PRECISE4Q

PREDICTIVE MODELLING IN STROKE

DELIVERABLE - RESUBMISSION

Project Acronym: **Precise4Q**

Grant Agreement number: **777107**

Project Title: **Personalised Medicine by Predictive Modelling in Stroke for better Quality of Life**

D2.3 – Decision of build of the data warehouse

Revision: 2.0

Authors and Contributors	Catalina Martínez Costa (MUG); Jose Antonio Miñarro Giménez (MUG); Nikola Lazovski (QMENTA); Paulo Rodrigues (QMENTA); Contributor: John D. Kelleher (TU Dublin)			
Responsible Author	Catalina Martínez Costa		**Email**	catalina.martinez@medunigraz.at
	Beneficiary	MUG	**Phone**	+4331638517880

Project co-funded by the European Commission within **H2020-SC1-2016-2017/SC1-PM-17-2017**		
Dissemination Level		
PU	Public, fully open	x
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified, information as referred to in Commission Decision 2001/844/EC	

Revision History, Status, Abstract, Keywords, Statement of Originality

Revision History

Revision	Date	Author	Organisation	Description
1.1	25.10.19	Catalina Martínez-Costa	MUG	Internal version for review by TU Dublin and DFKI
1.2	24.10.19	Nikola L.	QMENTA	Writing description of the QMENTA platform
1.3	24.10.19	Jose A.MG	UM	Writing and review
1.4	29.10.19	Catalina MC	MUG	Final writing and review
1.5	30.10.19	John D. Kelleher	TU Dublin	Machine learning module description
2.0	30.10.19	Catalina MC	MUG	Final review

Date of delivery	Contractual:	31.10.2018	Actual:	31.10.2019
Status	final **x** /draft ☐			

Abstract (for dissemination)	This document presents the architecture of the Precise4Q data warehouse. An implementation in two phases is described. During a first phase the running data warehouse provided by our industrial partner Qmenta will be adapted to our project, implementing the secure access to the heterogeneous data sources and their mapping to the Precise4Q data model, and integrating the different software modules provided by each project partner. During a second phase and hosted within the Qmenta platform but as an independent module, a semantic data warehouse using a graph database will be implemented, which is expected to better address the semantic functionalities provided by WP3, increasing the semantic power of the overall system. The semantic data warehouse will allow to export the data using the OMOP CDM format provided by the OHDSI community in order to provide a more standardized solution and align with a large and increasing community of researchers, health professionals, etc. Both, the semantic data warehouse and the export functionality to the OMOP CDM will be provided as open source tools. For a long-term scenario, we plan to have both data warehouses as independent platforms, where each data providing partner will be responsible for managing the graph-based data warehouse internally.
Keywords	(semantic) data warehouse, graph database, Digital Stroke Patient Platform, Qmenta platform, semantic data model, security and privacy framework, data interfaces, data quality assurance

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Table of Content

List of Figures

List of Tables

Executive Summary

This deliverable describes the architecture of the Precise4Q (P4Q) data warehouse, i.e. the Digital Stroke Patient Platform.

We will follow a two phases implementation strategy, in order to reduce the time to set up the data warehouse and aiming at combining the advantages of an industry solution with the advantages of a free data warehousing environment.

During a first phase the running data warehouse provided by our industrial partner QMENTA will be adapted to our project, implementing the secure access to the heterogeneous data sources and their mapping to the P4Q data model, and integrating the different software modules provided by each project partner: NLP module, harmonization / integration module, machine learning module, etc.

During a second phase and hosted within the QMENTA platform but as an independent module, a semantic data warehouse using a graph database will be implemented, which is expected to better address the semantic functionalities provided by WP3, increasing the semantic power of the overall system. Software modules like the learning / predictive modelling will be able to use both data warehouses depending on their needs.

The transition between both data warehouses will follow a pragmatic solution where some of the semantic functionalities might be implemented in QMENTA Cloud Platform for specific use cases when required.

In addition, the semantic data warehouse will allow to export the data using the OMOP CDM format provided by the OHDSI community in order to provide a more standardized solution and align with a large and increasing community of researchers, health professionals, etc. Both, the semantic data warehouse and the export functionality to the OMOP CDM will be provided as open source tools. For a long-term scenario, we plan to have both data warehouses as independent platforms, where each data providing partner will be responsible for managing the graph-based data warehouse internally.

1 Introduction

This deliverable describes the data infrastructure to be implemented in the P4Q project in order to support the Digital Stroke Patient Platform.

Within the project, heterogeneous and longitudinal data sources have to be managed. These data sources use different formats and with heterogeneous degrees of structure. Within the P4Q Data warehouse, clinical datasets will be harmonized and integrated in order to be accessible by predictive modellers. In order to provide a consistent use of the data and facilitate the integration and communication, we use a semantic representation of the data as input for the machine learning modelling and prediction tasks. The semantic representation infrastructure will be implemented as an independent module (Semantic Data Warehouse) but integrated in the Digital Stroke Patient Platform. Machine learning models and prediction results will be stored in the data warehouse infrastructure and, mainly, accessible from their own prediction modules.

We will follow a two phases implementation strategy. In order to reduce the time to set up the data warehouse and aiming at combining the advantages of an industry solution with the advantages of a free data warehousing environment, the QMENTA platform [1], provided by our industrial partner QMENTA, hosts the Digital Stroke Patient Platform and provides the basic functionalities related to the data infrastructure.

During a first phase the QMENTA running data warehouse will be adapted to our project, implementing the secure access to the heterogeneous data sources and their mapping to the P4Q data model, and integrating the different software modules provided by each project partner: NLP module, harmonization / integration module, machine learning module, etc.

