Kanta authorization guide for personal clients

Personal Health Record

29.1.2025

Kela

Kela
Kanta-palvelut
PL 450, 00056 Kela

FPA
Kanta tjänsterma
PB 450, 00056 FPA

Kanta
www.kanta.fi
kanta@kanta.fi

Change history

Version	Change	Author	Date
2.1	Clarified sections of the guide	Kela Kanta services	19.12.2024
2.0	The old guide is divided into two separate parts. This guide focuses exclusively on the authorization process for personal clients.	Kela Kanta services	17.5.2024

Contents

1 Introduction

This guide provides detailed descriptions of authorization profiles and flows for the My Kanta Personal Health Record (Finnish Kanta PHR) and Kanta Citizen access. It is designed for developers and implementers of applications that interface with Kanta services.

The authorization protocols vary based on the client type. This guide focuses exclusively on the authorization process for personal clients.

All Kanta-related material for application implementers is published on the Kanta pages.The endpoint addresses in this document are exemplary; actual addresses will be published by Kela. If you have any comments on this document, please provide feedback via kanta@kanta.fi.

1.1 Confidential client for personal use

Web applications that have server side business logic and that are capable of protecting application-specific client secrets. **Such applications can also be uniquely identified through client authentication with a client certificate, using mutual TLS. Examples of confidential clients are web-based customer portals.**

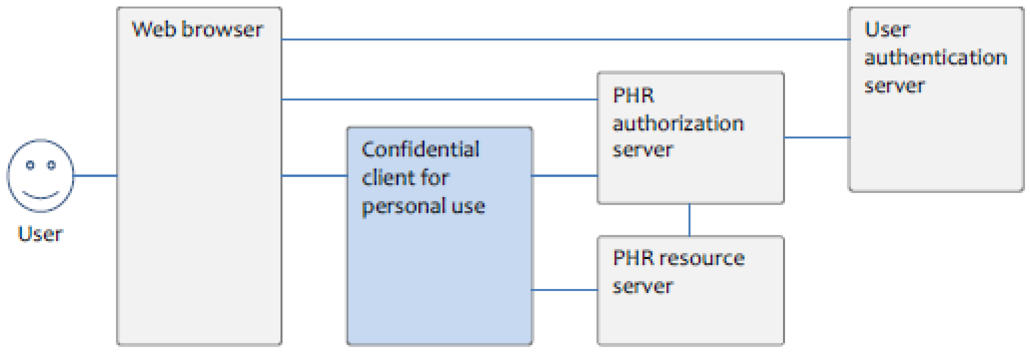
Web browser
User authentication server
PHR authorization server
Confidential client for personal use
User
PHR resource server

Figure 1: Data and control flows for confidential clients for personal use. Note that the vertical or horizontal order of the connectors is not necessarily the same as the order of phases in the authorization process.

In the authorization, the web browser uses confidential client for personal use, user authentication server, and Kanta authorization server. The client uses Kanta authorization server and resource server.

1.2 Personal client tokens

When the application requests a token and authorization, it is the application's responsibility to know which user the authorization request concerns, as the response will only include a pseudonym, not direct user identification.

The refresh token is user-specific and must be stored securely. For server-based applications, the refresh token expires after one year if the integration is not used. In this case, users must renew the application’s permissions.

The refresh token is always valid for one year after it has been used and applications cannot extend the validity of the refresh token without the user’s permission. This does not mean that the application cannot use the token for background queries. However, it is the application's responsibility to act with the user's permission.

2 Authorization code flow

The authorization code flow is used by confidential clients for personal use. The client retrieves a short-lived authorization code from the authorization endpoint of the authorization server, in order to exchange it later for a set of tokens at the token endpoint. The authorization endpoint is called when the client needs authorization from the user to access resources. This may occur the first time the client is used or if the client has not been granted a necessary scope to access a resource. The authorization and token endpoints are described in [RFC6749]. Note that the authorization and token URLs may differ from each other.

Auhtorization flow is also described in use case diagram. Read more about the use case diagram (Kanta.fi, in Finnish).

