable SSO.



A warning message appears. Read it carefully - restarting your environment may have some serious implications.

5. It is possible to switch the authentication method of the Connect for Analytics solution to OpenID Connect authentication.
Additional configuration of your BI Tool is required, for more information please contact Planon Support.
6. It is possible to switch the SDK authentication method to OpenID Connect authentication.
Additional configuration is required, for more information please see [OpenID Connect](#).
7. It is possible to switch the authentication method of WebDAV to OpenID connect authentication.
Additional configuration is required, for more information see [OpenID Connect > WebDAV](#).
8. It is possible to switch the authentication method for Mobile to OpenID connect authentication.
For more information see [OpenID Connect > Mobile](#).
9. Click **Save & rebuild**.
You have enabled SSO.



If you later disable SSO, the configuration will remain, but will be hidden.

OpenID Connect

It is possible to switch the Planon SDK to OpenID Connect (OIDC) authentication in the Environment management gadget.



This will currently break the Planon AutoCAD Plugin implementation, so if the Planon AutoCAD Plugin integration is used, do not switch your environment to OIDC authentication. This will be fixed in a newer version of Planon so that the Planon AutoCAD Plugin will support OpenID Connect in the near future.

The default behavior of the SDK is unchanged, this means if no additional configuration is done, form authentication and Planon access key is present.


Enabling OpenID connect disables form authentication. Planon Access key is optional supported in combination with OIDC, or Planon Access Key only. For more information, see the following table:

	Form Authentication	Planon Access Key	OpenID Connect
Option 1 (default)	Enabled	Enabled	Disabled
Option 2	Disabled	Enabled	Enabled
Option 3	Disabled	Disabled	Enabled
Option 4	Disabled	Enabled	Disabled

Installation

Planon Cloud configuration

1. Enable OpenID Connect authentication for SDK in the Environment Management gadget.

 In order to see this option, your environment must be running on the latest Cloud platform and SSO must be enabled.

2. In Keycloak, create a client with a self chosen client name (in the following image: *sdk-example1*). The root URL should be equal to the SDK interface URL.

Clients > Add Client

Add Client


Import

Client ID *

Client Protocol

Root URL

3. In the next screen, configure the client to meet up to your security policies and save the changes.

 ... Both **Client credentials** as well as **Authorization code flow** are supported.

... When using **Client credentials** flow make sure that **Service account** is enabled.

4. In Planon make sure a user is present that can be used by the configured client above. When **Client Credentials** flow is used, a service account user for the client must be present in Planon.

Example

If the client name is *sdk-example1*, than a user with the account name *service-account-sdk-example1* must be present and active in the Planon application.

Usage

To get access to the SDK service via OpenID Connect, take the following steps:

1. Retrieve an access token at the keycloak service via the Client created in the installation step.
2. Send this token as Bearer token to the Planon SDK service.

Troubleshooting

The following table lists a few common errors.

Error	Description
401 Unauthorized	Either no access token or an expired access token has been sent to the Planon application.
500 Internal error	The user account does not exist in the Planon application.

WebDAV

When enabling OpenID connect for WebDAV, in addition to the configuration mentioned in this article, you must also assign product definitions to the proper user groups.

When using Basic authentication, you can log on to the various WebDAV locations by using your environment's credentials.

After enabling OpenID connect for WebDAV, these credentials will no longer work. Instead, please assign the various product definitions for WebDAV to the relevant user groups.

The following WebDAV product definitions will be available:

- WebDAV
- WebDAV_Audit
- WebDAV_Backup
- WebDAV_PEET

- WebDAV_TMS
- WebDAV_Webservices

These product definitions will enable you to determine/authorize access to the various WebDAV locations.



Please note that assigning a WebDAV product definition to a user group is explicit. Without assigning WebDAV product definitions, no user can access WebDAV locations! See also: [Arranging access to Planon products](#) and subsequent articles.

Mobile

The Planon Live app will use offline tokens when OIDC has been enabled. The advantage of using offline tokens is that users need to authenticate a lot less.

Default behavior is that after initial log in, a user can use the app without further authentication once per 30 days. If the user uses the app at least once per 29 days he/she can use the app without re-authentication for maximum 180 days (from the initial log in).

If an administrator wants to change the default timings, this is configured in the Identity Broker environment under Clients / PlanonMobile / Advanced Settings.

- Client Offline Session Idle = 30 days (default)
- Client Offline Session Max = 180 days (default)



Please make sure the Offline Session times are always longer than 1 hour!! If set to a shorter timing unexpected behavior will occur.

Privacy sandbox compatibility

As of version L99, Planon provides the feature of Privacy sandbox compatibility. This feature ensures that your Planon cloud environment is compliant with upcoming deprecation of third party cookie support by Google (see [Privacy Sandbox for the Web](#) for more information).

- For environments enabling SSO the very first time, this feature is by default enabled.
- For existing customers already using SSO, additional configuration is required before this option can be enabled.

Prerequisites

The configuration of the Identity Provider (IDP) needs to be modified before you can enable this setting. Kindly request your IT organization to expand the current SSO configuration (configuration of your external Identity Provider (IDP)) for your Planon environment.

Request to add two additional redirect URLs alongside the existing allowed redirect URL.

- The first URL should be identical to the existing URL, but without the "-sso" part in the hostname.
- The second URL should be customized to match your custom domain (if no custom domain is configured, only the first additional URL is needed).

Example

Current redirect URL in the IDP configuration:

```
https://customerenvironment-prod-sso.planoncloud.com/auth/realms/planon/broker/saml/endpoint
```

First redirect URL to be added:

```
https://customerenvironment-prod.planoncloud.com/auth/realms/planon/broker/saml/endpoint
```

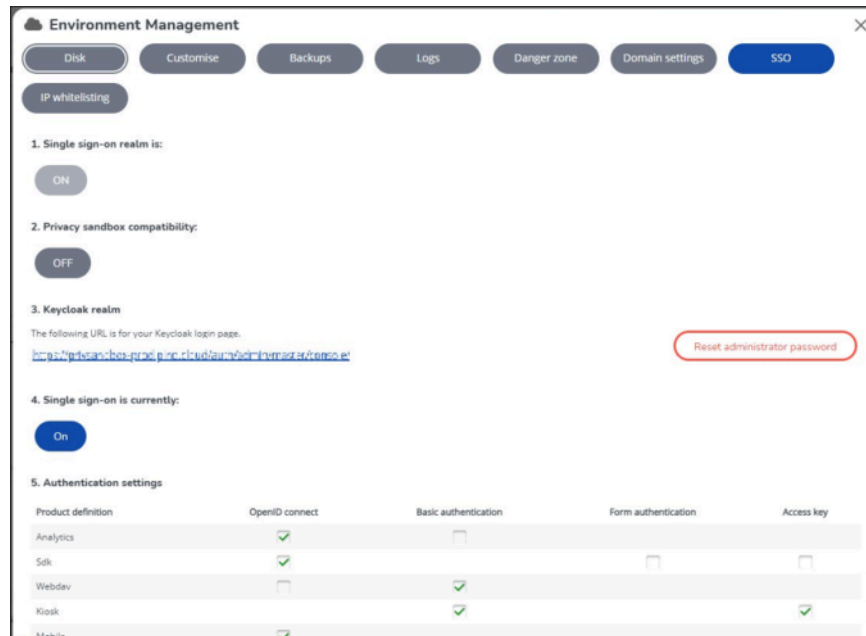
Optional, when a custom domain is configured (custom domain used in this example is `facilities.customer.com`):

```
https://facilities.customer.com/auth/realms/planon/broker/saml/endpoint
```

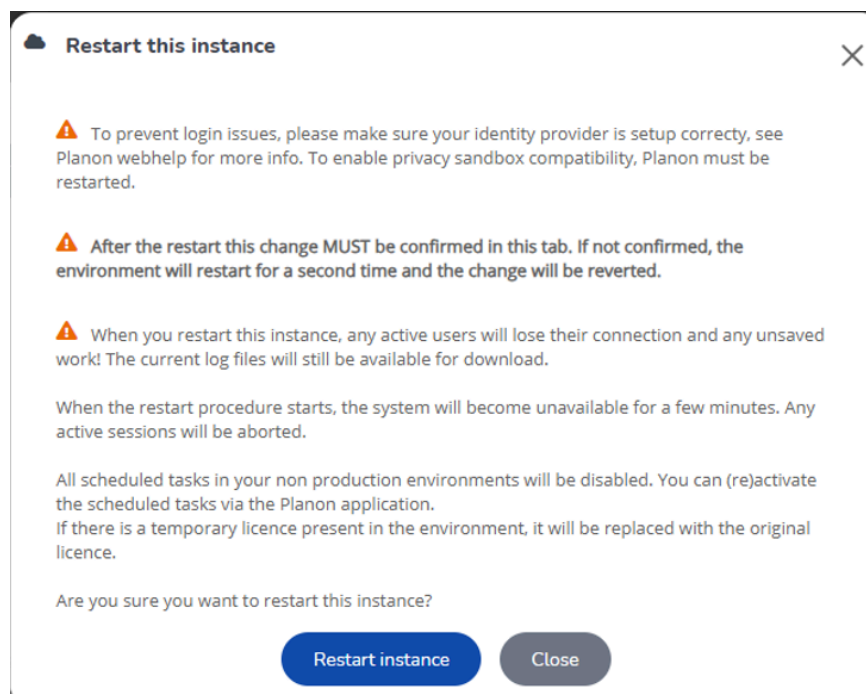
After the IT department confirmed the requested change, you can proceed.


Configuration

1. Enable the setting in the Environment Management gadget > **SSO** tab under the **Privacy sandbox compatibility** option by clicking the **OFF** button.



2. A pop-up will appear. Click **Restart Instance** to restart the environment.



 It is crucial that you log on to the Planon application and access the **Environment Management Gadget** within 10 minutes after the environment restarts. **Always Log on to your environment via Single Sign On (not as supervisor).**

On the **SSO** tab, please verify that the change has been successfully implemented. If the confirmation is not confirmed within 10 minutes after the environment has restarted, the environment will restart for a second time and the change will be reverted. This step is essential to prevent any

configuration errors resulting in making the Planon Cloud environment unusable.

