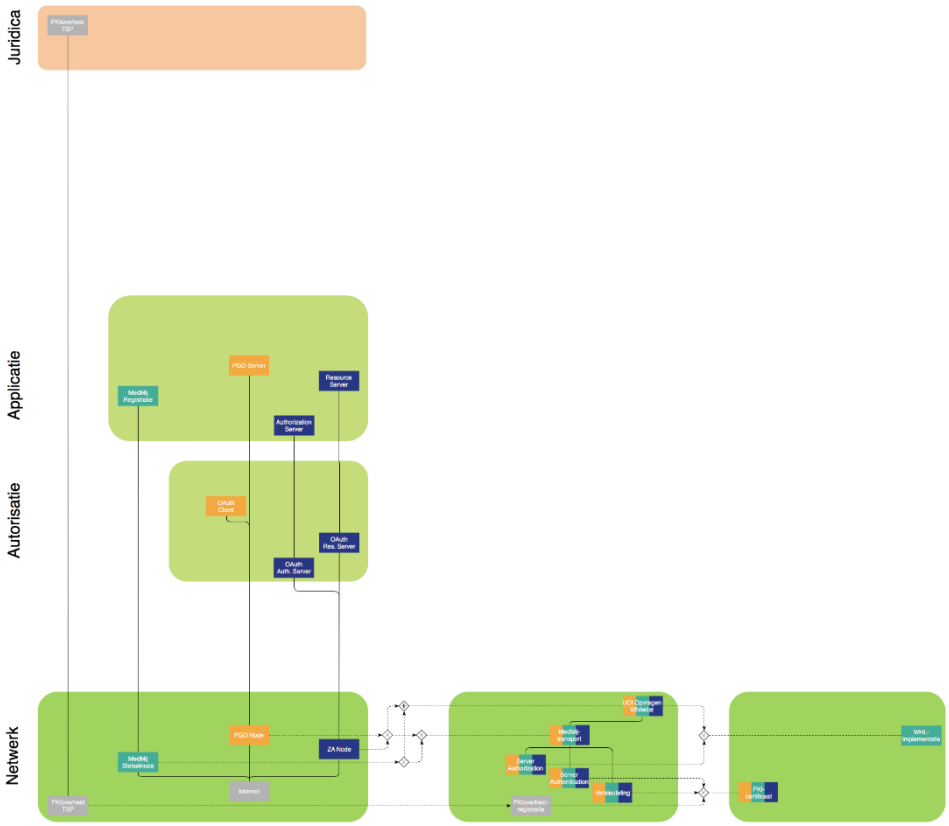
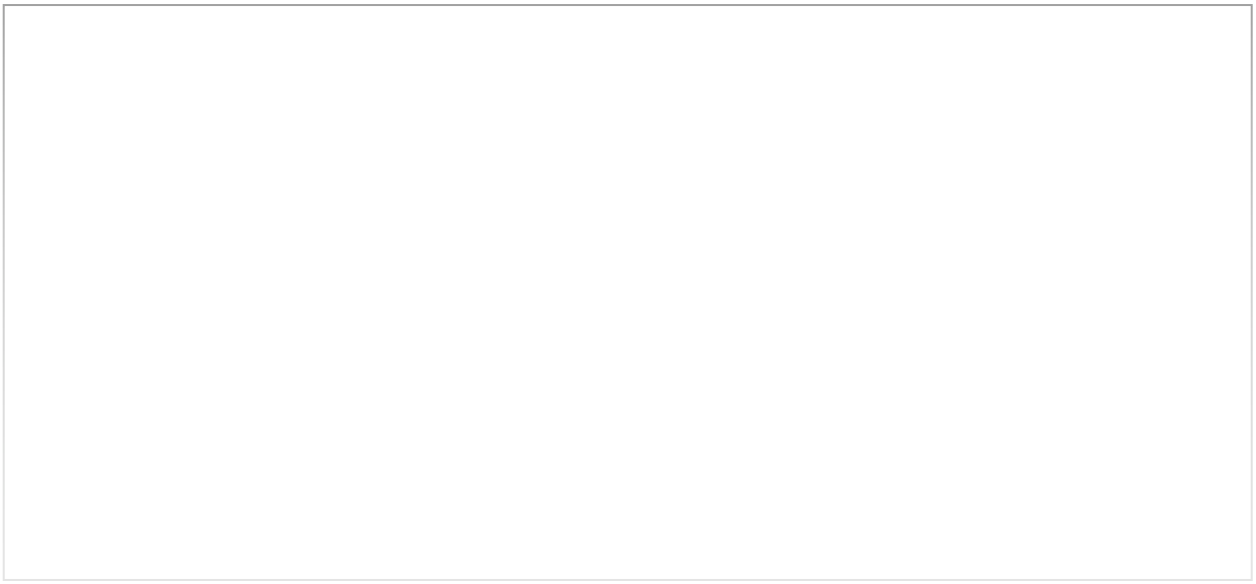
Network



