UCI Compile

Explanatory Notes

The figures below show the flowchart of the use case implementation *Compile*, from four different perspectives:

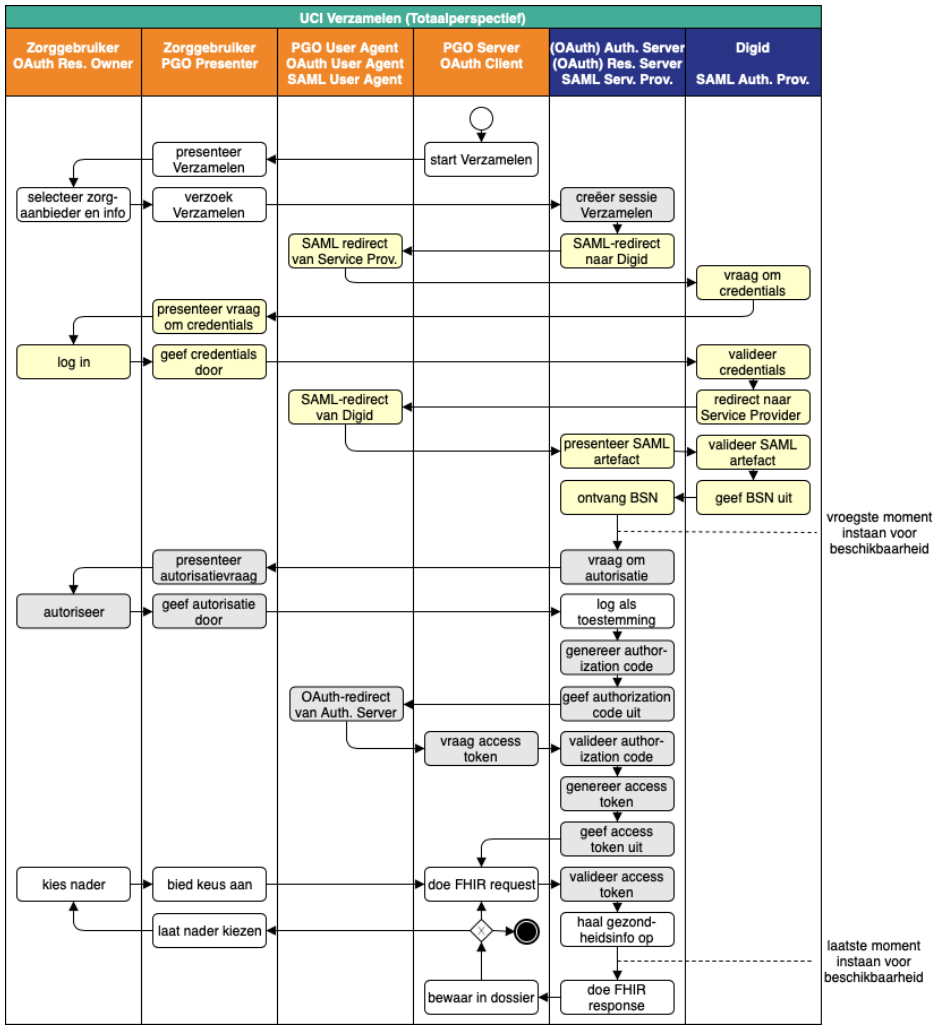
- the overall perspective, with both the happy flow and the exceptions;
- the perspective of the *PHE Server* (= *OAuth Client*), which comes under the *Individual's Service Provider*. In so far as the last-named are a participant in the MedMij Agreements System, they can thus read this figure as his mandatory participation in the use case implementation *Compile;*
- the perspective of the (*OAuth) Authorisation Server/*(*OAuth) Resource Server/ SAML Service Provider*, that comes under the *Care Provider's Service Provider*. In so far as the last-named are a participant in the MedMij Agreements System, they can thus read this figure as his mandatory participation in the use case implementation *Compile;*
- the perspective of the *Care User* (= *OAuth Resource Owner*).

The flowcharts only show the situation in which all actions are successful up to and including the ultimate compiling of the health information (this situation is known as the 'happy flow'). In line with the MedMij corporate identity, the three orange paths belong to the Individual's Domain and the two blue ones to the Care Provider's Domain. Many actions in the flowcharts are shown in colour. Together, the actions coloured in light grey form the authorisation flow in accordance with OAuth 2, whereas the actions coloured in light yellow together form the authentication flow in accordance with DigiD/SAML. In other words, these colours only refer to the standards used and say nothing about which component has to execute the step. Authentication is thus embedded in authorisation. In the flowcharts for the specific perspectives, it is only the actions in the path for that perspective that are named. The actions in the other paths are compressed and shown without names.