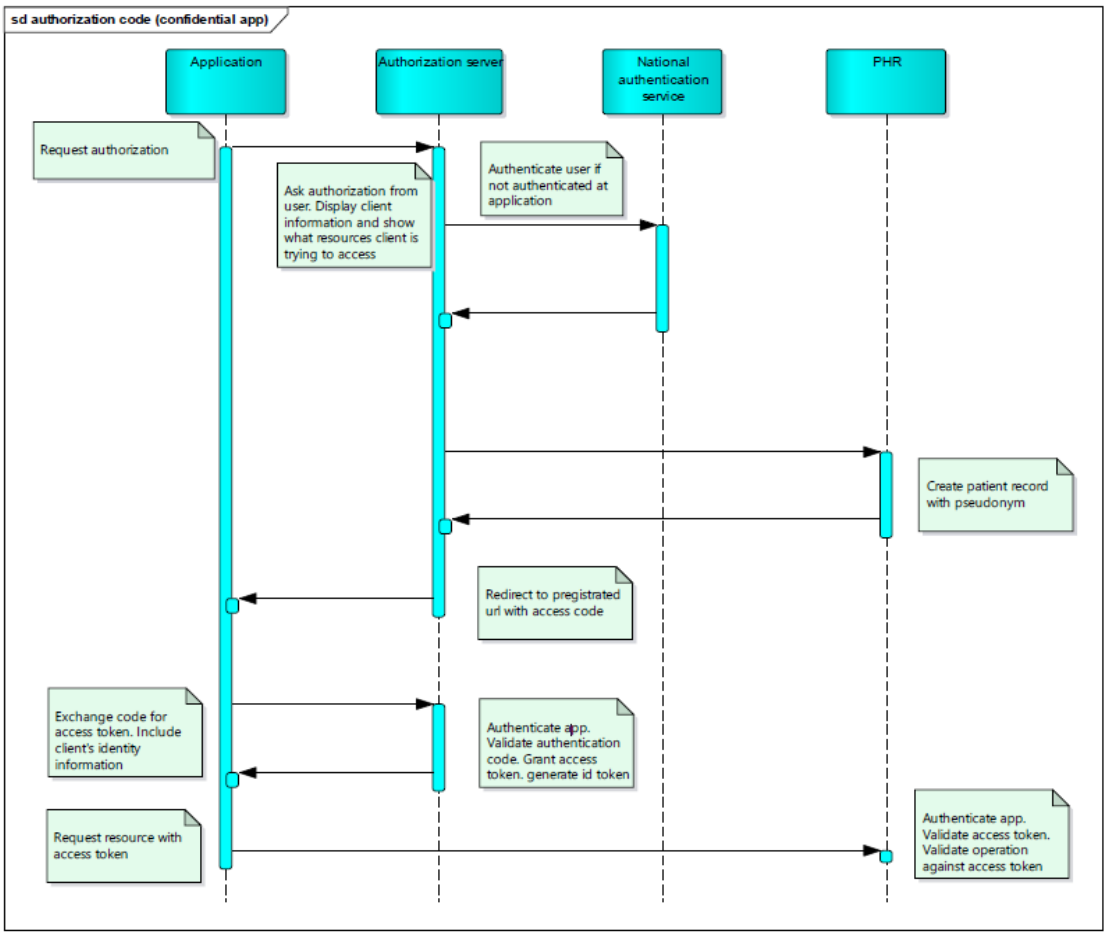
sd authorization code (confidential app)
Application
Authorization server
National authentication service
PHR
Request authorization
Ask authorization from user. Display client information and show what resources client is trying to access
Authenticate user if not authenticated at application
Create patient record with pseudonym
Redirect to pregistrated url with access code
Exchange code for access token. Include client's identity information
Authenticate app. Validate authentication code. Grant access token. generate id token
Request resource with access token
Authenticate app. Validate access token. Validate operation against access token

Figure 2: Authorization code flow.

The authorization endpoint is called when the client needs authorization from the user to access resources. This may be the first time the client is used or if the client has not been granted a scope that it needs to access a resource. First the client requests and receives a short-lived authorization code which will be then traded for a set of tokens at the token endpoint. Before issuing the code, the authorization server authenticates the user and requests the user to confirm the access to their Kanta account.

2.1 Authorization endpoint

2.1.1 Authorization request parameters

The user-agent MUST be requested to open the authorization endpoint's URL with the following parameters:

Name	Cardinality	Value
response_type	required	“code”
client_id	required	The identifier of the client instance. For confidential clients, the id is provided in the client registration process, and for public clients the id is generated through dynamic registration of instances at the registration endpoint.
redirect_uri	required	Must match the URI registered for the client software at the registration time. https://localhost is not allowed but for example https://127.0.0.1 is. The redirect_uri must be an absolute URI. The address must match the one registered for the application exactly.
scope	optional	The scopes that the client requests to be granted, separated using the +, %2B, or %20 characters. The client is not required to request all the scopes registered to it, the request may include only a subset of the registered scopes. If the value is skipped or empty, the authorization server will assume that the client is requesting all scopes registered for it.
state	required	The client MUST generate an unpredictable state parameter with