Explanatory Notes

In this layer, the roles (*Nodes*) on the MedMij network are determined and provided with responsibilities in the areas of encryption, authentication of *Nodes* and Authorisation of *Nodes*. This last point refers to the fact that it must in each case be established anew that a Node is entitled to be on the MedMij network. PKI certificates are used for encryption and authentication.



Basically, Authorisation can be included in the MedMij Framework in two ways:

- via these same PKI certificates, in which it can be seen from the certificate holder's domain name whether it relates to a *MedMij Node*, by demanding of it that this domain name has the form <service provider>.medmij.nl;
- or via a list of authorised *MedMij Nodes* (a whitelist) that MedMij manages itself.

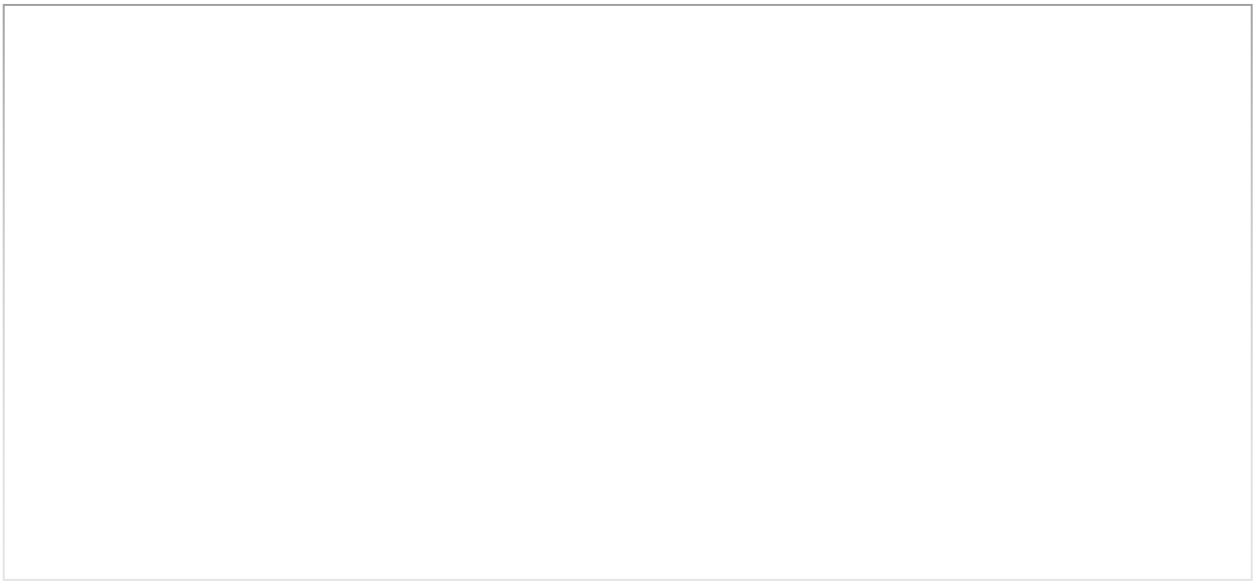
The advantages of the first option would be that:

- in this way, maximum use would be made of arrangements that are already necessary for other purposes too, namely for the use of PKI certificates;
- in this way, the degree of operational central involvement on the part of the MedMij Foundation is minimised, as are accordingly the associated costs and risks. With the whitelist option, the MedMij Foundation would itself have to start managing a list and making it available to all servers in order to make the operational data transfer possible. In the first option, only a name service is needed for the medmij.nl domain names. This last service is a well-standardised, well understood and easily outsourced service, that would result in lower costs, reduced risks and less dependency for the participants;
- in this way, MedMij will comply as far as possible with its architectural principle P6: MedMij only arranges what is necessary.

