Personalised Health Monitoring and Decision Support Based on Artificial Intelligence and Holistic Health Records

D3.7 – Standardisation and Quality Assurance of Heterogenous Data I

WP3 Personalised Holistic Health Records

Dissemination Level: Public
Document type: Report
Version: 1.0
Date: October 29, 2021

The project iHelp has received funding from the European Union's Horizon 2020 Programme for research, technological development, and demonstration under grant agreement no 101017441.

Document Details

Project Number	101017441
Project Title	iHelp - Personalised Health Monitoring and Decision Support Based on Artificial Intelligence and Holistic Health Records
Title of deliverable	Standardisation and Quality Assurance of Heterogenous Data I
Work package	WP3 Personalised Holistic Health Records
Due Date	October 31, 2021
Submission Date	October 29, 2021
Start Date of Project	January 1, 2021
Duration of project	36 months
Main Responsible Partner	UPRC
Deliverable nature	Report
Authors' names	George Manias (UPRC), Usman Wajid (ICE), Athanasios Dalianis (ATC), Eleftheria Kouremenou (UPRC), Ainhoa Azqueta (UPM)
Reviewers' names	Maritini Kalogerini, Athanasios Dalianis (ATC), Harm op den Akker (iSPRINT)

Document Revision History

Version History			
Version	**Date**	**Author(s)**	**Changes made**
0.1	2021-08-25	George Manias (UPRC)	Initial version and Table of Contents
0.2	2021-09-13	George Manias, Eleftheria Kouremenou (UPRC)	Initial content in Summary, Conclusion, Sections 1-3
0.3	2021-09-16	George Manias (UPRC)	Initial content in Sections 4-5
0.4	2021-09-30	Usman Wajid (ICE), Athanasios Dalianis (ATC), Ainhoa Azqueta (UPM)	Contributions in Sections 5, 6 and 7
0.5	2021-10-05	George Manias (UPRC)	Input Update and Internal Review, Annex A added
0.6	2021-10-19	Maritini Kalogerini (ATC), Athanasios Dalianis (ATC), George Manias (UPRC)	1st Internal Review and revision of the document
0.7	2021-10-22	Harm op den Akker (iSPRINT)	2nd Internal Review and revision of the document
1.0	2021-10-25	George Manias (UPRC)	Final version

Table of Contents

Table of Figures

Executive summary

This deliverable (titled “Standardisation and Quality Assurance of Heterogenous Data I”) describes the initial design and specifications of the iHelp Standardisation and Quality Assurance Mechanism. The Standardisation and Quality Assurance Mechanism will be a unified and integrated mechanism consisting of three (3) core sub-components, the Data Cleaner, the Data Qualifier, the Data Harmonizer, and two (2) integrated sub-components: the Primary Data Mapper and the Secondary Data Mapper, which will be responsible for providing the mapping operations between the raw data resources and the Holistic Health Records (HHRs) resources. This holistic mechanism is the core component that seeks to provide various cleaning, pre-processing and mapping functionalities and services on the incoming raw data. Specifically, it will provide to the wider research and innovation community a wide range of solutions for the cleaning, qualification, transformation, harmonization and mapping of raw healthcare-related data.

The current document aims to provide the initial design, the specifications and a concrete overview of how the proposed mechanism integrates with the overall architecture of the iHelp platform and other components in the Data Ingestion block, and specifically (i) how to retrieve the incoming data from the Data Gateways, (ii) how to interexchange data with the already identified project’s message bus and (iii) how to send the final processed, transformed and HHR aligned data to iHelp’s Data store solution.

To support the aforementioned functionalities, the iHelp Standardisation and Quality Assurance Mechanism specifications exemplify the respective sub-components, the overall data pipeline and workflow, the internal functionalities supported by each sub-component, the interaction points with different components as well as the technical details that will drive the implementation of this holistic mechanism. Finally, this document - in different subsections, e.g. in Section 5.1, - seeks to review the current state of the art in order to identify the baseline technologies and approaches for the realization of the implemented Standardisation and Quality Assurance Mechanism.

1 Introduction

iHelp aims to develop standardisation and quality procedures to ensure that the data modelling, transformation, and management operations will facilitate data sharing and analytics not only in the context of the iHelp project but also in the wider healthcare ecosystem. Given the challenge that in modern societies healthcare-related data are being obtained from various data sources and in divergent formats, this task and deliverable aim to provide technologies for dealing with this issue through the harmonization and transformation of the raw collected health data into the project's common HHR format, through finding common links or similarities between primary and secondary data types and available HHR resources. Moreover, these raw data also include missing values, non-matching words and partially overlapping concepts, hence state-of-the-art cleaning approaches and techniques aim to be utilized under the scopes of this task to provide cleaned and qualified data.

In summary, the Standardisation and Quality Assurance Mechanism will consist of five (5) sub-components with many different internal architectures and functionalities, in order to cover all the different requirements of the project's stakeholders and the overall data processing and data mapping procedures that will be implemented. In the following Sections of this document, the design, the internal architecture and the specifications of the iHelp Standardisation and Quality Assurance Mechanism will be analysed in detail.

