Authentication methods - current and new

Planon Software Suite

Version: L107

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

ℹ	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

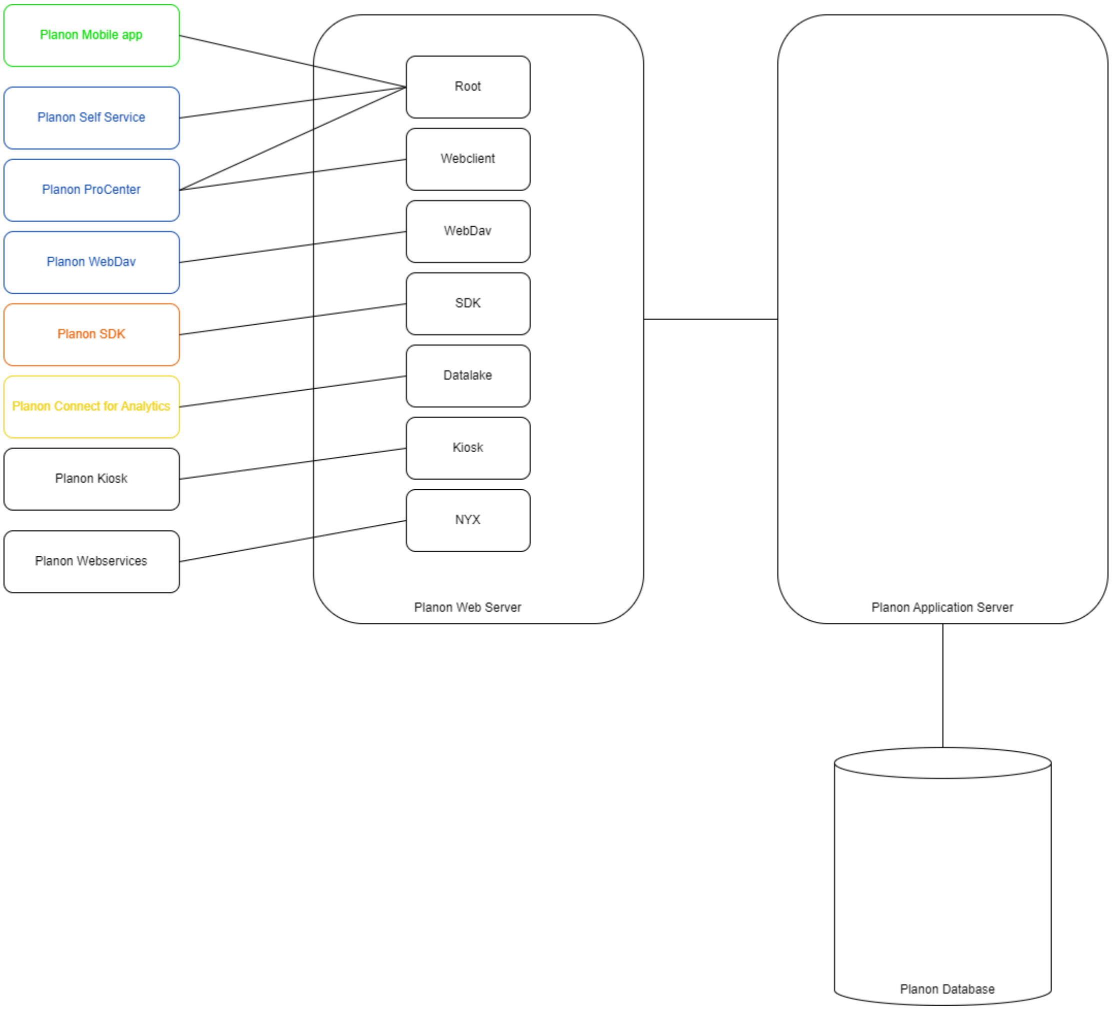
Authentication in Planon

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

Authentication methods - current and new

Introduction

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



Current supported authentication methods

The following table shows the current 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 Mobile apps	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 the Authentication webhelp page.

Migration to new OpenID Connect method

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 the following chapters.

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.

The **Keycloak** service can be connected 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.

Another option is to 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.