During a second phase and also hosted within this platform but as an independent module, a semantic data warehouse using a graph database will be implemented, which is expected to better address the semantic functionalities provided by WP3, increasing the semantic power of the overall system. Software modules like the learning / predictive modelling will be able to use both data warehouses depending on their needs.

The transition between both data warehouses will follow a pragmatic solution where some of the semantic functionalities might be implemented in QMENTA Cloud Platform for specific use cases when required.

In addition, the semantic data warehouse will allow to export the data using the OMOP CDM [2] format provided by the OHDSI community [3] in order to provide a more standardized solution and align with a large and increasing community of researchers, health professionals, etc. Both, the semantic data warehouse and the export functionality to the OMOP CDM will be provided as open source tools. For a long-term scenario, we plan to have both data warehouses as independent platforms, where each data providing partner will be responsible for managing the graph-based data warehouse internally.

In Chapter 2 we present an overview of the architecture of the Digital Stroke Patient platform. There the QMENTA platform is described, as hosting platform of the Digital Stroke Patient Platform, in terms of its privacy and security framework and data interfaces to access the data as well as quality assurance and the implementation of the P4Q data model. Then, we briefly describe each of the modules that will interact with the data warehouse, i.e. ETL module, NLP module, semantic mapping module, harmonization and integration module, and machine learning modelling and predicting module. We also describe the semantic data warehouse and its benefits in terms of data representation for the data semantic exploitation. Finally, Chapter 3 presents the conclusions of the decisions made regarding the architecture of the platform and, in particular, about the data warehouse and the semantic data warehouse.

2 Digital Stroke Patient Platform Architecture

The Digital Stroke Patient Platform sets out to minimise the burden of stroke for the individual and for society and to achieve personalised stroke treatment, addressing patient's needs in four stages: prevention, acute treatment, rehabilitation and reintegration. Thus, P4Q project will provide a web-based platform to allow researchers to test and apply the developed stroke models on their own data.

The Digital Stroke Patient Platform is integrated into the QMENTA platform and uses their functionalities to perform several analysis and data management tasks, such as text analysis, data integration, semantics and harmonization tasks, and machine learning modelling and prediction. Therefore, QMENTA provides the infrastructure to support the project data warehouse and the execution environment.

Figure 1 shows the elements that belongs to the Digital Stroke Patient Platform and how they interact with QMENTA infrastructure. The QMENTA platform encloses the Digital Stroke Patient Platform modules, the data warehouse, and also provides the security framework, the communication interface and functionalities for exploiting the content of the data warehouse, i.e. searching, filtering, modifying and storing.

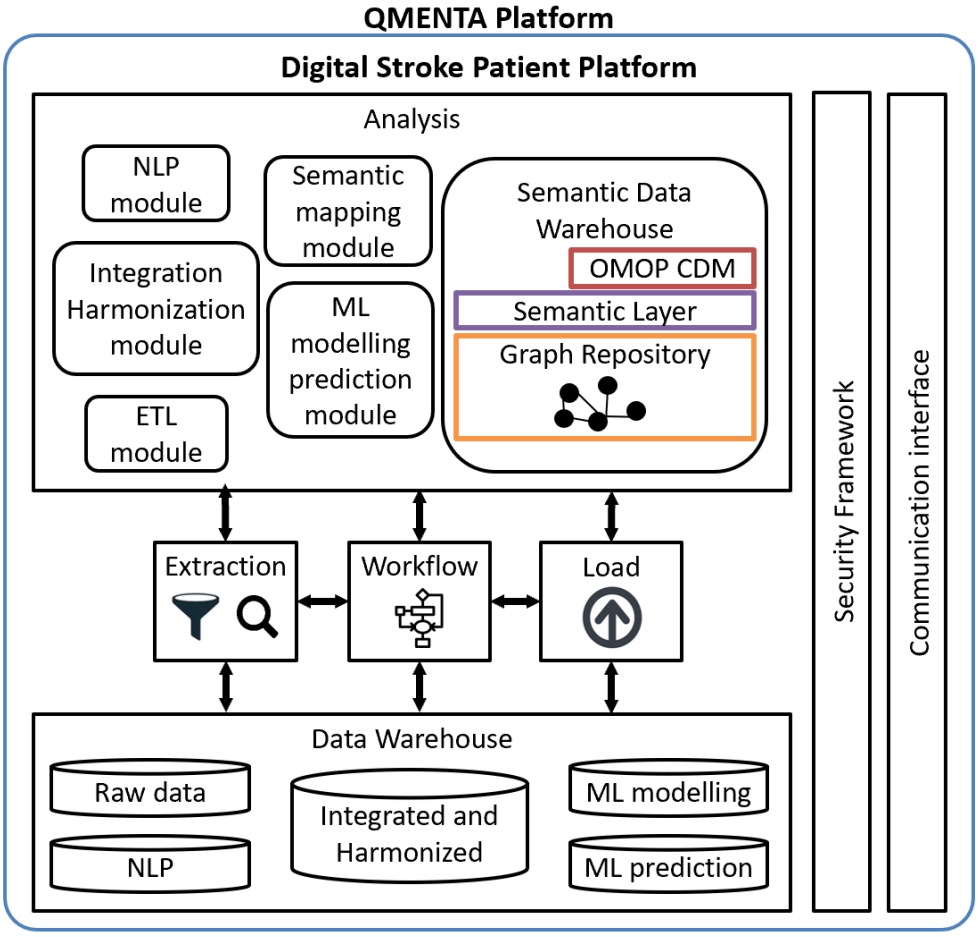

Figure 1 Digital Stroke Patient platform architecture

In the following we describe a general overview of the QMENTA platform: its data privacy and security framework; the data quality assurance process; the data interfaces and how its data model is extended to implement the P4Q data model.

Next, we will describe each of the modules of the Digital Stroke Patient Platform, including the semantic data warehouse and the standardized OMOP interface.

