

Meta model

Explanatory Notes

The meta model organises key concepts from the MedMij Framework. It is a conceptual data model, in the form of a UML class diagram. The meta model focusses on providing a coherent description of concepts and relationships, which amongst other things is used in a number of registers, catalogues and lists. Four of them are published by *MedMij Management* for operational usage by Service Providers:

- the *Care Providers List*, from which the *OAuth Client* can see which *Care Providers* are currently providing which *Data Services* and with which they can use to find the relevant technical addresses (URIs) of the *OAuth Authorisation Server* (two Endpoints: the *Authorisation Endpoint* and the *Token Endpoint*) and the *OAuth Resource Server* (the *Resource Endpoint*);
- the *Whitelist*, with which the *Nodes* accept each other as MedMij nodes;
- the *OAuthclient list*, with which the *OAuth Authorisation Server* can find a user-friendly name of the *OAuth Client* to use in the [consent declaration](#) or the [confirmation declaration](#);
- the *Data Service Names List*, from which the *OAuth Client* can see which *Depiction Names* the *Data Services* have that are at any moment available on the MedMij network.

a fifth, the *Catalogue*, is published by MedMij as an annex to the MedMij Framework, on [this page](#). Available logical models are available for all five of them, on a [separate page](#), which are implementations of the meta model.

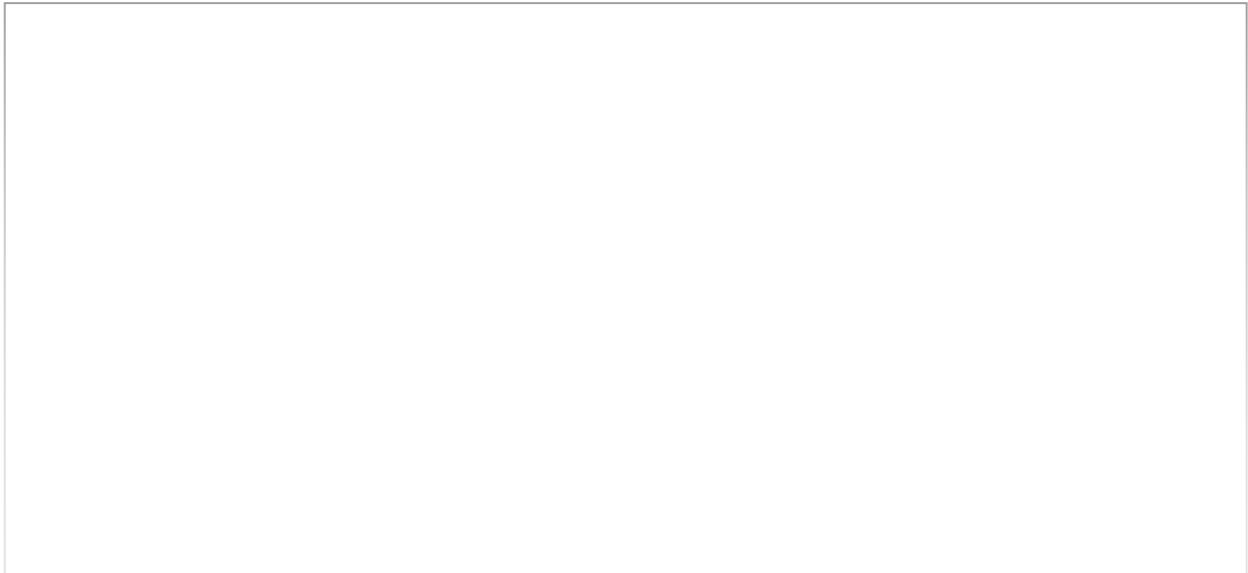
The meta model has been set up in a certain style, with association classes in particular. The advantage of this is that it means that the meta model remains as adaptable and expandable as possible. Common constructions, such as attributes and specialisation are all implementations of association classes. However, we want to leave implementation to the [logical models](#) and the technical models (the [XML schema descriptions](#)). A second advantage is that existence-dependent relationships become explicit. 'Existence-dependent' means that one class is meaningless without the other, and thus that the first-named class cannot exist without the last-named class. An association class is always existence-dependent on the two classes that associate it.

This modelling style has been deviated from in a couple of respects, namely through usage of:

- the uses relationship, especially in the *Information Standards* domain, because that domain is not managed by MedMij;

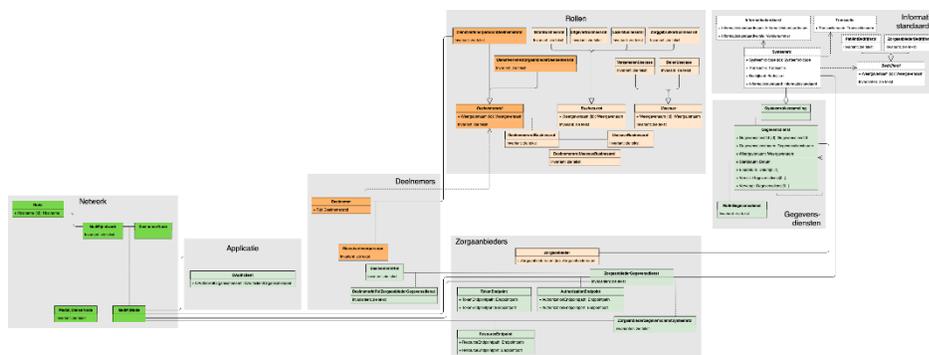
- the aggregation relationship, ditto;
- the object-oriented specialisation, namely where we give a summarised definition of *Participants Role*, *Business Role*, *Usecase* and *Organisational Role*;
- attributes for identification or specification.

In all these cases, association classes could also be used but that would unnecessarily complicate the model's presentation.