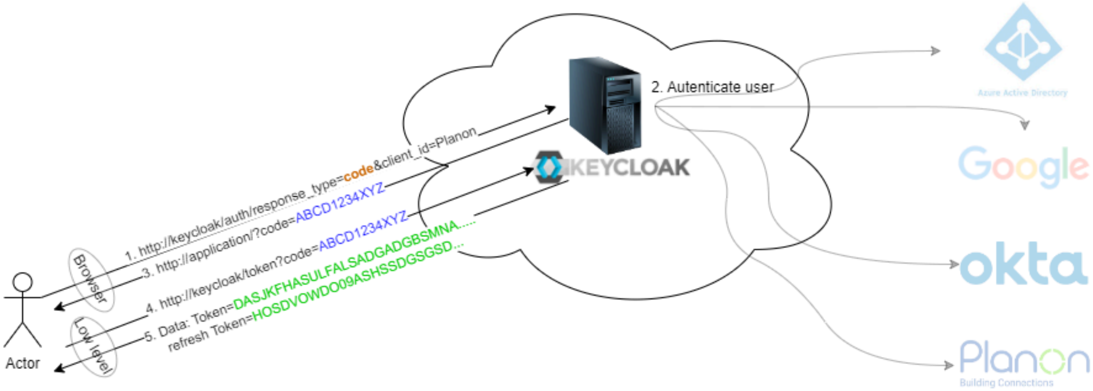
Browser clients

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

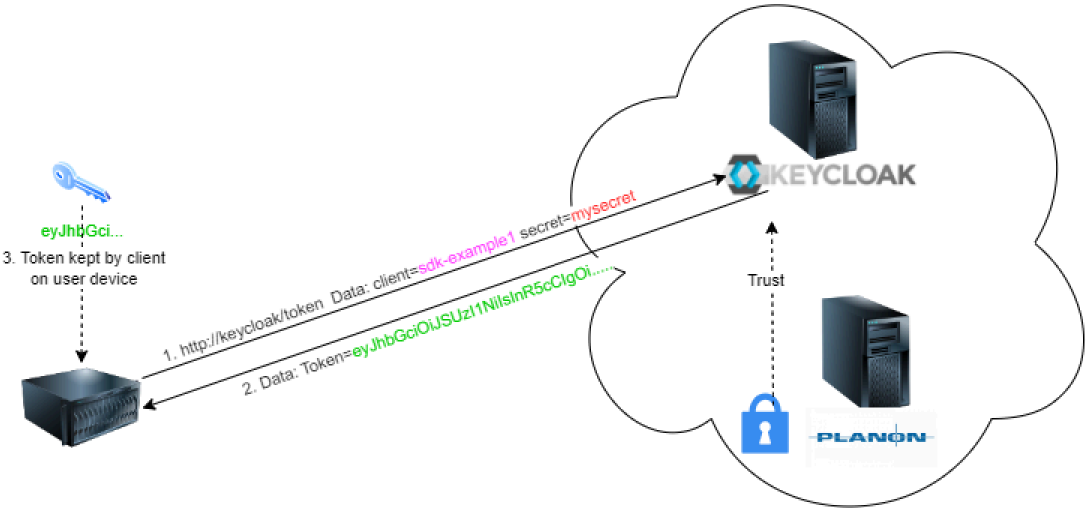
For mobile apps an *offline token* is generated which is a long-lived token. The default lifespan is 30 days. For an offline refresh token the default is 180 days. These authentication flows are called the **Authorization code flow**. By default, Planon uses **PKCE** (Proof key for code exchange) in 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:


2. Autenticate user
KEYCLOAK
1. http://keycloak/auth/response_type=code&client_id=Planon
3. http://application/?code=ABCD1234XYZ
4. http://keycloak/token?code=ABCD1234XYZ
5. Data: Token=DASJKFHASULFALSADGADGBSMNA.....
refresh Token=HOSDVOWDO09ASHSSDGSGSD...
Browser
Low level
Actor
Azure Active Directory
Google
okta
Planon
Building Connections

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**.


eyJhbGci...
3. Token kept by client
on user device
1. http://keycloak/token Data: client=sdk-example1 secret=mysecret
2. Data: Token=eyJhbGciOiJSUzI1NiIsInR5cCIgOi.....
KEYCLOAK
Trust
PLANON

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.

