Virtual Coaching Activities for Rehabilitation in Elderly

Call: H2020-SC1-2016-2017

Grant Agreement Number: 769807

Deliverable

D6.6 Reference Implementation Guide

Deliverable type:	Other
WP number and title:	WP6: Pathway System – Semantics driven Pathways Development and Execution Services
Dissemination level:	Public
Due date:	Month 42 – 28 February 2021
Lead beneficiary:	MYSPHERA
Lead author:	Álvaro Martínez (MYS)
Collaborators:	Kai Gand (TUD), Alexandru Vulpe (iSprint), George Matikas (iSprint), Luc Nicolas (ETHEL), Sofoklis Kyriazakos (AU)
Internal Reviewers:	Hannes Schlieter (TUD)

Document history

Version	Date	Author/Editor	Description
0.1	19.12.2020	Alvaro Martinez	TOC proposal
0.2	18.02.2021	Alexandru Vulpe, George Matikas	Inputs to section 3.7
0.3	18.02.2021	Karl Kreiner, Patrick Philipp	Inputs to section 3.2, 3.3
0.4	24.02.2021	Álvaro Martinez	Added the ontology overview
0.5	26.2.2021	Sofoklis Kyriazakos	Added input about the evaluation methodology of the reference profiles
0.6	26.02.2021	Kai Gand	Refined section 4.
0.7	26.02.2021	Alexandru Vulpe	Auth model in section 3.7.1
1.0	26.02.2021	Álvaro Martinez	Final version

Table of Contents

List of Figures

List of Tables

List of Acronym

1 Executive Summary

This deliverable contains the information resulting from task 6.5, 'Reference implementation guide', which reassembles results from deliverables 6.1 to 6.5, as well as some results from WP4, WP3 and WP7 to provide a reference implementation system.

We describe the components that make up the reference system from a technical and functional perspective. Also, we provide an overview of the different configurations that can be composed using a cherry-picking strategy. This provides a guideline on how these building blocks can be combined to build up different setups offering a different set of functionalities. We summarize the components used in defined profiles that could be reasonably built. These are archetypical implementation scenarios targeted at users interested in different kinds and extents of "pathway modelling" use cases.

The reference system is a subset of the whole vCare Coaching Platform, focussing only on the core pathways functionalities of *rehabilitation care pathway template* creation, instantiation for a specific user, instance execution, and instance adaptation. Additionally, WP7 provides the data middleware and the authentication/authorization framework. To give this reference architecture a SaaS perspective, the vCaaS API is added on top of the reference system to uniformly abstract the access to the different services.

In general, this deliverable continues the work presented in Deliverable 6.5. In addition to the test system, we want to add reflections on possible implementation scenarios for the pathway-related sub-functionalities of the overall solution. This is to allow interested parties to build up on pathway services as the core governing instrument of the vCare solution. We hope the current reference implementation guide could, in turn, spread the idea of clinical pathways as central element of the processual control of eHealth measures in general.

2 INTRODUCTION

This deliverable contains the information resulting from task 6.5 - reference implementation guide -which reuses the results from deliverables 6.1 to 6.5, as well as some results from WP4, WP3 and WP7 to provide a reference implementation system (see figure 1 for an overview of the primary relations).

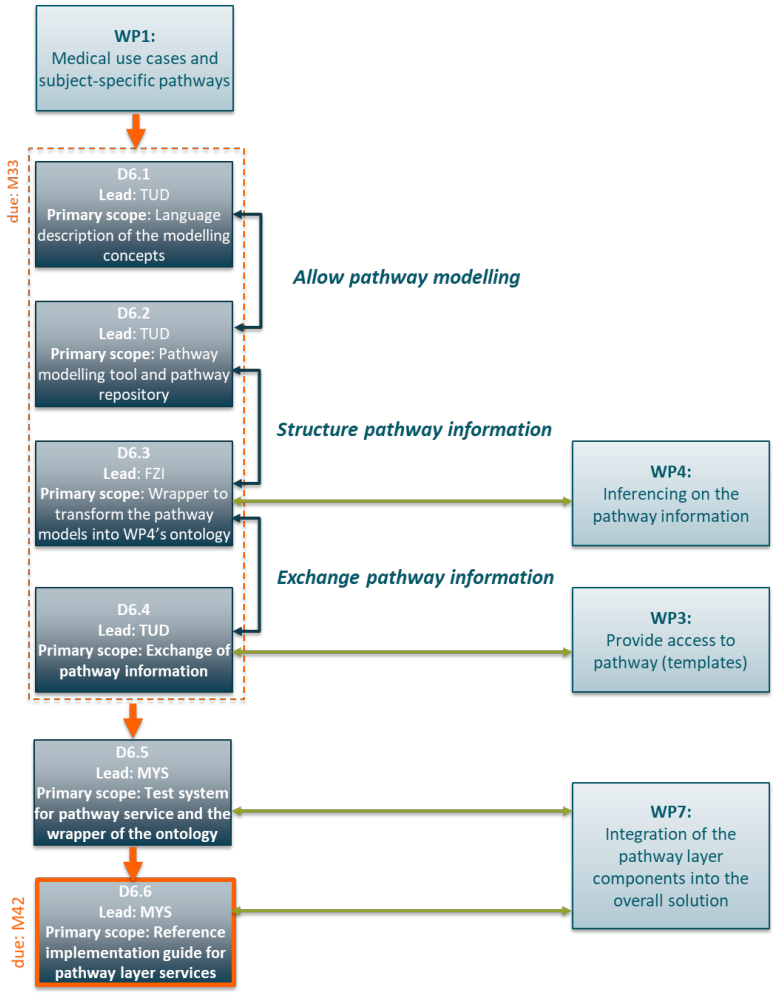

Figure 1 - Relationship between deliverables in WP6

The reference system is a subset of the whole vCare Coaching Platform, keeping only the core pathways functionalities:

- Template creation covered by results from WP6.
- Instantiation for a specific user, covered by results from WP3.
- Instance execution covered by WP4.
- Instance adaptation covered also by WP4.

Additionally, WP7 is providing the minimal infrastructure to enable the interactions of their results. In more detail, WP7 provides the data middleware and the authentication/authorization framework.

To give this reference architecture a SaaS perspective, the vCaaS API is added on top of the reference system to uniformly abstract the access to the different services.

In general, this deliverable continues the work presented in Deliverable 6.5[1], a test system demonstrating the full pathway cycle, from design to execution for a specific patient. The main goal of that demonstrator was to show to a wider audience the concept envisaged in vCare project in an easy and friendly manner. During the internal project validations, Living Lab phase and Pilot phase, refinements to the existing pathways as well as new ones are being added and tested to the catalogue of available pathways.

[1] https://pathways.vcare-project.eu/

3 REFERENCE SYSTEM

The reference system that we propose in this guide is a combination of the results already provided in previous deliverables in this work package, working in collaboration with other blocks provided in the scope of vCare project that focus on the pathway-related elements. Figure 2 shows the different building blocks that are required to implement the reference system.

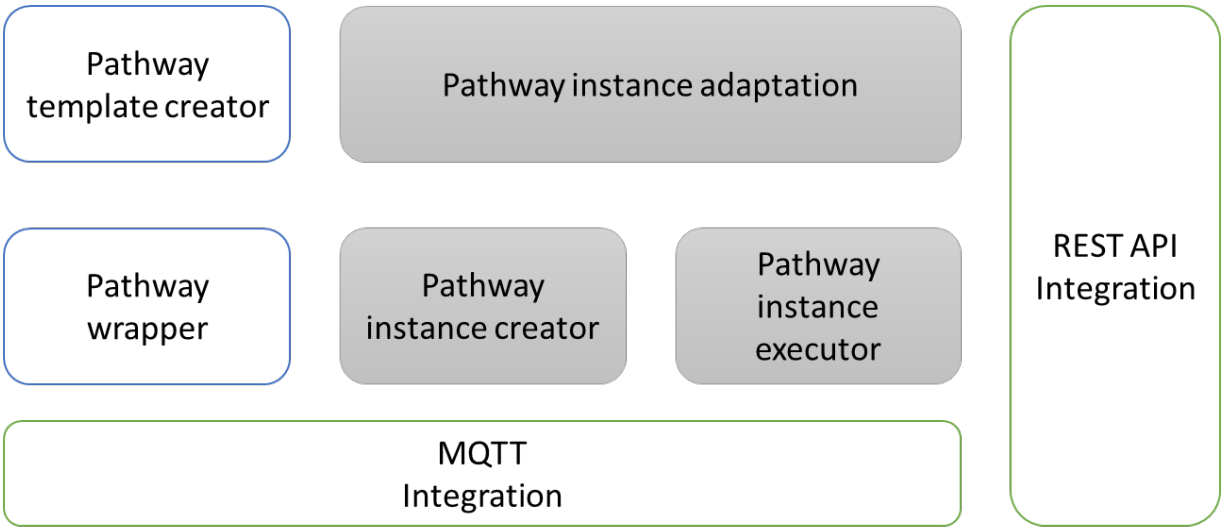

Figure 2 - Reference system

The reference system contains 7 main elements:

1) The *pathway template creator* (combination of a graphical modeler tool and a data repository), which is the building block in charge of creating and editing pathways. In our reference implementation, the graphical part makes use of BPMN (Business Process Model and Notation) notation and the repository conforms to FHIR[2] R4.
2) A *pathway wrapper*, being the element that transforms the FHIR model into a domain specific model. Custom implementations of this block allow plugin into the reference system services that work with models that are not strictly compliant to FHIR.
3) A *pathway instance creator*. The building block that can receive a pathway and tailor the contained actions to a specific individual.
4) The *pathway instance executor*, being the component that executes the different actions that are defined in a specific pathway instance.
5) The *pathway instance adaptor*, which complements the executor in adapting the running instance based on the feedback received to the actions already executed.
6) The *data integration broker*, which is an optional component to link the previous blocks to each other, decoupling the service-to-service communication through a data broker.
7) The *service integration API*, which is the block that allows abstracting the whole reference system as a service to a third party.

[2] Fast Healthcare Interoperability Resources (FHIR) https://www.hl7.org/fhir/

In the reference system, we are not taking into account non-functional requisites such as security, which is an additional layer on top of this reference system.