2.1 The QMENTA platform

QMENTA platform is a cloud-based platform that uses Artificial Intelligence (AI) techniques and allows to easily share data, results and methods with collaborators in a secure manner across jurisdictions during clinical studies and trials. QMENTA expertise is focused on machine learning and deep neuroimaging knowledge techniques. The scalable and convenient use of the AI techniques saves valuable time while providing researchers and practitioners quantitative evidence for their projects.

All data is safely stored in compliance with HIPAA [4] and Title 21 CFR Part 11 [5]. The privacy of patients' private health information is respected at all times with automated de-identification as well as end-to-end encryption.

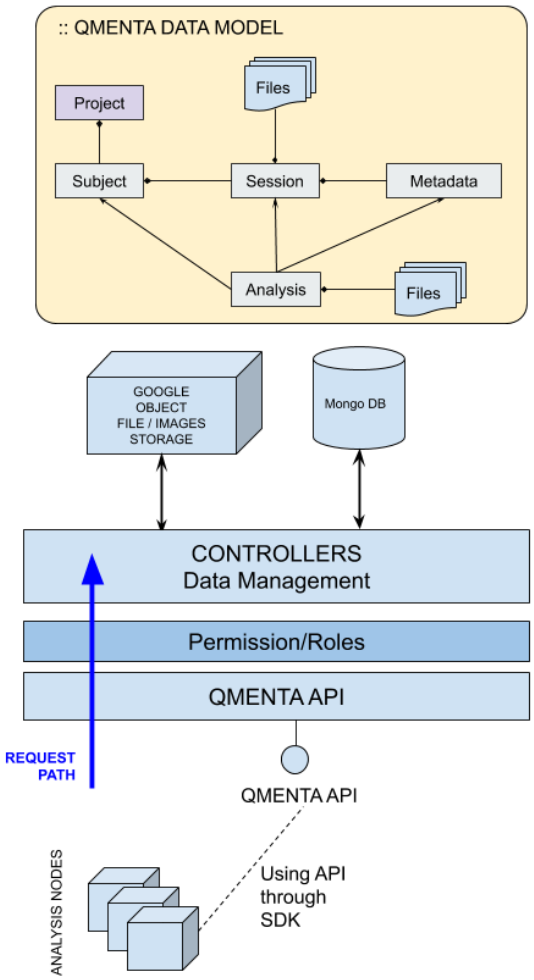

Figure 2 QMENTA Cloud Platform architecture

As shown in Figure 2, each request for fetching, storing or altering data goes through several functional layers: upon authenticating the user is provided a temporary token which is used to identify the users, their roles and permissions regarding the projects they have access to. The data stored in the QMENTA platform is always validated and stored in compliance with the general

QMENTA data model. Physically, the data is stored in the cloud providers offering worldwide data storage locations (See MongoDB [6] & Data Storage Location objects within Figure 2).

2.1.1 Data privacy and security framework

The platform manages protected health information (PHI) contained in medical images and related data in secure segregated network and encrypted data transmission.

When a user visits the platform, all data is transferred in encrypted format with HTTPS [7] protocol. The access to the platform is allowed only for authorized users through username and password and QMENTA administrators monitor the usage to deactivate invalid user accounts. Furthermore, QMENTA retains audit logs to track the activity on the platform as required by HIPAA.

Communication between client's web browser and the cloud infrastructure is through an encrypted and authenticated channel, with a strong protocol (TLS 1.2 [8]), a strong key exchange (ECDHE_RSA [9]), and a strong cipher (AES_256_GCM [10]).

In order to enable fine-grained access to data and related operation, QMENTA platform provides flexible role-permission model offering the possibility to define various roles regarding all aspects of a study and its data. For example, a user with a contributor role in one study can upload data, but it cannot delete any data; while a contributor role in another study can allow uploading data and delete only the data owned by the contributor.

The selection of data location is important to comply with the EU-US Privacy Shield [11]. The Shield highly regulates the transfer of personal data from EU to US and recommends limiting the transatlantic data transfer only for necessary situations. QMENTA platform helps comply with these privacy regulations. If you run your R&D activities in the EU, for example, you can choose to store your data in an EU region to avoid transatlantic data transfer.

QMENTA cloud platform provides built-in and automated anonymization of all uploaded data, removing PHI from uploaded DICOM [12] image files regarding the requirements to protect patient identity.

2.1.2 Quality assurance

QMENTA platform applies several levels of quality assurance:

Upon uploading, the system initiates an analysis that goes over each file, analyses its content, labels it properly and decides whether to keep it or not. This way the system transforms the unstructured files into an organized useful form.

Next, as a project option, the system initiates another automatic quality check that checks if the uploaded session data adhere to the protocol definition. A protocol adherence definition is a set of rules that each session data within a project have to fulfil in order to pass the quality check.

Apart from automatic quality check, the system allows project users to be assigned the role for quality assurance allowing inspection and validation of data. Both manual and automatic quality checks can mark the data with pass or fail labels.

In addition, QMENTA platform also performs regular automatic checks that estimate the overall quality of the data.

2.1.3 Data Interfaces

The functionality of the QMENTA platform is provided by an API accessed through an HTTPS protocol. The API describes a set of platform endpoints each referring to a particular functionality. This set can be divided into several logical subsets of endpoints, each subset referring to one or more entities of the QMENTA data model:

- Session data management endpoints
  - This set of endpoints is used for storing, altering and searching session data representing session files, session metadata, and metadata definition.
- Files management endpoints
  - This set of endpoints is used for uploading, storing, altering, and searching the files and the file metadata
- Analysis management endpoints
  - This set of endpoints is used for starting, labelling, and executing of analysis. Moreover, this set includes endpoints for registering external tools that can run in the QMENTA platform.
- Project management endpoints
  - This set enables creating and updating project configuration: managing of basic project data and managing of users within the project (such as sharing project with other users and assigning them roles).