For the overview, the meta model has been organised into a number of model domains: *Roles*, *Participants*, *Care Providers*, *Data Services*, *Information standards* and *Network*.

The names of the classes and the attributes all start with a capital letter. The rest of the names consist solely of lowercase letters, apart from where the rest of the name also occurs in the meta model as a separate name or if a proper name is used that requires something different. In other words, the meta model notes *OAuthClient*, because the name *OAuth* is a proper name in which the *A* is written as a capital letter, and because the name *Client* does not occur as a separate name in the meta model. The meta model notes *CareproviderDatSERVICE*, with a capital first *D*, because *Data Service* does indeed occur as a separate name.



Explanatory Notes

The MedMij management organisation keeps track of which *Organisations*, by entering into a *Participant's Agreement*, become a *Participant*. There are *Participants* in two Roles: *Service ProviderindividualParticipantsrole* and *Service ProvidercareproviderParticipantsrole*. These correspond with the respective roles *Individual's Service Provider* and *Care Provider's Service Provider* on the [legal layer](#).

Organisations use *Nodes* that they are the holder of. When an *Organisation* is a *Participant*, it will register such a *Node* as a *ParticipantNode* with the MedMij management organisation. On the *MedMijnetwork*, such a *ParticipantNode* appears as a *MedMijNode*. The *Hostnames* of these *MedMijNodes* are made available by the MedMij management organisation via the *MedMijnetwork*. The *MedMijNodes* use this list as a *Whitelist*, i.e. to determine whether a *Node* that presents itself is authorised to be active in the *MedMijnetwork*. This *Whitelist* appears as an implementation component only in the [logical models](#). This also applies for the *MedMijSystemNode*, which is the *Node* that *MedMij Management* uses to publish four lists. The *MedMijSystemNode* does not explicitly appear in the *Whitelist*, but it is indeed authorised to participate in the *MedMijnetwork*. What is more, without the *MedMijSystemNode* the *MedMijnetwork* cannot work.

For the *MedMijNodes* of *Participants* who are *Service Providerindividual* (or better put: for the *OAuth Clients* on the [application layer](#) during the authorisation phase of [UCI Compile](#) and [UCI Share](#)), the *OAuthclientlist* contains user-friendly names (*Organisationname*), in order to be used in the [consent declaration](#) and the [confirmation declaration](#). The *OAuthclientlist* is an implementation component and appears in the [logical models](#).

In the *Rolesmodel* domain, the *Participantroles*, *Businessroles* and *Usecases* appear that exist in this release of the MedMij Framework, along with their permitted combinations. In the *Participantsmodel* domain, the *Participants* in the MedMij Framework are addressed and for which *Care Providers* they provide which *Data Services*.

Data Services belong to a *Usecase* and have a validity period. In addition, by means of the attribute *Required*, it is required from some *Data Services* that, as a *Care Provider* that provides a *Data Service*, they must also provide certain other *Data Services*, too. This list will often be empty but it makes little sense, for example, to provide the *Share* of an agreement request without also providing the *Compile* of the response to it. The class *RoleInDatasevice* is used, via the *Participant*, to link the MedMij roles *ServiceproviderindividualParticipantsrole* and *ServiceProvidercareproviderParticipantsrole* with the corresponding roles that Nictiz has formulated in the *Informationstandards* domain, namely *PatientOrganisationalrole* and *CareproviderOrganisationalrole*.

The classes in the model domain *Information Standards*, including their names, must be understood in the sense in which Nictiz uses them in the context of the *Information Standards* that are permitted for use in MedMij. This is why the outlines of these classes are shown as dotted lines. An *Organisational Role*, which there are two of (*PatientOrganisationalrole* and *CareproviderOrganisationalrole*), are accepted by a *System Role*. Each *System Role* must have an *Information Standard*. *System Roles* are grouped into *System Role compilations* that together with a *Usecase* form a *Data Service*. A current example of a *System Role compilation* is a compilation of four *System Roles*, two of which (one for each relevant *Organisational Role*) exchange an overview of available PDF documents and two of which (again one for each relevant *Organisational Role*) exchange a PDF document from that overview. *Data Services* are provided as a unit (that is to say with their entire *System Role compilation*) to *Care Users*, and these users will also authorise them all at once.

At the bottom of the model, the link is made with the *Care Providers*. This model domain is the basis for the [logical model](#) of the *Care Providers List*. When a *Care Provider* provides a certain *Data Service* then a *Care Provider's Data Service* belongs to this, too. This class can be used to inform *Care Users* about which of the *Care Providers* provide which *Data Services*. Furthermore, within a *Data Service* there are one or more *System Roles*. This relationship is contained in the class *CareproviderDataseviceSystemrole*.

