Availability condition and accessibility condition

Explanatory Notes

The purpose of the availability condition and the accessibility condition - which come into force somewhere between the user authentication and the exchange of health information in UC/UCI Compile and UC/UCI Share respectively - is twofold:

1. They want to ensure that it may be assumed as quickly as possible after the authentication of the *Individual*, and in any case before health information is exchanged between *PGO Server* and *Resource Server*, that it may be assumed that two conditions have been fulfilled for the compiling or sharing of the relevant information, namely the existence of a current or former treatment relationship as a basis for it and the individual in question being at least 16 years old. It is the legal ultimate responsibility of the *Care Provider* to verify these conditions or to arrange for this to be done. With regard to the age issue, see also the Legal framework.
2. They give the *Care Provider* the opportunity - at his discretion - to impose additional one-off or systematic restrictions on the arranging of the compiling or sharing of information, for example for technical reasons or due to special situations, special patients or harrowing content.

In other words, the *Care Provider's Service Provider* guarantees - in letting the process proceed - that the treatment relationship is present and that the age is sufficient. How the *Care Provider's Service Provider* guarantees this (with the *Care Provider*) is entirely up to him. The following for instance may contribute to this guaranteeing:

- legal means, such as provisions in the service provision agreement between *Care Provider* and *Care Provider's Service Provider*;
- organisational measures relating to the way in which *Care Providers* manage the file, so that it can be seen from the file information, its organising or even from its mere presence, whether it is based on a treatment relationship;
- automated logic that for a certain *Individual* and a certain *Data Service* determines the receptiveness/availability at a certain *Care Provider*, this following on from organisational measures.

The MedMij Framework does not make it mandatory to explicitly administer the age data and the treatment relationship data. If the existence of a treatment relationship or a sufficient age can - on legal and/or organisational grounds - be implied by other data then the last-named data may also be used with this implication. This is why the MedMij Framework does not specify any logic for the conditions; instead, it solely lays down two necessary components of their post condition: the *Individual* is of sufficient age, and the existence of a current or former applicable treatment relationship.

If unavailability is found to be the case then this says nothing about the precise reason for it. It cannot even be concluded from this that either the treatment relationship is lacking or the individual is not old enough. This is because the *Care Provider* can also have refused for other reasons.

For reasons of data minimisation and user-friendliness, the availability condition and the accessibility condition will preferably become effective as soon as possible, namely immediately after the authentication of the *Individual* and still prior to the authorisation request (the early variant). Against this, the implementation of the conditions becomes easier if they do not need to become effective until the process has arrived at the *Resource Server* (the late variant).

The early and the late variant will be compared below from the perspectives of data minimisation and user-friendliness. Both issues must be viewed from the perspective of the entire use case and all roles involved, as choosing between the early and the late variant has consequences at multiple places simultaneously. The weighing-up for this issue distinguishes between four different situations, depending on two questions:

- Does the *Care Provider* consider the information to (ultimately or otherwise) be available/receptive or himself to be available/receptive to it?
- Does the *Individual* (ultimately or otherwise) give his/her consent?

By the way, the late variants differ subtly between both UC/UCI Compile and UC/UCI Share. In UC/UCI Share, comparatively speaking the late variant is an additional step earlier than in UC/UCI Compile. This is because otherwise a processing (namely: a placement) of health information by the *Resource Server* would take place even before it turned out that the *Care*

Provider was not receptive for this. In UC/UCI Compile, this can take place a step later, because the action to be prevented is only the exchange with the *PGO Server*.

In terms of data minimisation, the two variants can be compared as follows.

	(ultimately or otherwise) indeed available/receptive	**(ultimately or otherwise) not available/not receptive**
(ultimately or otherwise) indeed consent given	• If separate automated logic is used for a test of availability or receptiveness, the early variant requires additional data transfer compared to the late variant, namely between *Authorisation Server* and the component(s) that it addresses in order to execute this test. This data transfer does however take place entirely within the responsibility of a single controller (i.e. a party that has responsibility for processing); no provision takes place. • Only in the early variant does the *Authorisation Server* additionally learn that the treatment relationship and age are in order. In the late variant, it is only the *Resource Server* that learns this. This does not affect the fact that both come under the same ultimately responsible *Care Provider's Service Provider*.	• In contrast to the early variant, in the late variant all the data transfer unnecessarily takes place after the authentication (the consent request, the distributing of Authorisation code and access token and the addressing of the *Resource Server*). This data transfer extends across responsibility boundaries. • In the late variant, the *PGO Server*, unnecessarily learns more about the availability/receptiveness, and thus about the *Individual,* from the *Resource Server* than in the early variant from the *Authorisation Server*. In the early variant, the relevant exception can after all, viewed from the *PGO Server*, also be caused by failing authentication or refusal to give consent. In the late variant, however, the *PGO Server* does indeed come to know, through receipt of the unnecessary Authorisation code, that there is both a treatment relationship and an age that is sufficient.

(ultimately or otherwise) no consent		In contrast to in the early variant, in the late variant a superfluous consent request is made. This data transfer takes place across the relatively unsafe frontchannel.

The two variants can be compared with each other as follows in respect of user-friendliness:

	(ultimately or otherwise) indeed available/receptive	**(ultimately or otherwise) not available/ not receptive**
(ultimately or otherwise) consent indeed given	no difference	In the early variant, the individual is informed immediately, so that he/she: • does not need to carry out any unnecessary or confusing act (meaningless consent) that has legal consequences, as is the case in the late variant; • learns more precisely than in the late variant why an exchange has failed. In the late variant, this failing can occur for other reasons, so that the *Individual* would have to reply on support queries for clarification, which may even be directed at the *Care Provider* . In the early variant, while it's true that Exceptions 2, 3 and 4 are reported together in a single notification - in both UC/UCI Compile and UC/ UCI Share - to the *PGO Server*, which means that the latter cannot distinguish between failing authentication, failing authorisation and failing availability/receptiveness. However, the *Individual* does indeed know himself/herself the result of the authentication and authorisation, thanks to his/her prior direct interaction with the *Authorisation Server*, and accordingly can deduce from this combined notification - without the *PGO Server* knowing about this - whether there was failing availability/accessibility.

(ultimately or otherwise) no consent		In the early variant, the individual is informed immediately and does not need to carry out any unnecessary or confusing act (hollow rejection), in contrast to the late variant.

The cases where the *Care Provider* considers the information to be available/receptive (or himself available/receptive for the information), based on the reasonable behaviour of the *PGO Server*, are probably more numerous than those where this is not the case. On the other hand, the disadvantages of the early variant for the first-named cases are relatively minor, because the Care Provider's Domain and the *Authorisation Server* must already be sufficiently protected for other reasons, even if this is just due to the use made of the BSN. In addition, there is only additional data transfer in so far as automated logic is deployed that means that roles other than the *Authorisation Server*, and thus outside the MedMij Framework, are addressed for this.

In release 1.1.1, the MedMij Framework recommends the early variant, based on the aforementioned analysis. However, the MedMij Framework also permits the late variant, in order to give *Care Provider's Service Provider(s)* both the opportunity to link up quickly and the time to consider how the early variant could be implemented over time.