Additional functionality is provided by various modules within the main client application:

- Module for basic statistical analysis of the session data and their related metadata
- Module for project quality: This includes statistics for data homogeneity and QA coverage
- Module for visualization of various types of files such as DICOM / Nifti files, graphical images (e.g. jpg, png, and bmp), and CSV files.
- Module for visualization of analysis results: The QMENTA system offers a composer for organizing and configuring a pallet of widgets used to present different aspects of the analysis results.

2.1.4 Implementation of the Precise4Q data model

Before we describe the implementation of the P4Q data model inside the QMENTA platform, let explain the QMENTA data model. The QMENTA data model can be applied over various cases in which we observe events of interest (in which subjects of interest are involved) producing data that can be analysed.

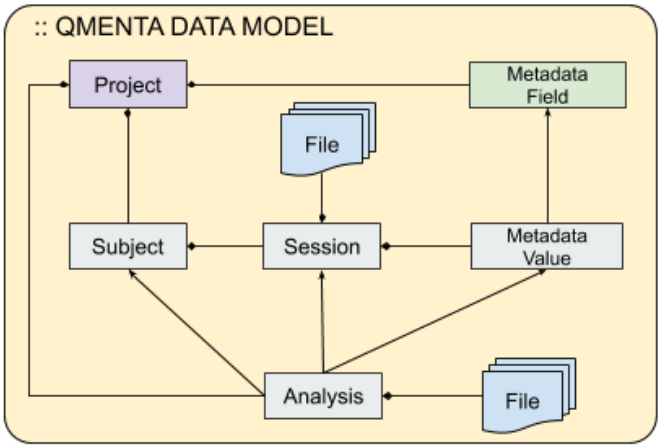

Figure 3 QMENTA Data Model

The QMENTA data model, as shown in Figure 3, is composed of the following entities:

- Project: The data of a study in the QMENTA system are organized in projects. The project has its own objective, events and subjects of interest. The project is the direct owner of the Subjects, Metadata Field definitions, and the Analysis performed over the data. The term study is used as a synonym.
- Subject: Each study observes any number of subjects. A subject can refer to a patient, but not only patients: it can refer to, for example, physical objects under investigation, or items produced in a factory.
- Session: A session is an event in which a subject is involved. The event is described with a number of metadata fields and their values. The term event can be used as a synonym for Session.
- Metadata Field: Metadata field is a definition of parameter of interest chosen on a project level.
- Metadata Value: This entity represents a value of a metadata field and it describes/measures some aspect of the event to which it refers.
- File: Files are data produced by the event. For example, the patient visits the hospital in order to make MRI scan. This produces MRI DICOM images that can be stored as files. Files can also represent unstructured data.
- Analysis: Events data, represented by the session data and its metadata values, can be analysed producing results in the form of outcome files, metadata fields, and their values.

The P4Q model offers a wide range of entities describing various types of statements outcome of different events (such as ClinicalProcedureStatement or ClinicalSituationStatement) involving subject(s) and actions. In order to store different kinds of statements, we extend the QMENTA data model (see Figure 4) with additional concepts such as Session Type, and we distribute the metadata fields across the session types. More specifically, for the P4Q project the session types will refer to phases (or further logical division) that measure (not necessarily the same) various parameters (i.e. metadata fields).

In addition, each metadata field can point to a particular coding system and a concept within that coding system. Regarding the P4Q project, a metadata field may refer to one or more SNOMED CT concepts.

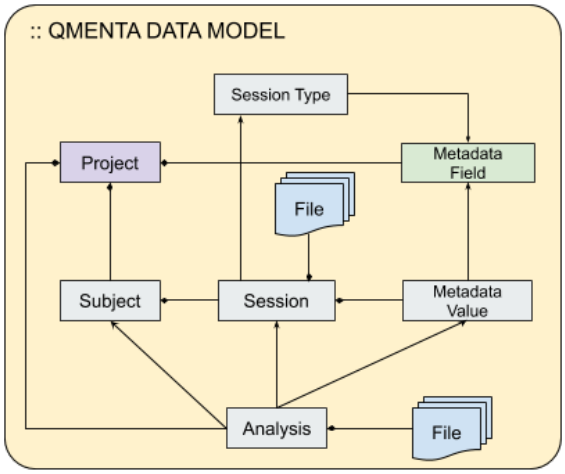

Figure 4 QMENTA Data Model after extension

At a first phase, the QMENTA platform will fit the structured and unstructured data in compliance with its own data model. However, its limited relational power will neglect some aspects of the data relations compared to the planned Semantic Data Warehouse and its underlying graph repository. The later one (i.e. Semantic Data Warehouse) as a model has the capacity to integrate the standardized ontologies, the standardized coding systems (in this case SNOMED CT) and the data itself, providing rich knowledge graph of data to be explored. This means that we will go from some-knowledge-loss model towards zero-knowledge-loss model. Moreover, the advanced search languages for graph databases will enable answering complex questions such as "finding all patients that had repeated stroke within a year if treated under some medication but has not been treated with other medication".

At a later phase, as mentioned before, the Semantic Data Repository will run in a separate execution environment (in this case a Docker container) as a service, and the QMENTA platform will use this service to search and fetch the data, instead of having them stored in its own data warehouse. The QMENTA platform will eventually provide a proxy for the Semantic Data Repository service thus using the overall security framework of the QMENTA platform.

2.2 Digital Stroke Patient data warehouse

A clinical data warehouse merges heterogeneous data sources in a central repository to provide input data for the different modules and to store their results.

It consists of different repositories and each one focuses on a different type of data (see Figure 1).