Responsibilities regarding the data handled in this use case implementation are, together with those of UCI Share, included on a separate page.

Overall perspective

Happy flow



The MedMij Agreements System recommends that the availability condition be made effective from the earliest stated moment. In release 1.1.1, the MedMij Agreements System allows that condition to be effective at a later moment but not later than the last time indicated in the figure.

Explanatory Notes

In each completion of the flow described in the diagram, there is in all cases only a single one of each of the roles named above.

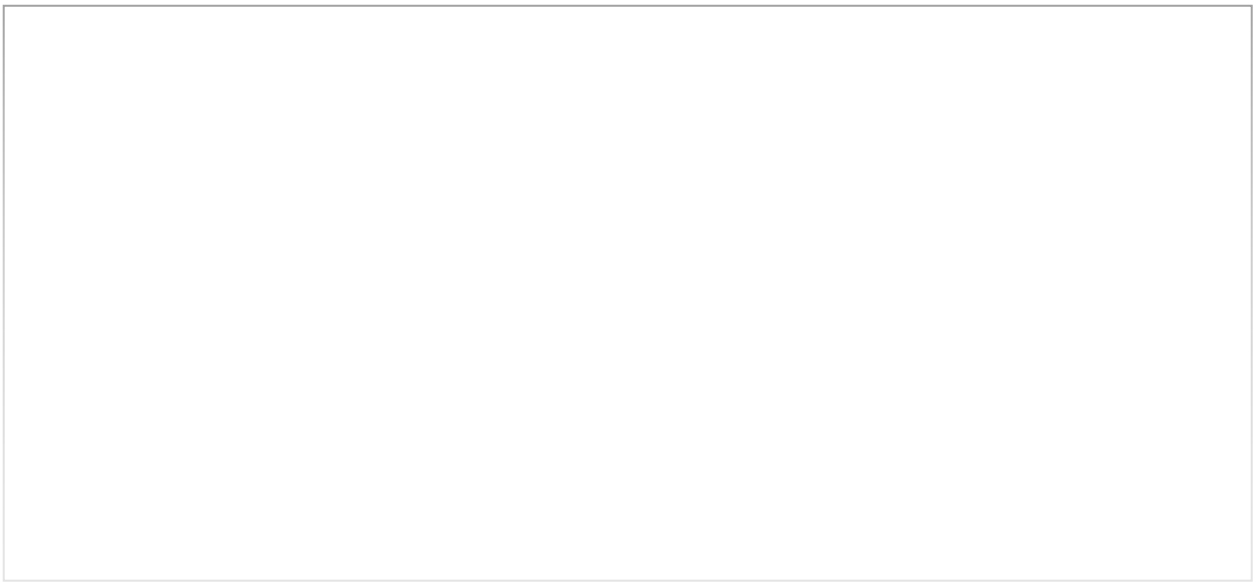
The flow has the following steps:

1. The *PHE Server* starts the flow by presenting the option in the *PHE Presenter* of the *Care User* to compile a particular *Data Service* from a certain *Care Provider*. This always relates to precisely one *Data Service* (a single scope, in OAuth terms). From the *Care Providers List*, the *PHE Server* knows which *Data Services* are provided by a *Care Provider*. If desired, the *Data Service Names* from the *Data Service Names List* are used.

