Architecture and technical specifications

Explanatory Notes

An essential part of the MedMij Framework relates to the responsibilities that the participants in the Framework have, each with in its own role, during the actual provision of the data transfer between the individual's domain and the care provider's domain. These responsibilities are included in the architecture and the technical specifications of the MedMij Framework, which are elaborated on in these pages. These responsibilities have been organised into a number of abstraction levels, inspired by the Nictiz interoperability model .

To start with, participants must together ensure that certain business processes take place between the individual's domain and the care provider's domain. These business processes relate to the compiling and sharing of care information and health information. On this abstraction level, there is not yet an automated handling of these processes but the responsibilities have only been formulated in terms of the content of these processes and of the health information that is handled in them. At this abstraction level, the process layer and information layer from the Nictiz interoperability model are combined in a single layer: Processes and Information.

At the next abstraction level , the Application layer, it is discussed that, and how, these business processes with the related health information that is handled therein must be performed, with the different roles working together to this end. It is the most complex layer, which has two special sublayers: one for the authentication of the *individual* and one for this authorisation of the information transfer.

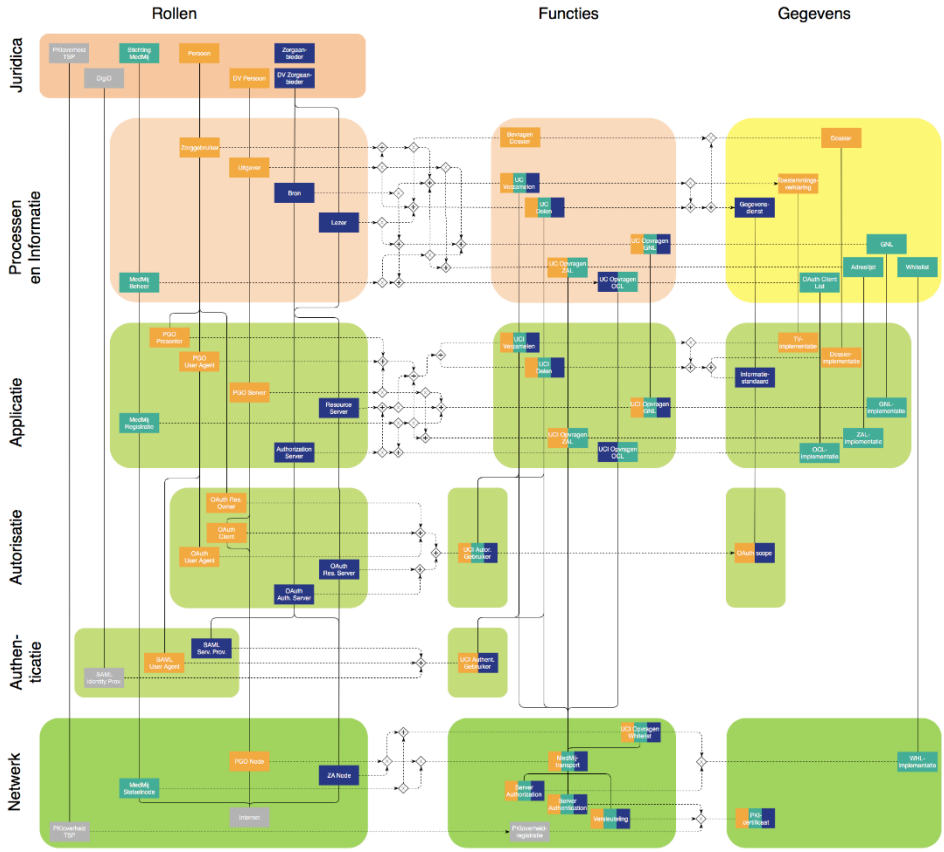
At the lowest abstraction level, the Network layer, the responsibilities are included in the area of the network infrastructure.

The diagram below identifies these abstraction levels. On each layer, the architectural elements are stated that are needed in the layer in question, along with their reciprocal links within and between the layers. The diagram on this page is not intended to specify the relationship between all details in one go. Instead, this is done step by step on the pages that belong to the specification layers; on the page for each layer, the cutaway of the diagram that matches that layer is repeated and dealt with. On this page, the diagram is only intended to fulfil two roles:

- to give a rough overview of the layers (and columns) of the architecture of the MedMij Framework, and
- to provide an index that allows the user to quickly find the layer for an architectural detail where this detail is discussed.