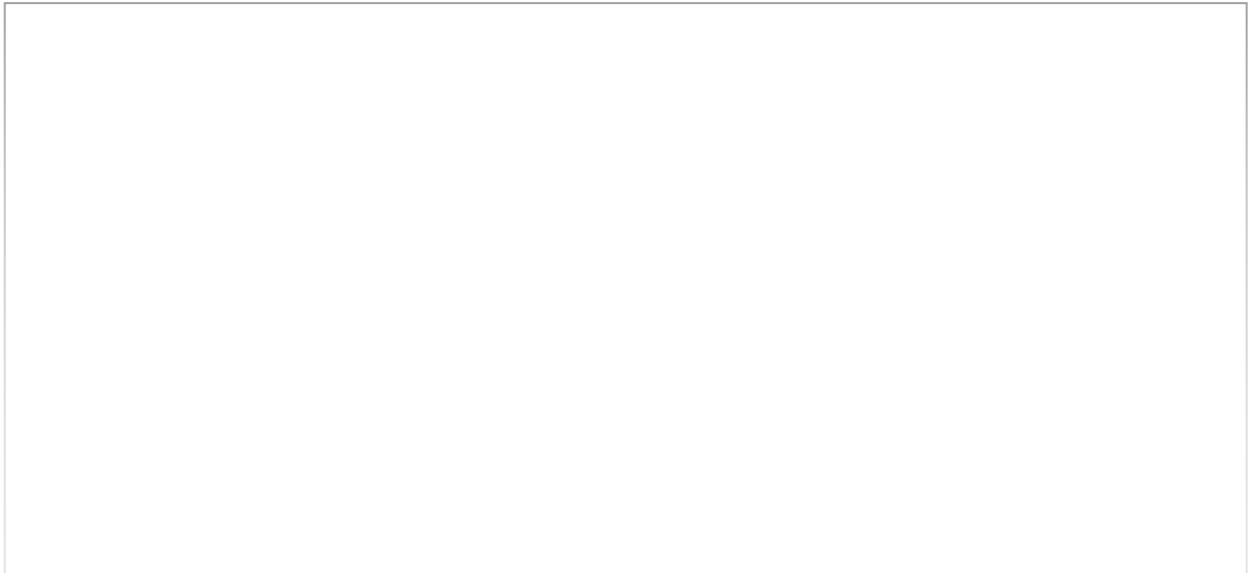
A *CareproviderDatasevice* must have a single *AuthorisationEndpoint* and a single *TokenEndpoint*, and a *CareproviderDataseviceSystemrole* must have a single *ResourceEndpoint*. For all three types of endpoints, the metamodel names the technical address (URI) with which they are addressed, namely:

- the *Endpointpath*, i.e. a first section of the path in the URI;
- the *Endpointport*, i.e. the (optional or other) port number that is used in the data transfer via the back-channel. This does not apply for the *AuthorisationEndpoint*, because this is addressed via the front-channel, which means that the standard IANA port for https is mandatory for this.

These components are compiled together with the *Hostname* of the relevant *MedMijNode* into a URI that is considered to be the address of the respective endpoint. This is done in the [logical model](#) (with invariants). The requirements for all these components and for the method used to

compile them into the URIs is described on the page [Data and performance in UCI Compile and UCI Share](#).

The same *Care Provider* may - for different *Data Services* - utilise services of various *Participants*. However, a single *CareproviderDatSERVICE* must have precisely one *ParticipantInRole*. This is why the class *ParticipantInRoleCareproviderDatSERVICE* has been included in the metamodel, namely in the *Participantsmodel* domain.

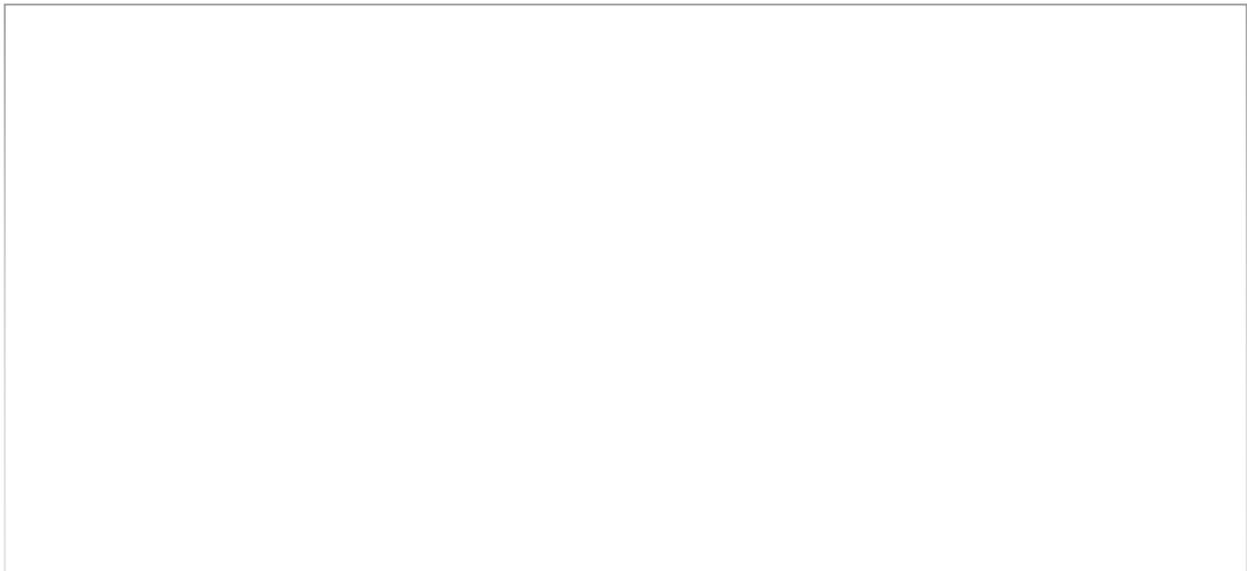


Invariants, i.e. restrictions that apply at all times, are shown at the bottom in a separate table. There are various types of these, which are named in the right-hand column:

- Summaries state that a certain class has a fixed number of explicitly named instances.
- Numerical relationships stipulate numerical requirements for the number of instances that a class has or for the relationship between the numbers in multiple classes.
- Local dependencies impose restrictions on the content-related relationships between attributes of the same class.
- Non-local dependency imposes restrictions on the content-related relationships between instances of different classes.
- Role connections limit the role combinations of different role classes. They correspond to (amongst other things) the role connections between the different layers.