The screenshot shows the 'Environment Management' interface with a top navigation bar containing buttons for Disk, Customise, Backups, Logs, Danger zone, Domain settings, and SSO. The SSO button is highlighted. Below the navigation bar, there is a section for 'IP whitelisting'. The main content area displays four configuration steps: 1. Single sign-on realm is: ON (button); 2. Privacy sandbox compatibility: Confirm (button) with a red warning message 'If not confirmed shortly, the environment will restart and this setting will be set back to OFF'; 3. Keycloak realm: The following URL is for your Keycloak login page. <https://planon-bes-poc.pln.cloud/auth/realms/planon/protocol/openid-connect/auth>; 4. Single sign-on is currently: On (button). A 'Reset administrator password' button is visible on the right side of the interface.

i After confirmation, please be aware that the option will be permanently enabled and cannot be disabled. Note: the improved feature [Activate Privacy Sandbox](#) will automatically be enabled when **Privacy Sandbox Compatibility** is confirmed.

This screenshot is identical to the one above, showing the 'Environment Management' interface with the same configuration steps for SSO. The 'Privacy sandbox compatibility' step is now 'ON' instead of 'Confirm', and the 'Single sign-on is currently' step is 'On'.

4. If a logoff URL is set for the environment, ensure it is updated to reflect the changes made.

If a custom domain is not used, remove the "-sso" part from the URL.

If a custom domain is configured, adjust the hostname to match the custom domain.

By following these steps, your Planon Cloud environment is future-proofed for the phasing out of third-party cookies.

Configuration

The following section describes how to configure SSO for Planon Cloud.

Using SAML

The following sections describe how to configure SAML SSO.

The Planon identity broker solution

Planon Cloud supports Single Sign On (SSO) based on SAML2, a process that allows users to authenticate themselves against an external customer side Identity Provider (IdP) rather than obtaining and using a separate user name and password handled by Planon Cloud.

Currently, only Service Provider initiated SSO is supported in the Planon Cloud.

To enable the customer to configure the SSO setup, Planon Cloud introduces an Identity Broker solution. The login information for this Identity broker solution will be provided to you by your Planon contact person.

For each Planon Cloud environment (Development, Test, Acceptance, Production), a separate Identity Broker solution is available.

The SSO feature is provided as a self service.

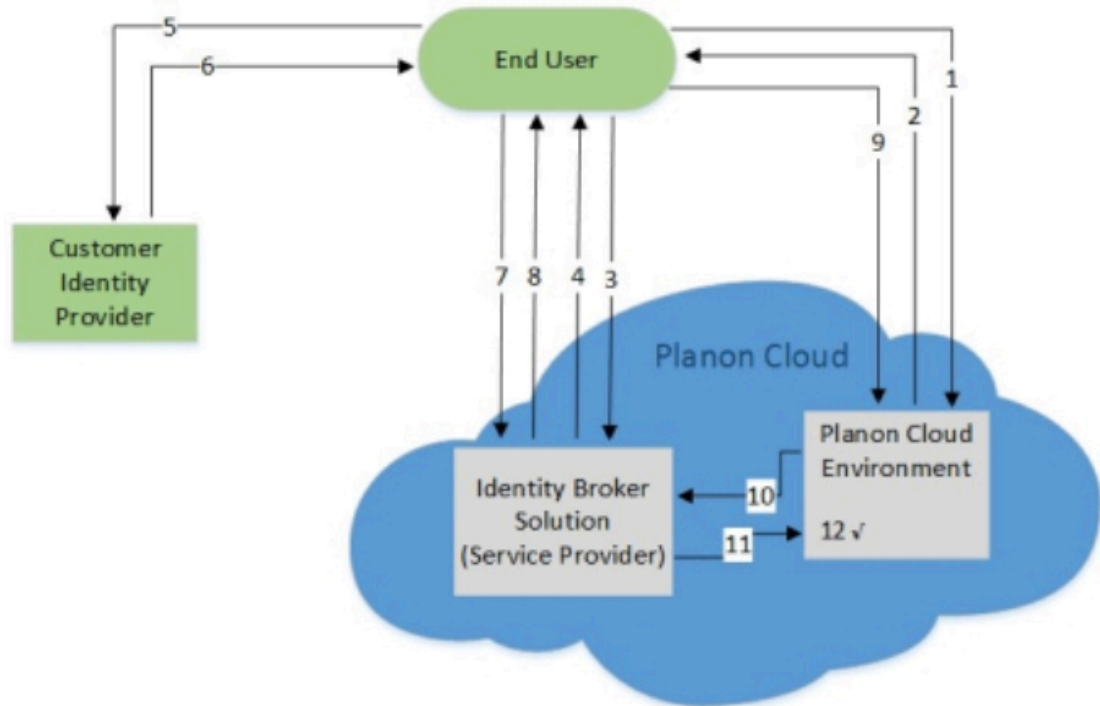
The customer can turn off/on and configure this feature at will. Planon recommends to check with a consultant /ICC person.

The SSO flow

A Single-Sign-On-enabled Planon Cloud environment consists of various components.

- End user
- Planon Cloud environment
- Identity Broker solution
- Customer Identity Provider (IdP)

The following image describes the connection between these various components in Planon Cloud:



The numbers in the diagram correspond with the steps below.



In this process, the end user is a browser.

Only a few of these actions will result in something that an actual end user will see. These steps are marked with an * and a short explanation is given on what the end user could experience.

1. End users request a resource from the Planon Cloud The service provider performs a security check on behalf of the target resource. (If a valid security context at the Identity Broker (service provider) already exists, skip steps 2–9).

End user experience: enter the Planon URL in the browser or click on a link that points to the Planon Cloud.

2. Planon Cloud Environment responds with a redirect to the Identity Broker solution.
3. End user requests login Identity Broker solution*.

End user experience: If manual login is enabled (default for all non-Prod environments), all users are moved to KeyCloak automatically and can be used to login. The user can click on the link to be redirected to IDP (step 4). If SAML Identity Provider is the default login method, the user is automatically redirected to IDP (step 4).

4. The Identity Broker responds with a redirect to the Identity Provider.
5. The end user requests login from the Identity Provider*.

End user experience: User views the login page or is automatically logged in to the Identity Provider, depending on the configuration at the customer.

6. After a successful login at the Identity Provider, the end user is redirected to the Identity Broker.
7. The end user visits the Identity Broker with a SAML post.
8. The Identity Broker responds with a redirect to the Planon Cloud.
9. Post configured attribute to Planon Cloud.
10. Planon Cloud checks if user session is valid at Identity Broker solution.
11. Identity Broker solution confirms, when the session is valid.
12. Only after the valid session confirmation, the user can access the requested resource.

End user experience: The user sees the requested resource at Planon Cloud. If the user name is unknown in the Planon Cloud Environment, an access failed message will be displayed.

Prerequisites - SAML assertion to be sent to Planon

The Identity Broker Solution requires a SAML response that contains the following two components:

A **NameID** (including a mandatory format description).

A separate **SAML attribute** that contains the identifier to map to Planon. (so not the NameID itself!)

In the example below, these mandatory components appear in bold.

The following excerpt is an anonymized sample of a SAML post to Planon:

```
<samlp:Response ID="_0216c6ce-7f8c-4e22-b6ca-d4cb9c6fc431"
  InResponseTo="ID_dbe02f23-e90a-4b04-a8ab-8af19632c7b5" Version="2.0"
  IssueInstant="2015-09-01T20:55:33.525Z"
  Destination="https://xx-yyy.planoncloud.com/"
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" >
  <saml:Issuer
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">urn:xyz:saml:idp</saml:Issuer>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod
        Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
```

```

<SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"
/>

<Reference URI="#_0216c6ce-7f8c-4e22-b6ca-d4cb9c6fc431">

  <Transforms>

    <Transform

      Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />

    <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">

      <InclusiveNamespaces PrefixList="#default samlp saml ds xs xsi"

        2001/10/xml-exc-c14n#" />

    </Transform>

  </Transforms>

  <DigestMethod

    xmlns="http://www.w3.org/Algorithm="http://www.w3.org/2000/09/xmldsig#sh
    a1" />

    <DigestValue>WrxQ8DfeSzygwXgKFbLLuK/iPvI=</DigestValue>

  </Reference>

</SignedInfo>

<SignatureValue>....</SignatureValue>

<KeyInfo>

  <X509Data>

    <X509Certificate>...</X509Certificate>

  </X509Data>

</KeyInfo>

</Signature>

<samlp:Status>

  <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success" />

</samlp:Status>

<saml:Assertion Version="2.0" ID="_e6db33a1-0724-4474-bdde-a9628e8223e0"

  IssueInstant="2015-09-01T20:55:33.525Z"

  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">

```

```

<saml:Issuer>urn:xyz:saml:idp</saml:Issuer>

<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">

  <SignedInfo>

    <CanonicalizationMethod

      Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />

    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"

      />

    <Reference URI="#_e6db33a1-0724-4474-bdde-a9628e8223e0">

      <Transforms>

        <Transform

          Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />

        <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">

          <InclusiveNamespaces PrefixList="#default saml ds xs xsi"

            org/2001/10/xml-exc-c14n#" />

          </Transform>

        </Transforms>

        <DigestMethod

          xmlns="http://www.w3.org/2000/09/xmldsig#"

          Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"

          />

        <DigestValue>Y1ksPiFQl6Mzh0nJrMNO2OMDtEI=</DigestValue>

      </Reference>

    </SignedInfo>

    <SignatureValue>...</SignatureValue>

    <KeyInfo>

      <X509Data>

        <X509Certificate>...</X509Certificate>

      </X509Data>

    </KeyInfo>

  </Signature>

</saml:Signature>

```



```

<saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameidformat:
persistent">username</saml:NameID>

<saml:SubjectConfirmation
  Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">

  <saml:SubjectConfirmationData NotOnOrAfter="2015-09- 01T20:58:33.541Z"
    Recipient="https://xx-yyy.planoncloud.com/"
    InResponseTo="ID_dbe02f23-e90a-4b04-a8ab-8af19632c7b5" />

</saml:SubjectConfirmation>

</saml:Subject>

<saml:Conditions NotBefore="2015-09-01T20:52:33.525Z"
  NotOnOrAfter="2015-09-01T20:58:33.525Z">

  <saml:AudienceRestriction>

    <saml:Audience>https://xyyyy.
      planoncloud.com/auth/realms/environment-test</saml:Audience>

  </saml:AudienceRestriction>

</saml:Conditions>

<saml:AuthnStatement AuthnInstant="2015-09-01T20:55:33.541Z"
  SessionIndex="_e6db33a1-0724-4474-bdde-a9628e8223e0">

  <saml:AuthnContext>

    <saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Passwo
      rd</saml:AuthnContextClassRef>

  </saml:AuthnContext>

</saml:AuthnStatement>

<saml:AttributeStatement>

  <saml:Attribute Name="email">

    <saml:AttributeValue
      xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi
        ="http://www.w3.org/2001/XMLSchema-instance">USERNAME@email.com</saml:At
      tributeValue>

  </saml:Attribute>

```

```
</saml:AttributeStatement>

</saml:Assertion>
```

For more information on how to check a SAML assertion, please see [SSO troubleshooting](#).