The explanation under the diagram also discusses what the columns, colours and lines in the diagram mean and prepares the reader for the viewing of the detailed pages.

The complete architecture of the MedMij Framework is provided in the figure below.



Explanatory Notes

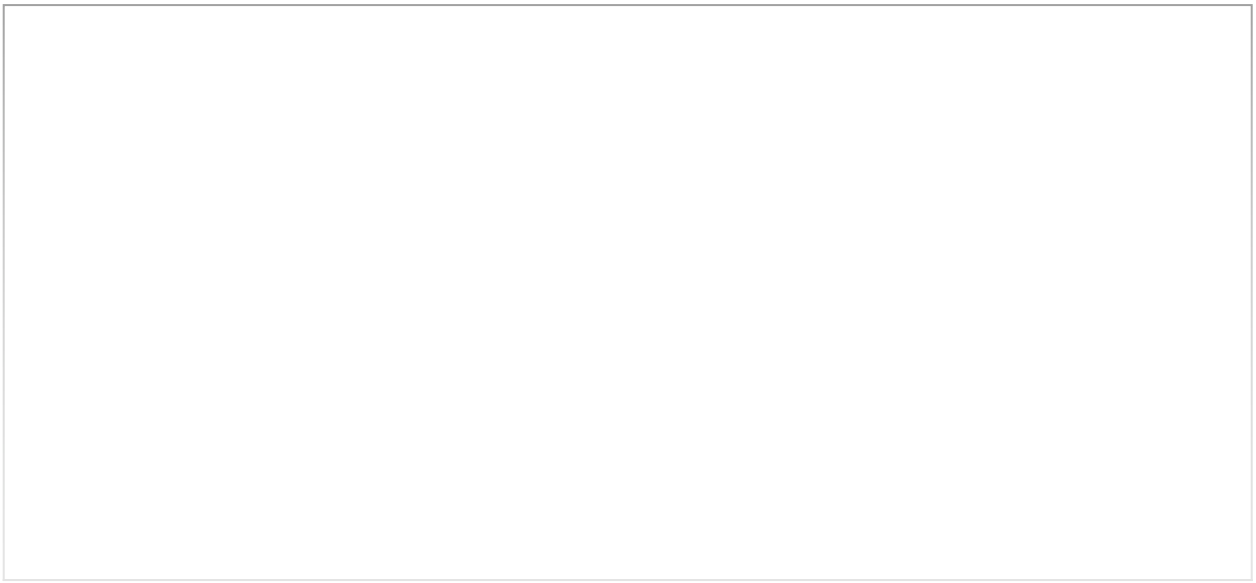
Three columns have also been used in the architecture, namely: roles, functions and data. In each layer there are roles specific to that layer that perform functions specific to that layer with the help of data specific to that layer. This is precisely the reason why the process layer and the information layer from Nictiz's interoperability model are combined into a single layer that also includes a roles column. Because it concerns architecture for a framework, not for a solution, the roles column plays a key role in the coherence of the entire architecture. Roles are bundles of responsibilities, not components. These responsibilities are linked to functions to be performed (second column), which in turn make use of data (third column).

Furthermore, the application layer has been refined by isolating two separate sublayers: an authorisation layer and an authentication layer. This is because for both of these two issues, standards are used that have their own roles structure, with which an explicit connection must be realised. Additionally, it is possible in this way to give the agreements that specifically arise from the design of this standard a recognizable and manageable place.

An additional level has been placed above the processes-and-information layer, namely: Legal. This layer only has the roles column, not the other two columns. The latter are namely

discussed in the pageAgreements and legal relationships. This layer is only intended for the linking - role by role - of the architecture with the legal part of the MedMij Framework, so that it is made clear which architectural and technical responsibilities are linked to which legal roles.