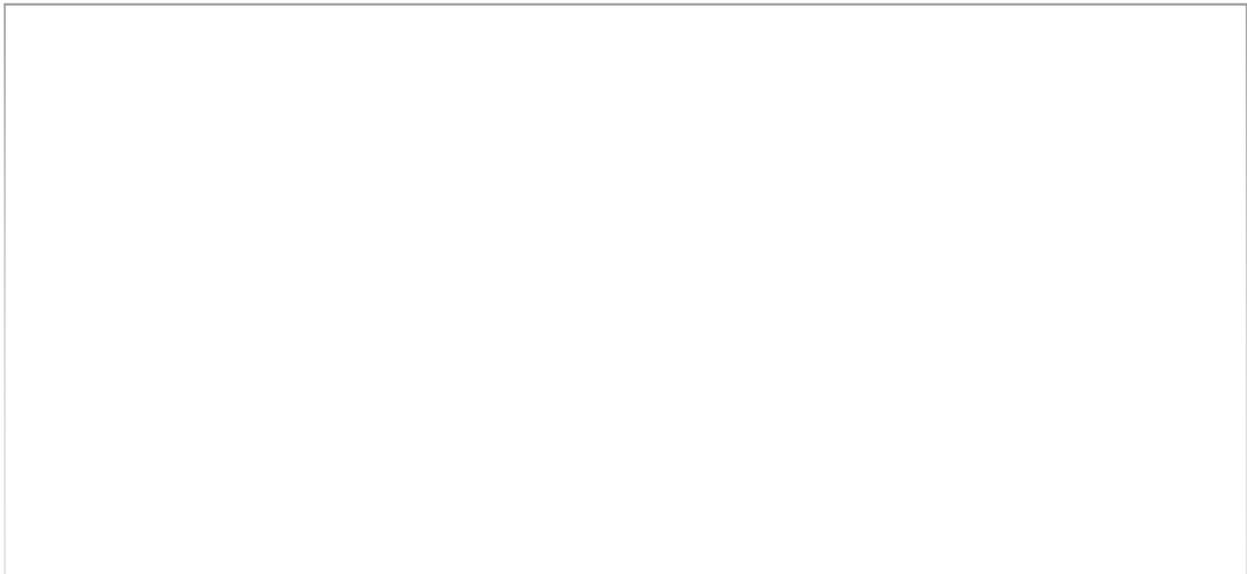
The classes in the metamodel belong to the different [layers](#) in the architecture of the framework. The relevant layer is shown by colouring in the classes. Only in the case of the Nictiz classes in the *Register of Information Standards* have we not done this.



It is clear from this metamodel how addressing is dealt with in the MedMij Framework. The addressing system consists of three components:

- MedMij care provider names for *Care Providers*, as described in responsibility 13 in the [Processes and Information layer](#);
- *Data Services* with *System Roles* as included in the *Catalogue* or *Register of Information Standards* respectively;

- Each *Care Provider* has for each *Care Provider's Data Service* (which they provide via a *Care Provider's Service Provider*) a single *AuthorisationEndpoint* and a single *TokenEndpoint* and for each *CareproviderDataproviderSystemrole* within it a single *ResourceEndpoint*. The endpoints each have a URI as a technical address.



Where the metamodel refers to periods, limited by a start and an end, this start and end must be interpreted as being begin moments. If it relates to a start date and end date, as is the case in the attributes of *Data Service*, the begin moments of these dates are accordingly meant, at 00h00m00. The start is interpreted as the begin of the validity and the end as the begin of the invalidity. This is why the validity runs from start to end (and not 'up to and including'). This also means that if the end is lacking that the validity extends for an indefinite period of time.

Invariants

The above diagram is organised by (existence-) dependencies between classes. Within this organisation, there are also consistency requirements that are imposed on the instances of these classes. These are the invariants that are shown in the table below. An invariant expresses the fact that an instance of the relevant class does not exist if it does not comply with the invariant. The table makes no other statement about how the monitoring of this consistency is implemented. In many implementations, temporary inconsistencies are permitted and are only refused or remedied later on. There are many ways to do this but the MedMij Framework wants to allow great leeway in the way in which the consistency in registrations is guaranteed.

The path expressions in the invariants consist of names separated by full stops. A step is always taken from a certain class to a class on whom the first-named class is directly existence-

dependent. The name of the side of the association about which the step is taken is deemed to bear the name of the relevant endpoint of the association, i.e. of the step's destination.

Relates to instances of class ...	Invariant	Model domain	Explanatory Notes	Nature
<i>Organisational Role</i>	Each <i>Organisational Role</i> is either <i>PatientOrganisationalrole</i> or <i>Care ProvidersOrganisationalrole</i> .	<i>Information Standards</i>	This is an exclusive summary.	summary
<i>Organisational Role</i>	For each <i>Organisational Role b</i> it is true that: IF(<i>b : PatientOrganisationalrole</i> THEN <i>b.Depictionname</i> = "Patient"; <i>b : Care ProvidersOrganisationalrole</i> THEN <i>b.Depictionname</i> = "Care Provider"; IF NOT THEN ERROR)	<i>Information Standards</i>	This links the names of the subclasses to the depiction names.	local dependency
<i>SourceBusinessrole</i>	There is precisely one instance of this.	<i>Participants</i>	This is a 'loner' in the model.	numerical relationship
<i>Business Role</i>	For each <i>Business Role b</i> it is true that: IF(<i>b : SourceBusinessrole</i> THEN <i>b.Depictionname</i> = "Source"; <i>b : ReaderBusinessrole</i> THEN <i>b.Depictionname</i> = "Reader"; <i>b : PublisherBusinessrole</i> THEN <i>b.Depictionname</i> = "Publisher"; IF NOT THEN ERROR)	<i>Roles</i>	This links the names of the subclasses to the depiction names.	local dependency