- *Raw data repository* contains the clinical datasets provided by the data providing partners
- *NLP repository* contains the outcome of the text analysis and the semantic mapping modules
- *Integrated and Harmonized repository* contains the integrated and harmonized data represented according to the P4Q ontology.

- *Machine learning model repository contains the training datasets and the different versions of the resulting models.*
- *Machine learning prediction repository contains the validation datasets as well as prediction results associated with the different versions of the models in Machine learning model repository.*

The ETL and NLP modules will access the raw data repository in order to perform the text analysis and the outcome will be stored into the NLP repository. The integration and harmonization module will perform the integration and harmonization task and store the results into the Integrated and harmonized repository in order to instantiate the semantic data warehouse.

The semantic data warehouse module supports more sophisticated built-in reasoning queries that extends the functionalities of the data warehouse. Finally, the ML modelling/prediction module will query the data warehouse and the semantic data Warehouse in order to obtain training and testing datasets and store the generated models in the ML modelling repository and the validation results in ML prediction repository.

2.3 ETL module

ETL stands for Extraction, Transformation and Load. Extraction means to connect to different data sources and extracting the relevant data for analysis and research. In the Transformation phase, extracted data is transformed into a specific format based on certain rules, procedures, etc. In the Load stage, data is stored into the data warehouse according to certain schema. The Digital Stroke Patient platform will reuse QMENTA ETL tools in order to, in a first stage load the source datasets into the data warehouse following the QMENTA data schema, and in a later stage support the data extraction and storing into the other repositories and into the Semantic Data Warehouse.

2.4 NLP module

The Natural Language Processing (NLP) module consists of the project text analyser (P4Q-TAE) (cf. Deliverable 3.5), where the main goal is to take input as *unstructured clinical* texts and produce *structured* / *annotated* texts. The extracted knowledge can then be mapped to the ontologies for information access and for potential utility as features for downstream predictive modelling. The granularity of the information extracted can be at document, sentence, multi-word, token or at character level (see Figure 5 for different modules). P4Q-TAE is mainly implemented in Python but is modular to make calls to the external modules. Currently, it takes unstructured data in CSV format and transform to an internal JSON representation [13]. The modules will be integrated into the Digital Stroke Patient platform as a Docker container [14].

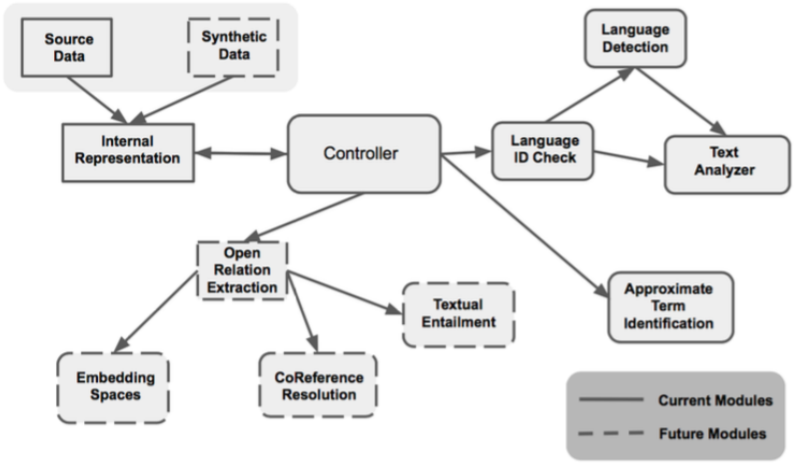

Figure 5. P4Q-TAE architecture

2.5 Semantic mapping module

The semantic mapping module is connected to the NLP module and maps the outcome of the text analysis and the structured data to the ontologies including data provenance information. The dictionaries created in WP3 will support this mapping process.

For structured data, manual mappings will be defined between the data source features and the ontology concepts. Unstructured and semi-structured data will be mapped to the ontologies using the P4Q-TAE text analyser.

2.6 Integration / harmonization module

Data mapped to the ontologies might differ in the use of scales, level of granularity etc. This module will bridge these heterogeneities supported by reasoning and rules (e.g. Glucose < 65 mg/dL = hypoglycemia) and will store the outcome data into the semantic data warehouse. The module will be integrated into the platform as a Docker container.

2.7 Machine Learning (ML) modelling module

The machine learning module will contain both the datasets and the models created through the activities carried out as part of WP4. The PRECISE4Q project has adopted an iterative approach to model development, with new versions of datasets and models created at each iteration through the development process. Consequently, for each modelling task, multiple dataset and model versions will be created and stored with the machine learning module. The creation, and updating, of dataset for model development will draw on the functionality afforded by the NLP, Semantic Mapping and Data Harmonization modules, and so the machine learning module will link with these other modules, frequently using them as part of the data pipeline used to query and retrieve data from the data warehouse. Similar to the NLP Module the machine learning module will be integrated into the Digital Stroke Patient platform as a Docker container [14].

2.8 Semantic Data Warehouse Architecture

Figure 6 depicts the architecture of the semantic data warehouse. The proposed solution will provide a maximum of interoperability and interfacing between QMENTA and open-source warehousing solutions, which includes the attempt to totally or partially host an open-source solution within the QMENTA Cloud Platform (Figure 1).

A graph-based repository based on the P4Q ontology implemented within WP3 (Task 3.2) will be built. The semantically harmonized data produced as outcome of Tasks 3.3-3.5 will be loaded into this repository.

One of the benefits of this graph-based repository is that it will allow a straightforward method for querying the warehouse content supported by some formal reasoning, and allowing performing more complex queries that take full advantage of the semantic relationships defined by the P4Q ontology, i.e. exploiting the hierarchy of concepts without the need of explicitly annotating each data entry with the full list of ancestor concepts. The query interface with reasoning functionality will be implemented in the semantic layer. It will facilitate users to query the graph repository using a more textual query and transform it to the query languages provided by the graph repository (i.e. SPARQL [15] and Cypher [16]) to consult the content of the data warehouse (Figure 6).