3.1 PATHWAY MODELLER

The vCare project's Pathways editor and Template management tool has been designed as an integral part of the vCare's ecosystem addressing primary functions of the overall solution. In terms of *Care Pathway Template Management*, physicians or other authorized roles are enabled to administrate and browse the templates of the rehabilitation care pathway[3]. Physicians can define new templates and update existing templates (*Care Pathway Template Editing*). The templates contain the typical procedure a patient with a disease should conduct as part of his/her treatment. The templates are used when a physician prescribes a treatment to a patient (*Care Pathway Instantiation*; done after assigning a template to a particular patient within the Professional Portal (as designed in the course of WP3)). This way, using inclusion and exclusion criteria, the physician selects and assigns a proper care program to the patient. During the assignment, the physician can set particular treatment parameters (i.e., target values to be reached by the patient for particular observation features or durations treatment activity should take) for the patient's specifics. Subsequently, the pathway information will be wrapped, its format will be transformed, it will be stored and processed in the ontology (see D6.3/WP4). This allows further functions as basis of inference mechanisms like *Agenda Management*, *Context Analysis,* (Automatic) *Care Pathway Adaptation/Personalisation,* based on the ontology. Along these lines, also the particular *Coaching* in terms of the execution of (coaching) services such as *rehabilitation games* or *e-learning activities* (see D5.1 for the initial description of the coaching services and the further activities and more detailed technical descriptions in D5.2[4] and D5.3[5] and in the course of WP5 at all) is assigned to the patient. The *Context Integration*'s sensors and interfaces are, in turn, the technical basis to monitor the patient when conducting the treatment measures.

As outlined in D6.2, the basal Pathway Template System is a combination of two interlinked components, the ***modeller,*** and the ***repository***:

- The web-based **model editor** allows to graphically overview and work on the care program. Here, BPMN[6] is used as graphical conceptual modelling language for clinical pathways. The bpmn.io[7] library is reused adding further modelling concepts according to the depiction prerequisites stated as relevant in the clinical domain (see D6.1).

[3] The pathway modelling is not limited to *rehabilitation* care pathways. This is just the project's focus. In terms of a generalisable solution in principle any kind of care procedure, inpatient or outpatient, could be modelled.

[4] https://vcare-project.eu/wp-content/uploads/2020/07/vCare_D5.2_new_0.24-final-2.pdf

[5] https://vcare-project.eu/wp-content/uploads/2021/02/D5.3-coaching-services-description_v1.0_final.pdf

[6] https://www.omg.org/spec/BPMN/2.0/

[7] https://github.com/bpmn-io

Template models itself are modified using either the provided modelling toolbar or the properties panel.

- The template **repository** browser provides an overview of available templates. It provides some basic functionalities to create new or edit existing template models and their metadata.

The HL7 FHIR-based implementation[8] enables the database persistence and REST-based access to the templates. To refine the generic HL7 FHIR standard data schema a profile[9] for vCare's rehabilitation care templates[10] is in place. For this, the FHIR PlanDefinition[11] as base resource is utilised (see D6.4[12]). The profile introduces specifications for mapping a template modelled with the editor (BPMN-XML) to the corresponding persisted FHIR resource (FHIR-XML).

3.1.1 INTERFACE

Finding a sound technique for representing treatment plans is a major issue, requiring modelling several elements such as actions, patient's information, decisions to be taken, constraints between activities, and temporal properties. Clinical pathways are a central instrument for the structured description of the clinically relevant procedures a patient should follow and serve as a baseline for the following inference mechanisms, scheduling and personalisation of the treatment. Since many years, there is a trend towards the usage of the *BPMN* for depicting business processes due to its broad acceptance in research and industry[13]. In principle clinical guidelines are quite similar to business processes when it comes to managing the correct way for the patient's situation at every instant, for every condition. This is the way a process modelling languages like BPMN work as well and why it is used for the Pathway Modeller as well. Moreover, BPMN is defined as official ISO standard. The BPMN provides a well-defined meta model, which is formulated by the most prevalent meta modelling language MOF. This especially facilitates model exchangeability, tool integration and the reuse of existing modelling. BPMN partly supports the derivation of computer-interpretable workflow models and is appropriately embedded within an execution-oriented framework which is promising in regard of clinical decision support systems. Moreover, BPMN provides a large set of generic process modelling concepts, which implicate great opportunities in regard of expressiveness. Additionally, BPMN provides an explicitly defined, lightweight extension mechanism for domain-specific extension and adaptation.

Motivated by the stated benefits of BPMN, the standard modelling language has been extended to be fully able to express the relevant clinical contexts and to satisfy the requirements regarding the representation of the pathways both visual and informative.

---

[8] https://hapifhir.io
[9] https://www.hl7.org/fhir/profiling.html
[10] https://simplifier.net/helict/ModeledPlanDefinition
[11] http://hl7.org/fhir/r4/plandefinition.html
[12] D6.4's annex also contains an exemplary implementation of the PlanDefinition and CarePlan as the two resources used in the vCare FHIR profile to illustrate how this looks like. See: https://vcare-project.eu/wp-content/uploads/2021/02/vCare_D6.4_final.pdf
[13] Kirchner, K., Malessa, C., Scheuerlein, H., & Settmacher, U. (2014). *Experience from collaborative modeling of clinical pathways*. 13–24.

To start the graphical procedure modelling one needs to access the “Template Gallery” to overview the available and create new rehabilitation care pathway templates. The existing ones can be deleted, opened, or edited with regard to their meta data or the actual process model content. The elements available for modelling (representing the desired or possible treatment procedure from the clinical point of view) can be chosen on basic modelling panel (see figure 3). It is also possible to alter already placed nodes or to further link elements. The templates can be stored and exported. Also, the properties of single pathway template elements can be altered directly in the modeler (e.g., the configuration of a time-based features, like foreseen frequencies of an activity).

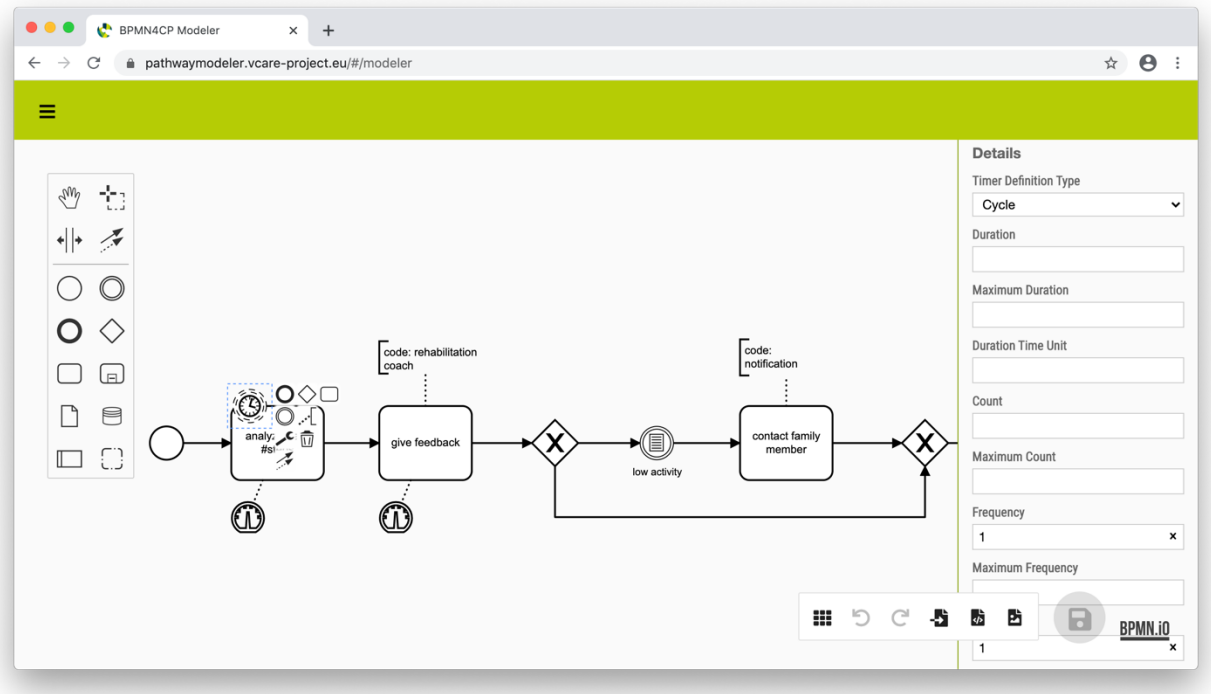

Figure 3 - Screenshot of the vCare-Modeler showing the creation of an exemplary and simplified rehabilitation care pathway template for a cardiological case.

The basic modelling elements of the BPMN also usable here are (figure 4):

- Pool/lane: contains the model elements from the viewpoint of an organisational role (caregiver/patient)
- Start/End Event: defines where a process starts resp. ends.
- Activity: allow modelling the actual tasks/actions to be performed
- Sub-process: allows reusability and grouping.
- Parallel Gateway: all the following process flows have to be executed.
- Decision Point: only one of the following process flows is being executed based on a decision.
- Timer Event: is triggered by a defined timer, which can be a date, a duration or a cycle.
- Conditional Event: is triggered if a given condition is evaluated as true.
- Events can stand alone or be attached to a task/sub-process.
- When attached to a task, events can be either interrupting or non-interrupting.

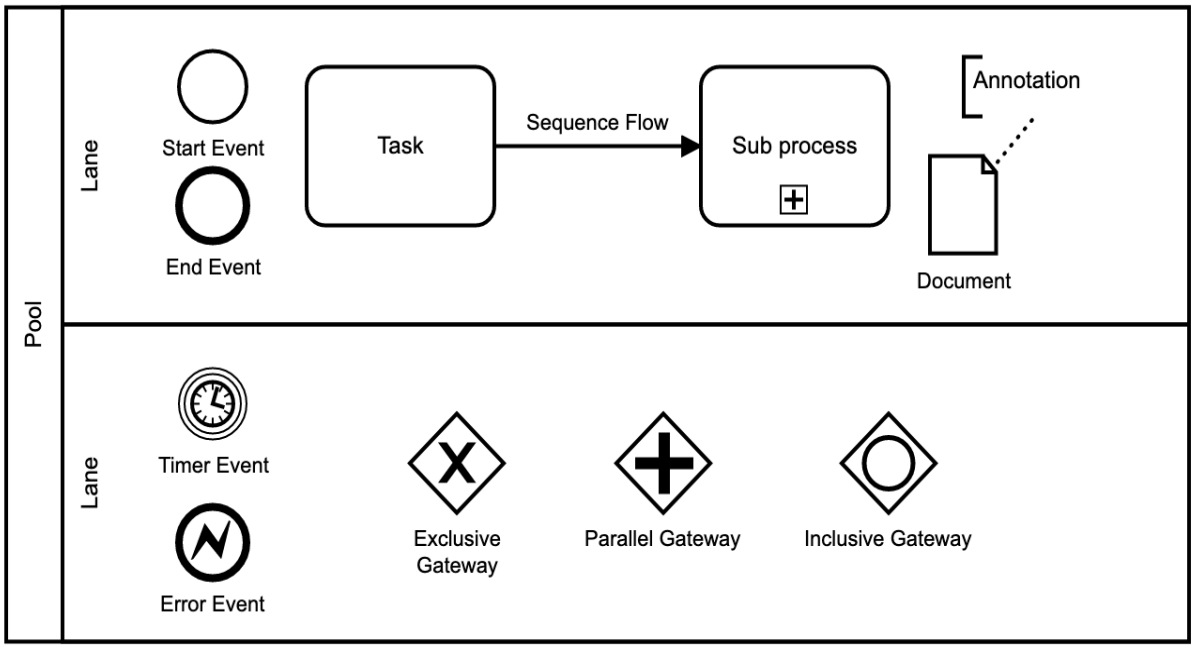

Figure 4 - Basic BPMN elements

Combining these elements factual represents the care procedure or pathway modelling to create the pathway templates.