<i>ParticipantInRole</i>	For each <i>ParticipantInRole d</i> it is true that: <i>d.Participant.Participantsrole</i> and <i>d.RoleInDatasevice</i> . <i>ParticipantsroleUsecaseBusinessrole.Participantsrole</i> are identical.	<i>Participant s</i>	The relevant <i>Participant</i> can only register for roles that the <i>Catalogue</i> provides.	non-local dependency
<i>ParticipantInRoleCareproviderDatasevice</i>	For each <i>ParticipantInRoleCareproviderDatasevice d</i> it is true that: <i>d</i> . <i>CareproviderDatasevice.Datasevice = d.ParticipantInRole.RoleInDatasevice.Data Service</i>	<i>Care Providers</i>	A <i>Participant</i> can only assert authority about the inclusion - in the <i>Care Providers List</i> - of a <i>Data Service</i> with a <i>Care Provider</i> if this <i>Participant</i> is also approved for this <i>Data Service</i> in MedMij.	non-local dependency
<i>Individual's Service Provider</i>	There is no more than a single instance of this with a single <i>Participant</i> , and precisely one of them if the <i>Participant's Role</i> of the last-named is of the type <i>Service ProviderindividualParticipantsrole</i> .	<i>Participant s</i>	A <i>Participant</i> is called a <i>Service Providerindividual</i> when and only when they play the applicable role.	numerical relationship
<i>Participant's Role</i>	For each <i>Participant's Role d</i> it is true that: IF(<i>d : Service ProviderindividualParticipantsrole</i> THEN <i>d.Depictionname = "Individual's Service Provider"</i> ; <i>d : Service ProvidercareproviderParticipantsrole</i> THEN <i>d.Depictionname = "Care Provider's Service Provider"</i> ; IF NOT THEN ERROR)	<i>Roles</i>	This links the names of the subclasses to the depiction names.	local dependency

<p><i>ParticipantsroleBusinessrole</i></p>	<p>There are precisely three instances of this, namely:</p> <ul style="list-style-type: none"> • a single one such that <i>ParticipantsroleBusinessrole.Participantsrole : Service ProviderindividualParticipantsrole</i> and <i>ParticipantsroleBusiness. Businessrole : PublisherBusinessrole;</i> • a single one such that <i>ParticipantsroleBusinessrole.Participantsrole : Service ProvidercareproviderParticipantsrole</i> and <i>ParticipantsroleBusiness. Businessrole : SourceBusinessrole;</i> and • a single one such that <i>ParticipantsroleBusinessrole. Participantsrole : ServiceprovidercareproviderParticipantsrole</i> and <i>ParticipantsroleBusiness. Business Role : ReaderBusinessrole;</i> 	<p><i>Roles</i></p>	<p>This is where the two legal roles <i>Care Provider's Service Provider</i> and <i>Individual's Service Provider</i> are linked to the Business Roles <i>Publisher, Source</i> and <i>Reader</i>.</p>	<p>role connection</p>
--	--	---------------------	--	------------------------

<p><i>ParticipantsroleUsecaseBusinessRole</i></p>	<p>This class consists of precisely one instance for each combination of an instance <i>d</i> of <i>ParticipantsroleBusinessrole</i> and an instance <i>u</i> of <i>UsecaseBusinessrole</i> for which it is true that: <i>d.BusinessRole = u.BusinessRole</i>.</p>	<p><i>Roles</i></p>	<p>This is where all (namely four) suitable combinations are made of <i>ParticipantsroleBusinessrole</i> and <i>UsecaseBusinessrole</i>. This relates to: ServiceproviderIndividual/Publisher/Compile, ServiceproviderIndividual/Publisher/Share, ServiceproviderCareprovider/Source/Compile and ServiceproviderCareprovider/Reader/Share.</p>	<p>summary</p>
<p><i>ShareUsecase</i></p>	<p>There is precisely one instance of this.</p>	<p><i>Roles</i></p>	<p>This is a 'loner' in the model.</p>	<p>numerical relationship</p>
<p><i>Service ProviderindividualParticipantsrole</i></p>	<p>There is precisely one instance of this.</p>	<p><i>Roles</i></p>	<p>This is a 'loner' in the model.</p>	<p>numerical relationship</p>
<p><i>Service ProvidercareproviderParticipantsrole</i></p>	<p>There is precisely one instance of this.</p>	<p><i>Roles</i></p>	<p>This is a 'loner' in the model.</p>	<p>numerical relationship</p>
<p><i>Data Service</i></p>	<p>There are none or more <i>Data Services</i>.</p>	<p><i>Data Services</i></p>	<p>It is possible at any moment to have no <i>Data Services</i>.</p>	<p>numerical relationship</p>

<i>Data Service</i>	For each <i>Data Service g</i> it is true that: <i>g.Start date</i> is before <i>g.End date</i> .	<i>Data Services</i>	Otherwise the validity period makes no sense.	local dependency
<i>Data Service</i>	For each <i>Data Service g1</i> and <i>g2</i> it is true that: IF <i>g2</i> occurs in <i>g1.Required</i> THEN (<i>g2</i> is stated as a <i>Data Service</i> in <i>Catalogue</i> AND <i>g1.Start date</i> is not before <i>g2.Start date</i> AND <i>g1.End date</i> is not after <i>g2.End date</i>)	<i>Data Services</i>	A valid <i>Data Service</i> cannot require a non-existent or invalid <i>Data Service</i> . A missing <i>End date</i> (because it is optional) means "for an indefinite period of time" and is after each "specified period of time".	local dependency
<i>Data Service</i>	For each <i>Data Service g</i> it is true that: <i>g.Data Servicename</i> is a concatenation of <i>g.Usecase.Depictionname</i> , <i>g.Depictionname</i> and the first two number sequences (in so far as present and with the separating full stop) of <i>g.Systemrole.Informationstandard.Informationstandardversion</i> , with a space as delimiter.	<i>Data Services</i>	This standardises the naming of the <i>Data Services</i> . Note that only the first two number sequences of the <i>Information Standard version</i> are taken over. All further number sequences (for patches for example) are considered to be non-significant.	non-local dependency
<i>Data Service</i>	For each two different <i>Data Services g1</i> and <i>g2</i> it is true that: <i>g1.Data Servicename</i> \neq <i>g2.Data Servicename</i>	<i>Data Services</i>	In this way, different <i>Data Services</i> are not confused with each other by having the same name.	local dependency
<i>ReaderBusinessrole</i>	There is precisely one instance of this.	<i>Roles</i>	This is a 'loner' in the model.	numerical relationship