1.1 Objective of the Deliverable

The main objective of this deliverable is to provide the ground for the realization of the iHelp Standardisation and Quality Assurance Mechanism and to report the work that has been conducted in the context of task T3.4 ("Standardisation and Quality Assurance of Heterogeneous Data") at this phase of the project. On top of this, it will introduce and analyse the internal architecture and data workflow, the initial design and the specifications of this holistic mechanism, and its development will be performed based on them. In addition, the deliverable will outline the main sub-components and their internal functionalities and clarify the reason for their implementation and existence. Finally, this deliverable seeks to describe the overall positioning of the Standardisation and Quality Assurance Mechanism into the iHelp platform and analyse its integration and relation with other project's technical components and mechanisms.

1.2 Structure of the Deliverable

This document is structured as follows: Section 1 introduces the deliverable and its main objectives, while Section 2 introduces the initial architecture, the objectives and the overview of the Standardisation and Quality Assurance Mechanism, discussing its scope and the basic concepts and key features of its design. Afterwards, the next sections introduce the five (5) main sub-components of the Standardisation and Quality Assurance Mechanism. Specifically, Section 3 covers the data cleaning, data verification and data validation aspects focusing on the Data Cleaner sub-component, Section 4 focuses on the data qualification functions of the Data Qualifier sub-component, while Section 5 focuses on transformation and harmonisation solutions to support interoperability and transformation of incoming data through the analysis and description of the Data Harmonizer sub-component. On top of this, Section 6 and Section 7 focus on the primary data mapping and the secondary data mapping functionalities respectively, hence the sub-components of the Primary Data Mapper and the Secondary Data Mapper, that are incorporated and strongly integrated with the Data Harmonizer sub-component, will be further analysed. Finally, Section 8

concludes the document and states any future work and deliverable concerning the task T3.4 (“Standardisation and Quality Assurance of Heterogeneous Data”).

2 Standardisation and Quality Assurance Mechanism

This section describes the scope and the overall architecture of the iHelp Standardisation and Quality Assurance Mechanism, while it also places this holistic mechanism into the iHelp data ingestion pipeline and further analyses its correlation and integration with other technical components and mechanisms of the project.

2.1 Overview

Nowadays, the healthcare domain faces various challenges related to the diversity and variety of data, their huge volume, and their distribution, thus processing and analysis of these data become more and more complex and challenging. Hence, approaches, applications and solutions to address issues that derive from the wealth of Big Data are vitally important. The collection, the quality estimation, as well as the interpretation and the harmonization of the data, that derive from the existing huge amounts of heterogeneous medical devices and data sources, face a dramatic increase of interest in the healthcare domain. To this end, to address all these issues this specific task of the iHelp project has four objectives and targets. On one hand it seeks to assure the incoming data accuracy, integrity, and quality, while on the other hand it seeks to perform different types of transformation, interoperability, and integration operations on the raw data. On top of this, this task aims – through the utilization of specific measures and rules – to ensure data quality and to provide a predictive selection mechanism for achieving data sources' reliability during runtime and for providing the decision whether a connected data source will be considered as reliable or not. Finally, advanced data mappers will be introduced and implemented to provide an automated structure mapping mechanism between data resources and widely known and approved in the healthcare domain HHR resources and data model (K., M., M., + 19) following the mapping approaches that will be identified under the scopes of task T3.1 ("Data Modelling and Integrated Health Records") of the iHelp project.

2.2 Internal Architecture

As presented in the above sub-section the Standardisation and Quality Assurance Mechanism seeks to enhance the quality of the incoming data, the interoperability and harmonization of them and the extraction of valuable information and knowledge out of them. To this end, three (3) core sub-components have been identified and are incorporated into the internal workflow and architecture of the Standardisation and Quality Assurance Mechanism. These specific three (3) subcomponents are being depicted in Figure 1 below and are the Data Cleaner, the Data Qualifier and the Data Harmonizer. The implementation and utilization of the above-mentioned components aim to enhance the value of the incoming/raw data.