Activating Keycloak



The following Keycloak configuration is an example only. No rights can be derived. Screenshots and example may differ from your situation. If you need assistance in setting up your local specific configuration of keycloak, please contact your account manager.

Take the following steps to activate Keycloak.

1. Turn on SSO by using the Environment Management gadget, save the URL and user name/password.
2. Open the URL saved in step 1 and log in with the initial credentials.

3. At the first login, you will be prompted to complete your profile.

Provide a valid email address. A verification email will be sent to you to enable your account.

4. You will be prompted to change your password.

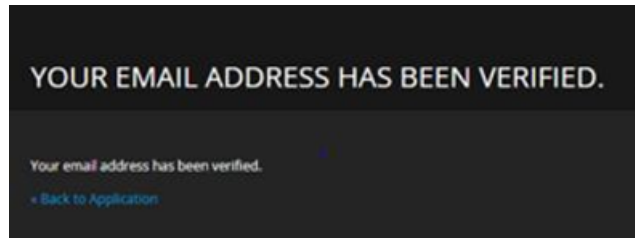


You can use the password saved at step 1 as the **New Password**. This way the password can always be looked up in the **Environment Management** gadget.

A verification email will be sent to you to the email address you have provided. This email contains an activation link for your account.

Note that the link in this email will expire within 5 minutes.

5. After verifying the email address, you can log in to the Planon Cloud Identity Broker solution.



Configuring Keycloak



The following Keycloak configuration is an example only. No rights can be derived. Screenshots and example may differ from your situation. If you need assistance in setting up your local specific configuration of keycloak, please contact your account manager.

Take the following steps to configure Keycloak.

1. In the menu on the left panel, select **Identity Providers (IdP)**.
2. Select SAML in the list to modify the preconfigured settings.
3. Modify the details in the data section, you can configure the settings here (the Redirect URI is automatically set for you):

4. The information for the fields under SAML Config need to be provided by the customer.

These are the details of the Identity Provider (IDP) on the Production environments (and recommended on Non-Production environments):

- **Want Assertions Signed** must be **ON**
 - **Validate Signature** must be **ON**.
5. Click **Save** to add the configuration to the **Identity Broker solution**.
 6. Click the **Mappers** tab. Click on **attributetoplanon**.

Identity Providers > salesforce > Identity Provider Mappers > attributetoplanon

Identity Provider Mapper attributetoplanon

ID	c26991bc-caed-404c-9e6d-bcb8e3e4725c
Name *	attributetoplanon
Mapper Type	Attribute Importer
Attribute Name	please change this
Friendly Name	
User Attribute Name	user.attributes.plnuid

7. Modify the **Attribute Name** with the correct IDP SAML attribute.

This will also be provided by the customer.

Do not fill the field **Friendly Name** and do not modify the field **User Attribute Name**.

8. Click **Save** to activate the updated attribute mapper to the configuration.

Replacing the certificate

For enhanced control over their own Cloud environments, customers can further tweak single-sign-on configuration.



The following Keycloak configuration is an example only. No rights can be derived. Screenshots and example may differ from your situation. If you need assistance in setting up your local specific configuration of keycloak, please contact your account manager.

1. In the Environment Management gadget > SSO tab, log on to Keycloak by clicking the Identity broker URL.

The Keycloak console appears.

2. In the left panel, select Identity Providers (IdP).
3. Select SAML in the list to modify the preconfigured settings.
4. Create a backup of the data in the fields Single Sign-On Service URL and Validating X509 Certificates.

5. Replace the values in these fields with the *Single Sign-On Service URL* and *X509 Certificate* provided by the Identity Provider.
6. Click Save.

The changes are active directly and can be tested immediately. To do this, close the browser completely and open a new session to validate the login.



For more information on configuring Keycloak, see Keycloak's [Server Administration Guide](#).

Rollback

Should the credentials provided in the fields **Single Sign-On Service URL** and **Validating X509 Certificates** not function correctly for any reason and the previous values need to be reinstated, replace the values with those you backed up earlier. This will reactivate the former settings.

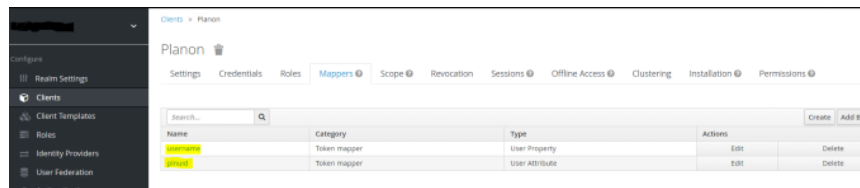
Rearranging the mappers

After configuring the identity provider, please make sure the mappers for the Planon client are in the correct order.

To find and check the current order of mappers, proceed as follows:

1. Click on Clients in the left panel, and click on the Planon client.
2. Open the Mappers tab, and make sure the order is:
 - username
 - plnuid

Example



Name	Category	Type	Actions
username	Token mapper	User Property	Edit Delete
plnuid	Token mapper	User Attribute	Edit Delete

If these mappers are not in the correct order, delete and recreate them in the correct order. (You may need to do this a couple of times to get it right).

Service Provider metadata



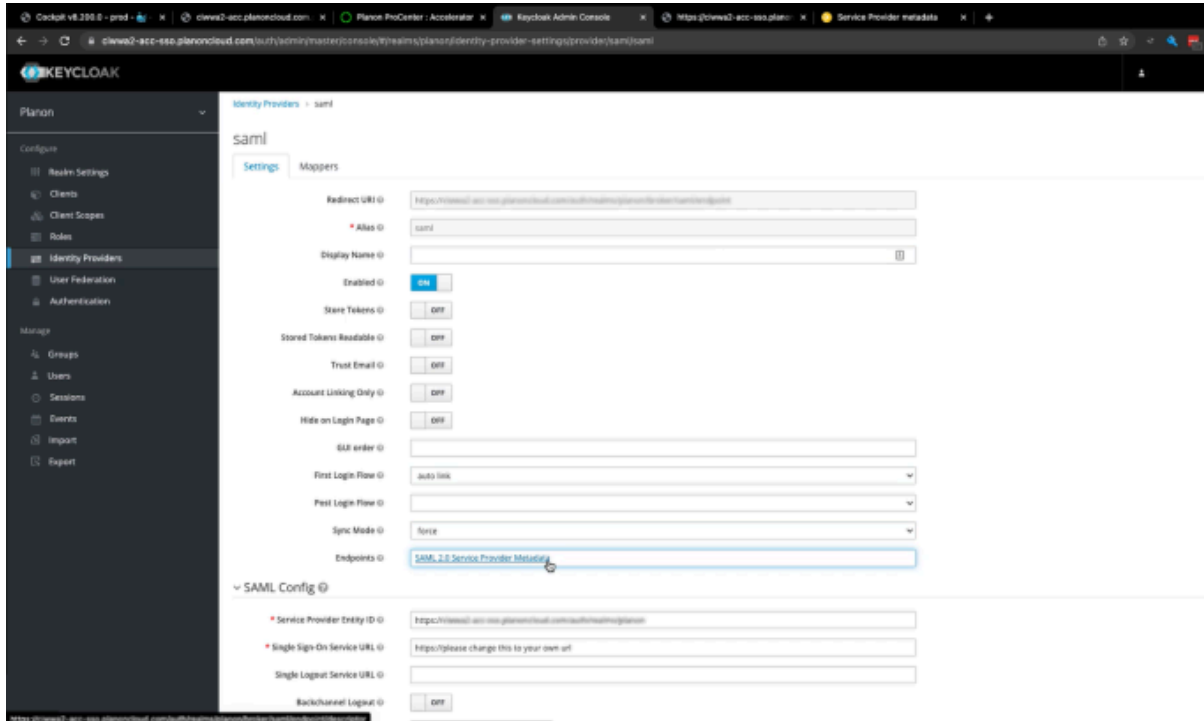
The following Keycloak configuration is an example only. No rights can be derived. Screenshots and example may differ from your situation. If you need assistance in setting up your local specific configuration of keycloak, please contact your account manager.

You must also send some details on SSO - the Service Provider Metadata - to the customer. If there are any configuration changes in the metadata, they can be exported via the Identity Broker solution.

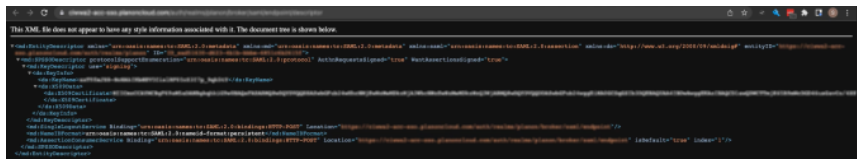
1. In the menu on the left panel, select **Identity Providers (IDP)**.