<i>MedMijnetwork</i>	There is precisely one instance of this.	<i>Network</i>	This is a 'loner' in the model.	numerical relationship
<i>MedMijSystemNode</i>	There is precisely one instance of this.	<i>Network</i>	Without <i>MedMijSystemNode</i> no <i>MedMijNetwork</i> and no <i>Whitelist</i> .	numerical relationship
<i>Node</i>	The hostname of a Node contains a domain name that is a fully-qualified domain name, in accordance with RFC3696, section 2 .	<i>Network</i>	This is a measure to combat the risk 4.4.1.4 from RFC 6819.	local dependency
<i>OAuthclient</i>	For each <i>OAuthclient o</i> it is true that: <i>o.OAuthclientOrganisationname</i> complies with the OAuthclientnamepolicy .	<i>Application</i>	See the OAuthclientnamepolicy .	local dependency
<i>PatientOrganisationalrole</i>	There is precisely one instance of this.	<i>Information Standards</i>	This is a 'loner' in the model.	numerical relationship
<i>RoleInDataservice</i>	This class consists of precisely a single instance <i>r</i> for each combination of an instance <i>d</i> of <i>r.ParticipantsroleUsecaseBusinessrole</i> and an instance <i>g</i> of <i>r.Data Service</i> for which it is true that: <i>g.Usecase = d.UsecaseBusinessrole.Usecase</i>	<i>Participants</i>	In this way it is ensured that the <i>Usecase</i> that belongs to the relevant <i>Data Service</i> corresponds with the <i>Usecase</i> that belongs to the <i>ParticipantsroleUsecaseBusinessrole</i> . For each time that this is the case, this class has an instance.	non-local dependency

<p><i>System Role</i></p>	<p>For each <i>System Role s</i> it is true that: IF <i>s.Organisationalrole : PatientOrganisationalrole</i> THEN it is true that for all <i>RoleInDatSERVICE r:</i> (IF <i>s in r.Data Service.TransactionCompilation</i> THEN <i>r.ParticipantsroleUsecaseBusinessrole.Participantsrole : Service ProviderindividualParticipantsrole</i>)</p>	<p><i>Participant s</i></p>	<p>This links the MedMij role <i>Individual's Service Provider</i> to the Nictiz role <i>Patient</i>.</p>	<p>role connection</p>
<p><i>System Role</i></p>	<p>For each <i>System Role s</i> it is true that: IF <i>s.Organisationalrole : CareproviderOrganisationalrole</i> THEN it is true that for all <i>RoleInDatSERVICE r :</i> (IF <i>s in r.Data Service.TransactionCompilation</i> THEN <i>r.ParticipantsroleUsecaseBusinessrole.Participantsrole : Service ProvidercareproviderParticipantsrole</i>)</p>	<p><i>Participant s</i></p>	<p>This links the MedMij role <i>Care Provider's Service Provider</i> to the Nictiz role <i>Care Provider</i>.</p>	<p>role connection</p>
<p><i>PublisherBusiness Role</i></p>	<p>There is precisely one instance of this.</p>	<p><i>Roles</i></p>	<p>This is a 'loner' in the model.</p>	<p>numerical relationship</p>
<p><i>Usecase</i></p>	<p>For each <i>Usecase u</i> it is true that: IF(<i>u : CompileUsecase THEN u.Depictionname = "Compile";</i> <i>u : ShareUsecase THEN u.Depictionname = "Share";</i> IF NOT THEN ERROR)</p>	<p><i>Roles</i></p>	<p>This links the names of the subclasses to the depiction names.</p>	<p>local dependency</p>

<p><i>Usecase Business Role</i></p>	<p>There are precisely four instances of this, namely:</p> <ul style="list-style-type: none"> • a single one such that <i>UsecaseBusinessrole.Businessrole : PublisherBusinessrole</i> and <i>UsecaseBusinessrole.Usecase : CompileUsecase</i>; • a single one such that <i>UsecaseBusinessrole.Businessrole : PublisherBusinessrole</i> and <i>UsecaseBusinessrole.Usecase : ShareUsecase</i>; and • a single one such that <i>UsecaseBusinessrole.Businessrole : SourceBusinessrole</i> and <i>UsecaseBusinessrole.Usecase : CompileUsecase</i>; and • a single one such that <i>UsecaseBusinessrole.Businessrole : ReaderBusinessrole</i> and <i>UsecaseBusinessrole.Usecase : ShareUsecase</i>. 	<p><i>Roles</i></p>	<p>Here it is determined which <i>Business Roles</i> participate in which <i>Usecases</i>.</p>	<p>summary</p>
<p><i>CompileUsecase</i></p>	<p>There is precisely one instance of this.</p>	<p><i>Roles</i></p>	<p>This is a 'loner' in the model.</p>	<p>numerical relationship</p>
<p><i>Care ProvidersOrganisationalrole</i></p>	<p>There is precisely one instance of this.</p>	<p><i>Information Standards</i></p>	<p>This is a 'loner' in the model.</p>	<p>numerical relationship</p>