2. The *Care User* makes explicit his selection and gets the *OAuth User Agent* to send a compile request to the *Authorisation Server*. The address of the authorisation endpoint is taken from the *CPL*. The redirect URI indicates where the *Authorisation Server* must hereafter redirect the *OAuth User Agent* to (with the authorisation code).
3. The *Authorisation Server* now begins the OAuth flow (in his role as *OAuth Authorisation Server*) by creating a session.
4. The *Authorisation Server* (now in the role of *SAML Service Provider*) starts the SAML flow by redirecting the browser to *DigiD*, without providing a redirect URI, which indicates to where *DigiD* must later return the *OAuth User Agent*, after the logging in of the *Care User*.
5. DigiD asks the *Care User* - via his *PHE Presenter* - for his login details.
6. When these are correct then *DigiD* redirects the *OAuth User Agent* back to the *Authorisation Server*, whilst giving him a retrieval certificate: the SAML artefact.
7. With this retrieval certificate, the *Authorisation Server* retrieves the BSN (citizen service number) directly from *DigiD*.
8. The earliest moment then comes when the *Authorisation Server* guarantees that the *Care Provider* - for the relevant *Data Service* - has any health information of this *Individual* available at all; otherwise, the happy flow terminates. Part of this is that the *Individual* must be at least 16 years old for this. See a separate page for detailed explanatory notes.
9. If this is successful then the *Authorisation Server* presents - via the *PHE Presenter* to *Care User* the question of whether the last-named permits him to send the requested personal health information to the *PHE Server* (as *OAuth Client*). Under the flowchart it is specified which information, and from where, is processed by the *OAuth Authorisation Server* in the authorisation request to be submitted to the *Care User*.
10. Upon agreement, the *Authorisation Server* logs this as consent, generates an authorisation code and sends this as a retrieval certificate, by means of a browser redirect, along with the redirect URI received in step 1, to the *PHE Server*. The *Authorisation Server* sends with it the local state information that it received in the initial step of the *PHE Server*. The last-named recognises in it the request that it must associate the authorisation code with.
11. The *PHE Server* not only interprets this authorisation code as a retrieval certificate, but also deduces from it that the consent has been given and logs the obtaining of the retrieval certificate.
12. The *PHE Server* applies again to the *Authorisation Server* with this retrieval certificate but now without the intermediation of the *OAuth User Agent* for an access token.
13. The *Authorisation Server* now generates an access token and sends it to the *PHE Server*.
14. The *PHE Server* is now ready to send the request for the health information to the *Resource Server*. It obtains the address of the resource endpoint from the *CPL*. It places the access token in the message and ensures that no BSN is included in the message.
15. The *Resource Server* checks whether the received token grants entitlement to the requested resources and retrieves them from underlying sources or other sources. The final moment then comes at which the Resource Server must guarantee that the Care Provider has the health information available for the relevant Data Service. If this

information is available then the *Resource Server* sends it in a FHIR response to the *PHE Server*. If it isn't then the *Resource Server* terminates the happy flow

16. and will also retain the received health information in the personal file. If the *Data Service* that the *Care User* has authorised consists of multiple *Transactions* then the *PHE Server* may subsequently ask the *Resource Server* for the *Transactions* still remaining, possibly after new user interaction. This can be done as long as the access token is valid.



When implementing the condition of availability at the *Care Provider* for the health information to be compiled, it makes sense to take privacy requirements into consideration. If the *Care Provider's Service Provider* were to create new data compilations for the availability condition then a processing always takes place under the responsibility of a single *Care Provider*. The combining of processing or inadequate segregation must be avoided. This can only be deviated from under the explicit instruction of the *Care Provider*(s) and require a careful weighing-up beforehand, due to the associated privacy risks.

Exceptions

Explanatory Notes

The table below describes the situations where there are exceptions. They may be viewed as the counterparts of the exceptions of the use case Compile. All exceptions are discovered by the *Authorisation Server* or the *Resource Server*. In this version of the MedMij Agreements System, it has been determined that they always lead to the fastest possible termination of the flow by all roles involved. However, the other roles must first be involved about this. In order to prevent the *PHE Server* from obtaining information about the existence of handling relationships without consent having being given (already or otherwise) for this then the distinction between the exceptions 2, 3 and 4 must not be made by the *PHE Server*.

This table only contains the exceptional situations in respect of which the MedMij Agreements System lays down its own requirements for the implementation. The specification of OAuth 2.0 also contains more generic exceptional situations, such as the situation in which the redirect URI turns out to be invalid. These exceptional situations must be implemented too.

Number	Implements the exception	Exception	Action
UCI Compile 1	UC Compile 1	*Authorisation Server* finds the received request to be invalid.	*Authorisation Server* informs *PHE Serv* exception. *PHE Server* informs *Care Us*
UCI Compile 2	UC Compile 2	*Authorisation Server* cannot establish the identity of the *Care User*.	*Authorisation Server* informs *PHE Serv* exception. *PHE Server* informs *Care Us* progress can be made on his request b cause of this completely aside.
UCI Compile 3	UC Compile 3	*Authorisation Server* establishes at any moment that no health information of the *Individual* is available at the *Care Provider* for this *Data Service*. See separate page for detailed explanatory notes.	
UCI Compile 4	UC Compile 4	The authorisation request is denied.	
UCI Compile 5	UC Compile 5	*Authorisation Server* cannot establish the authorisation.	*Authorisation Server* informs *PHE Serv* exception. *PHE Server* then informs *Ca* this.
UCI Compile 6	UC Compile 6	The validation of the authorisation code by *Authorisation Server* fails.	*Authorisation Server* informs *PHE Serv* exception. *PHE Server* then informs *Ca* this.
UCI Compile 7	UC Compile 6	The validation of the access token by *Resource Server* fails.	*Resource Server* informs *PHE Server* ab exception. *PHE Server* then informs *Ca* this.