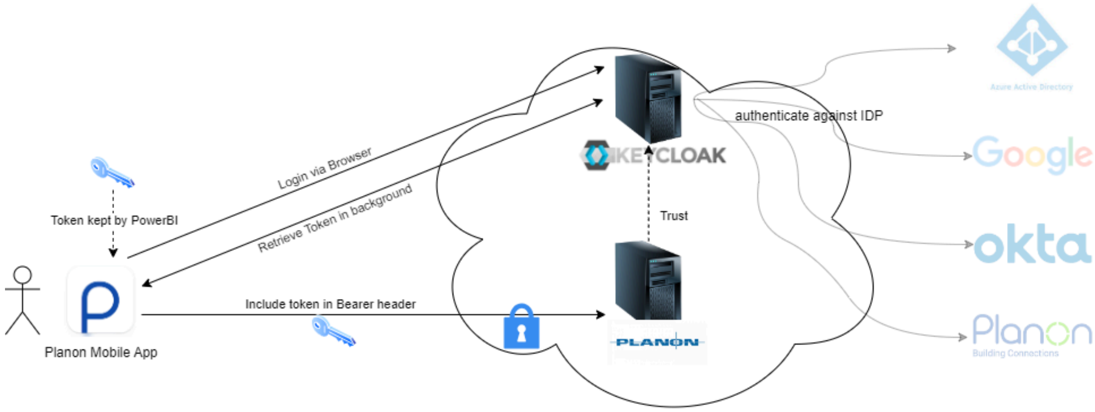
> ℹ Default Keycloak configuration is present in your environment. Make sure the **Offline session times** are always longer than one hour! Shorter session times might result in unexpected behavior.

Client type	openid-connect
Client ID	planon-mobile-app
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 in 15 minutes
Client Token Idle	Inherits from realm setting
Client Token Max	Inherits from realm setting
Client Offline Token Idle	Expires in 30 days

Client Offline Token Max	Expires in 180 days
Proof Key for Code Exchange Code Challenge Method	S256
Authentication, Required action: Welcome the user*	Enabled

* Requires installation of Plugin to Keycloak for on-premise installations.

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.

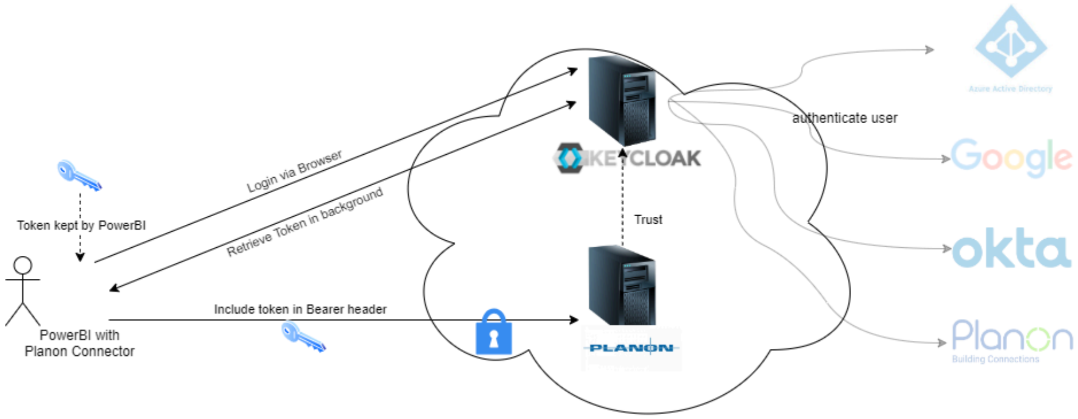
To use Connect for Analytics with OIDC you must configure your cloud environment via the **Environment management gadget** on the **SSO** tab and enable **OpenID Connect** for the Planon Connect for Analytics solution. Additional **Keycloak** configuration is needed. You must add a public client with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