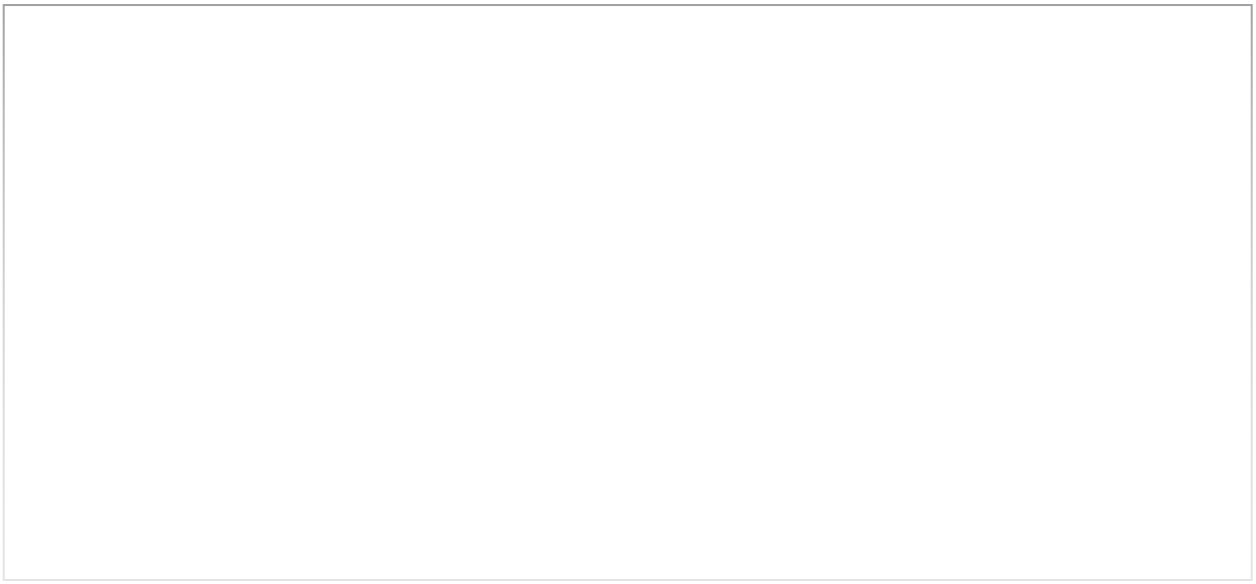
In the authentication layer, it is not necessary to make further agreements about data. The specifications of DigiD's SAML interface can be entirely relied on for this. This is why this column is lacking in the architecture.



The colours of the large areas correspond to the colours that Nictiz assigns to the relevant architectural aspects in itsinteroperability model. The colours of the architectural elements (the small rectangles) indicate in which domain the relevant architectural element is placed. First of all, MedMij's corporate identity has been retained, so that:

- orange represents the individual's domain;
- blue represents the care provider's domain, and
- green represents the MedMij domain.

The grey colour represents external roles used by the MedMij Framework. Where multiple colours are combined, this indicates that the domains work together in the relevant architectural element.



The vertical lines in the architecture connect the roles, functions and data between the different layers.

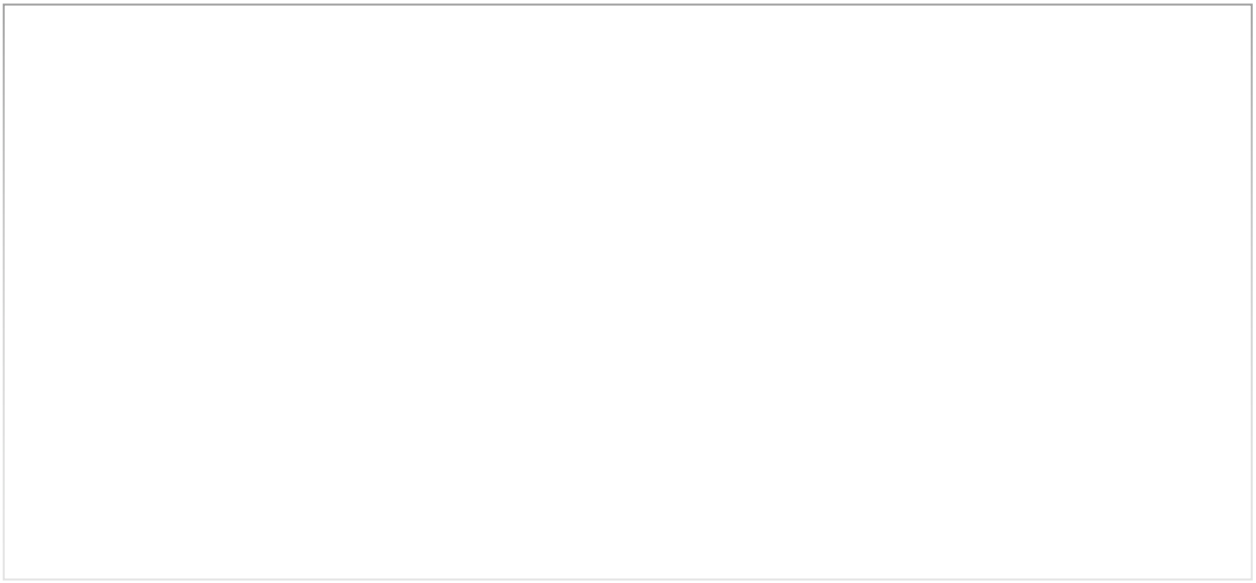
In the MedMij Framework, the roles are sets of responsibilities that belong together. They occur on each layer of the architecture, from the Legal layer via the Processes-and-Information layer and the Application layer up to and including the Network layer. The roles are linked to each other between two adjacent architectural layers. Each role on the one layer is paired with one or more roles on the layer below. In this way, the role links form the backbone of the MedMij Framework architecture.