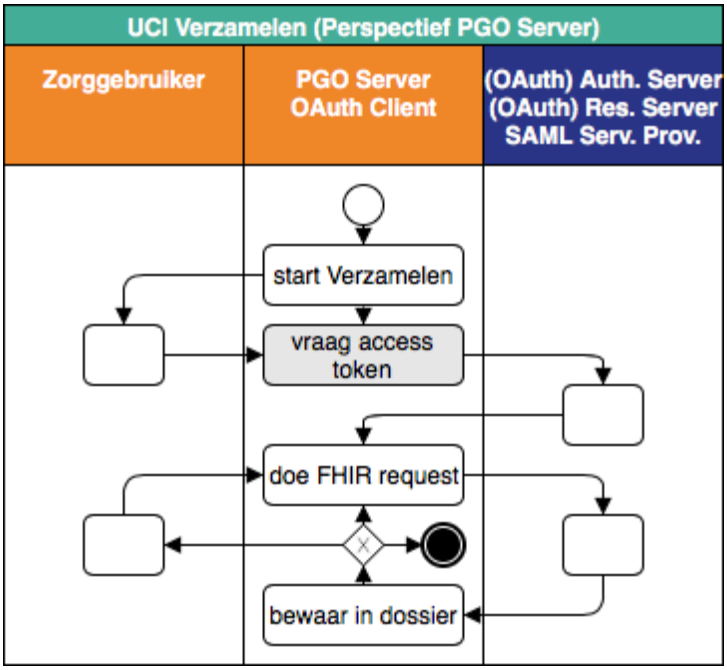
UCI Compile 8	UC Compile 5	*Resource Server* cannot retrieve the requested information at all or in time from the underlying systems.	*Resource Server* informs *PHE Server* ab exception. *PHE Server* then informs *Ca* this.

Specific perspectives

Perspective of PHE Server (happy flow)

Explanatory Notes

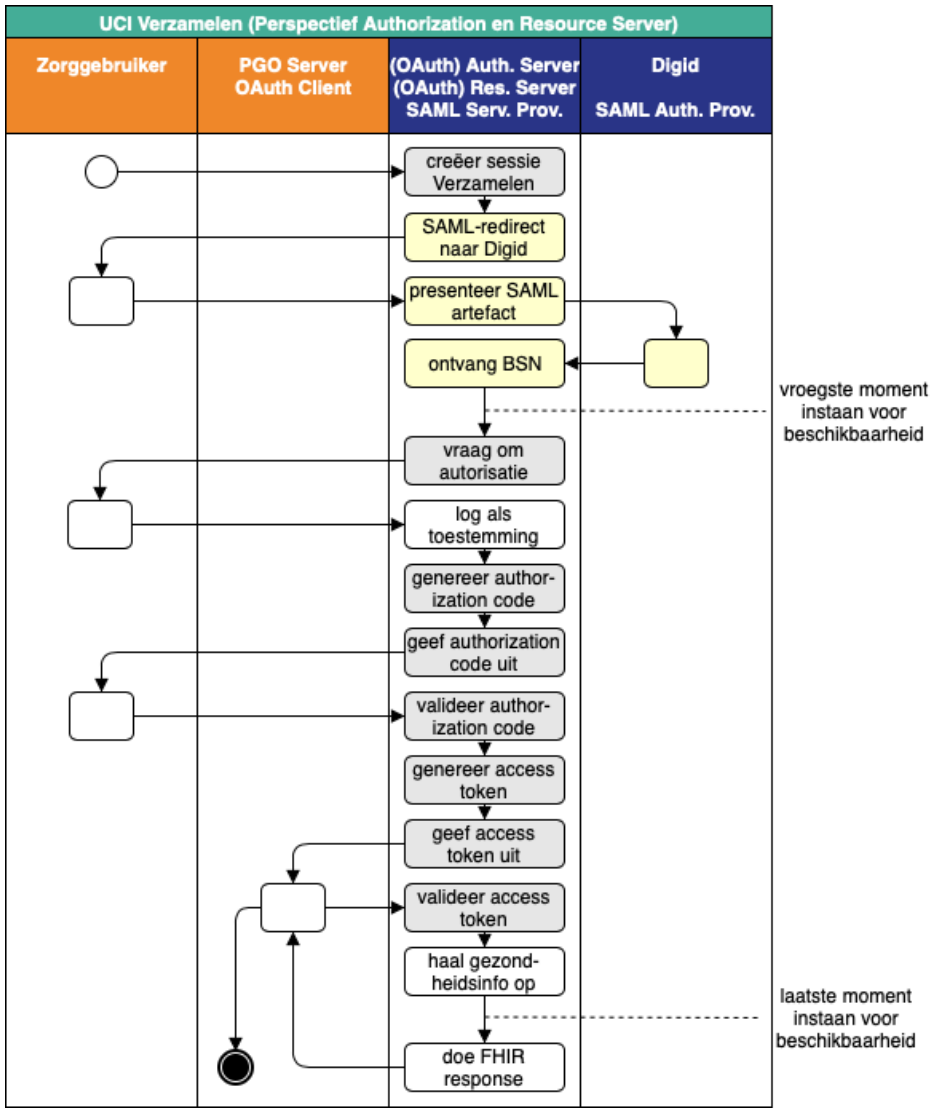
Please find below the same flowchart but now from the perspective of the *PHE Server*. In other words, all in-between steps that are not visible to the *PHE Server* are shorted. *Care User* is "hidden behind the browser" and *DigiD* "behind the *Authorisation Server*".