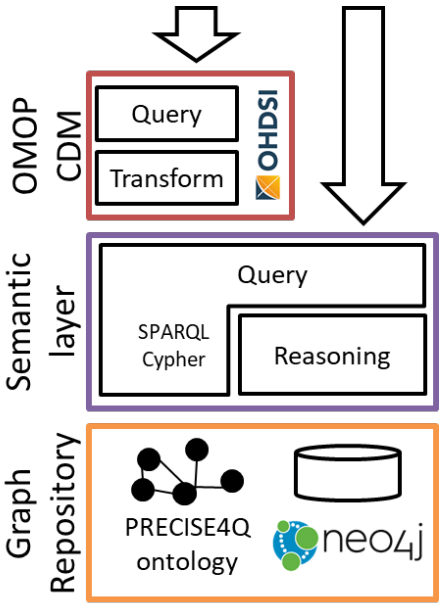

Figure 5 Semantic Data Warehouse architecture

2.8.1Information storage

Within the semantic data warehouse, the information storage architecture distinguishes the following layers, which correspond with the data repositories described in section 2.2.

- Raw data: This is the bottom layer consisting of the uploaded datasets into the QMENTA platform without being processed yet.
- Semantically Annotated data: On top of the raw data is the annotated data. This is the data obtained as outcome of the text analysis. It consists of text annotated with ontologies / terminologies.
- Ontologies / Semantic models: These are the inter-related ontologies required to represent the data from the different datasets. These ontologies can evolve based on the data sources needs. This corresponds to the P4Q ontology (cf. deliverables 3.1 and 3.2) which includes standardized ontologies like SNOMED CT [24] and other local ones created in the context of P4Q and based on users' needs.
- Harmonized and Integrated data: At the top layer is the data outcome of the harmonization and integration module and represented as instances of a knowledge graph built based on the previous ontologies / semantic models.

2.8.2 Graph database

For the implementation of the semantic data warehouse we have chosen a property graph database to model the proposed ontologies. A graph database allows the underlying data model to evolve. We use Neo4J [17] as technology as it is the current state of the art property graph database [18]

NoSQL databases and especially graph databases are increasingly gaining popularity as they provide better performance when handling interconnected data compared to relational databases and they are more flexible regarding the data model. This has become especially relevant with the growing size of available data and their increase in complexity (i.e. Big Data) [19].

In relational databases complex queries are usually built as complicated joins, resulting in low performance queries. In graphs, we talk about nodes connected by edges or relationships and properties. This kind of database is simpler and more powerful when the meaning is in the relationship between the data. Relational databases can easily handle direct relationships, but not

indirect ones. A graph is designed to traverse indirect relationships and still maintain the performance.

RDF databases or triple stores represent data in terms of subject-predicate-object triples. They are a type of directed graph database that among others, differ with them in the way they model the graph. In triple stores the "nodes" (i.e. subject and object) tend to be primitive datatypes while in graph databases they are containers that correspond to objects in a domain. They tend to be more an academic product with scalability problems while graph databases are more adopted in industry.

Neo4j is one of the most popular graph databases in areas such as health, government, automotive production, military area, among others [17]. It is open-source (Community edition) and is implemented in Java. It has its own query language named Cypher, a declarative language inspired by SQL.

In order to import the P4Q ontology and the data into Neo4j we have used the Neosemantix plugin [20]. This plugin allows the use of RDF in Neo4j together with some inferencing capabilities.

This plugin implements the following mapping rules:

- Subject of triples are mapped to Neo4j nodes. A node in Neo4j representing an RDF resource will be labelled :Resource and have a property uri with the resource's URI
- Predicates of triples are mapped to node properties in Neo4j if the object of the triple is a literal
- Predicates of triples are mapped to relationships in Neo4j if the object of the triple is a resource

Additionally RDF instances are linked to RDF classes by using categories (in RDF corresponds to rdf:type).

Neo4j allows not only to import the RDF data instances but also the ontologies to which they conform. Figure 6 depicts the class hierarchy of the SNOMED CT concept "Finding of life event (finding)".

- An owl:Class corresponds to a node label in Neo4j
- An owl:DatatypeProperty corresponds to an attribute in Neo4j
- An owl:ObjectProperty describes a relationship in Neo4j
- The rdfs:domain and rdfs:range properties specify the source class and the target class or XMLSchema datatype of the property

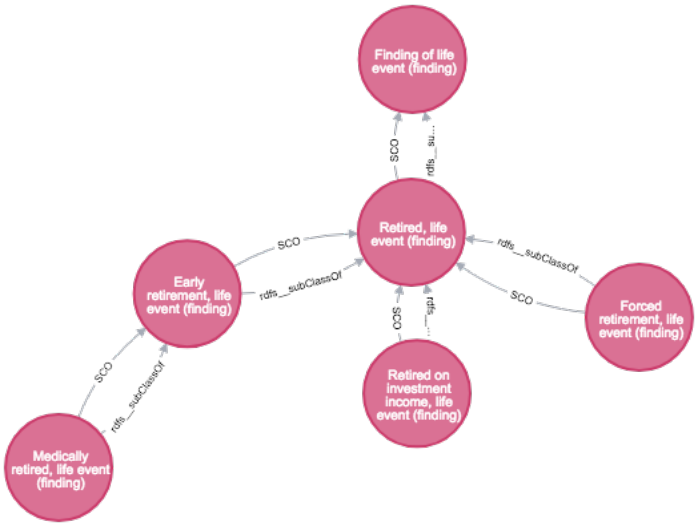

Figure 6 Excerpt of the SNOMED CT ontology. Concept hierarchy "Finding of life event (finding)". SCO stands for subClassOf