3.1.2 REST API

The endpoint for the REST-based access is provided by a web servlet. It provides the basic CRUD-operations for pathway templates (FHIR PlanDefintion). This is also the access point to gather the care pathway information for the components that further process, specify, and instantiate the templates. Firstly, the template information allows the assignment of a particular pathway instance (a FHIR CarePlan[14]) by the physician for a particular patient. Secondly, the pathway wrapper can then also request this information to transfer it into the ontology's structure in order to perform reasoning queries and personalization based on reinforcement learning. The particular pathway instance can then automatically be updated to adapt to the patient's needs, status, performance, or preferences.

Overall, a PlanDefinition is a pre-defined group of actions to be taken circumstances flexible enough to be used to represent a variety of workflows, as well as clinical decision support. PlanDefinitions hierarchically structure groups of action definitions (activities to be performed), and each group defines additional behaviour, relationships, and applicable conditions between the actions in the overall definition. Moreover, it is not only possible to describe what care procedures should take place but also when and whether/under what conditions this should be done. This again fits well to the conditional and time-related elements as identified in the clinical pathways. Based on a PlanDefinition, a specific CarePlan can be generated for a patient. During this transformation step each element is processed and dynamic patient-specific parameters can be set. Furthermore, optional elements of the PlanDefinition can be selected. The subsequent process of applying a PlanDefinition to a particular context typically

[14] See: https://www.hl7.org/fhir/careplan.html; Details for the vCare-specific adaptations can be found in D6.4: https://vcare-project.eu/wp-content/uploads/2021/02/vCare_D6.4_final.pdf

produces request resources representing the actions that should be performed, relating to a grouping within a CarePlan to capture relationships between the resulting request resources.

Figure 5 shows the HAPI FHIR [15] server's interface for accessing the details of the PlanDefinition resource. This is the technical representation of the graphically modelled rehabilitation care pathway template. Here, options how to encode or summarise the resource can be set or the processing/response details of the server or results of queries can be displayed (in figure 5, the PlanDefinition details are shown as JSON code). Furthermore, FHIR relies on the standardization of resource structures and interfaces. This may be considered a violation of REST principles[17] but is key to ensuring consistent interoperability across diverse systems.

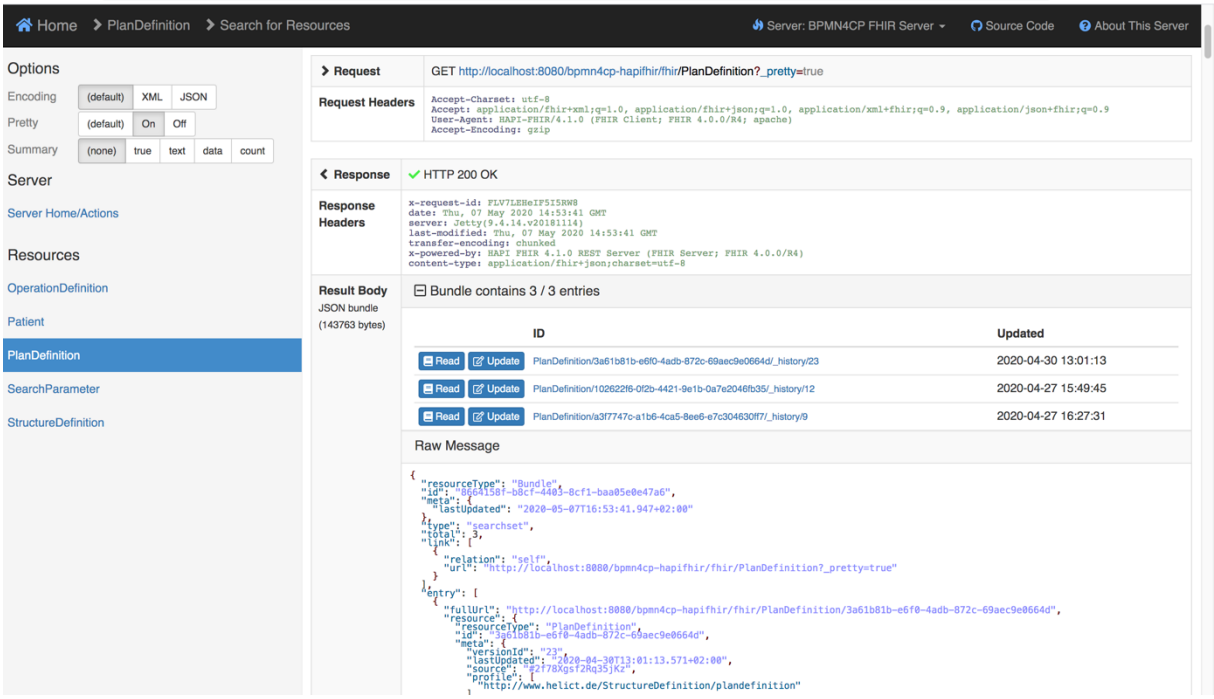

Figure 5 - Screenshot showing the HAPI FHIR server's interface accessing the PlanDefinition details.

Two principal information flow scenarios and related interaction modes can be differentiated to better protect system internals (sensitive data):

(1) The altering access to the rehabilitation care pathway templates by means of the Pathways editor and Template management tool. This "internal" servlet is to implement the changeability of the pathway templates for users with respective authorisation (i.e., the physician that re-assesses the rehabilitation care pathway template's

[15] See: https://hapifhir.io
[17] See: https://www.hl7.org/fhir/http.html

appropriateness). Primary allowed operation interactions: *read*[18], *search*[19], *history*[20], *create*[21], *update*[22], *delete*[23]

(2) The flow of pathway information to system components external to the pathway layer. This particularly relates to the professional portal and the knowledge layer where a further processing of the rehabilitation care pathway information takes place. Optionally, further external services (following the as-a-Service concept) could be connected as well. Here, only a consuming access is foreseen. Exceptions could be made to access the operations when leveraging complex authentication and authorization frameworks (as offered by Keycloak) and thus have no risk of unduly compromising the pathway information. Primary allowed operation interactions: *read*[24], *search*[25], *history*[26]

3.2 PATHWAY SEMANTIC WRAPPER

The semantic wrapper for clinical pathways has been thoroughly introduced in D6.3. It maps the elements of a pathway, defined as FHIR Plan Definition, to the vCare knowledge layer to enable (i) reasoning activities to personalize the pathway for a particular patient and (ii) trigger the respective activities via the vCare Coaching Services provided by AIT.

The implementation of the wrapper expects files in well-defined JSON which serves as a data-exchange format between the clinical pathways representation format as defined in D6.4[27] and the semantic wrapper. During serialization, all information about the object and its attributes is preserved. In concrete terms, the information is stored during serialization via so-called dictionaries.

All elements of the FHIR Plan Definition are lifted to a semantic representation based on the FHIR standard, which defines an interface to the Resource Description Framework (RDF). The key element of the mapping is the Request Groups [28], i.e., the definitions of the planned actions for a patient. The semantic wrapper maps each action attribute, such as its id, the planned weekly frequency, or the defined conditions, to the knowledge layer by parsing the available JSON representation, iterating through the available elements and creating and persisting novel semantic triples (i.e., <subject, predicate, object>-pairs).

A conceptual overview of the final version of the vCare ontology schema is presented figure 6.

[18] See: https://www.hl7.org/fhir/http.html#read
[19] See: https://www.hl7.org/fhir/plandefinition.html#search
[20] See: https://www.hl7.org/fhir/http.html#history
[21] See: https://www.hl7.org/fhir/http.html#create
[22] See: https://www.hl7.org/fhir/http.html#update
[23] See: https://www.hl7.org/fhir/http.html#delete
[24] See: https://www.hl7.org/fhir/http.html#read
[25] See: https://www.hl7.org/fhir/plandefinition.html#search
[26] See: https://www.hl7.org/fhir/http.html#history
[27] See: https://vcare-project.eu/wp-content/uploads/2021/02/vCare_D6.4_final.pdf
[28] See: https://www.hl7.org/fhir/requestgroup.html