Perspective of Authorisation Server/Resource Server (happy flow)

Explanatory Notes

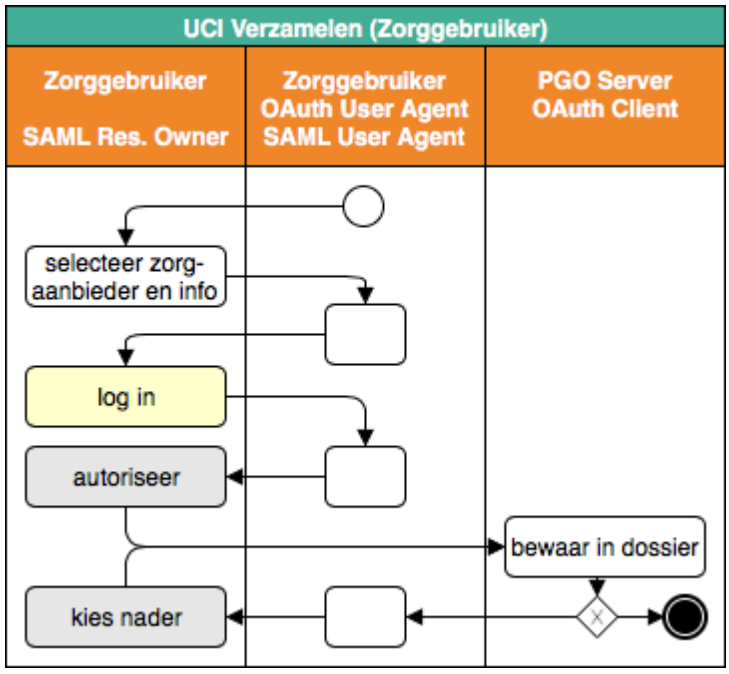
Please find below the same flowchart but now from the perspective of the *Authorisation/Resource Server*. In other words, all in-between steps that are not visible to the *PHE Server* are shorted. *Care User* is "hidden behind the browser".



Perspective of Care User (happy flow)