<i>Care Provider</i>	Each <i>Care Provider</i> has at least one <i>CareproviderDatSERVICE</i>	<i>Care Providers</i>	If not then the inclusion of the <i>Care Provider</i> in the <i>Care Providers List</i> is pointless.	numerical relationship
<i>Care Provider</i>	Each <i>Care Provider</i> has with each <i>Data Service</i> no more than a single <i>CareproviderDatSERVICE</i> .	<i>Care Providers</i>	In this way, the <i>OAuth Client</i> can - for the combination of a <i>Care Provider</i> and a <i>Data Service</i> - find the <i>AuthorisationEndpoint</i> and <i>TokenEndpoint</i> in the <i>Care Providers List</i> .	numerical relationship
<i>Care Provider</i>	For each <i>CareproviderDatSERVICE zg1</i> and for each <i>Data Service g</i> in <i>zg1.Data Service.Required</i> , it is true that: there is a <i>CareproviderDatSERVICE zg2</i> , so that <i>zg1.Careprovider = zg2.Careprovider</i> and <i>zg1.DatSERVICE = g</i> .	<i>Care Providers</i>	In this way it is ensured that, if a <i>Care Provider</i> provides a <i>Data Service</i> that requires another one then this <i>Care Provider</i> also provides the other one.	non-local dependency

<p><i>Care Provider</i></p>	<p>For each <i>CareproviderDatSERVICE zg1</i> and for each <i>Data Service g</i> in <i>zg1.Data Service.Replaces</i>, it is true that:</p> <p>there is, unlike for <i>zg1</i>, <u>no</u> <i>CareproviderDatSERVICES zg2</i>, so that <i>zg1.Careprovider = zg2.Careprovider</i> and</p> <p>either <i>zg2.DatSERVICE</i> follows <i>g</i> or <i>g</i> follows <i>zg2.DatSERVICE</i>.</p> <p>'Follows' is (recursively) defined as follows here: <i>Data Service h1</i> follows <i>Data Service h2</i> if either <i>h1=h2</i> or else <i>h2</i> follows a <i>Data Service</i> in <i>h1.Replaces</i>.</p>	<p><i>Care Providers</i></p>	<p>In this way it is ensured that a <i>Care Provider</i> cannot provide two <i>Data SERVICES</i> where one of them replaces the other (either directly or indirectly). The indirect relationship is the reason why a recursive definition is needed.</p>	<p>non-local dependency</p>
<p><i>CareproviderDatSERVICE</i></p>	<p>For each <i>CareproviderDatSERVICE.DatSERVICE</i>. <i>TransactionCompilation.TransactionSystemrole s</i> for which it is true that <i>s.Organisationalrole = CareproviderOrganisationalrole</i>, it is true that there is a <i>CareproviderDatSERVICESystemrole z</i> so that <i>z.Systemrole = s</i>.</p>	<p><i>Care Providers</i></p>	<p>If it is the case in the Catalogue that a <i>System Role</i> for <i>Care Providers</i> belongs to a certain <i>Careprovider's</i> provided <i>Data Service</i> that is provided on this certain <i>Care Provider's</i> behalf then it is on behalf of this same <i>Care Provider</i> that this <i>System Role</i> must be provided, too.</p>	<p>non-local dependency</p>
<p><i>CareproviderDatSERVICE</i></p>	<p>Each <i>CareproviderDatSERVICE</i> has precisely one <i>AuthorisationEndpoint</i>.</p>	<p><i>Care Providers</i></p>	<p>In this way, the <i>OAuth Client</i> can - with the combination of a <i>Care Provider</i> and a <i>Data Service</i> - find the <i>AuthorisationEndpoint</i> in the <i>Care Providers List</i>.</p>	<p>numerical relationship</p>

<p><i>CareproviderDataser vice</i></p>	<p>Each <i>CareproviderDataser vice</i> has precisely one <i>TokenEndpoint</i>.</p>	<p><i>Care Provi ders</i></p>	<p>In this way, the <i>OAuth Client</i> can - with the combination of a <i>Care Provider</i> and a <i>Data Service</i> - find the <i>TokenEndpoint</i> in the <i>Care Providers List</i>.</p>	<p>nume rical relati onshi p</p>
<p><i>CareproviderDataser vice</i></p>	<p>Each <i>CareproviderDataser vice</i> has precisely one <i>ParticipantInRoleCareproviderDataser vice</i>, and in such a way that <i>d.ParticipantInRole.Participant.Role = Service ProvidercareproviderParticipants role</i>.</p>	<p><i>Care Provi ders</i></p>	<p>In this way it is clear which <i>ParticipantInRole</i> is responsible for a <i>CareproviderDataser vice</i>, and that it relates to a <i>Care Provider's Service Provider</i>.</p>	<p>nume rical relati onshi p and role conne ction</p>

<p><i>CareproviderDatasevice</i></p>	<p>For each <i>CareproviderDatasevice z</i>, for each <i>AuthorisationEndpoint a</i> of <i>z</i> and for each <i>ParticipantinRoleCareproviderDatasevice d</i> of <i>z</i> it is true that: there is a <i>MedMijnNode m</i> such that: <i>a.AuthorisationEndpointuri</i> contains <i>m.ParticipantNode.Node.Hostname</i> and <i>d.ParticipantinRole.Participant = m.ParticipantNode.Participant</i></p>	<p><i>Care Providers</i></p>	<p>This complicated invariant ensures that the URI of each <i>AuthorisationEndpoint</i> contains the hostname of a <i>MedMijnNode</i> that is from the same <i>Participant</i> that is also providing the relevant <i>CareproviderDatasevice</i>. Although the invariant refers to "each <i>AuthorisationEndpoint</i> of <i>z</i>" and to "each <i>ParticipantinRoleCareproviderDatasevice d</i> of <i>z</i>", there is only a single one of both of them for each <i>CareproviderDatasevice</i>. This is regulated by other invariants, but this invariant does want to be dependent on them.</p>	<p>non-local dependency</p>
--------------------------------------	---	------------------------------	--	-----------------------------