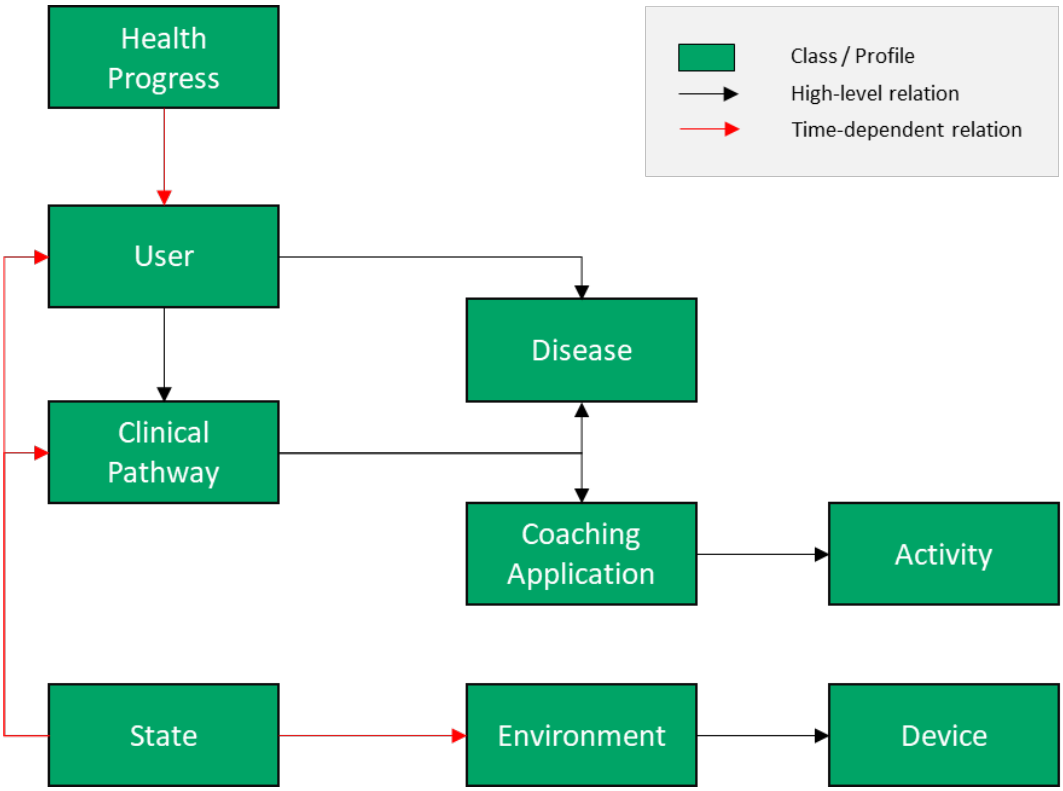

Figure 6 - vCare Ontology Schema Overview

The overview illustrates the core vCare classes and their high-level relationships with the aim to integrate all available vCare-related information. The green rectangles represent classes and profiles, which are connected via their relationships. Black arrows represent standard, high-level relationships between classes, where the direction of the arrow implies a dependency. Red arrows imply a dependency on time, i.e., we require timestamps in order to reassemble "slices" of the respective profiles.

The classes of the vCare ontology schema can be directly mapped to vCare profiles, where the clinical pathway class maps to the treatment plan profile. The state class represents a novel profile, which is essential for integrating time-dependent information needed for analysis and recommendation algorithms.

They define the required structure of the individual vCare profiles, such that it becomes explicit which information in which format has to be entered or generated. In addition to the structure, the identification of concepts and their instanced via URIs (i.e., unique resource identifiers) enables to link vCare knowledge to external ontologies in order to enable mutual enrichment.

Complete ontology details are available at D4.3[29].

3.3 PATHWAY INSTANCE CREATOR

The pathway is instantiated through a clinical professional through a dedicated user interface, the professional portal. The clinician enrols a patient for coaching services by selecting a top-level pathway (e.g., Heart failure). The pathway is then transformed into a FHIR-compatible

[29] https://vcare-project.eu/wp-content/uploads/2021/02/vCare_D4_3_Baseline_ontology_final_version.pdf

care plan for the patient registered. The clinician can then further customize the care plan for the patient on an activity level, whereas individual activities can be switched on and off and configured with respect to length, duration, and schedule. The care plan itself is provided to the pathway instance creator through a FHIR API providing interfaces for (a) retrieving, (b) updating and (c) viewing a history of all changes to the care plan.

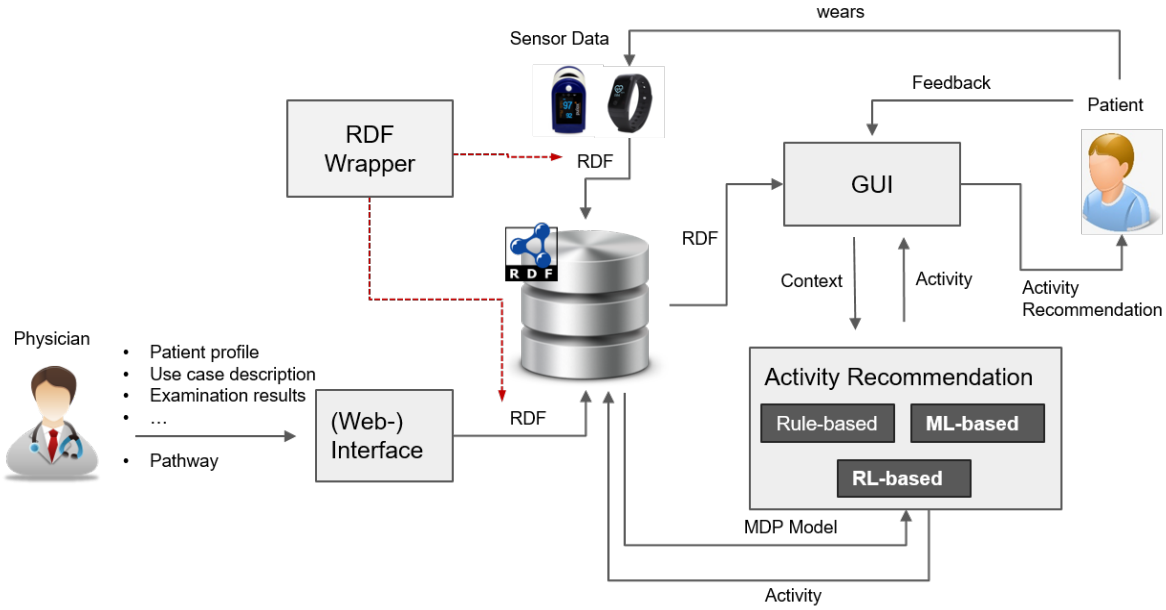

Figure 7 - pathway instantiation, execution and adaptation

3.4 PATHWAY INSTANCE EXECUTOR

The pathway instance executor exploits the integrated pathway information received from the pathway semantic wrapper to execute the correct vCare Coaching services at the correct point in time.

More specifically, the pathway execution service implements a scheduler which maps the service calls for the current pathway for a patient (i.e., care plan) to specific points in time during the day. The plan is continuously monitored with respect to pathway instance adaptation, as dwelled upon next.

3.5 PATHWAY INSTANCE ADAPTATION

The pathway instance executer is continuously re-calculating the most up-to-date plan for an individual patient to enable pathway instance adaptation.

Pathway instance adaptation consists of two stages: (i) expert rule-based adaptation and (ii) machine learning-based adaptation. For stage (i), the knowledge layer uses a curated set of expert rules to adapt the patient goals (e.g., the targeted number of steps for a given day or the targeted score of serious games), the number of times physical activities should be conducted during the week as well as which activities should be played. For stages (ii), the knowledge layer learns patient-specific machine learning model which perfects the set of expert rules in order to be better adjusted for an individual patient.

From an architectural point of view, the pathway instance adapter adapts the internal representation of the current patient-specific pathway (i.e., care plan) and makes it available to be queried by the Coaching services (in plain JSON format according to FHIR standards).

One example of instance adaptation based on the fall risk could be:

Use Case: UC2

Linked Measure: patient classify at fall risk (by Physician or by Machine Learning)

Rules:

If (*Faller*)

→ VC suggests e-learning session about "the best tips for avoiding falls", "how to properly climb stairs", "how to properly sit and stand-up from a chair".

Machine Learning:

ML can help here to predict risk factors. Every risk leads to new interactions with the patient in order to warn him/her. The following information integrated together can be used to identify fall risk:

- Clinical evaluation score (Conley scale >= 2)
- Home localization (bathroom and bedroom are room to high risk)
- Sit to stand action (hazard phase)
- Time (night)
- Age (>80 years)
- Mood (Depression)

If VC thanks to ML perceives (through collected data) that the patient has a Fall Risk, it can:

- Remind the use of walking stick every stand up.
- Suggest e-learning session with informative and proactive guide on falls prevention.
- Suggest e-learning session with informative and proactive guide on falls prevention based on the most frequented rooms (data from Environmental Context)
- Motivate patient to perform a Serious Game concerning balance rehabilitation.

3.6 DATA INTEGRATION ENGINE

The reference systems described in section 3 is a composition of standalone services which require a gluing middleware to communicate to each other. This middleware is based on a combination of REST endpoints and MQTT (Message Queuing Telemetry Transport) messages.

REST is used by the 'pathway creation' tools, as these do not interact directly with the normal operation of the rest of the services, they just supply the static template that is used to create instances.

The other components do provide real time communication using MQTT, which allows a more flexible data exchange than REST calls.

The MQTT exchange strategy is based on two authorization principles:

- User data separation: a user can only subscribe to its own data, having no access to the data from other users.
- User groups isolation: users can be arranged in independent groups. An MQTT client will have access to just one group by policy.

The first authorization principle is solved by defining a topic strategy based on the user identifier, in combination with topic access policies. Topic policies are, at the same time, divided in two categories:

- Policies for end-user clients (e.g., an application requesting the careplan for a specific patient), which grant access to topics prefixed with the patient id.
- Policies for backend services running batch processes (e.g., the pathway instance adaptor, which might get the pathway instances from all patients to perform the adaptation tasks), which have access to all the users in a group.

The second authorization principle is ensured by providing separate MQTT broker tenants for each group. This supplies a physical separation of data, since each tenant will only handle the data for a specific group.

For the reference implementation we provide a RabbitMQ[30] broker for MQTT communication, and OAuth2 token-based authentication and authorization mechanism. A remote client willing to connect to the MQTT broker must identify first against an OAuth2 server. In our reference implementation we use Keycloak[31] as the auth server, with some custom mappers to insert specific token claims to the access token.

Figure 8 shows an example of the provided access token for a client which can only access data for a unique patient:

```
{
  "exp": 1614183964,
  "iat": 1614097564,
  "jti": "ae9db0fc-9148-4c6f-873a-b93414f65e91",
  … (shortened for clarity)
  "groups": [
    "/techlabs"
  ],
  "rabbit_extra_permissions": [
    "rabbitmq.read:techlabs/*/vcare.50c091ba-3b0b-4ae4-8236-0ce05cb8b1ff.*",
    "rabbitmq.write:techlabs/*/vcare.50c091ba-3b0b-4ae4-8236-0ce05cb8b1ff.*",
    "rabbitmq.configure:techlabs/*/vcare.50c091ba-3b0b-4ae4-8236-0ce05cb8b1ff.*"
  ]
}
```

Figure 8 - Sample oauth2 access token

[30] https://www.rabbitmq.com/
[31] https://www.keycloak.org/

The most important claims in the token are:

- "groups": this claim controls the groups (and therefore, the MQTT tenant to connect) this client is able to see.
- "rabbit_extra_permissions": permissions granted to this client when accessing the MQTT tenant. These permissions are separated in three categories:
    - Read: ability to subscribe to topics.
    - Write: ability to publish to topics.
    - Configure: ability to create/destroy topics.