2. Select the Identity Provider just created.
3. Click the link in the **Endpoints** field.

It may not directly be apparent that this is a link, but if you hover over the field, the URL will be displayed at the bottom of your browser.



Clicking the link opens the metadata page in your browser:



4. Share this URL with the customer IDP administrator to establish a trusted relation between IDP and the Service Provider.

If the logon page is enabled, you can still automatically be redirected to the desired IDP by adding the following parameters to the URL:

?kc_idp_hint=<IDP Alias>

Example

https://customer-prod.planoncloud.com/?kc_idp_hint=saml

If a redirect to the default IDP is enabled, you can go to the login page by entering a different value.

Example

https://customer-prod.planoncloud.com/?kc_idp_hint=lmas

Custom domain allowance

When adding a custom domain to your Planon Cloud environment additional configuration needs to be done in Keycloak to be able to use Planon via the custom domain in combination with Single Sign on.

Take the follow steps to configure Keycloak.

Procedure

1. In the menu on the left panel select **Clients**.
2. In the list that is displayed, select **Client ID** Planon.
3. Add the Custom Domain URL in the **Valid Redirect URIs** by typing the URL followed by `/*`

The screenshot shows the 'Client details' page for a client named 'Planon' in Keycloak. The 'Settings' tab is selected. The 'General Settings' section includes fields for 'Client ID' (Planon), 'Name', and 'Description'. The 'Always display in UI' toggle is turned off. The 'Access settings' section includes fields for 'Root URL', 'Home URL' (https://<name>.planoncloud.com), and 'Valid redirect URIs'. The 'Valid redirect URIs' field contains three entries: 'https://<name>.planoncloud.com/*', 'https://<custom domain url>/*', and 'https://<after logoff url when configured>'. A link 'Add valid redirect URIs' is visible at the bottom of the list. On the right side, there is a 'Jump to section' menu with links to 'General Settings', 'Access settings', 'Capability config', 'Login settings', and 'Logout settings'.

4. Click **Save** to add the redirect URL.

Logging out of Planon Cloud

This URL makes a user log off from Planon, sends a log off request to the Identity Broker solution and redirects the user to the given redirect URL.

The redirect URL must be configured in Identity Broker solution.

Procedure

1. Login to the Identity Broker solution.
2. On the left side, select **Clients**.
3. Select **Planon**.
4. Add the value of the redirect URL in the **Valid Redirect URIs** by typing the URL followed by `/*`

Clients > Planon

Client Planon

Settings | Credentials | Roles | Mappers | Scope | Revocation | Sessions | Clustering | Installation

Client ID: Planon

Name: Planon

Enabled: ☒ ON

Consent Required: ☐ OFF

Direct Grants Only: ☐ OFF

Client Protocol: openid-connect

Access Type: confidential

* Valid Redirect URIs:

- https://<name>.planoncloud.com/*
- http://www.planonsoftware.com

Base URL: https://<name>.planoncloud.com


Admin URL:

Web Origins:

- https://<name>.planoncloud.com/*


Save Cancel

5. Click **Save** to add the redirect URL.

 Note that logging out of Planon Cloud but not from IDP only works if you do not configure a Single Logout Service URL on the Identity Provider page.

KeyCloak secure configuration considerations

This section lists a number of security considerations that can enhance your security level when using Planon Single Sign On (SSO).

 Please be aware that these configuration settings are considerations that highly depend on the customers' requirements, their Identity Provider and the security policies within the customers' organization. Only IT staff that is trained in these configurations should deploy these considerations or contact Planon for consultancy.

Authentication

External identity provider

When delegating authentication to an external identity provider (IdP) you should consider the following:

Subject	Description
Local account password	When a user logs in using an external identity provider, KeyCloak will create an account in it's local store.

Subject	Description
	<p>By default, it is possible for users to set a password on this account and use the user name and the KeyCloak local password to login.</p> <p>As this bypasses the external identity provider, this may be undesired.</p> <p>This behavior can be disabled at two places:</p> <ul style="list-style-type: none"> • Configure > Authentication > Required actions and disable Update password. When this is disabled, users can no longer set the password on the local KeyCloak account. • Configure > Authentication > Flows > Browser and disable the forms. When this is disabled, the password screen can no longer be used. Please be aware that this option will also disable all local keycloak accounts just as supervisor.
Forcing external IdP login	<p>What also could be considered is to make the login via an external IdP mandatory in the browser flow by setting the Identity Provider Redirector to required; this way, you cannot authenticate against other sources than your own IdP.</p> <p>This can be configured by going to:</p> <p>Configure > Authentication > Browser and configure the Identity Provider Redirector as Required.</p>

Using Planon user federation

It is possible to authenticate using the Planon system as an authentication source. Credentials entered in the KeyCloak user name and password fields are validated against the Planon credential store.

Subject	Description
Local account password	<p>When a user logs in using the Planon user federation, KeyCloak will create an account in it's local store. By default, it is possible for users to set a password on this account and use the user name and the KeyCloak local password to login. This bypasses the Planon user federation check so this may be undesired.</p> <p>This can be configured by going to:</p> <p>Configure > Authentication > Required actions and disable Update password.</p>

Subject	Description
	When this is disabled, users can no longer set the password on the local KeyCloak account.

KeyCloak local account password

Subject	Description
Password policy	<p>When using the local KeyCloak passwords, it is advised to set a password policy. This can be done in:</p> <p>Configure > Authentication > Policies > Password policy.</p> <p>Here, you can add policies for the different aspects of the passwords. Planon recommends setting the password policy in accordance with your organization's security policies.</p>
Brute force protection	<p>Brute force detection will be enabled by default. However, customers can set up their own metrics if desired.</p> <p>The Brute force detection settings can be found under:</p> <p>Configure > Realm settings > Security defences > Brute force detection</p>

General settings

Subject	Description
Multi-factor admin	<p>We strongly recommend to set up multi-factor authentication on the admin account. This can be done by:</p> <p>User name (top right of your screen) > Manage account > Account security > Signing in > Two-factor authentication</p>
Security headers	<p>Customers can set up their own security headers if desired.</p> <p>The security headers settings can be found under:</p> <p>Configure > Realm settings > Security defences > Header</p>

Testing the solution

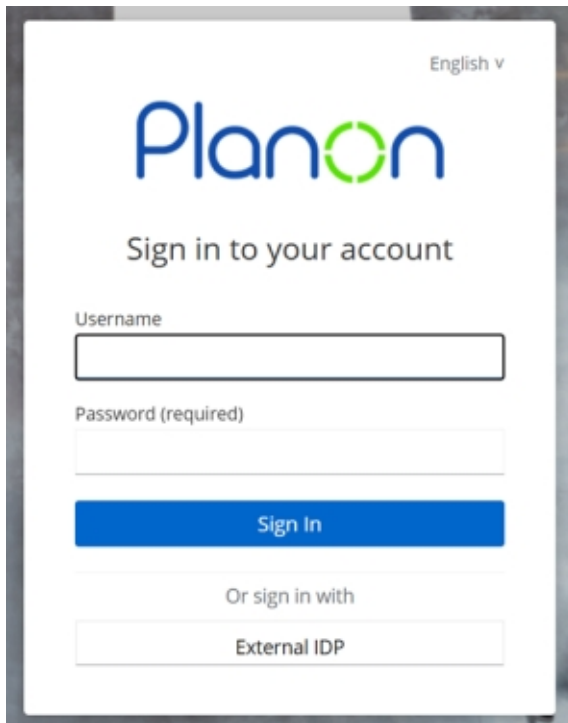
Prerequisite

Ensure that the UID for the test user is present as an account in the Planon Cloud environment for a full working test.

Procedure

1. Visit the main URL of the Planon Cloud environment. The default login page is being replaced by the Identity Broker login page.

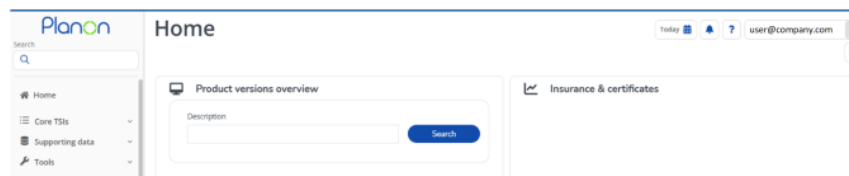
For production, the configured IDP will be configured as default and no manual login will be possible. The users will be automatically redirected to the IDP page.



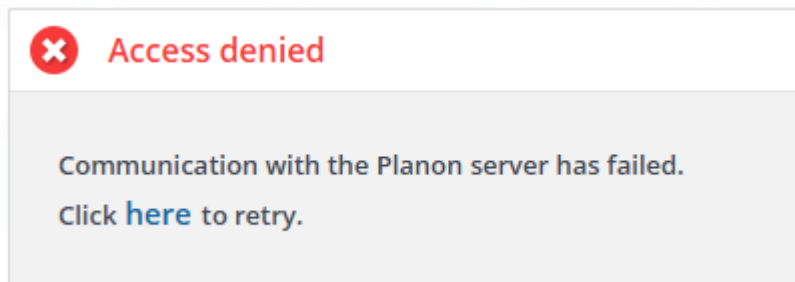
2. To check the SSO solution, click on the alias you configured in the Identity Broker (only for environments other than Production).

You will be redirected to the configured IDP and - if authenticated already - automatically logged in.

You will be logged in to Planon with the configured UID (in the example, the user e-mail address).



If you do not see the Planon screen, but an Access denied message, it means that the SSO login was successful but the users UID could not be resolved as a Planon Cloud account.



SSO troubleshooting

For troubleshooting the SSO configuration, Planon recommends to use Mozilla Firefox in combination with the add-on **SAML Tracer**. This add-on lets you read the messages being sent between the end user (browser), the Identity Broker (Service Provider) and the IDP (Identity provider at customer side).