Despite this, the second option was chosen, because the check on the hostnames and the certificates needed for the first option would only result in undesirable side effects. The following options were explored in this regard:

- The MedMij management organisation becomes **Registration Authority (RA)** in PKIoverheid, in respect of all relevant Certificate Authorities (CAs). However, PKIoverheid does not have this option.

- The MedMij management organisation issues a **domain declaration**, so that participants can themselves request a subdomain under .medmij.nl from a CA. In this way, the management organisation can indeed influence the issuing of a certificate but it is not possible to revoke it later unless there has been misuse. After all, there is no legal relationship between the owner of the domain (the management organisation) and the CA.
- In a similar way to the way in which some parties issue professional certificates, a **customised certification service** is conceivable. The product terms and conditions (valid from the date of the certificate application) then explicitly stipulate that if the registration in an external register is cancelled, the certificate will be revoked by the CA. This requires the register holder (the management organisation) to pass on changes to all CAs. This does not become cost-attractive until there are a considerable number of certificate holders, which there will not be in MedMij for the time being.
- MedMij could set up its **own PKI environment** (different from PKIOverheid). This option was not explored further, due to the complexity and responsibility that would rest on the shoulders of the management organisation.
- The MedMij Foundation could itself be **holder** of all certificates, whereby participants are mandated for management tasks relating to their own subset of certificates. The Foundation can revoke certificates. Identification of the service provider to the user is not possible, because the certificates are in the name of MedMij Foundation.
- A **custom field** could be used in certificates. The MedMij Management Organisation could be allowed to control the way in which this field is dealt with. This probably requires arrangements to be made with all CAs. This gives you control over the issuing of certificates but does not give the management organisation any options for having the certificate revoked.



The table below summarises how the security functions security, authentication and Authorisation are organised in the responsibilities in this layer. With Authorisation, the

distinction drawn between incoming and outgoing data transfer is made because in these two cases, the identification of the other *Node* takes place differently.

	frontchannel data transfer	**outgoing backchannel data transfer**	**incoming backchannel data transfer**
encryption according to TLS, with PKIoverheid certificate	always		
identification on the basis of ...	redirect_uri or *Care Providers List*		PKIoverheid certificate
authentication, on the basis of PKIoverheid certificate, of ...	only the TLS server	TLS client and TLS server	
Authorisation on the basis of checks against the *Whitelist*	no	prior to the TLS handshake	see responsibility 14a

Roles

1. Functionality in the *MedMij network* :

- each *PGO Server*, including his *OAuth* role*,* , functions on one or more *PGO Nodes*. In the case of frontchannel data transfer, each *PGO Server* uses a single *PGO Node*, namely one with a hostname that is stated for this *PGO Server* on the *OAuth Client List*.
- each *Authorisation Server*, including his *OAuth* role, functions on one or more *CP Nodes*;
- each *Resource Server*, including his *OAuth* role, functions on one or more *CP Nodes*;
- precisely one *MedMij System Node* functions, on which *MedMij Registration* functions.