  The last part of the permission is a regular expression defining which topics are accessible. In this case, the pattern specifies that accessible topics are these starting with the word 'vcare', followed by the patient identifier '50c091ba-3b0b-4ae4-8236-0ce05cb8b1ff', followed by anything else ('*' pattern).

The following topics have been defined for different kinds of interactions[32]:

Table 1 - List of available used MQTT topics

Interaction name	Description	QoS
DIALOG_CLIENT_COMMAND_MESSAGE	message that is sent to the dialog management from the output device containing information about the result of the taken actions	2
DIALOG_CLIENT_COMMAND_REQUEST	message that is sent to the output device from the dialog management containing information about actions to take	2
DIALOG_CLIENT_MODE_MESSAGE	message that is sent to the dialog management from the output device containing information about the client's output mode	2
DIALOG_CLIENT_MODE_REQUEST	message that is sent to the dialog management from the output device containing information about the client's output mode	2
DIALOG_CLIENT_STATUS_MESSAGE	message that is sent to the dialog management from the output device containing information about the client's status	2
DIALOG_EVENT_TRIGGER	message to trigger dialog management event with its event name. It contains the input for the dialog management model in written text form	2

[32] QoS is related to MQTT's delivery policy: https://mosquitto.org/man/mqtt-7.html

Interaction name	Description	QoS
DIALOG_INPUT_TEXT	message that triggers the dialog management. It contains the input for the dialog management model in written text form	2
DIALOG_OUTPUT	message that is sent from dialog management to the output device with values for various modalities	2
DIALOG_OUTPUT_REQUEST	message that is sent the dialog management with output for various modalities to be sent to the output device with values for various modalities	2
DIALOG_PING	message sent to the dialog management to check if the dialog management is alive	2
DIALOG_PONG	message sent from the dialog management to acknowledge that the dialog management is alive	2
GAME_RESULT	message that is sent from the gaming module to indicate a specific game was finished	2
GAME_SESSION_COMPLETED	message that is sent from the gaming module to indicate a specific session of games has been completed	2
GAME_SESSION_PLANNED_REQUEST	message that is sent to the UI professional module to get a list of planned sessions related to a specific user	2
GAME_SESSION_PLANNED_RESULT	message that is sent from the UI professional module as a response to a previous request related to a specific use	2
GAME_SESSION_PLAYED_REQUEST	message that is sent to the gaming module to get a list of played sessions related to a specific user	2
GAME_SESSION_PLAYED_RESULT	message that is sent from the gaming module as a response to a previous request related to a specific user	2
GAME_SESSION_START	message that is sent to the gaming module to start a specific session of games	2
GAME_SESSION_STARTED	message that is sent from the gaming module to indicate a specific session of games was started	2

Interaction name	Description	QoS
HEALTH_QUESTIONNAIRE_REQUEST	message that is sent to the coaching services to perform a questionnaire for a specific user	2
HEALTH_QUESTIONNAIRE_RESULT	message that is sent from the coaching services with the result of a questionnaire for a specific user	2
HEALTH_STATUS_MESSAGE	message that is sent to the coaching services with the health status for a specific user	2
HEALTH_STATUS_REQUEST	message that is sent from the coaching services to request the health status for a specific user	2
HEALTH_THERAPY_ENDED	message that is sent from the therapy module to indicate a specific therapy was ended	2
HEALTH_THERAPY_START	message that is sent to the therapy module to start a specific therapy	2
HEALTH_THERAPY_STARTED	message that is sent from the therapy module to indicate a specific therapy was started	2
HEALTH_THERAPY_TASKS_REQUEST	message that is sent to the KRF to retrieve a list of pathway related tasks for a patient	2
HEALTH_THERAPY_TASKS_RESPONSE	message that is sent from the KRF containing a list of pathway related tasks for a patient	2
REMINDER_MESSAGE	message that is sent to the coaching services to remind the user e.g., about a specific task	2
REMINDER_RESPONSE	message that is sent back to the origin of the reminder message and contains an optional confirmation or reply from the user.	2
REMINDER_POSTPONE	message that is sent from the coaching services to indicate that a reminder should be postponed	2
SUPPORT_SERVICE_MESSAGE	message that is sent from the supporting services with data for a specific user	2
SUPPORT_SERVICE_REQUEST	message that is sent to the supporting services to request data for a specific user	2

Interaction name	Description	QoS
MOTIVATION_TEXT_MESSAGE	message that is sent to the service which requested a motivational message	2
ELEARNING_CONTENT_MESSAGE	message that is sent to the e-learning coaching service to trigger an e-learning session with specific content	2
STANDBY_MESSAGE	message that is sent communication modules to inform about an enabled/disabled standby mode	2
MOTIVATION_TEXT_REQUEST	message that is sent to the motivational message service for generating a context related message for the user	2
MOTIVATION_OUTPUT_REQUEST	message that is sent to the motivational message module which generates an output message directly based on the data and context provided	2
ELEARNING_CONTENT_REQUEST	message that is sent to the knowledge representation layer to retrieve content for an e-learning session for a specific user	2
USER_FEELING_STATUS_MESSAGE	message that is sent to the knowledge representation framework to store the user's subjective feeling	2
USER_MESSAGE	message that is sent to the coaching services to inform the user	2
LOCATION_MESSAGE	message that is sent from the data services with the location for a specific user	2
LOCATION_REQUEST	message that is sent to the data services to request the location for a specific user	2
POSITION_MESSAGE	message that is sent from the data services with the position for a specific user	2
POSITION_REQUEST	message that is sent to the data services to request the position for a specific user	2
CAREPLAN_ASSIGNED	message issued when a new careplan is assigned to a patient	2
CAREPLAN_MODIFIED	message issued when a careplan has been modified and clients are notified with an update	2

Interaction name	Description	QoS
CAREPLAN_UNASSIGNED	message issued when a careplan is removed/unassigned from a patient	2
CURRENT_CAREPLAN_REQUEST	message issued by a client requesting the active careplan of a patient	2
CURRENT_CAREPLAN_RESPONSE	message issued to response a careplan request	2
CAREPLAN_HISTORY_REQUEST	message issued by a client requesting the history of treatments assigned to a patient	2
CAREPLAN_HISTORY_RESPONSE	message responding a careplan history request	2
PATIENT_PROFILE_CREATED	Used to broadcast new patient profiles created in vCare	2
PATIENT_PROFILE_UPDATED	Used to broadcast updates to a patient profile	2
PATIENT_PROFILE_DELETED	Used to notify the deletion of a user profile	2
PATIENT_PROFILE_REQUEST	Used to request a specific patient profile	2
PATIENT_PROFILE_RESPONSE	Used to response with a specific patient profile	2
OBSERVATION	Used to exchange data about a specific observation type. The <<obsType>> variable correspond to any vital signs/attribute defined in the vCare ontology (D4.3)	2
OBSERVATION_RESPONSE	Used to answer to a request	2
CAREPLAN_ACTIVITY_STARTED	Used to notify that a certain activity in the careplan has started	2
CAREPLAN_ACTIVITY_FINISHED	Used to notify that a certain activity in the careplan has finished	2
DEVICE_ASSIGNED	Used to assign a device to a patient	2
DEVICE_UNASSIGNED	Used to unassigned a device to a patient	2
PATIENT_DEVICE_LIST_REQUEST	used to request the device list for a patient	2
PATIENT_DEVICE_LIST_RESPONSE	Used to answer a previous device list request	2
PATIENT_CLINICAL_STATE_REQUEST	Used to request the clinical state of a patient	2
PATIENT_CLINICAL_STATE_RESPONSE	Used to answer to a clinical state request	2
PATIENT_CLINICAL_HISTORY_REQUEST	Used to request the clinical history of a patient	2

Interaction name	Description	QoS
PATIENT_CLINICAL_HISTORY_RESPONSE	Used to answer to a clinical history request	2
PATIENT_PERSONAL_STATE_REQUEST	Used to request the personal state of a patient	2
PATIENT_PERSONAL_STATE_RESPONSE	Used to answer to a personal state request	2

Additional documentation and examples are available at https://middleware.vcare-project.eu/docs/

3.7 REST API ACCESS

To allow a convenient access to the vCare pathway information, the vCare-as-a-Service has defined basic CRUD operations for managing the different stages of a pathway instance.

There exist eHealth platforms that support a large number of services, however many might not be linked with a clinical pathway, as supported by vCare. By means of the vCare-as-a-Service, an eHealth platform user will be able to subscribe to the vCare platform and "link" his/her disease with the recommended clinical pathway. vCare services will be "consumed" by the eHealth platform that implements the vCare API and in this way the internal decisions making of an eHealth platform (e.g., for rehabilitation/coaching) will be more efficient.

For the vCaaS pathway management, table 2 provides an overview of the vCaaS operation and interfaces that are offered to provide access to the general clinical pathway information/PlanDefinition details for external services.

Table 2 - Interactions the vCaaS service offers to 3rd parties to interact with Pathway instances.

Repository's RESTful base	
`http:// middleware.vcare-project.eu/vcaas/vcaas-v1` This is the vCaaS base URL of the RESTful API. All the endpoints are available at this base.	
Object	**Description**
getClinicalPathway	**Overview** Retrieve a list of clinical Pathways
	Endpoint `GET /clinicalPathways[/{clinicalPathwayId}]` **Parameters** `{clinicalPathwayId}` – Optional. If specified, only the clinical pathway with matching clinicalPathwayId is returned, or none if no match. **Query string parameters**

none

Response

`HTTP 200` – Response body contains a collection of JSON of the clinical Pathways (PlanDefinition resource) corresponding to the patient profile or the JSON of the clinical Pathway identified by the clinicalPathwayId

`HTTP 400` – Returned, if the request is in a bad format. Typically, it means the clinicalPathwayId is not in the correct format.

`HTTP 401` - Returned, if the authorization information is missing or invalid. Typically, it means the requesting entity is not allowed to view the clinicalPathwayId (patient profile does not match)

`HTTP 404` – Returned, if there are no clinical Pathways belonging to the patient profile or the clinicalPathway with the specified clinicalPathwayId does not exist in the repository.