<p><i>CareproviderDataser vice</i></p>	<p>For each <i>CareproviderDataser vice z</i>, for each <i>TokenEndpoint t</i> of <i>z</i> and for each <i>ParticipantinRoleCareproviderDa taservice d</i> of <i>z</i> it is true that: there is a <i>MedMijnNode m</i> such that: <i>t.TokenEndpointuri</i> contains <i>m.ParticipantNode.Node.Hostna me</i> and <i>d.ParticipantinRole.Participant = m.ParticipantNode.Participant</i></p>	<p><i>Care Provi ders</i></p>	<p>This complicated invariant ensures that the URI of each token endpoint contains the hostname of a <i>MedMijnNode</i> that is from the same <i>Participant</i> who is also providing the relevant <i>CareproviderDataser vice</i>. Although the invariant refers to "each <i>ResourceEndpoint r</i> of <i>z</i>" and to "each <i>ParticipantinRoleCarep roviderDataser vice d</i> of <i>z</i>" there is only a single one of both of them for each <i>CareproviderDataser vice</i>. This is regulated by other invariants, but this invariant does want to be dependent on them.</p>	<p>non- local depe ndenc y</p>
--	--	---------------------------------------	--	--

<p><i>CareproviderDatasevice</i></p>	<p>For each <i>CareproviderDatasevice z</i>, for each <i>CareproviderDataseviceSystemrole zs</i> of <i>z</i>, for each <i>ResourceEndpoint r</i> of <i>z</i> and for each <i>ParticipantinRoleCareproviderDatasevice d</i> of <i>z</i> it is true that: there is a <i>MedMijNode m</i> such that: <i>r.ResourceEndpointuri</i> contains <i>m.ParticipantNode.Hostname</i> and <i>d.ParticipantinRole.Participant = m.ParticipantNode.Participant</i></p>	<p><i>Care Providers</i></p>	<p>This complicated invariant ensures that the URI of each resource endpoint contains the hostname of a <i>MedMijNode</i> that is from the same <i>Participant</i> who is also providing the relevant <i>CareproviderDatasevice</i>. Although the invariant refers to "each <i>TokenEndpoint a</i> of <i>z</i>" and to "each <i>ParticipantinRoleCareproviderDatasevice d</i> of <i>z</i>" there is only a single one of both of them for each <i>CareproviderDatasevice</i>. This is regulated by other invariants, but this invariant does want to be dependent on them.</p>	<p>non-local dependency</p>
<p><i>CareproviderDataseviceSystemrole</i></p>	<p>Each combination of a <i>CareproviderDatasevice</i> and a <i>System Role</i> has no more than a single <i>CareproviderDataseviceSystemrole</i>.</p>	<p><i>Care Providers</i></p>	<p>In this way the <i>OAuth Client</i> can - with the combination of a <i>Care Provider</i>, a <i>Data Service</i> and a <i>System Role</i> - find the <i>ResourceEndpoint</i> in the <i>Care Providers List</i>.</p>	<p>numerical relationship</p>
<p><i>CareproviderDataseviceSystemrole</i></p>	<p><i>CareproviderDataseviceSystemrole.Systemrole.Organisationalrole = CareproviderOrganisationalrole</i></p>	<p><i>Care Providers</i></p>	<p><i>Care Providers</i> can only provide <i>System Roles</i> that are intended for <i>Care Providers</i>.</p>	<p>non-local dependency</p>

<i>CareproviderDataser viceSystemrole</i>	Each <i>CareproviderDataser viceSystemrole</i> has precisely one <i>ResourceEndpoint</i> .	<i>Care Provi ders</i>	In this way the <i>OAuth Client</i> can - with the combination of a <i>Care Provider</i> , a <i>Data Service</i> and a <i>System Role</i> - find the <i>ResourceEndpoint</i> in the <i>Care Providers List</i> .	nume rical relati onshi p
<i>CareuserBusinessrol e</i>	There is precisely one instance of this.	<i>Roles</i>	This is a 'loner' in the model.	nume rical relati onshi p

Basic classes

Basic class	Definition
<i>Date</i>	In accordance with the type <code>xs:date</code> , as specified in XML schema 1.0 .
<i>Endpointport</i>	See the addressing responsibilities on the page Data and performance in UCI Compile and UCI Share .
<i>Endpointpath</i>	See the addressing responsibilities on the page Data and performance in UCI Compile and UCI Share .
<i>DataserviceId</i>	String of at least one, and a maximum of 30, character(s).
<i>Dataservicename</i>	String of at least three, and a maximum of 50, characters.
<i>Hostname</i>	See the addressing responsibilities on the page Data and performance in UCI Compile and UCI Share .
<i>Information Standard name</i>	String of at least three, and a maximum of 50, characters.
<i>OAuthclientOrganisationname</i>	In accordance with the applicable OAuthclient name policy .
<i>Systemrolecode</i>	String of at least one, and a maximum of 30, character(s).
<i>Transactionname</i>	String of at least three, and a maximum of 50, characters.

<i>Version number</i>	One or more number sequences, each consisting of one or more digits 0 to 9 inclusive, separated by a full stop
<i>Depictionname</i>	String of at least three, and a maximum of 50, characters.
<i>Careprovidername</i>	In accordance with applicable CareProvidersnamepolicy.