Name	Cardinality	Value
		at least 128 bits of entropy for each user session. The authorization server will include the state value when redirecting the user-agent to the redirect URI. The client MUST validate the state value for any request sent to its redirect URI.
lg	optional	The language parameter provided by the application redirects the user to identify and authorize. Accepted and supported languages are: • For the Finnish page, use 'fi', 'fi-FI', 'fi-SE', or 'fi-xx' • For the Swedish page, use 'sv', 'sv-SE', 'sv-FI', or 'sv-xx'. • For the English page, use 'en' or 'en-xx'. If the parameter is unknown, the user will be redirected to the English authorization page. If the parameter is missing from the request, the user will be redirected to the Finnish authorization page.
code_challenge	required	Random string generated according to RFC7636 (https://www.rfc-editor.org/rfc/rfc7636#section-4.2)

Name	Cardinality	Value
code_challenge_method	required	“S256”

All parameters MUST be "application/x-www-form-urlencoded" formatted as defined in the Appendix B of [RFC6749].

2.1.1.1 Example of authorization request:

https://phrauth.kanta.fi/authorize?response_type=code&client_id=8d415 da7-bec9-44a3-8979-105ea5bf8ee4&redirect_uri=fi.sw-vendor.app%3A%2Fafter-auth& scope=patient%2FObservation.read+patient%2FObservation.write%20patient%2FMedicationAdministration.read&state=adfh56kiwshti2k4

2.1.2 Actions after the equest

After verifying the parameters of the call, the authorization server will redirect the user-agent to Suomi.fi e-Identification, the Finnish national citizen authentication service. If the application uses the same authentication service that will show the citizen notification about authenticating to Kanta. After successful authentication of the user, the authorization server shows the scopes that are requested by the client. Authorization server also shows which user is identified. If the identified user isn’t the one using the service, it’s possible to switch the user and authorization server redirects the user-agent to Suomi.fi e-Identification. The user is asked to confirm the scopes.

If an error occurs during the authorization process, such as an invalid client identifier or a denied access request, the authorization server must not redirect the user-agent to an invalid redirect URI. Instead, the user-agent is redirected to the Kanta.fi error page. If the authorization process is denied because the user rejects the authorization, the user-agent is redirected back to the application. (see RFC6749 4.1.2.1. for details).

The authorization server will assign each user a pseudonym to be used with the service. Pseudonyms are random UUID identifiers that are directly associated with the Finnish national identification numbers of the same persons. Clients will never receive original national identification numbers from the service. User’s pseudonym remains the same, for example when the application is authorized again.

After generating any required pseudonyms, the authorization server will redirect the user-agent to the redirect URI (client's redirection endpoint) provided in the authorization request.

2.1.3 Authorization reponse parameters

The table below describes the parameters included in the response sent by the authorization server after a successful authorization request.

Name	Cardinality	Value
code	required	The short-lived authorization code generated by the authorization server.
state	required	The value of the state parameter exactly as supplied by the client in the authorization request. The client MUST validate the state value for any request sent to its redirect URL and check whether it matches a submitted authorization request.

2.1.3.1 Example of autorization response

fi.sw-vendor.app:/after-auth?code=ahui560zxs12n3dq&state=adfh56kiwshti2k

The authorization code is valid for 5 (five) minutes.

2.2 Token endpoint

After receiving an authorization code through the redirect call performed by the authorization server in the previous step, the client accesses the token endpoint in order to receive an access token and a refresh token. The client presents the authorization code along with its own credentials to the authorization server's token endpoint to obtain the said set of tokens. When an access token is expired, the client can request a new access token by presenting a valid refresh token.

2.2.1 Access token request parameters

The client trades the code for an access token, a refresh token and an ID token via a POST call to the Kanta authorization server's token endpoint URL.

The following parameters are supplied with the call:

Name	Carddinality	Value
client_id	required	The id of the client
grant_type	required	"authorization_code"
code	required	The short-lived authorization code received from the authorization server.
redirect_uri	required	MUST match the URI used in the authorization request.
code_verifier	required	Random string generated according to RFC7636 (See more about the rfc)

All parameters MUST be "application/x-www-form-urlencoded" formatted as defined in the Appendix B of [RFC6749].

2.2.1.1 Example of access token request

POST https://phrauth-token.kanta.fi/phr-authserver/token
HTTP/1.1

```
Content-Type: application/x-www-form-urlencoded
grant_type=authorization_code
&code=4lKCd5
&redirect_uri=https%3A%2F%2F127.0.0.1
&client_id=4393ab31-7753-472b-af74-dcb8b7b64c93
```

2.2.2 Access token response parameters

The Kanta authorization server will return a JSON structure that includes an access token or a message indicating that the authorization request has been denied.