Sample request

```
curl -X GET "http://middleware.vcare-project.eu/vcaas/vcaas-v1/clinicalPathways" -H "accept: application/json"
```

Sample response

```
HTTP 200 OK

{

  "id":" 0a598c1d-1e84-4a29-bbc2-ad192a1f9526"

  "scheduled_activity": [

    {

      "activity": [

        {

          "name": " memoryTrainingA ",

          "ref_activity": " <http://memory-games.com/A> ",

          "textual_description": " training the memory of the patient ",

          "evidenceIndicator": " GameScore ",

          "supervised_by": " clinical_department ",

          "treatment_type": " Cognitive",

          "difficulty_level": " medium "

        }

      ],

      "active_device": " device_name ",
```

<table>
<tr><td></td><td>

<pre>
      "start_time": "11:00",
      "end_time": "12:00",
      "goal": "success",
      "result": " activity result ",
      "documented_supervision": "supervisor information"
    }
  ]
}
</pre>

</td></tr>
<tr><td>getClinicalPathway</td><td>

Overview:

Retrieve a clinical pathway

Endpoint

<pre>GET /clinicalPathways/{clinicalPathwayId}</pre>

Parameters

`{clinicalPathwayId}` – Required. The clinical pathway with matching clinicalPathwayId is returned, or none if no match.

Query string parameters

<pre>none</pre>

Response

`HTTP 200` – Response body contains the JSON of the clinical Pathway identified by the clinicalPathwayId

`HTTP 400` – Returned, if the request is in a bad format. Typically, it means the clinicalPathwayId is not in the correct format.

`HTTP 401` - Returned, if the authorization information is missing or invalid. Typically, it means the requesting entity is not allowed to view the clinicalPathwayId (patient profile does not match)

`HTTP 404` – Returned, if the clinicalPathway with the specified clinicalPathwayId does not exist in the repository.

Sample request

<pre>curl -X GET "http://middleware.vcare-project.eu/vcaas/vcaas-v1/clinicalPathways/a3f7747c-a1b6-4ca5-8ee6-e7c304630ff7" -H "accept: application/json"</pre>

Sample response

<pre>HTTP 200 OK

{</pre>

</td></tr>
</table>

	"scheduled_activity": [
	{
	"activity": [
	{
	"name": " memoryTrainingA ",
	"ref_activity": " <http://memory-games.com/A> ",
	"textual_description": " training the memory of the patient ",
	"evidenceIndicator": " GameScore ",
	"supervised_by": " clinical_department ",
	"treatment_type": " Cognitive",
	"difficulty_level": " medium "
	}
	],
	"active_device": " device_name ",
	"start_time": "11:00",
	"end_time": "12:00",
	"goal": "success",
	"result": " activity result ",
	"documented_supervision": "supervisor information"
	}
	]
	}

addClinicalPathway	**Overview**: Add Clinical Pathways
	Endpoint `POST /clinicalPathways` **Request body:** A valid clinical Pathway (PlanDefinition) **Query string parameters** none **Response**

<table>
<tr><td></td><td>

HTTP 201 – The response body contains the created clinicalPathway (PlanDefinition) resource as it would be requested by a GET.

HTTP 400 – Returned if the request is in a bad format. Typically, it means the JSON is not a valid clinical Pathway.

HTTP 401 – Returned, if the authorization information is missing or invalid. Typically, it means the requesting entity is not allowed to create a clinicalPathway (no rights)

HTTP 404 –

HTTP 409 – Returned, if there is a name conflict for the clinicalPathway Id.

Sample request

```
curl -X POST "http://middleware.vcare-project.eu/vcaas/vcaas-v1/clinicalPathways" -H "accept: application/json" -H "Content-Type: application/json" -d "{ \"scheduled_activity\": [ { \"activity\": [ { \"name\": \"memoryTrainingA\", \"ref_activity\": \"<http://memory-games.com/A>\", \"textual_description\": \"training the memory of the patient\", \"evidenceIndicator\": \"GameScore\", \"supervised_by\": \"clinical_department\", \"treatment_type\": \"Cognitive\", \"difficulty_level\": \"medium\" } ], \"active_device\": \"device_name\", \"start_time\": \"11:00\", \"end_time\": \"12:00\", \"goal\": \"success\", \"result\": \"activity result\", \"documented_supervision\": \"supervisor information\" } ]}"
```

Sample response

```
HTTP 201 CREATED

{

  "resourceType": "PlanDefinition",

  "id": " 8d24826e-77cd-4e10-971f-883bb449f6ca",

  ... further content of the created clinicalPathway

}
```

</td></tr>
<tr><td>updateClinicalPathway</td><td>

Overview

Update clinical Pathway

</td></tr>
<tr><td></td><td>

Endpoint

```
PUT /clinicalPathways/{clinicalPathwayId}
```

Parameters

</td></tr>
</table>

`{clinicalPathwayId}` – Required. The clinical pathway with matching clinicalPathwayId will be updated.

Request body:

A valid clinical Pathway (PlanDefinition)

Query string parameters

`none`

Response

`HTTP 200` – The response body contains the updated clinicalPathway (PlanDefinition) resource as it would be requested by a read.

`HTTP 400` – Returned if the request is in a bad format. Typically, it means the JSON is not a valid clinical Pathway.

`HTTP 401` – Returned, if the authorization information is missing or invalid. Typically, it means the requesting entity is not allowed to delete the clinicalPathway (no rights)

`HTTP 403` – Returned, if the server understands the request but refuses to authorize it

`HTTP 404` – Returned, if the clinicalPathwayId does not exist.

`HTTP 409` – Returned, if there is already another operation on the same resource.

Sample request

```
curl -X PUT "http://middleware.vcare-project.eu/vcaas/vcaas-v1/clinicalPathways/8d24826e-77cd-4e10-971f-883bb449f6ca" -H "accept: application/json" -H "Content-Type: application/json" -d "{ \"scheduled_activity\": [ { \"activity\": [ { \"name\": \"memoryTrainingA\", \"ref_activity\": \"<http://memory-games.com/A>\", \"textual_description\": \"training the memory of the patient\", \"evidenceIndicator\": \"GameScore\", \"supervised_by\": \"clinical_department\", \"treatment_type\": \"Cognitive\", \"difficulty_level\": \"medium\" } ], \"active_device\": \"device_name\", \"start_time\": \"11:00\", \"end_time\": \"12:00\", \"goal\": \"success\", \"result\": \"activity result\", \"documented_supervision\": \"supervisor information\" } ]}"
```

Sample response

```
HTTP 200 OK

{

  "resourceType": "PlanDefinition",

  "id": "8d24826e-77cd-4e10-971f-883bb449f6ca",
```

	`... further content of the updated clinicalPathway` `}`
deleteClinicalPathway	**Overview** Delete a clinical pathway

	Endpoint
	`DELETE /clinicalPathways/{clinicalPathwayId}`
	Parameters:
	`{clinicalPathwayId}` – Required. The clinical pathway with matching clinicalPathwayId will be deleted.
	Response
	`HTTP 200` – OK
	`HTTP 400` – Returned, if the request is in a bad format. Typically, it means the clinicalPathwayId is not in the correct format.
	`HTTP 401` – Returned, if the authorization information is missing or invalid. Typically, it means the requesting entity is not allowed to update the clinicalPathway (no rights)
	`HTTP 403` – Returned, if the server understands the request but refuses to authorize it.
	`HTTP 404` – Returned, if the clinicalPathwayId does not exist.
	`HTTP 409` – Returned, if there is already another operation on the same resource.
	Sample request
	`curl -X DELETE "http://middleware.vcare-project.eu/vcaas/vcaas-v1/coachingMessages/9770af79-8911-4098-8592-3199dcb955f7" -H "accept: application/json"`
	Sample response
	`HTTP 200 OK`

3.7.1 AUTHORIZATION CONSIDERATIONS

The REST API will be the main entry point to the reference system, so this API has not only to provide proper functions to access the rest of components, but a set of built-in authorization mechanisms.

The authorization chosen for this purpose is OAuth scopes embedded in OAuth tokens.

In order to grant/deny access to a pathway resource, we need to:

- Define policies.
- Define permissions.
- Apply policies to permissions.

- Associate permissions to a scope or resource (or both).

In order to do this, there are several options, but what we have considered within the project is to define individual policies, then group all the related policies under an aggregated policy (a policy of policies) and then associate that aggregated policy with scope-based permission.

The API, therefore, uses scopes that refer to the object it grants access to, followed by the class of actions on that object they allow (e.g. `pathway:write`).

The list of available objects is mapped to the different elements in the reference system.

There are currently only two classes of action:

- read: Reading the full information about a single resource.
- write: Modifying the resource in any way e.g., creating, editing, or deleting.

For example, to request access to the list of available pathways defined, the API will request `pathway:read`.

Table 3 – Roles available within the pathway system

Role	Description
admin	Administer the configuration of the reference system
pathway	Access the pathway subsystem
wrapper	Access the wrapper
Instance.creator	Create an instance of a pathway
Instance.executor	Execute an instance of a pathway
Instance.adapter	Modify an instance of a pathway

Table 4 – OAuth scopes

OAuth scope	API methods	Roles
pathway:read	getClinicalPathway	admin, pathway
pathway:write	addClinicalPathway, updateClinicalPathway, deleteClinicalPathway	admin, pathway
wrapper:read	getWrapper	admin, wrapper, pathway
wrapper:write	addWrapper, updateWrapper, deleteWrapper	admin, wrapper, pathway
instance:read	getInstance	admin, instance.executor
instance:write	addInstanceCreator, updateInstance, deleteInstance	admin, instance.creator, instance.adapter

Note: API methods are pending design and implementation, list is not final

3.7.2 EXAMPLE OF INTERACTION

The example of interaction between a 3rd party system and vCare's pathway system is split into the authorization part, which should be carefully designed and implemented and the business logic taking place after access has been granted to the pathway system.

Authorization flow

The first example of interaction is generic, highlighting an unnamed 3rd party system and a vCare pathway resource. The 3rd party system is assumed to have the need to access a certain resource from the pathway system (e.g., instance creator or the wrapper) and to have either read or write or both permissions. With this, the flow is presented in figure 10.