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

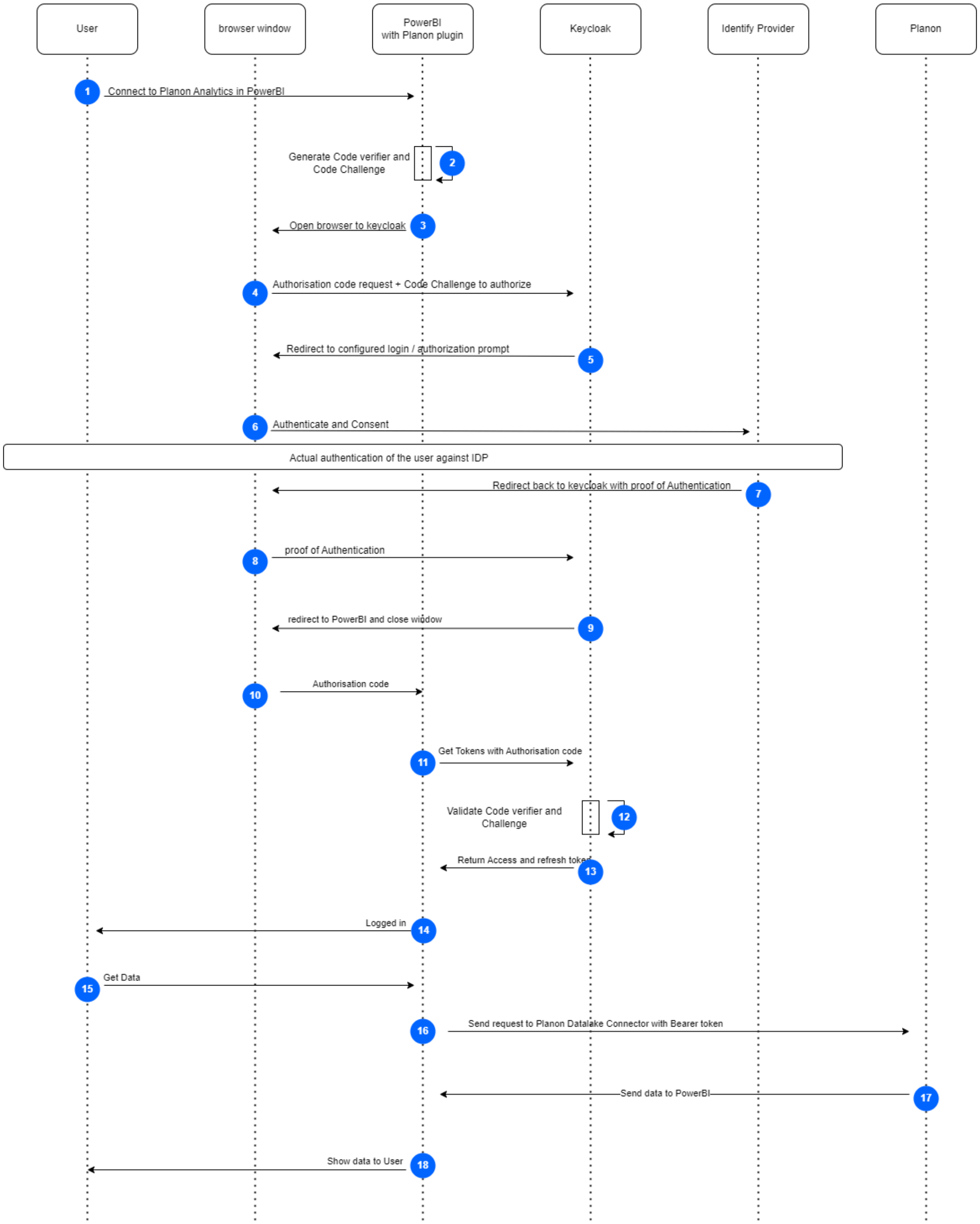
Proof Key for Code Exchange Code Challenge Method	plain

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:


User
browser window
PowerBI with Planon plugin
Keycloak
Identify Provider
Planon
1
Connect to Planon Analytics in PowerBI
Generate Code verifier and Code Challenge
2
3
Open browser to keycloak
4
Authorisation code request + Code Challenge to authorize
Redirect to configured login / authorization prompt
5
6
Authenticate and Consent
Actual authentication of the user against IDP
Redirect back to keycloak with proof of Authentication
7
8
proof of Authentication
redirect to PowerBI and close window
9
10
Authorisation code
11
Get Tokens with Authorisation code
Validate Code verifier and Challenge
12
Return Access and refresh token
13
Logged in
14
15
Get Data
16
Send request to Planon Datalake Connector with Bearer token
Send data to PowerBI
17
Show data to User
18

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.