Explanatory Notes

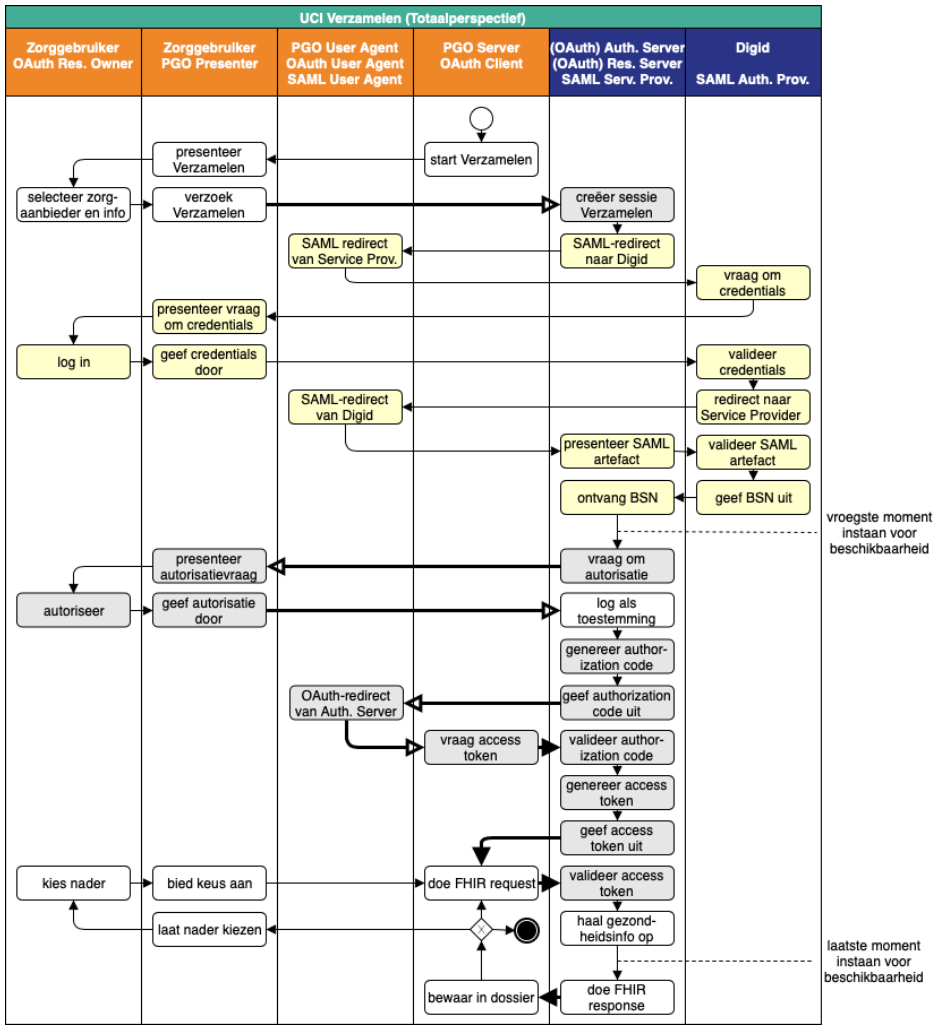
Please find below the same flowchart but now from the perspective of the *Care User*. In other words, all in-between steps that are not visible to the *Care User* are shorted. Almost everything is "hidden behind the browser". We have only kept the final step of *PHE Server* visible, because the retention of the compiled health information has meaning for the *Care User*. The *PHE Server* will probably let the *Care User* know that the compiling was successful but is not obliged to do so.



Frontchannel and backchannel

Explanatory Notes

In the flowchart below for UCI Compile, the thick arrows show the *MedMij data transfer* and include the five cases of frontchannel data transfer (open arrowhead) and four cases of backchannel data transfer (closed arrowhead).


UCI Verzamelen (Totaalperspectief)
Zorggebruiker OAuth Res. Owner
Zorggebruiker PGO Presenter
PGO User Agent OAuth User Agent SAML User Agent
PGO Server OAuth Client
(OAuth) Auth. Server (OAuth) Res. Server SAML Serv. Prov.
Digid SAML Auth. Prov.
start Verzamelen
presenteer Verzamelen
selecteer zorg-aanbieder en info
verzoek Verzamelen
creëer sessie Verzamelen
SAML redirect van Service Prov.
SAML-redirect naar Digid
vraag om credentials
presenteer vraag om credentials
log in
geef credentials door
valideer credentials
SAML-redirect van Digid
redirect naar Service Provider
presenteer SAML artefact
valideer SAML artefact
ontvang BSN
geef BSN uit
vroegste moment instaan voor beschikbaarheid
presenteer autorisatievraag
vraag om autorisatie
autoriseer
geef autorisatie door
log als toestemming
genereer author-ization code
OAuth-redirect van Auth. Server
geef authorization code uit
vraag access token
valideer author-ization code
genereer access token
geef access token uit
kies nader
bied keus aan
doe FHIR request
valideer access token
haal gezond-heidsinfo op
laat nader kiezen
laatste moment instaan voor beschikbaarheid
bewaar in dossier
doe FHIR response