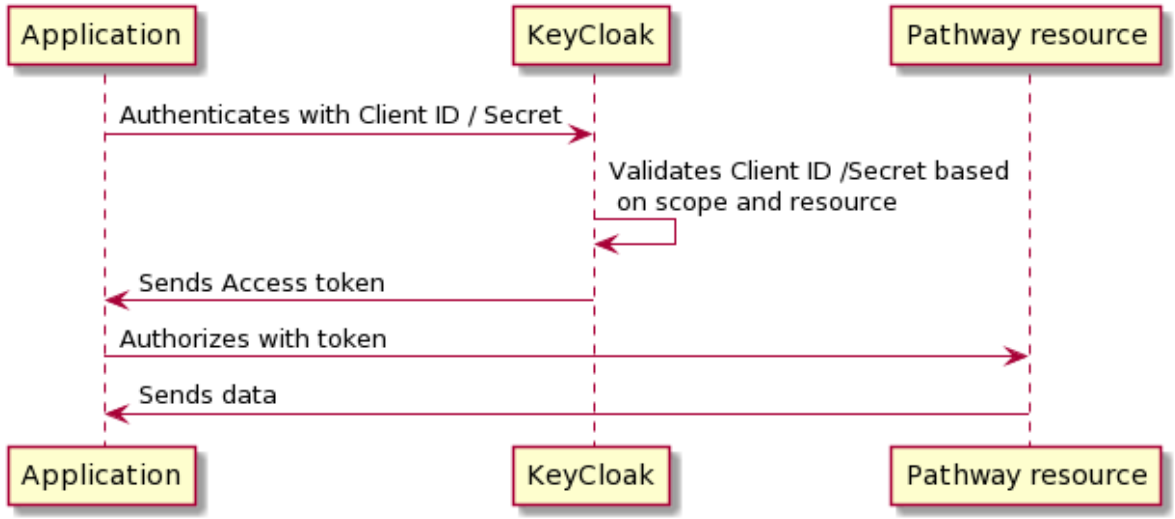

Figure 9 – authorization flow between 3rd party system and vCare pathway system

As an example of the interaction, we show how the authorization flow between a CC2U application and a pathway resource is taking place via the vCaaS API. The application will request from vCaaS access to the resource, together with its existing token (internal to CC2U) and the vCaaS logic takes care of matching the secret and generating the access token (assuming, again, that the application has the right permissions). This is depicted in *figure* 11.

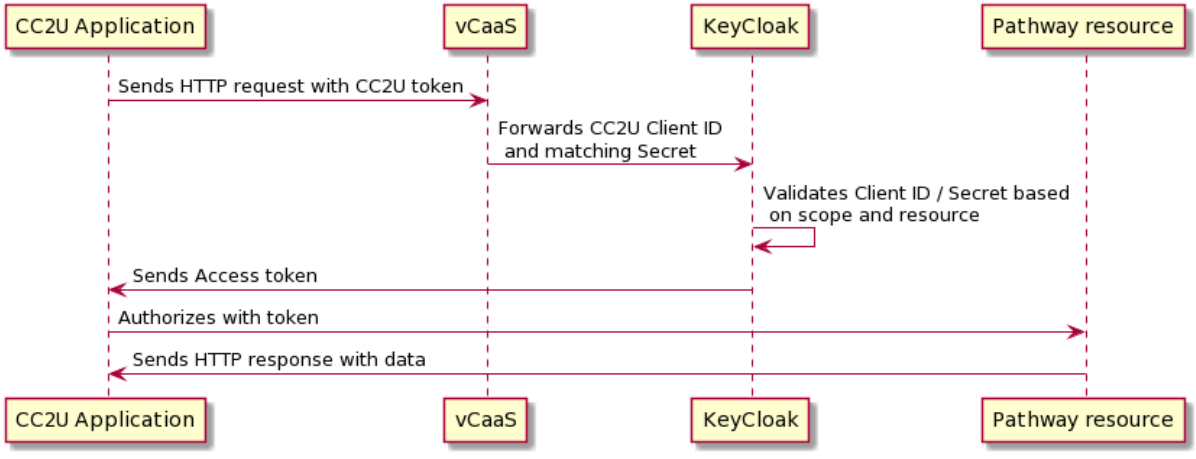

Figure 10 – authorization flow between a CC2U application and vCare pathway system

Interaction between CC2U and vCare pathway system

As a showcase of the possible interaction between an external 3rd party system and the vCare pathway system, we show in *figure* 11 how the CC2U Daily Functioning Monitoring Application uses the pathway information.

This application category, as the name indicates, concerns monitoring of all activities that the CC2U user performs during the day inside the home environment. Data can be acquired automatically or manually inputted by the user.

The CC2U Daily Functioning Application relies on the service with the same name, which will call the vCaaS API to extract the status of the user from the vCare pathway system, which in turn will respond with the Plan Definition that will contain the current and/or planned action of the user.

Based on this action of the user from the care plan, the monitoring application will be enriched with information concerning the pathway of the user, in addition to all the other monitoring information contained in the CC2U system.

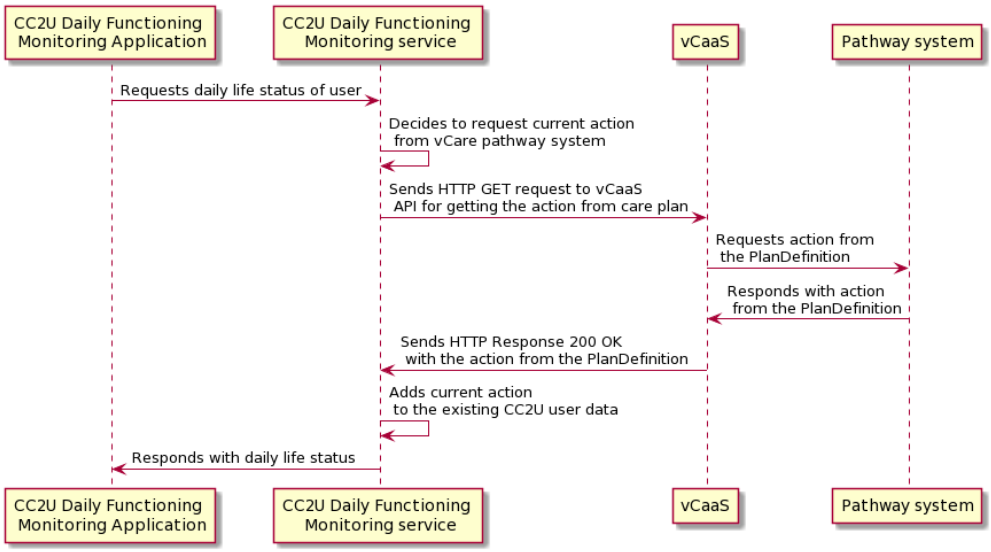

Figure 11 – Example of CC2U Daily Functioning Monitoring Application using vCare pathway information via vCaaS API.

4 TECHNICAL AND ORGANIZATIONAL MEASURES

In the previous sections we have described the components that make up the reference system from a technical and functional perspective. In this section we move one step further and provide an overview of the different configurations that can be composed using a *cherry-picking* strategy.

Simply clarify that the reference implementation provided in this document **is not intended as a "exploitable result" or as a running system**, but a guideline on how these building blocks can be combined to build up different setups offering a different set of functionalities.

It is not either the purpose of this document to enter into an in-depth discussion about exploitation and licensing models, since the components described are part of a bigger system whose scope goes well beyond the scope of this document and will be tackled in a future document (D9.10).

In the rest of the section, we will assume a Software-as-a-Service approach (SaaS) to develop a set of profiles that can be composed by selecting a set of building blocks providing the following set of functionalities:

Table 5 - basic functionalities in the reference system

Functionality	Description	Component(s)
Pathway templates	A tool to create a pathway using BPMN notation and modelled into FHIR resources	Pathway creator
Pathway mapping into vCare model	Translation of pathways expressed in FHIR model to the vCare internal model	Pathway wrapper
Pathway instantiation	Service that takes a generic pathway template and provides personalization based on patient profile and treatment parameters	Pathway instance creator
Pathway execution	Service that executes in a timely manner the actions in a pathway instance	Pathway instance executor
Pathway adaptation	Service that considers information about the progression of the treatment to modify currently available action or plan the actions for a later period	Pathway instance adaptor
Integration engine	A MQTT communication broker to link the internal communication between modules	Mqtt broker
Access REST API	A set of rest methods to access to the functionality provided by the different components	vCAAS API

4.1 REFERENCE PROFILES

Table 6 summarizes the components used in each profile that could be reasonably built. These are archetypical implementation scenarios a user interested in different kinds and extents of "pathway modelling" use cases could be interested in to have. In the sub-sections below, we will outline the profiles' core ideas.

Table 6 - reference system profiles detail

Profile	Templates	Mapping	Instance	Execution	Adaptation	MQTT	REST
#1	✓	✗	✗	✗	✗	✗	✓
#2	✓	✓	✗	✗	✗	✗	✓
#3	✗	✗	✓	✗	✗	✓	✓
#4	✗	✗	✓	✓	✓	✓	✓
#5	✓	✓	✓	✓	✓	✓	✓

4.1.1 PROFILE 1: PATHWAY TEMPLATES

This profile is the simplest one supplying access to the Pathway creation tool and its REST API (see figure 12).

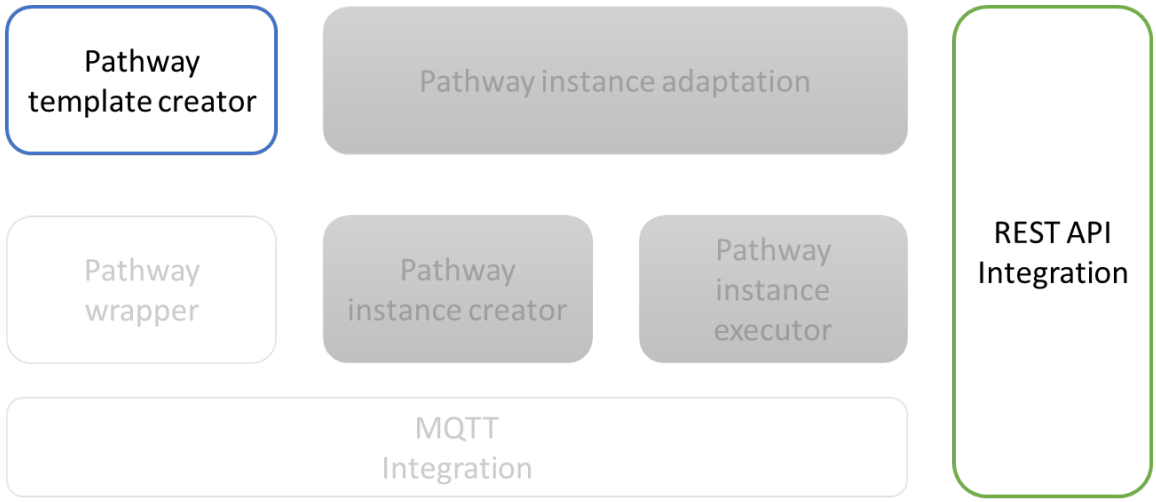

Figure 12 - reference profile 1

This profile is suitable for those scenarios in which there is no need to instantiate the pathway but rather have a tool to formalize templates for different pathologies. Through the REST API, these templates can be exported as a set of FHIR resources, either in JSON or XML formats.