Figure 7 shows an excerpt of the graph-based representation of a data instance representing a clinical statement describing that a patient requires assistance with all daily activities. The statement corresponds to an instance of the following OWL expression according to the Precise4Q ontology using Manchester syntax [21] (OWL object properties in bold and OWL classes in italics):

sdm:ClinicalStatement
and **btl:represents** some *sct:RequiresAssistanceWithAllDailyActivitiesFinding*
and **btl:isOutcome** of some *sdm:ClinicalProcess*
and **btl:hasPart** some *sct:KnownPresent*
and **btl:hasPart** some *sct:CurrentOrSpecifiedTimeQualifierValue*
and **btl:hasPart** some (*sdm:InformationAboutSubjectOfInformation*
and **btl:represents** some *sct:SubjectOfRecordPerson*)

Table 1 OWL instance of a clinical statement about a patient that requires assistance with all daily activities (sct: prefix for the SCT ontology; sdm: prefix for the Semantic Data Model ontology; btl: prefix for the BioTopLite2 ontology)

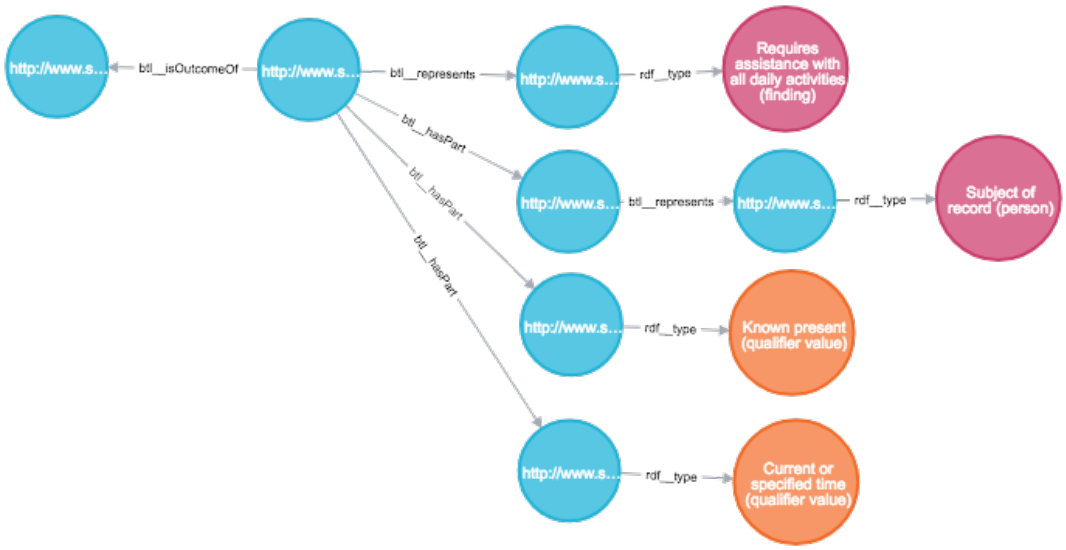

Figure 7 Excerpt of an instance of a clinical statement representing that the patient requires assistance with all daily activities. Orange and Purple nodes represent SNOMED CT concepts. Blue nodes are named RDF individuals

Neo4j allows some inferencing capabilities. By inference / reasoning we understand the process of getting information from Neo4j database that is not explicitly stored. As an example, if you have in Neo4j some nodes labelled as anticoagulant and some others as Heparin, which is a type of anticoagulant, then you would like that Neo4j DB would apply this reasoning on the fly and return both anticoagulant and heparin nodes when you query for anticoagulant. This functionality is implemented in Neo4j by the neosemantix plugin.

Within P4Q harmonized and integrated data will be stored in Neo4j in order to be queried by predictive modellers. For querying the data, Neo4j provides the Cypher language, however we will provide an easier query interface to the modellers.

2.8.3 Standardized interface, OMOP CDM

In order to provide a more standardized solution and align with a large and increasing community of researchers, health professionals, etc. a transformation module to export the content of the semantic data warehouse into a repository that implements the OMOP Common Data Model (CDM) from the OHDSI community will be built by reusing existing open-source tools provided by this community.

This is possible given the P4Q semantic data model focuses on representing data semantics and not data structure and thus is independent of any particular implementation.

The repository will be hosted first in the QMENTA Cloud Platform as a docker running internally in their infrastructure, but with the long-term perspective of being deployed in a dedicated server and managed by each partner interested in accessing the semantic functionalities. This development will be based on open-source solutions which allow its free use during the project and beyond. This could be of benefit for the long-term perspective of the platform and in case other data providing partners might join.

The Observational Health Data Sciences and Informatics (OHDSI) is an international collaborative whose goal is to create and apply open-source data analytics solutions to a large network of health databases to improve human health and wellbeing [22]. The OHDSI team comprises academics, industry scientists, health care providers, and regulators that among others aim to create reliable scientific evidence through large-scale analysis of observational health databases for population-level estimation and patient-level predictions [23]. OHDSI grew out of the Observational Medical Outcomes Partnership (OMOP) [24] in the US, created to inform about the appropriate use of observational healthcare databases for studying the effects of medical products.

A centrepiece of the OMOP project was the development of the OMOP Common Data Model (CDM) [2] to represent heterogeneous healthcare data in a standardized way. Once data has been converted to the OMOP CDM (see list of tables in Table 2), OHDSI open-source tools can be used to extract evidence.