The JSON structure includes the following parameters:

Name	Cardinality	Value
access_token	required	The access token issued by the authorization server
token_type	required	Fixed value: Bearer
expires_in	required	Lifetime in seconds of the access token, after which the token SHALL NOT be accepted by the resource server
scope	required	Scope of access authorized.
refresh_token	required	Token that can be used to obtain a new access token
sub	required	Application user’s pseudonym
id_token	optional	Authenticated patient pseudonym identity, added if openid scope exist

The access token is used in all calls to the resource server to obtain resources. The token parameter shall be sent as the Bearer http-header to the FHIR resource server as defined in section 2.1 in RFC6750.

2.2.2.1 Example of access token response

```
{
"access_token":"eyJhbGciOiJSU...8h0eQ",
"token_type":"Bearer",
"expires_in":3599,
"scope":"patient/Observation.read+patient/Observation.write+openid",
"sub": "44a12254-b28d-42f9-8bec-4a468473ef9f",
"refresh_token":"eyJhbGciOiJSUzI1NiI...ZGFicmE=",
"id_token":"eyJhGciOiJSUzI1NiI...ESL0eIX7eg1_DA"
}
```

3 Expiration of authorization and tokens

Authorizations and tokens may expire for various reasons:

- If the refresh and access tokens remain unused, they are valid for up to one year from the date of issuance.
- In the event of changes to the notification informing users about wellbeing data stored in Kanta Services, the authorization remains valid for 6 months following the change. The user always acknowledges the information during the authorization process. If the information changes, the user must accept the new information through the OmaKanta interface. Once accepted, the active period is restored to normal.
- The user can expire the authorization in OmaKanta.

Once the refresh token has expired, the application must be re-authorized.

4 Supported scopes in Kanta authorization

Scopes supported by the Kanta authorization can be divided into the scopes that grant access to specific FHIR resources stored on the resource server and to scopes that allow applications to obtain other information and keep the authorization active.

4.1 User scopes for data access

Scopes than can be granted to access resources on the resource server are defined similarly to SMART on FHIR scopes.

To read a resource you need to have the patient/Resource.read –scope. For writing, updating and deleting the resource patient/Resource.write scope is needed. A scope is needed only for the main resource type, contained resources that are inline in the resource to be read or written follow the scope of the resource that they are part of. Referenced resources are subject of the scope of their respective type.

All requested scopes that can be authorized by user to the application, need to be registered for the application at the registration time. Only registered scopes are allowed to request authorization for. All scopes that are included in the access token need to be authorized by the user.

All scopes accepted by the Kanta Authorization server are listed in this Simplifier.net project (Simplifier.net)

4.2 Techical scopes

There are in addition the user scopes that provide access to protected resources on the server some more technical scopes. These are the “offline_access” scope and “openid” scope.

The “offline_access” is defined in OpenId Core specification and enables the client to request new access token after expiration using the refresh token granted at the authorization time. The “openid” is defined in OpenId core specifications and enables the client to identify the user.

5 Security considerations

All transactions MUST be protected in transit by TLS, as described in BCP195 [BCP195].

All clients MUST adhere to the applicable recommendations in the Security Considerations sections of [RFC6749] and the OAuth 2.0 Threat Model and Security Considerations document [RFC6819].

Additionally, all clients MUST follow the relevant recommendations in the OWASP Mobile Security Project's Secure Mobile Development Guidelines [OWASP].

6 References

[RFC6749] Hardt, D. The OAuth 2.0 Authorization Framework, RFC 6749, DOI 10.17487/RFC6749, October 2012.

[RGC6819] T. Lodderstedt, M. McGloin, P. Hunt.OAuth 2.0 Threat Model and Security Considerations, RFC6819. January 2013.

[RFC7517] M. Jones. JSON Web Key (JWK), RFC 7517. May 2015.

[OWASP] OWASP Mobile Security Project, Mobile Application Coding Guidelines https://owasp.org/www-project-mobile-app-security/

[RFC7523] M. Jones, B. Campbell, C. Mortimore. JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants. RFC 7523. May 2015.

[ARGONAUT] Argonaut Project. Cross-Organization Data Access Profile. Working draft of a OAuth 2.0 profile to support the EHR-to-EHR use case. December 2015. https://github.com/smart-on-fhir/smart-on-fhir.github.io/wiki/cross-organizational-auth

[CONNECTCORE] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, C. Mortimore. OpenID Connect Core 1.0 incorporating errata set 1. November 2014. http://openid.net/specs/openid-connect-core-1_0.html

[BCP195] Y. Sheffer, R. Holz, P. Saint-Andre. Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Tarnsport Layer Security (DTLS). May 2015.

[SUOMI.FI] https://www.suomi.fi/page/about-eidentification

Mutual X.509 Transport Layer Security (TLS) Authentication for OAuth Clients
https://tools.ietf.org/html/draft-campbell-oauth-tls-client-auth-00