A role is emphatically not a component or system. While it is true that many roles are realised by components and systems, but the precise way in which this is done, and how much components architecture or system architecture is used for this is for the *Service Provider* to decide, as long as the latter plays its roles properly on all layers, that is to say, carries the responsibilities for these roles. In this way, Service Providers, in both domains, are given all the scope they need to choose a business model as they see fit, within which subcontractors are given all the scope they need, as long as the ultimate responsibility for the MedMij Framework continues to lie inalienably with the *service provider*.

For a *Service Provider* , there must in other respects be maximum freedom to arrange a single role on the one level with multiple roles on the level below it. However, conversely it must continue to be clear, on all layers, that a single *service provider* is responsible for each role. In other words, multiple roles cannot be depicted on a single lower one. It is indeed possible for multiple roles to be realised by a joint system, as long as their individual ultimate responsibilities remain intact.

In other words, in general a single role on the higher architectural layer will be fleshed out with one or more affiliated roles in the layer below. Conversely, however, a single role on the lower

architectural layer belongs to a single affiliated role on the higher layer. Because the Nodes at Network level are identified using a hostname, it can always be read from the logging at Network level which participant is responsible for which event.



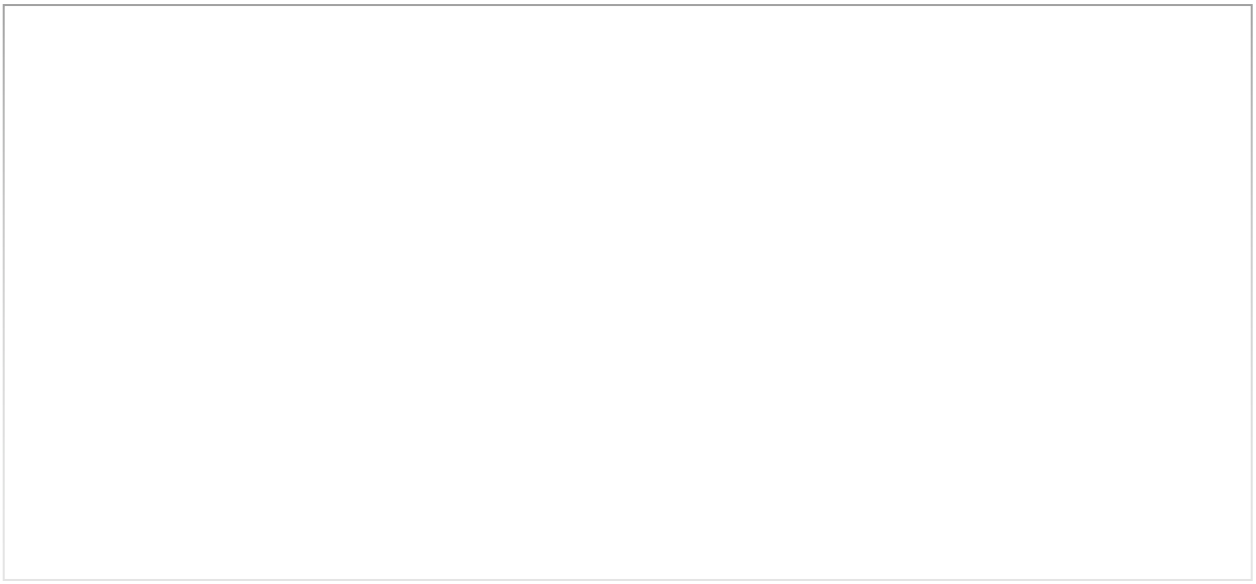
The horizontal dotted lines indicate which roles perform which functions, and respectively which functions use which data. To prevent a confusing tangle of dotted lines, the figure uses joins and splits. Joins and splits are indicated by small diamond shapes. A join (juncture) is characterised by multiple incoming arrows and one outgoing arrow and a split (division) by one incoming and multiple outgoing arrows.