Make sure the SAML Tracer is enabled when visiting the Planon Cloud environment. All http messages will be recorded. If a message contains a SAML request, it is highlighted and the SAML request can be viewed in the SAML tab. Please ensure that the SAML assertion sent by the Identity Provider meets the prerequisites.

Common issues

- No format in **NameID**
- No separate SAML attribute present (this is not needed when the **NameID** is used as the identifier)

Keycloak

It is possible to configure Keycloak to validate the entered user name and password against the accounts stored in the Planon database.

Having this in place renders a Planon Cloud environment suitable for OpenID Connect authentication without having to use an external authentication source (IDP).

Prerequisites

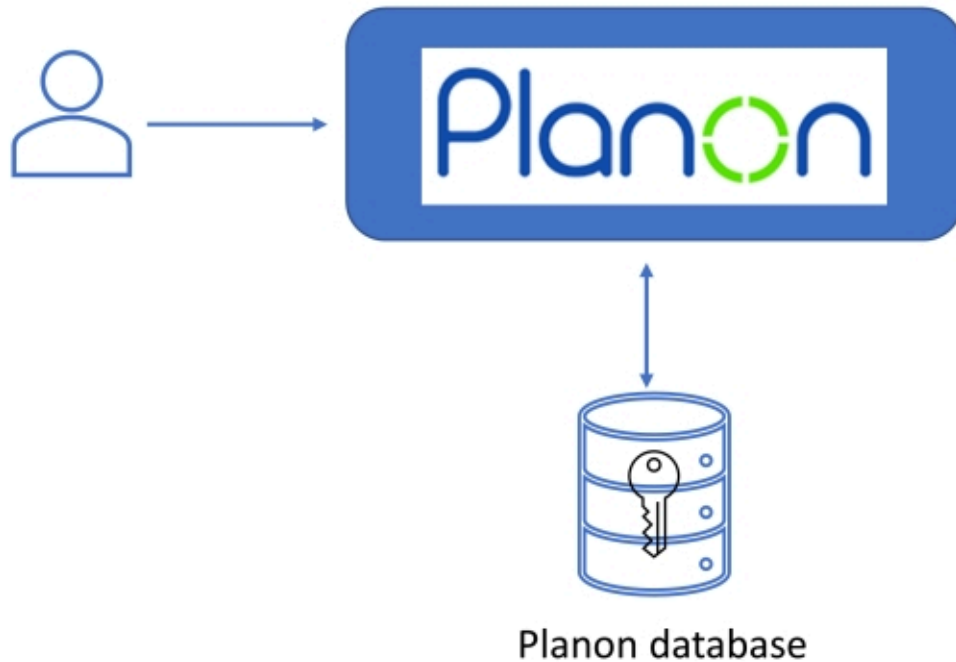
Before configuring this feature, please note the following requirements:

- This feature is currently only available on Planon Cloud environments.
- The environment must have Keycloak enabled (**SSO** tab in the [Environment management gadget](#)).
- The Planon version L92 or later.

Overview

When a Cloud environment is delivered, the following way of authentication is the default configuration.

 This configuration is also the default **on-premise** configuration; it uses form authentication for the Planon environment.



Keycloak

Enabling Single Sign On (**SSO**) on a Cloud environment introduces Keycloak authentication. Keycloak can be configured to use different authentication *sources*.

January 1, 2025


On January 1, 2025, Planon User federation will be implemented. Please note that this is release independent (from L92 onwards). By default this automatically applies to newly configured SSO environments.

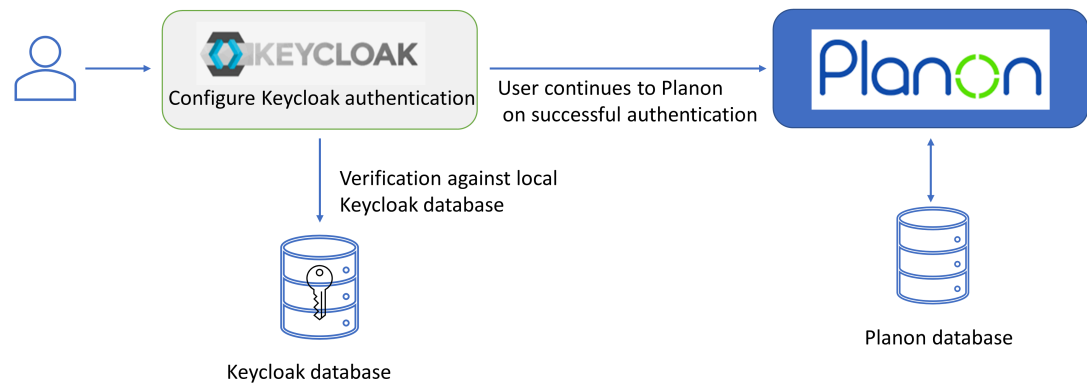
- Planon User federation applies to new and existing customers.
 - For existing customers who have configured SSO (in the past), nothing will change.
 - For existing customers who have not yet configured SSO, this will apply.
 - For new customers, this automatically applies.
- What it means:

If a customer is using a Planon environment (L92 or later) and enables SSO (Keycloak) after January 1, 2025, the current user passwords will remain working via the Planon User provider in Keycloak.

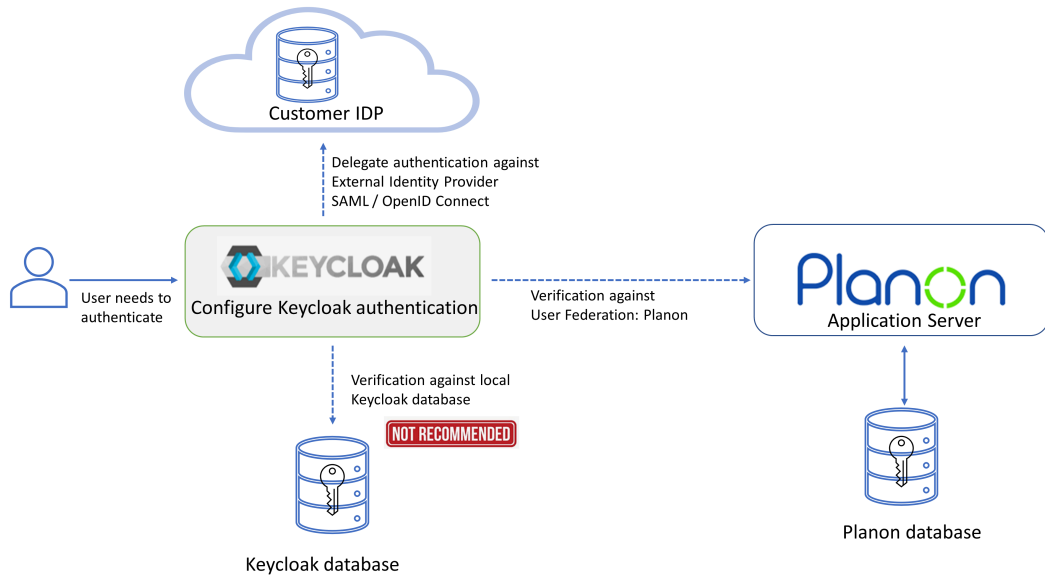
Keycloak authentication

By default, the authentication via Keycloak is configured as follows.

 This configuration needs to be adjusted by the customer according to the customer's specific (security) requirements. The default configuration already contains all existing users that are able to log in to Planon.



The following diagram shows the possible configuration options for authenticating users. This includes the configuration that needs to be applied by the customer.



Customer's options

The customer can choose to:

- Add accounts to the Keycloak database for users to authenticate against Keycloak.



This is not recommended!

- Add Planon provider in Keycloak under User Federation (current out-of-the-box-solution).

This way, users authenticate against the account in Planon database via Keycloak

- Add a external IDP under Identity Providers in Keycloak and disable Planon User Federation.

This way, users authenticate only against the external IDP of the customers choice via Keycloak.



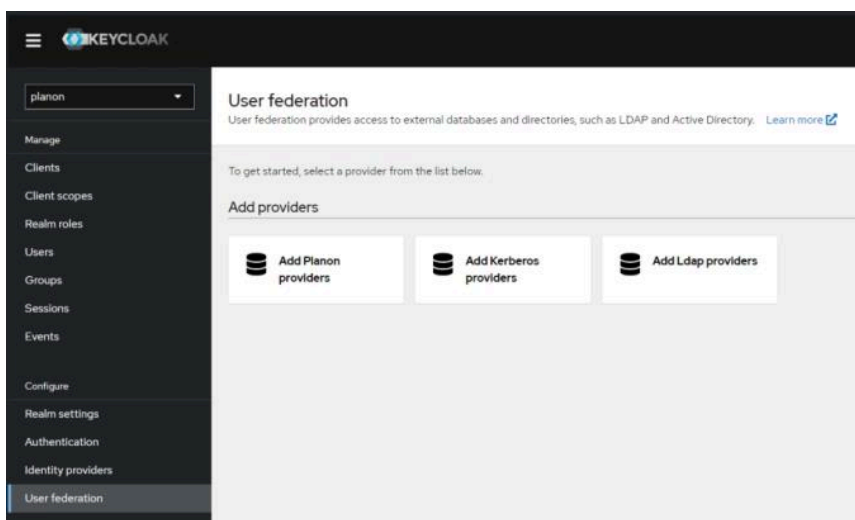
This is the recommended solution.

Configuring Planon User federation

Proceed as follows to configure User Federation: Planon.