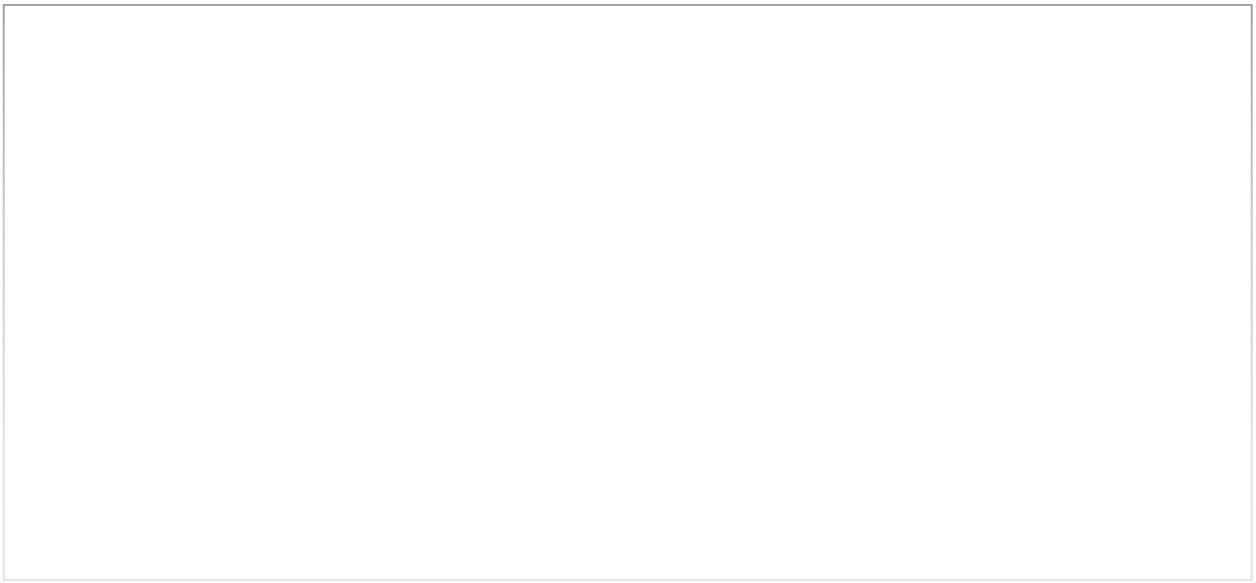
Explanatory Notes

For details of the general principles regarding the numerical relationships between the roles, see the page Architecture and technical specifications.

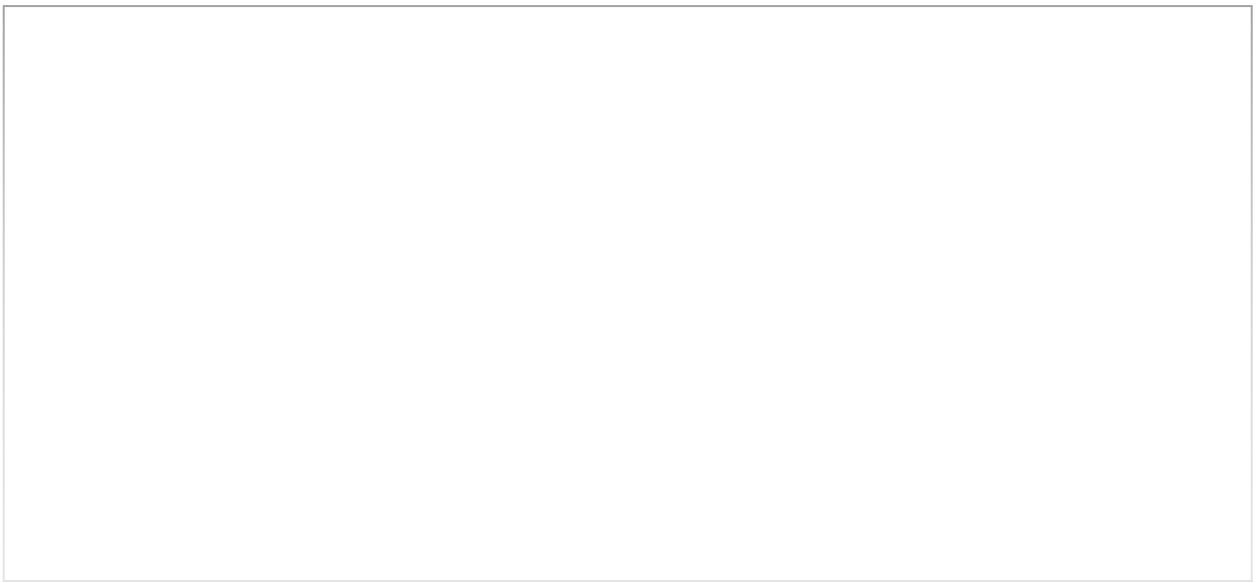
The exception to this regarding the frontchannel data transfer is necessary so that the *OAuth Client List* can function. In other words, it is possible for a *PGO Server* to deploy different certificates for frontchannel and backchannel data transfers, as long as the *OAuth Client List*

contains the same hostname in the certificate for frontchannel data transfer that is stated in the redirect URI regarding OAuth. The *OAuth Client List* is used by the *Authorisation Server* for the consent request (in UCI Compile) and the confirmation request (in UCI Share).

See also responsibility 4 on the page Data and performance in UCI Compile and UCI Share.