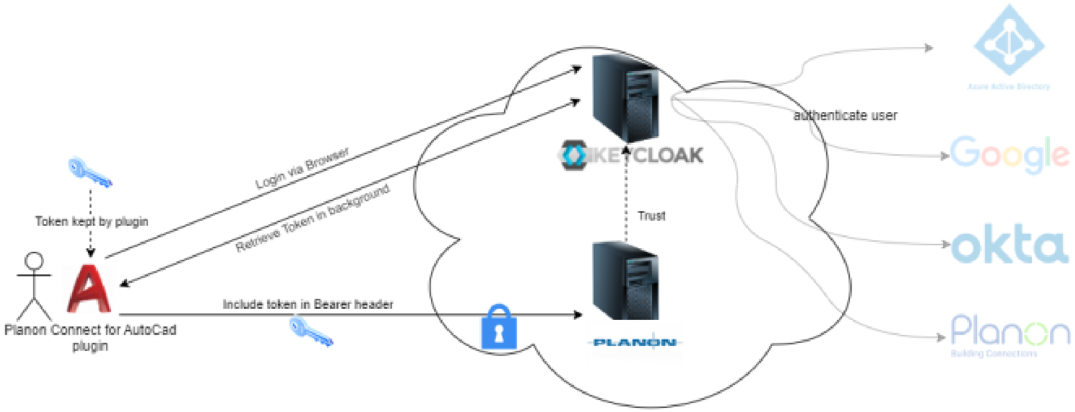
Default Keycloak configuration is present in your environment. Make sure the **Offline session times** are always longer than one hour! Shorter session times might result in unexpected behavior.

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 in 15 minutes
Client Token Idle	Inherits from realm setting
Client Token Max	Inherits from realm setting
Client Offline Token Idle	Expires in 30 days
Client Offline Token Max	Expires in 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

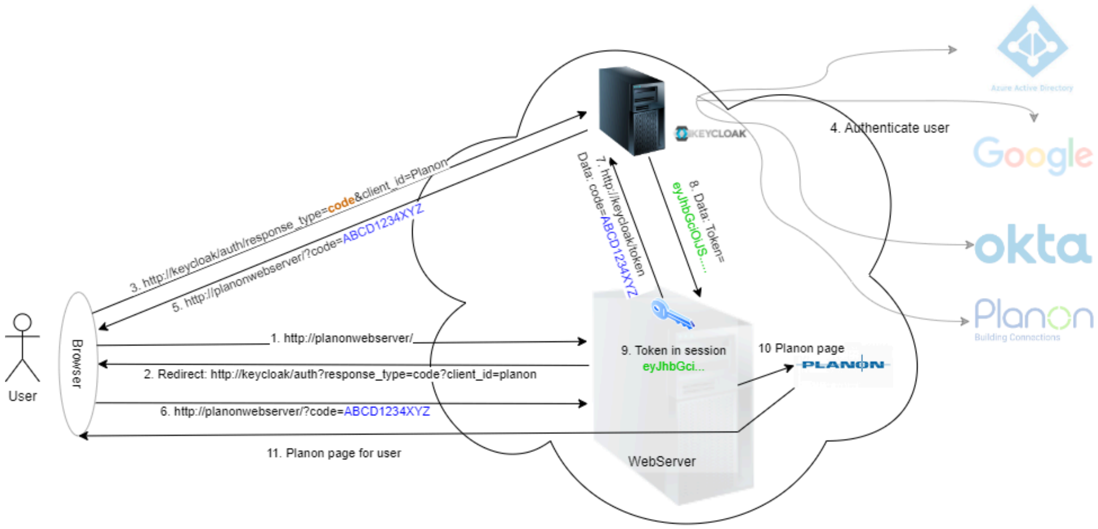
Planon ProCenter consists of a Web application and Planon Self-Service. Both solutions are configured via the same authentication method.

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.

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:

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:

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)