4.1.2 PROFILE 2: EXTENDED PATHWAY MODELLER

The second profile (see figure 13) is similar to profile #1 but adding the capability of translating the templates in FHIR format to the ontological representation defined by vCare.

This profile is suitable for end systems that are willing to implement the instantiation, execution, and adaptation on their own.

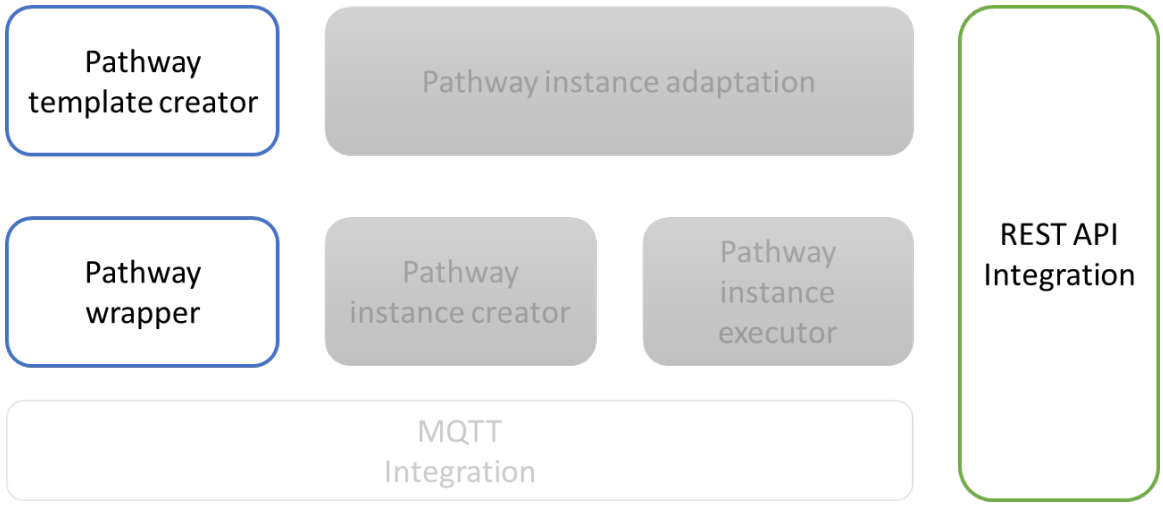

Figure 13 - reference profile 2

4.1.3 PROFILE 3: INSTANCE CREATION

In profile 3 (see figure 14), we open up the reference system for a third party that has its own template repository and simply wants to use the instantiation process. The execution and adaptation are handled by the third-party system.

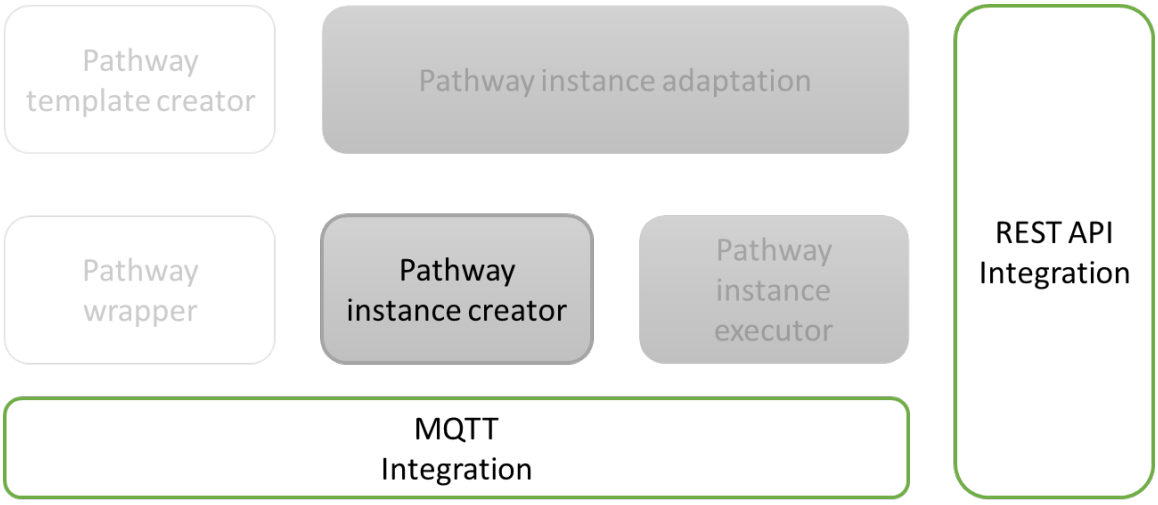

Figure 14 - reference profile 3

4.1.4 PROFILE 4: INSTANCE CREATION AND ADAPTATION

In profile 4 (figure 15), we extend profile 3 by adding the execution and adaptation capabilities. Using this profile, a third-party system can instantiate and execute its own pathway templates.

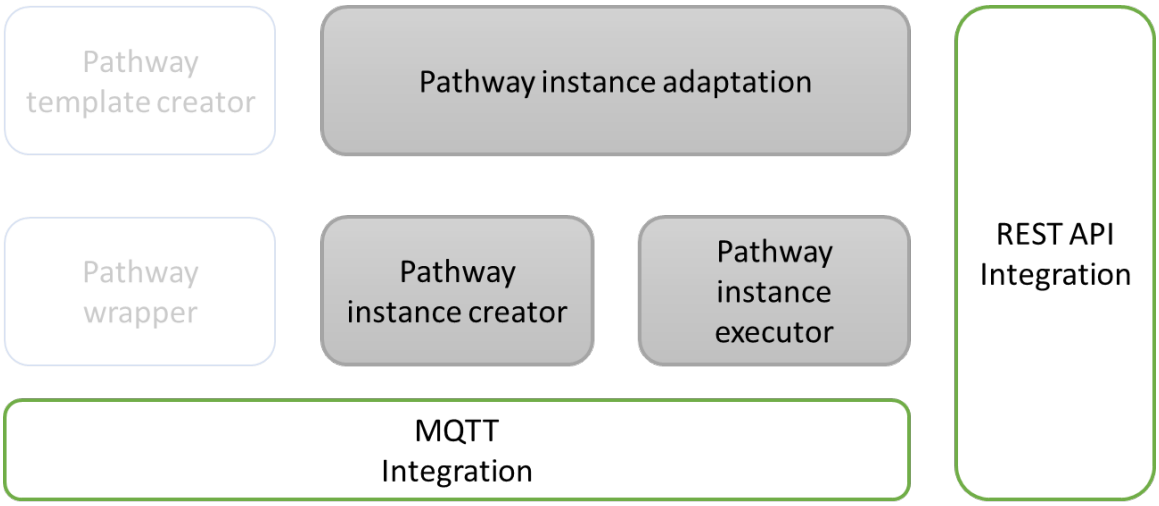

Figure 15 - reference profile 4

4.1.5 PROFILE 5: FULL PATHWAY CAPABILITIES

Profile 5, as depicted in figure 2, provides a complete reference system. This profile is suitable for end system that want to leverage the full power of vCare by just plugin it into their own interaction and monitoring devices.

4.2 EVALUATION OF PROFILES

4.2.1 METHODOLOGY FOR FEEDBACK AND EVALUATION

To assess the suitability and perceived added value of the suggested profiles, it would be advisable to conduct a survey (e.g., by making use of a sufficiently experienced focus group and by utilizing both qualitative and quantitative modes of investigation) to evaluate some of the profiles with the possible modes:

1. Feedback Questionnaire with pre-defined answer options (e.g., User Experience Questionnaire (UEQ) questionnaire).
2. Interviews with questions in free form.

The outcome of the evaluation aims to provide useful information for:

- Usefulness of profiles.
- Applicability in certain domains.
- Meeting expectations of stakeholders.
- Interest for future validation.
- Usability and User Experience.

4.2.2 USABLE METHODS FOR FEEDBACK AND EVALUATION

Interviews: Can be conducted with the members of the focus groups, so as to evaluate the qualitative aspects from the reference profiles.

Feedback Questionnaire: Here, the feedback analysis relies on a questionnaire that will be made available over Google Forms following the demonstration/presentation of profiles.

In terms of a particular feedback questionnaire we could make use of the **UEQ**. It could fit well here as this is also part of WP1's methodology to assess the coaching system as a whole. The UEQ is "a fast and reliable questionnaire to measure the User Experience of interactive products"[34]. It covers an impression of user experience. Both classical usability aspects (efficiency, perspicuity, dependability) and user experience aspects (originality, stimulation) are measured. More details on the UEQ have already been introduced in D1.5, section 7.5.

Here, a *short version* of the UEQ would be sufficient as well. This short version of the UEQ consists of just 8 items as displayed in table 7.

Table 7 - Short UEQ items

obstructive	o o o o o o o	supportive
complicated	o o o o o o o	easy
inefficient	o o o o o o o	efficient
confusing	o o o o o o o	clear
boring	o o o o o o o	exciting
not interesting	o o o o o o o	interesting
conventional	o o o o o o o	inventive
usual	o o o o o o o	leading edge

[34] https://www.ueq-online.org

5 CONCLUSIONS AND FUTURE WORK

In this deliverable we have presented the reference pathway system as a subset of the whole vCare Coaching Platform. The goal of this reference system is to provide a simple guideline on the architectural elements that would be needed to create a Coaching System for a similar scenario as the one covered by vCare. In general, this deliverable continues the work presented in Deliverable 6.5. In addition to the test system, we want to add reflections on possible implementation scenarios for the pathway-related sub-functionalities of the overall solution. This is to allow interested parties to build up on pathway services as the core governing instrument of the vCare solution. We hope the current reference implementation guide could, in turn, spread the idea of clinical pathways as central element of the processual control of eHealth measures in general.

The results from this deliverable should be taken into account for future exploitation strategies still to be defined by WP9.

Also, during the internal project validations, Living Lab phase and Pilot phase, refinements to the existing pathways as well as new ones are being added and tested to the catalogue of available pathways. So, the developments regarding the pathway system are not fully finalised by now, but no fundamental change is expected anymore. Any adaptations in the course of the testing phases will then be reported in D6.7.