There is precisely one *MedMij System Node* in the *MedMij network*. Without this *MedMij System Node* there is no *MedMij network*.



In line with choices made in the Process and Information Layer, in the care providers domain only the *CP Nodes* occur in the *MedMij network*. This means that for instance the underlying xISs will not use the *MedMij network* to communicate with the *CP Node*. This data transfer is

hidden behind the *CP Node*. All routing needed for this is processed by the server implementations and takes place without the MedMij Framework seeing it.

2. On a single:

- *PGO Node* functions a single *PGO Server*;
- *CP Node* functions either a single *Authorisation Server* or a single *Resource Server* or the combination of a single *Authorisation Server* and a single *Resource Server*.

Explanatory Notes

For details of the general principles regarding the numerical relationships between the roles, see the page Architecture and technical specifications.

3. One or more *PKIoverheid TSP*s act as *PKIoverheid TSP*.

Responsibilities

TLS and certificates

1. All the data transfer across the *MedMij network* is protected with Transport Layer Security (TLS). Only TLS versions and TLS algorithms are used that are classified as "good" in the ICT security guidelines for Transport Layer Security (TLS), version 1.0, dated 3 November 2014 issued by the NCSC. Using TLS False Start is prohibited.

Explanatory Notes

Usage of TLS False Start is prohibited, in order to prevent content-related processing of exchanged data taking place before authentication and Authorisation have been successful for the exchange in question (see below).

2. In order to be able to authenticate and authorise themselves on the *MedMij network* where and in the way that the MedMij Framework requires this, each *PGO Node*, each *CP Node* and the *MedMij System Node* can - in relation to the TLS data transfer as referred to in responsibility 1 - submit a PKIoverheid certificate, namely a server certificate from a *PKIoverheid TSP*.

3. All certificate holders undertake to comply with the requirements from the PKIoverheid system that apply to them. A single organisation may have multiple certificates.

Explanatory Notes

The decision to opt for the PKI standard fits in with principle P19 of the MedMij Framework. There are other ways to ensuring trust in a network of automated systems but these are by no means as tried-and-trusted as PKI, which is supported and tested worldwide by governments and market players.

Using the PKI standard raises the issue of which PKI system(s) can or must be used. Such a PKI system provides for a hierarchy of organisations that issue certificates, such that the trustworthiness of the certificates of such an organisation rests on the trustworthiness of the organisation directly above it in this hierarchy, because the certificates of the lower-in-hierarchy have been signed by those of the higher-in-hierarchy. At the top of such a hierarchy there is what is known as the root Certificate Authority (root CA) which cannot derive its trustworthiness from a higher authority and signs its own (master and other) certificates and in this way is a mainstay of the trust placed in the entire relevant PKI system.

The MedMij Framework could have opted to set up a PKI system specifically for MedMij but the cost of doing so, both for itself and for its participants, do not provide sufficient benefits when it is the case that another suitable PKI system is available. After all, participants and their services could become involved in frameworks other than that of MedMij. In addition, such a choice does not fit in with principle P6.

Because the MedMij network is a critical infrastructure from both a national and social point of view, with strict requirements set for its trustworthiness, the MedMij Framework opts for the only PKI system currently available whose trustworthiness is ultimately founded on a single Dutch public-law legal entity, namely PKIoverheid with the State of the Netherlands as root CA. In this way, the governance of the root CA is transparent and accessibly allocated.

In other words, when it comes to the trust that the MedMij Framework provides its participants with, this is based in part on the PKIoverheid system, on the schedule of requirements adopted by that system for the TSPs involved in that system, and on the certification hierarchy of PKIoverheid. Participants in the MedMij Framework must accordingly obtain servicecertificates from a TSP that is affiliated with PKIoverheid that (i.e. the TSP) is right for that participant.

Function: *Encryption*

4. On the *MedMij network* , all the data transfer is encrypted in line with TLS, as referred to in responsibility 1.

Function: *Server Authentication*

5. During the TLS handshake referred to in responsibility 1, the TLS server does the following to the TLS client in the server hello step:

- in the case of backchannel data transfer, the TLS server always submits a request for a certificate. If the TLS client does not hand over a certificate in response then the handshake is immediately terminated.
- in the case of frontchannel data transfer, the TLS server will never submit a request for a certificate.