Two characters are used in the small diamond shapes:

- A multiplication sign stands for exclusive, which means that only one of the incoming arrows (at joins) or outgoing arrows (at splits) are simultaneous.
- A plus sign stands for inclusive, which means that all incoming arrows (at joins) or outgoing arrows (at splits) are always simultaneous.

For example, in the layer *Processes and Information*, the *MedMij Management* role is involved in:

- six use cases, namely: *UC Compiling*, *UC Sharing*, *UC Request ZAL*, *UC Request OCL*, *UC Request WHL* and *UC Request GNL* but not simultaneously (i.e. are exclusive).
- in the use case *UC Request ZAL*, simultaneously (inclusive) with the role *Publisher*.



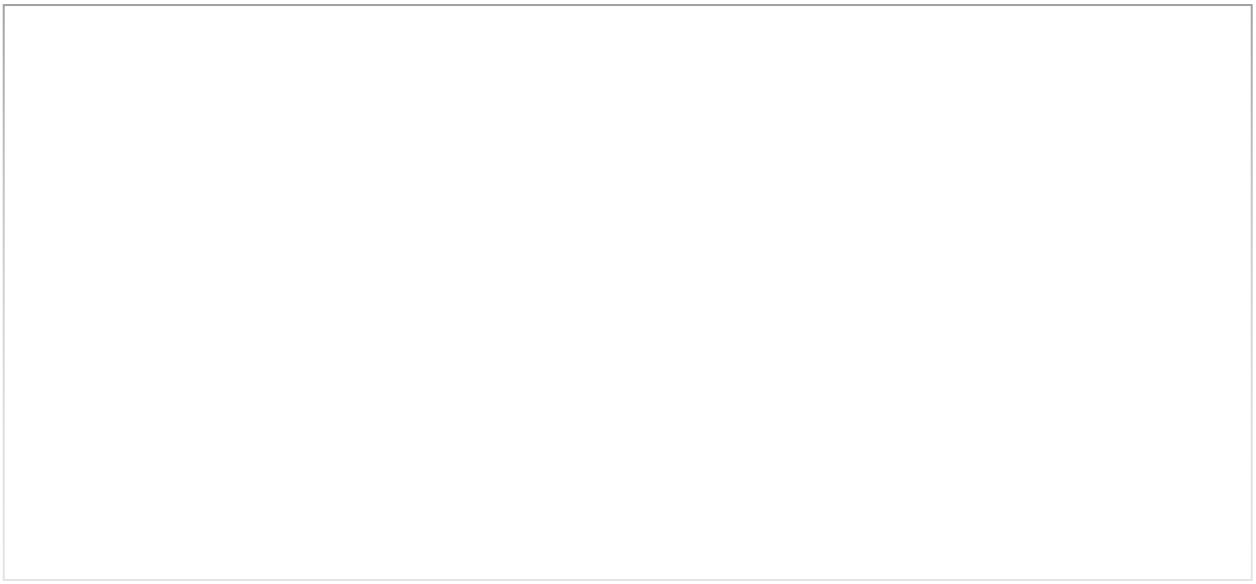
For each level, the agreements are elaborated on in a separate page:

- Legal
- Processes and Information
- Application, including Authentication and Authorisation
- Network

Each page may have subpages for subissues. These agreements always consist of:

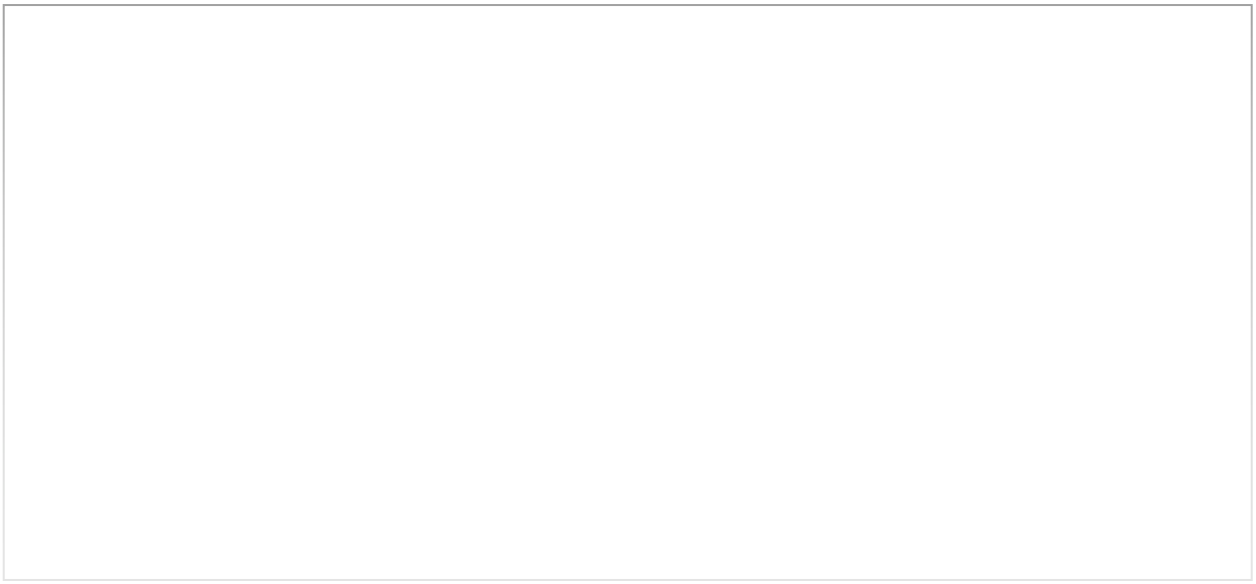
- the identification of the roles on this layer/sublayer and the tie(s) from these roles to the roles on the layer above;
- the responsibilities that the roles on this layer/sublayer have in performing certain functions with certain data.

A separate page Information Models, with three subpages, specifies the conceptual structure of (all or part of) the concept device of the architecture of the MedMij Framework and translates this via logical models into technical models of a number of components. In this way, the interoperability of the MedMij network at a technical level is guaranteed.



In many cases, reference to a specification is made in the responsibilities. This can be a use case specified specifically for MedMij, for example, but is often also a standard, especially for information. The specification will not be written out in detail in the responsibility itself; it will instead be referred to. For example, there is no need to always adjust the responsibility for any detail adjustments in the specification. This would, definitely with standard specifications, result in an undesirable management burden on the framework.

In the first instance, as a rule the roles and responsibilities are concisely and thoroughly formulated. Only in the second instance are they elaborated upon. Accordingly, the approach taken is not that of providing a narrative explanation of the system but to provide a set of agreements, article by article. This makes the architecture suitable for use as an extension of the participant's agreement. The very first question to ask is: *What is the agreement?* In the second instance, questions such as: *Why was this chosen?* and*What does this agreement mean?* are important.



Where in the description of the architecture, the roles and responsibilities contained therein and the explanatory notes for them, use a name to refer to the architectural components, as they occur in the diagram above, the name is written *in italics* and with a *Capital letter* . This also applies for the path expressions in the invariants for the Information Models. Variables in the path expressions are written in *italics*, too, but start with a lower-case letter.

Some architecture components are also represented by a class, attribute, element or type in the Information Models. Because the spelling of the names in the Information Models is more formal, the naming used in these Models may differ a bit from that used in the rest of the architecture, namely in the use of spaces and capital letters. In the Information Models, all names begin with a capital letter. In addition, capital letters may appear in the middle of a name if, but only if, the remaining part of the name there also appears as a separate name.

Technical code fragments are quoted in monospace.