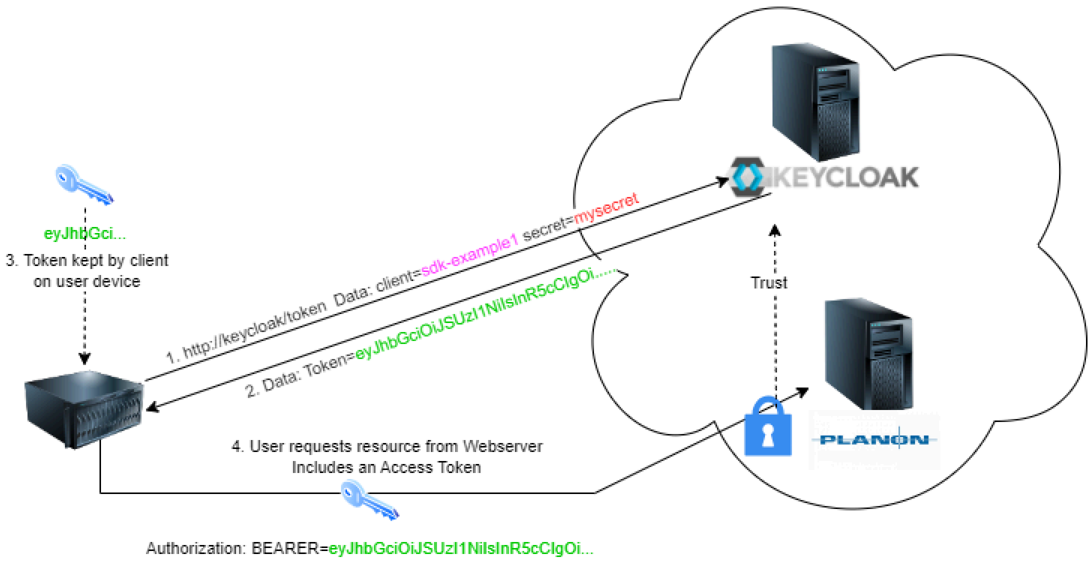
When client credentials flow is used, a *service account user* must be present in Planon. Example

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.

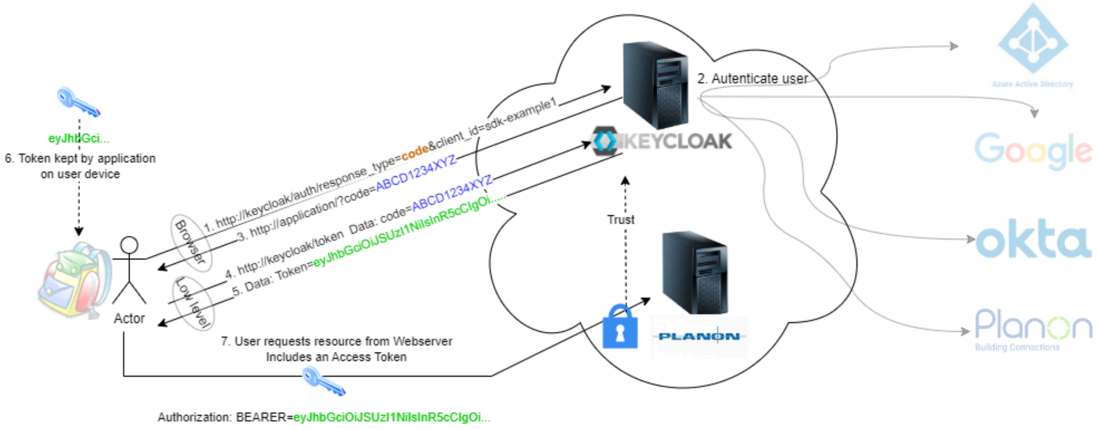
1. To get access to the SDK service via OpenID Connect, take the following steps: Retrieve an access token at the **Authentication service** via the client created in the installation steps.
2. Send the access token as a **Bearer token** to the Planon SDK service.

Technical information - SDK

Client credentials flow:


eyJhbGci...
3. Token kept by client on user device
1. http://keycloak/token Data: client=sdk-example1 secret=mysecret
2. Data: Token=eyJhbGciOiJSUzI1NiIsInR5cCIgOi.....
KEYCLOAK
Trust
PLANON
4. User requests resource from Webserver Includes an Access Token
Authorization: BEARER=eyJhbGciOiJSUzI1NiIsInR5cCIgOi...

Authorization code flow:


eyJhbGci...
6. Token kept by application on user device
Actor
Browser
Low level
1. http://keycloak/auth/response_type=code&client_id=sdk-example1
3. http://application/?code=ABCD1234XYZ
4. http://keycloak/token Data: code=ABCD1234XYZ
5. Data: Token=eyJhbGciOiJSUzI1NiIsInR5cCIgOi.....
2. Autenticate user
KEYCLOAK
Trust
PLANON
Azure Active Directory
Google
okta
Planon
Building Connections
7. User requests resource from Webserver Includes an Access Token
Authorization: BEARER=eyJhbGciOiJSUzI1NiIsInR5cCIgOi...

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.

Index