Explanatory Notes

In the case of backchannel data transfer, therefore, two-way authentication takes place; with frontchannel data transfer, one-way authentication.

6a. *CP Node, PGO Node and MedMij System Node* validate during the TLS handshake at the beginning of a TLS session whether it is a PKIoverheid certificate and check, with the *Certification Authority* on the basis of OCSP, whether the received certificate is valid. If a single one of these checks fails or in the absence of a check result, the certificate will not be accepted and the TLS session will not be started.

6b. With due observance of responsibility 6a, *CP Node*, *PGO Node* and *MedMij System Node* accept both G2 and G3 certificates from each other by:

- trusting the root certificates State of the Netherlands Root CA - G2 and State of the Netherlands Root CA - G3, as published on https://cert.pkioverheid.nl;
- recognising and trusting all *PKIoverheid TSP certificates* and domain certificates under the respective G2 and G3 hierarchies, in so far as they, according to https://cert.pkioverheid.nl:
    - come under the G3 Domain Organisation Services and are of the type Server or else come under the G2 Domain Organisation;
    - have not been revoked.
- regularly consulting the respective G2 and G3 Certificate Revocation Lists on https://crl.pkioverheid.nl and no longer trusting the certificates listed therein;
- processing and using - within 10 working days - changes that apply to G2 and G3 that are set out on the pages https://cert.pkioverheid.nl and https://crl.pkioverheid.nl.

Explanatory Notes

PKIoverheid TSPs issue certificates under various root certificates of the State of the Netherlands, namely an old one (G2) and a new one (G3). The classification of certificate types (domains) differs between G2 and G3. The validity of G2 certificates ends on 22 March 2020 at the latest. Although no more G2 certificates with a validity of three years can accordingly be issued, it has to be possible for the time being to use older G2 certificates for MedMij purposes. In other words, where PKIoverheid certificates have to be accepted, it has to be possible for them to be G2 or G3 certificates. The MedMij Framework has no reason to impose additional restrictions on the way in which PKIoverheid makes the transition from G2 to G3.

Responsibility 6b corresponds to requirements relating to the Framework eRecognition, with the proviso that EV certificates do not have to be accepted in the MedMij Framework. The more comprehensive validation of certificate holders that is deployed for EV certificates is envisaged in the acceptance process for participants in the MedMij Framework.

Function: *Server Authorisation*

Distribution of the *Whitelist*

7. The *MedMij System Node* provides *PGO Node* and *CP Node* with a use case implementation (*UCI Request WHL*) to request the current version of the *WHL implementation* . The roles involved use the relevant flowchart for this.

Explanatory Notes

The *WHL implementation* is the implementation of the *Whitelist* in XML.

8. The participation of the *MedMij System Node* in *UCI Request WHL* is available at least 99.9% of the time. *MedMij Registration* allows - once the participation of *MedMij System Node* in the use case becomes unavailable - a maximum of eight hours (480 minutes) to elapse before it is available again.

9. *PGO Nodes* and *CP Nodes* obtain the most recent *WHL implementation* from *MedMij System Node* at least every fifteen minutes (900 seconds).

10. The *MedMij System Node* has stelselnode.medmij.nl as its hostname. The *MedMij System Node* is not on the *WHL implementation*; however, for the check made against the *Whitelist implementation* it is considered to be on it all the same.

Explanatory Notes

By authorising the *MedMij System Node* in this way for MedMij data transfer, it is ensured that even in error situations or bootstrap situations a *PGO Node* or *CP Node* can address the *MedMij System Node* in order to retrieve a *WHL implementation*.

11. *PGO Nodes and CP Nodes* validate each newly obtained *Whitelist* against the Whitelist's XML schema description. This XML schema description is a technical implementation of the MedMij meta model. All hostnames on the *Whitelist* are fully-qualified domain names, in accordance with RFC3696, section 2

12. The technical security of the data transfer that takes place in relation to *UCI Request WHL* deploys *Encryption*, *Server Authentication* and *Server Authorisation*, in line with the provisions in this Network layer.

Using the *Whitelist*

13. *CP Node, PGO Node and MedMij System Node* allow backchannel data transfer to pass through the *MedMij network* when and only when they have established that the hostname of the other *Node* is present on the most recent *Whitelist* .