Procedure

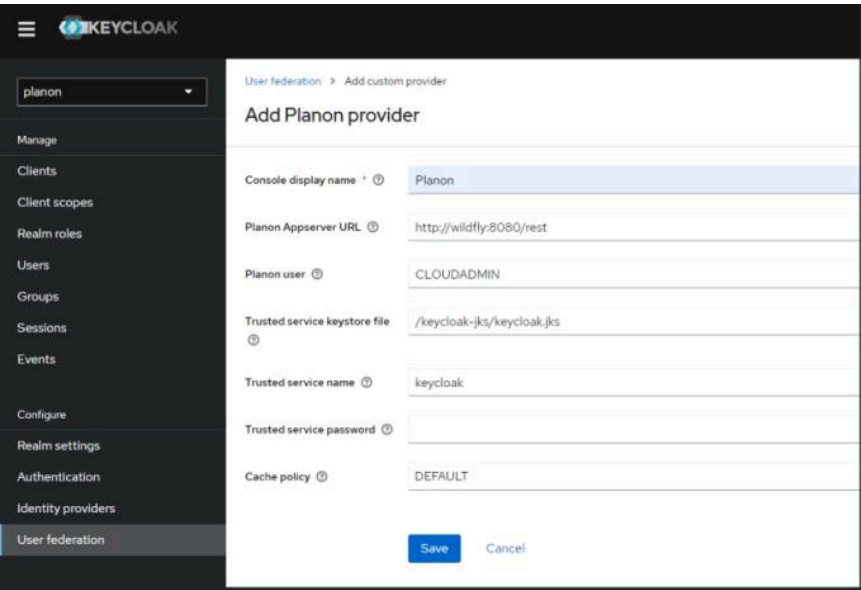
1. Add a **Planon provider** in the section User Federation of KeyCloak.



2. Fill out the settings.

In a Planon Cloud environment only the Console display name needs to be entered, all other fields are pre-configured.

Most of them are initialized correctly for Cloud environments so they need not be changed.



The following table provides a description of the required settings:

Field	Description
Console display name	The name of the user federation in the Keycloak configuration.
Planon AppServer URL	The location of the Planon backend that can handle user validation. For Cloud, the default value is fine.
Planon user	The user that is used to perform queries to the Planon backend. This should be an active user.
Trusted service keystore file	The user federation uses a trust between Keycloak and the Planon backend. This file contains the keys. For Cloud, you can leave this as is.
Trusted service name	The user federation uses a trust between Keycloak and the Planon backend. This is the name of that trusted service.

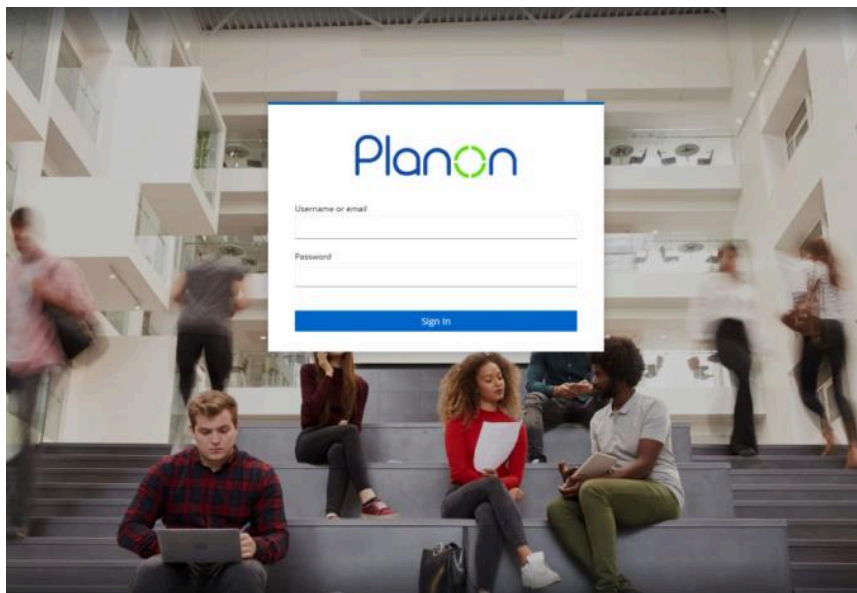
Field	Description
	For Cloud, you can leave this as is.
Trusted service password	The user federation uses a trust between KeyCloak and the Planon backend. This field can override the password.
	For Cloud, you can leave this as is.
Cache policy	Sets the caching policy. Default should be fine.


3. Click **Save** to apply your settings.

A check with the configured backend is carried out to verify that all configured values are correct. When a mismatch is detected, the save will fail and an error with specific error information will be displayed.

If saving is successful, the user federation is activated.

End users can now log in via Keycloak with the user name and password that are stored in the Planon backend.



 Managing users can still be done through the Planon **Accounts** TSI. For more information, see the [Planon Webhelp](#).

Recommendations

- It is not recommended to combine the Planon user federation and Keycloak local password verification. For this reason, we recommend

to disable the **Update password** required action on the authentication section of the Keycloak configuration.

- It is recommended to switch to the OpenID Connect authentication method for applications that support this as is described in [Planon Webhelp](#)

Limitations

- Additional configuration is required to be able to use Planon's [Forgotten password](#) functionality together with Planon user federation:

Configuring forgotten password functionality

After configuring Planon Authentication via Keycloak, Planon's Forgotten password functionality will no longer be accessible via the login screen.

If you want to make the forgotten password functionality available for the end users, please provide a link to reset the forgotten password on a central system of your organization. This link to reset forgotten password is:

Default: `https://<cloud name>.planoncloud.com/forgotpassword/page`

If a custom domain is configured: `https://<custom domain url>/forgotpassword/page`

- Planon user federation is currently only available on Cloud environments.

On-premise

This section describes the information relevant to setting up authentication for on-premise environments.

Single Sign-On

This chapter describes how to configure Planon Web Client and the web server to support Single Sign-On (SSO) for on-premise customers on a Kerberos-enabled domain using:

- [WAFFLE](#) - only for Windows platform
- [Apache SPNEGO implementation](#)
- [Tomcat Keycloak adapter](#) - to be used to configure SSO via a Keycloak Server.



...

Single sign-on is enabled using a Kerberos domain, but it can also be another authentication mechanism. WAFFLE or the Apache SPNEGO implementation takes care of the selection of the accepted mechanism. WAFFLE or the Apache SPNEGO implementation may come with a login prompt because it is not Kerberos but another mechanism.

... This chapter is NOT an installation guide or manual for Kerberos security, WAFFLE authentication, SPNEGO authentication or Keycloak.

[WAFFLE](#)

[Apache SPNEGO implementation](#)

[Configuring browsers](#)

[Troubleshooting SPNEGO and WAFFLE](#)

[Tomcat Keycloak adapter](#)

WAFFLE

WAFFLE (**W**indows **A**uthentication **F**unctional **F**ramework **L**ight **E**dition) is a native Windows authentication framework consisting of two C# and Java libraries that perform functions related to Windows authentication, supporting Negotiate, NTLM and Kerberos.

This solution only works for the Windows platform. The application server, web server and client must all be in the same domain.



For more information, for example about configuration options and enabling extra logging for Waffle, see: <https://github.com/dblock/waffle/blob/master/Docs/tomcat/TomcatSingleSignOnValve.md>



For troubleshooting Waffle, see: <https://github.com/Waffle/waffle/blob/master/Docs/Troubleshooting.md>

[How WAFFLE SSO authentication works](#)

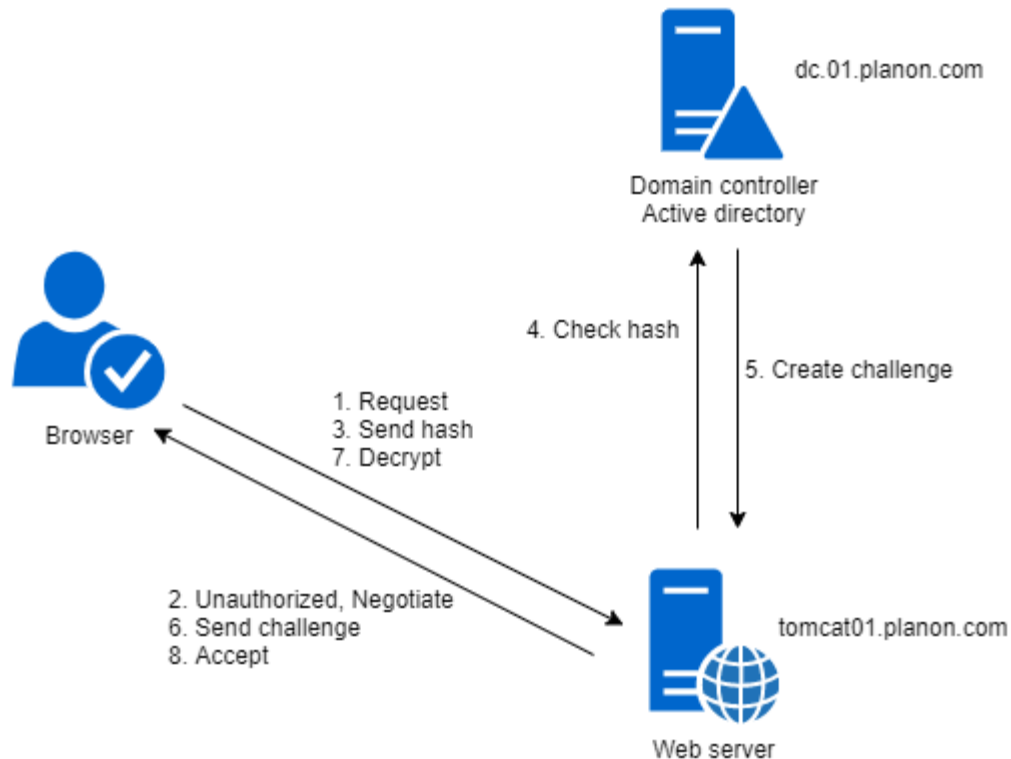
[Configuring the web server](#)

[Use domain user for the web server service](#)

How WAFFLE SSO authentication works

This image depicts the working of WAFFLE SSO authentication.

i **i** A precondition for WAFFLE SSO is that the user request comes from a device that is logged on to the domain.



1. The browser sends a request to the web server.
2. The web server replies with *unauthorized* and proposes negotiations.
3. The client browser gets the user's credentials that were used to log into Windows, takes its hash and sends it to the server.
4. When receiving the hash, the server looks up the user store and identifies the user.

i **i** There is no keytab file needed as is the case for SPNEGO.