Model Domain	**Table Names**
Standardized Clinical Data Tables	PERSON, OBSERVATION_PERIOD, VISIT_OCCURRENCE, VISIT_DETAIL, CONDITION_OCCURRENCE, DEATH, DRUG_EXPOSURE, PROCEDURE_OCCURRENCE, DEVICE_EXPOSURE, MEASUREMENT, NOTE, NOTE_NLP, SURVEY_CONDUCT, OBSERVATION, SPECIMEN, FACT_RELATIONSHIP
Standardized Health System Data Tables	LOCATION, LOCATION_HISTORY, CARE_SITE, PROVIDER
Standardized Health Economics Data Tables	PAYER_PLAN_PERIOD, COST
Standardized Derived Elements	DRUG_ERA, DOSE_ERA, CONDITION_ERA
Standardized Vocabularies	CONCEPT, VOCABULARY, DOMAIN, CONCEPT_CLASS, CONCEPT_RELATIONSHIP, RELATIONSHIP, CONCEPT_SYNONYM, CONCEPT_ANCESTOR, SOURCE_TO_CONCEPT_MAP, DRUG_STRENGTH

Table 2 Tables in the OMOP Common Data Model v6.0

In the following a brief description of some of the open-source tools offered by the OHDSI community and that could be of interest for P4Q:

- Achilles: a standardized database profiling tool for database characterization and data quality assessment [25]
- Athena: web application for distributing and browsing the standardized vocabularies for the OMOP CDM [26]

- Patient level prediction: An R package for building patient level predictive models using data in Common Data Model format [27]

3 Conclusions

In this deliverable we have provided a description of the architecture of the P4Q data warehouse. An implementation in two phases has been described. The first phase will last until approx. month 22 of the project after which will start the implementation of the semantic data warehouse. The transition between both data warehouses will follow a pragmatic solution where some of the semantic functionalities might be implemented in QMENTA Cloud Platform for specific use cases when required. The semantic data warehouse will allow to export the data using the OMOP CDM format provided by the OHDSI community in order to provide a more standardized solution and align with a large and increasing community of researchers, health professionals, etc. Both, the semantic data warehouse and the export functionality to the OMOP CDM will be provided as open source tools. For a long-term scenario, we plan to have both data warehouses as independent platforms, where each data providing partner will be responsible for managing the graph-based data warehouse internally.

References

1. QMENTA Platform. https://www.qmenta.com/qmenta-labs/ (accessed October 2019)
2. OMOP Common Data Model v6.0 Specifications. https://github.com/OHDSI/CommonDataModel/wiki (accessed October 2019)
3. OHDSI Vision. http://ohdsi.org/who-we-are/mission-vision-values/ (accessed October 2019)
4. Atchinson, Brian K.; Fox, Daniel M. (May–June 1997). "The Politics Of The Health Insurance Portability And Accountability Act". Health Affairs. 16 (3): 146–150. doi:10.1377/hlthaff.16.3.146.
5. TITLE 21--FOOD AND DRUGS ADMINISTRATION, PART 11 -- ELECTRONIC RECORDS; ELECTRONIC SIGNATURES. Available online (Last accessed October 2019): https://www.ecfr.gov/cgi-bin/text-idx?SID=3ee286332416f26a91d9e6d786a604ab&mc=true&tpl=/ecfrbrowse/Title21/21tab_02.tpl
6. MongoDB. Available online https://www.mongodb.com/ (accessed October 2019)
7. RFC2818 - HTTP Over TLS (HTTPS). https://tools.ietf.org/html/rfc2818 (accessed October 2019)
8. RFC5246 - The Transport Layer Security (TLS) Protocol Version 1.2. https://tools.ietf.org/html/rfc5246 (accessed October 2019)
9. RFC8422 - Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier. https://tools.ietf.org/html/rfc8422 (accessed October 2019)
10. RFC5288 - AES Galois Counter Mode (GCM) Cipher Suites for TLS. https://tools.ietf.org/html/rfc5288 (accessed October 2019)
11. US-EU Privacy Shield. https://www.privacyshield.gov/ (accessed October 2019)
12. Digital Imaging and Communications in Medicine - DICOM. https://www.dicomstandard.org/current/ (accessed October 2019)

13. The JSON Data Interchange Syntax. https://www.ecma-international.org/publications/standards/Ecma-404.htm (accessed October 2019)
14. Docker container. Available online (last accessed October 2019): https://www.docker.com/
15. W3C SPARQL 1.1 Query Language. Available online (last accessed October 2019): https://www.w3.org/TR/sparql11-query/
16. Cypher, The Graph Query Language. Available online (last accessed October 2019): https://neo4j.com/cypher-graph-query-language/
17. The Neo4j Graph platform. Available online: https://neo4j.com (accessed October 2019)
18. DB-Engines Ranking. Available online: https://db-engines.com/de/ranking/graph+dbms (accessed October 2019)
19. Raghupathi, W., Raghupathi, V., 2014. Big data analytics in healthcare: promise and potential. Health Inf. Sci. Syst. 2 (1), 3.
20. Neosemantics plugin. https://github.com/neo4j-labs/neosemantics (accessed October 2019)
21. OWL 2 Web Ontology Language Manchester syntax. https://www.w3.org/TR/owl2-manchester-syntax/ (accessed October 2019)
22. Hripcsak, George, et al. "Observational Health Data Sciences and Informatics (OHDSI): opportunities for observational researchers." Studies in health technology and informatics 216 (2015): 574.
23. OHDSI Vision. (Accessed October 2019) http://ohdsi.org/who-we-are/mission-vision-values/
24. Overhage, J. Marc, et al. "Validation of a common data model for active safety surveillance research." Journal of the American Medical Informatics Association 19.1 (2011): 54-60.
25. Achilles. https://www.ohdsi.org/web/wiki/doku.php?id=documentation:software:achilles (accessed October 2019)
26. Athena. http://athena.ohdsi.org/search-terms/terms (accessed October 2019)
27. OHDSI Methods library. https://www.ohdsi.org/methods-library/ (accessed October 2019)