Explanatory Notes

In the case of frontchannel data transfer, no Server Authorisation takes place.

14a. The *Node* that

- is to become the TLS client carries out the check referred to in responsibility 13 against the *Whitelist* prior to the start of the TLS handshake. If this check cannot be carried out or else delivers a negative result then the TLS handshake is not started.
- is the TLS server carries out in its entirety the check referred to in responsibility 13 against the *Whitelist* prior to the beginning of what is the next step of the *OAuth Authorisation Server* or *OAuth Resource Server* according to the specifications of UCI Compile and UCI Share. This requirement is called 'sequence'. If the check against the *Whitelist* cannot be carried out or else delivers a negative result then the process is terminated immediately and no start is made to the execution of this next step. In this case, the check against the *Whitelist* is successful if and only if at least one of the following names appears on the *Whitelist* that (i.e. the name) is taken from the certificate provided by the TLS client, namely the Common Name or one of any Subject Alternative Names that there are.

14b. In so far as the *Care Provider's Service Provider* opts to carry out the check against the *Whitelist* after the TLS handshake has ended then this check is separate in logic terms from the next step referred to. The required sequence can be shown using code inspections, penetration tests and inspections of logs.

Explanatory Notes

In the case of outgoing data transfer, the TLS client envisaged can already carry out the check against the *Whitelist* before they initiate the TLS handshake, because they have already identified the envisaged TLS server in order to know who they have to address at all. However, in the case of incoming data transfer the TLS server cannot identify the TLS client that presents himself until during or after the TLS handshake, doing so by means of the certificate that they must receive in accordance with responsibility 2. A hostname must then occur that can be found in the *Whitelist*. By permitting names other than the Common Name to contain the hostname authorised by MedMij, such as a Subject Alternative Name, the MedMij Framework gives participants the option to re-use certificates for multiple MedMij nodes or for purposes other than participation in MedMij.

The earliest - and at first sight the most secure - moment to carry out the check against the *Whitelist* is in that case during the TLS handshake, between the receipt of the certificate from the TLS client and the envisaged sending of the Finished message. If this check cannot be carried out or else delivers a negative result then instead of the Finished message the exception access_denied is sent. Although section 7.2.2 of the TLS specification provides for this possibility, many standard implementations do not. It is sometimes possible to modify these standard implementations but doing so may create new security risks, due for example to the complexity of managing customised modifications to standard implementations.

This is why the MedMij Framework wants to offer more leeway regarding implementation, without however accepting the risk that content-related information from a TLS client will start being processed that originates from a TLS client before the check against the *Whitelist* has ensured that this TLS client was authorised for MedMij data transfer. Because there are multiple ways to implement this even after the TLS handshake has ended, the MedMij Framework does not require any fixed architectural variant (such as with a reverse proxy) for this but does set the requirement regarding sequence, in addition to that of having a logical separation. It has to be possible to show this by means of code inspection, penetration tests and inspections of logs.

15. If a *Whitelist* check in relation to responsibility 14 cannot be carried out or else delivers a negative result then this will terminate the progress of the execution of the use case implementation, with this exception being treated as if it were the next content-related exception in accordance with the tables of exceptions on UCI Compile or UCI Share respectively, provided that the relevant Application roles do not inform each other about this.

Explanatory Notes

In this way, an exception on the Network level also becomes meaningful on the Application level. If the *Whitelist* check is unsuccessful then this indicates an untrustworthy counterparty, which is why the latter is not informed about this.

Domain Name System

16. Each *Individual's Service Provider*, each *Care Provider's Service Provider* and *MedMij Management* ensures, in their role as DNS Server or client thereof, in the public Domain Name System that - with regard to the hostnames of the *MedMij Nodes* or *MedMij System Node* respectively for which they are responsible- that the name records that belong to this hostname are signed in accordance with DNSSEC.

17. The *MedMij System Node* and each *MedMij Node*, in his role as DNS resolver in the Domain Name System, checks whether the name records received have been signed in accordance with DNSSEC and validates this in accordance with DNSSEC. Both this check and validation must be successful; if they are not then they decide not to connect with the relevant hostname.

Explanatory Notes

Usage of DNSSEC (RFC 4033, RFC 4034, RFC 4035) reduces the vulnerability of the Domain Name System to DNS spoofing, for example.