5. An unique and encrypted challenge is created.
6. The server sends the challenge to the browser. That challenge can be only decrypted using the user's password.
7. The browser decrypts the challenge with the user's credentials and sends the response back to the server.
8. The server checks whether the response for the challenge is correct and serves the user request if the answer is correct. If the answer is wrong, the server denies the access to the requested resources and sends the unauthorized message.

Configuring the web server

Adapt the following files located in: ..\Server\tomcat-.*\conf\Catalina\localhost

1. Open Root.xml and remove the Realm (PlanonRealmLogin) and FormAuthenticator valve (PnMessageFormAuthenticator). Add the following realm and valve below the AccessKeyValve:

```
<Valve className="waffle.apache.NegotiateAuthenticator" />
```

```
<Realm className="org.apache.catalina.realm.CombinedRealm">
```

```
<Realm className="waffle.apache.WindowsRealm" />
```

```
<Realm appName="PlanonRealmLogin" className="nl.planon.tomcat.PnMessageJaasRealm"
```

```
  userClassNames="nl.planon.cerebeus.PnUser"
```

```
  roleClassNames="nl.planon.cerebeus.PnRole"
```

```
  allRolesMode="authOnly"/>
```

```
</Realm>
```

The accounts used to log in to Planon should have the following user name:

- NetBios name

For example: "planon\username" or "planon.com\username"

Use domain user for the web server service

If you want to use the service account for the server service in combination with WAFFLE the HTTP SPN for the web server should be set to the service account. Run the following command to set the SPN:

```
setspn -U -S http/planonserver.domain.ext serviceaccount@domain.ext
```



Note that you need to be a member of the Domain Admins, Enterprise Admins or have been granted the permission to set the SPN. By default the http SPN is not set, so Waffle will use the host SPN (default linked to Local System). After setting the SPN, you cannot use Waffle with Local System anymore until you remove the SPN again.

Apache SPNEGO implementation

Apache SPNEGO refers to implementations or modules within the **Apache HTTP Server** that support **SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism)** for authentication. SPNEGO is a standard that is primarily used to negotiate authentication mechanisms such as **Kerberos** in web environments.

SPNEGO enables secure **single sign-on (SSO)** by negotiating the authentication protocol between a client (such as a browser) and a server.

For more background information, see [Windows Authentication How-To](#).



SPNEGO SSO using Kerberos does not work on localhost. You cannot try out your SSO configuration on the web server. Negotiation will take the unsupported NTLM authentication instead of localhost.

How SPNEGO SSO authentication works

[Generating a key tab](#)

[Amending the Tanuki configuration file](#)

[Configuring web server](#)

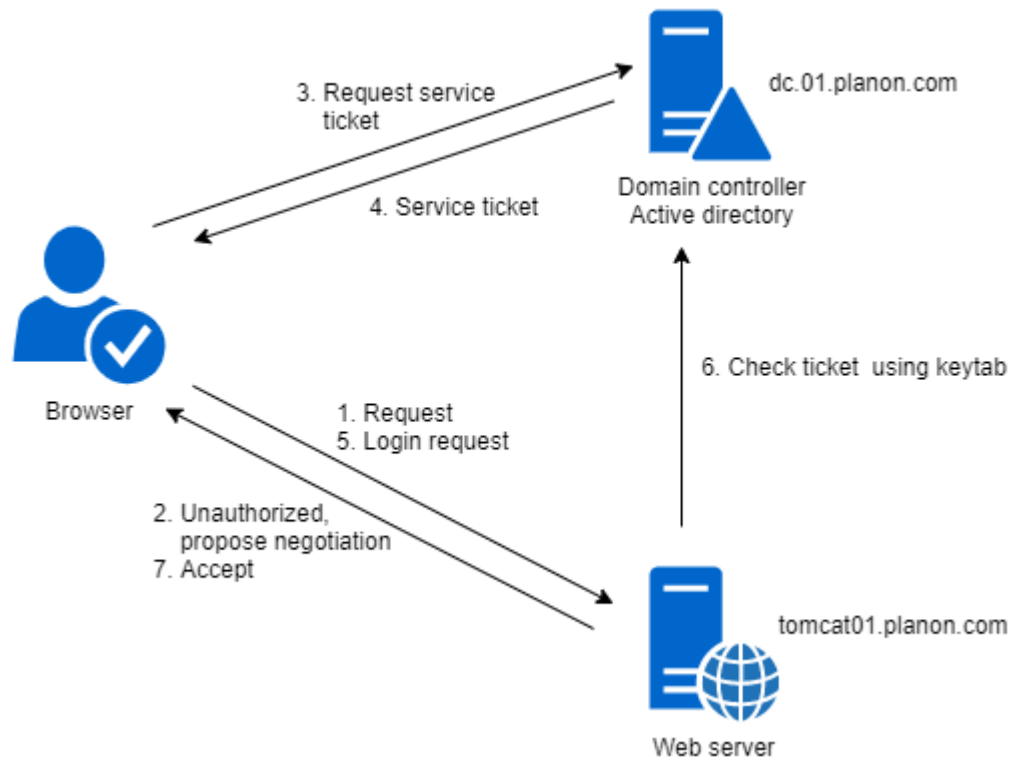
[Planon Web Client configuration](#)

[Verifying the configuration](#)

[Enabling logging](#)

How SPNEGO SSO authentication works

The following image depicts the working of SPNEGO SSO authentication.



1. The browser sends a request to the web server.
2. The web server replies with *unauthorized* and proposes negotiations.
3. The browser decides to go with Kerberos (because configured). The browser takes the client ticket from the local ticket cache, and uses that ticket to request a service ticket for HTTP/tomcat01@PLANON.COM from the domain controller.
4. The domain controller validates the client ticket and returns the service ticket.
5. The browser sends a login request to the web server.
6. The web server verifies the ticket of the client against the keytab.
7. If the ticket is validated, the server accepts the login request.

Generating a key tab

Because the web server is responsible for authenticating the caller (the user), it needs to authenticate itself against the security domain. When such a service needs to authenticate itself, this is typically done by using a keytab.

A keytab is a file with trusted (private credentials) information associated with a domain user who is mapped to the service for which the keytab is valid. The user's name is then mapped to a SPN (Service Principal Name).

For the web server to be able to authenticate a Planon user, a keytab needs to be generated.



Generating a keytab can only be done by a user with administrator's privileges on a domain controller.

A keytab creation is done by executing the ktpass executable.

Given a domain realm PLANON.COM, the following steps have to be performed:

1. Create a Kerberos user for the web server. If you are using PPJC SSO, make sure you create different users for the application server and the web server.



In the example below tomcat01 is used

2. Set the option the user does not need to change the password.
3. Set the option the password never expires.
4. Map the service principal name to the user account.
`setspn -A HTTP/HostName.planon.com tomcat01`

The HostName should be the FQDN of the web server, used in the browser to connect to the web server.

Do not use the CNAME record it must be the host name.

5. On the domain controller, open a command line and create a keytab:
`ktpass /out tomcat.keytab /mapuser tomcat01@PLANON.COM /princ HTTP/HostName.planon.com@PLANON.COM /pass tomcat01pass /kvno 0 -ptype KRB5_NT_PRINCIPAL -crypto all`
 - Type the command in one line or create a .bat and run it.
 - The password (/pass tomcat01pass) will override the password of the user created in Step 1.
 - Make sure the service principal name is unique; otherwise Single Sign On will not work.
 - We recommend to save this command, so you can refer to/reuse it later.
6. Copy the file to the web server host (for example to ...\\Server\\tomcat-*\\).



Because the file contains sensitive security information and is only needed by the web server, it is advisable to restrict access to the file to the user running the web server.

Amending the Tanuki configuration file

To enable Single Sign On, you must add a parameter to the Tanuki configuration file.

1. Stop the Tanuki service.
2. Go to ...\\Server\\tanuki\\webserver\\conf and open the tomcat-wrapper-default.conf file.
3. Locate the # Java Additional Parameters section and add the following parameter (and increment the sequence number):
`wrapper.java.additional.nr=-Djava.security.krb5.conf=AbsolutePath\\Server\\tomcat-*\\conf\\`

Configuring web server

To access Planon Web Client, you need to have the Planon role.

Open the web server's login configuration file in a text editor. This file is named `jaas.config` and is located: `... \Server\tomcat-*\conf`

1. In the `jaas.config`, add the following jaas configurations at the end of the file:

```
com.sun.security.jgss.krb5.accept {
```

```
    com.sun.security.auth.module.Krb5LoginModule required
```

```
    doNotPrompt=true principal="HTTP/HostName.planon.com@PLANON.COM"
```

```
    useKeyTab=true
```

```
    keyTab="AbsolutePath/Server/tomcat-*/tomcat.keytab"
```

```
    isInitiator=false
```

```
    storeKey=true;
```

```
};
```

2. Update the "com.sun.security.jgss.krb5.accept" configuration with your configuration.
3. Update the `server.xml` located in `... \Server\tomcat-*\conf`. Remove the single sign on valve:
`<Valve className="org.apache.catalina.authenticator.SingleSignOn" requireReauthentication="true"/>`

Planon Web Client configuration

1. Open **ROOT.xml** located in: ...\\Server\\tomcat-.*\\conf\\Catalina\\localhost

If you do an update, this file will not be overwritten.

2. Remove the Realm (PlanonRealmLogin) and FormAuthenticator valve (PnMessageFormAuthenticator). Add the following realm and valve below the AccessKeyValve:

```
<Realm className="org.apache.catalina.realm.CombinedRealm" allRolesMode="authOnly">
```

```
<Realm className="nl.planon.tomcat.SPNegoRealm"
```

```
    stripRealmForGss="false"
```

```
    allRolesMode="authOnly"/>
```

```
<Realm className="org.apache.catalina.realm.JAASRealm"
```

```
    appName="PlanonRealmLogin"
```

```
    userClassNames="nl.planon.cerebeus.PnUser"
```

```
    roleClassNames="nl.planon.cerebeus.PnRole"
```

```
    allRolesMode="authOnly"/>
```

```
</Realm>
```

```
<Valve className="org.apache.catalina.authenticator.SpnegoAuthenticator"/>
```



If you do not want to use user names including the domain, you must set `stripRealmForGss="true"`.

3. Restart the Tanuki service.

Verifying the configuration

You can perform the following checks to verify the configuration is correct:

1. Run the setspn tool to see that the requested SPN is not duplicate. If it is duplicate, Single Sign On will not work:

```
setspn -T * -X
```

2. List the keytab and see that the content is as expected. Use java's klist:

```
<install>\Planon*\Server\jdk-*\bin\klist -t -K -e -k <your keytab filename including path>
```

Example outcome with keytab=tomcat.keytab:

Key tab: tomcat.keytab, 1 entry found.

[1] Service principal: HTTP/HostName.planon.com@PLANON.COM

KVNO: 0

Key type: 23

Key: 0x4e150649cdf4b2b394ccefbcd08d709a Time stamp: Jan 01, 1970 01:00

3. Try to login with the keytab using java's kinit:

```
PathToServer\jdk-*\bin\kinit -t PathToTomcat.keytab HTTP/  
HostName.planon.com@PLANON.COM
```

If successful the return message for this call will be:

"New ticket is stored in cache file:..."

Enabling logging

By default you do not see all the logging that is related to single sign-on.

The following steps describe exactly what should be done to enable it.

1. Stop the Tanuki service.
2. Go to and open `..\Server\tomcat-*\conf\logging.properties`
3. Search for and uncomment:
`org.apache.catalina.authenticator.level = FINE`
4. Set the following properties to FINE:
 - `java.util.logging.ConsoleHandler.level`
 - `1catalina.org.apache.juli.FileHandler.level`
5. Save and close the file.
6. Restart your service. Logging will subsequently be available in the logs directory: `..\Server\tomcat-*\logs`

Configuring browsers

A browser only allows Kerberos authentication if it is a trusted site. Browsers do not always support SSO on the same server as the webserver. They will fall back to normal login with login dialog. If you encounter this try login from a different machine

[Configuring Internet Explorer](#)

[Configuring Chrome](#)

[Configuring Firefox](#)

Configuring Internet Explorer

Open the internet options and check the following configuration:

1. Trusted sites list
 - a. On the Security tab.
 - b. Click on the Trusted Sites icon.
 - c. Click the Sites button.
 - d. Add the site to the list.
2. Security level
 - a. On the Security tab.
 - b. Click the Custom Level button.
 - c. Search in the list for User Authentication > Logon and enable one of the following options:
Automatic logon with current username and password or Automatic logon in intranet zone.
 - d. For Miscellaneous enable: Web sites in less privileged Web content zones can navigate into this zone.
3. Integrated windows authentication
 - a. On the Advanced tab.
 - b. For Security enable: Integrated Windows Authentication is set.



All these settings expect a restart of the Internet Explorer.

Configuring Chrome

Open the internet options and check the following configuration:

1. Trusted sites list
 - a. On the Security tab.
 - b. Click on the Local intranet icon.
 - c. Click the Sites button.
 - d. Click the Advanced button.
 - e. Add the site to the list.

Configuring Firefox

The site must be explicitly added to the configuration:

1. In the URL bar go to about:config
2. Filter on network.negotiate-auth.trusted-uris
3. Add the URL to this list.

Troubleshooting SPNEGO and WAFFLE

Question

“Error 400 – Bad Request” on the client.

If you enable FINE logging on Catalina you will get the following errors:

Error parsing HTTP request header

Request header is too large

Answer

When a user is a member of a large number of active directory groups the Kerberos authentication token for the user increases in size.

The HTTP request that the user sends contains the Kerberos token in the header, and the header size increases as the number of groups goes up.

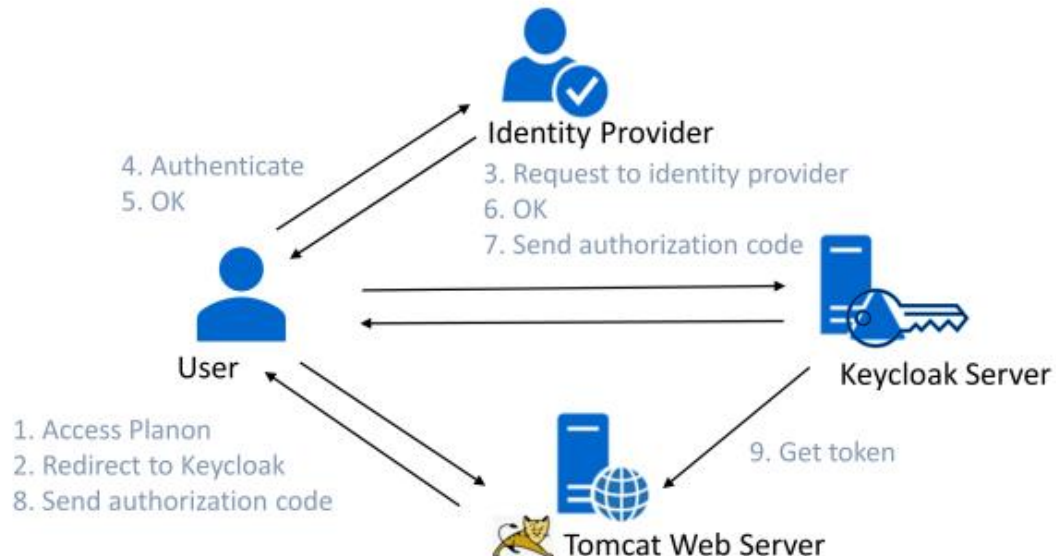
Enlarge the `maxHttpHeaderSize` in the connector you use in the `tomcat-*/conf > server.xml` file.

Tomcat Keycloak adapter

Keycloak is a tool for identity and access management and can be used in front of the Planon application to handle authentication instead of using a standard Planon login.

Authorization is still applied via the Planon application.

The following image depicts the communication with Keycloak.



The Tomcat keycloak adapter and Planon Tomcat Keycloak adapter are installed in the following location: ...\\Server\\tomcat-*.\\lib

i ... We only deliver the Keycloak adapter to make it possible to connect Planon to the Keycloak Server to configure other authentication options as SPNEGO, Waffle or Planon login.

... How to configure the Keycloak Server is not part of this.

... For details about how to install and configure Keycloak Server see:

[Keycloak json](#)
[Configuring the web server](#)

Keycloak json

Generate the Keycloak.json file with the Keycloak server and place it in the following location:

...\Server\tomcat-*\conf\keycloak.json

Extend the content of the Keycloak.json file with the following parameter and value:

"principal-attribute" : "**preferred_username**"

Configuring the web server

To make the Planon ProCenter Web application available for SSO, adapt the following files located in:

...\Server\tomcat-*\conf\Catalina\localhost

1. Open **Root.xml** and remove:

- a. the ForgotPasswordLoginValve valve.
- b. the PnMessageFormAuthenticator valve.

2. Add the following valve above the ExcludingRoleValve valve:

```
<Valve className="nl.planon.tomcat.keycloak.KeycloakAuthenticatorValve"/> <Parameter  
name="keycloak.config.file" value="AbsolutePath/Server/tomcat-*/keycloak.json" />
```

Index

A

- Activating Keycloak 34
- Authentication
 - browser clients 8
 - Cloud configuration 27
 - Configuration 19
 - Environment Management gadget 19
 - Introduction 6
 - mobile apps 9
 - On-premise 50
 - Planon Cloud 19
 - Planon SDK 9
 - system integration 9
- Authentication set-up
 - overview of technical clients 6

B

- Browsers: configure 64

C

- Chrome: configure 66
- Configuration: view 62
- Configuring Keycloak 35
- Connect for Analytics
 - authentication via OIDC 12
 - authentication with OIDC 11
 - Authorization code flow 11
 - Keycloak 12
 - technical info OIDC 12
- Custom domain allowance 39

D

- Domain user 55

F

- Firefox: configure 67
- Forgotten password 50

I

- Identity Broker Solution 28
- Identity provider 8, 28
- Internet Explorer: configure 65

K

- Key tab: generate 58

- Keycloak 8
 - Limitations 50
 - Recommendations 49
- Keycloak json 70

L

- Logging out of Planon Cloud 39
- Logging: enable 63

M

- Mappers 37

N

- NameID 30

O

- OIDC
 - browser clients 8
 - introduction 8
 - mobile apps 9
- OpenID Connect 8, 21
 - OIDC 8
 - Planon database 44

P

- Planon Authentication
 - default client configuration 10
- Planon Connect for AutoCAD
 - Keycloak configurations 14
- Planon mobile app
 - Keycloak 10, 11
 - OIDC authentication method 10
 - technical OIDC info 11
- Planon ProCenter authentication
 - OIDC technical info 16
- Planon SDK
 - OpenID Connect 16
 - PKCE 16
- Privacy sandbox compatibility 24
- ProCenter
 - OIDC 15
 - Planon Self-Service 15
 - Planon Web client 15

R

- Replace certificate 36

S

- SAML 27, 30, 36
- SAML attribute 30
- SAML post 30
- SAML2 28
- SDK
 - Authorization code flow 18
 - Client credentials flow 18
- Secure configuration
 - Considerations 40
- Service Provider metadata 37
- Single Sign-On Webclient and PSS2 51
- SPNEGO 57
- SPNEGO implementation 56
- SPNEGO SSO authentication 57
- SSO 20, 44
 - On-premise customers 51
- SSO authentication 53, 57
- SSO flow 28

T

- Tanuki configuration file: amend 59
- Testing the solution 42
- Tomcat Keycloak adapter
 - SSO 69
- Troubleshooting 44
 - SPNEGO 68
 - WAFFLE 68
- Troubleshooting OIDC 19

U

- UID 42
- User federation
 - Planon 47

V

- Valid Redirect URI 39

W

- Waffle 53
- WAFFLE 52
- WAFFLE SSO authentication 53
- Web Client: configure 61
- Web server
 - Keycloak
 - Configuring 71
- Web server: configure 54, 60