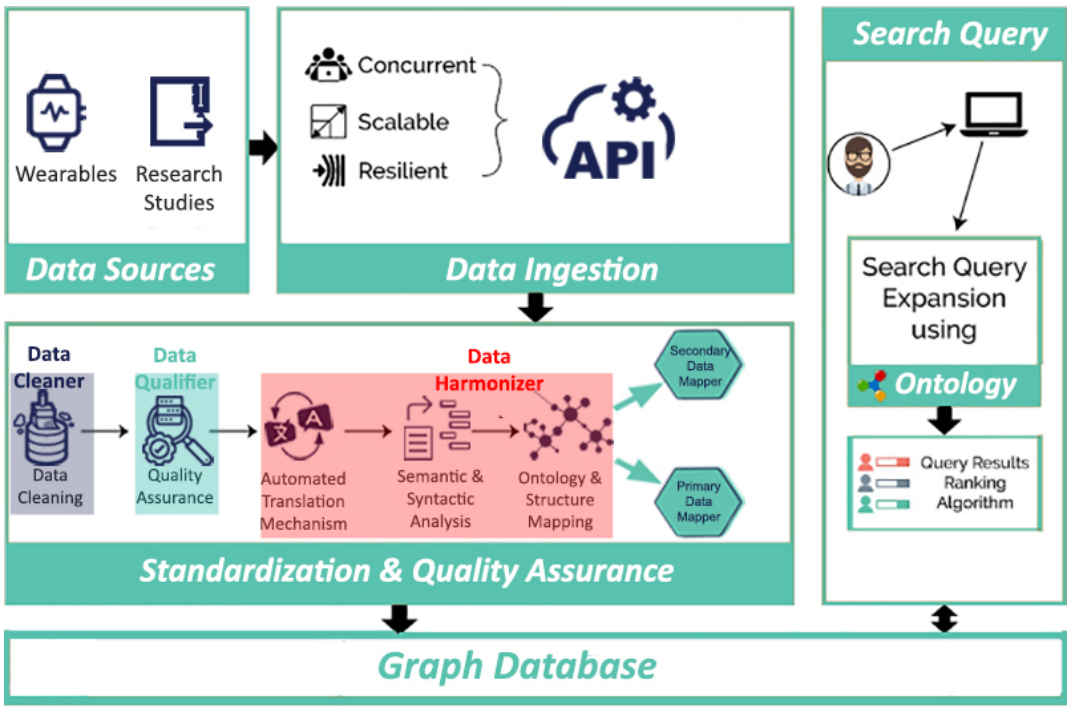

Figure 1: Standardisation & Quality Assurance mechanism.

Following the step of the Data Ingestion, one of the preliminary steps of this mechanism is to deal with data derived from unreliable data sources, or from reliable data sources that may have produced uncleaned and faulty data. Thus, from the very beginning of the overall processing pipeline it aims to clean all the collected data and to measure and evaluate the quality of both the connected data sources and their produced data, so as to finally keep only the reliable data that comes from only reliable data sources. To successfully achieve that, the mechanism exploits two (2) separate modules, the Data Cleaner sub-component and the Data Qualifier sub-component. Sequentially, in the harmonization phase, the interpretation and transformation of the collected cleaned and reliable data takes place through the implementation and utilization of the Data Harmonizer. The latter incorporates three (3) sub-mechanisms, the Automated Translation Mechanism, the Semantic and Syntactic Analysis mechanism and the Ontology and Structure Mapping Mechanism, in order to transform the cleaned and reliable data and to provide translated and syntactic and semantic interoperable data. Finally, the transformed and interoperable data are mapped to the common HHR format through the utilization of the Primary and Secondary Data Mapper, external sub-components and closely integrated with the Data Harmonizer, as also depicted in Figure 1 above. The specifications, internal architectures and implementation procedures of all the above introduced sub-components will be further explained and detailed in the next sections of this deliverable. At this point, it should be noted that this deliverable includes only a brief introduction and description regarding the Primary and Secondary Data Mapper sub-components, as these sub-components will start being designed and implemented after the final modelling and establishment of the common HHR model and format that will be followed across the iHelp project and platform and that will be introduced in the context of task T3.1. Moreover, as the project progresses and in collaboration and alignment with the parallel work that is being implemented under the scopes of T3.1, more details about the implementation and architecture of these two sub-components will be outlined.

2.3 Overall Objectives

Data have long been a critical asset for organizations, businesses, and governments and their analysis is of major importance for every stakeholder in order to be able to handle and extract value and knowledge out of them. The advances in the fields of IoT, cloud computing, edge computing and mobile computing have led to the rapidly increasing volume and complexity of data, thus the concept and term of Big Data have experienced enormous interest and use over the last decade. The spectacular growth in the creation, storage, sharing and consumption of data during the last decade indicates the need for modern organizations to fuse advanced analytical techniques with Big Data in order to deal with them and to get significant value from them. Hence, Big Data and their analysis facilitate personalised healthcare and risk assessment, therefore clinicians are able to identify patients who are eligible for appropriate treatments, which results in savings of time and cost. On top of this, R&D on personalised healthcare makes diagnostics smarter and more targeted, like in the case of Pancreatic Cancer, while early identification and personalised treatments can help in the design of improved screening programs and can also allow people to live longer, healthier and more productive lives. The ability to identify which preventative measure and intervention is delivering the desired impact can massively help in the development of new diagnostic and treatment regimes. Especially when it comes to the healthcare domain, the successful exploration and interpretation of all these data play a vital role (J., F. + 20). On top of this, healthcare data are available in different formats (e.g. images, signals and wavelengths) and may derive from different healthcare stakeholders (i.e. patients themselves, healthcare professionals, etc.). Hence, many healthcare organizations find themselves overwhelmed with data, but lacking truly valuable information. At the same time, due to the improvement in the automatic collection of data from medical devices and systems, researchers and analysts can monitor data or information that can be accessed in electronic configuration (Pan. 16).

Moreover, the term *Big Data* defines a two-fold meaning in these data. On one hand, it describes a change in the quality and type of data that modern healthcare organizations possess, which has potential impacts throughout the entire healthcare domain and stakeholders. On the other hand, it describes a massive volume of both structured and unstructured data that is huge and complicated to be processed using traditional database and software techniques (C., P. 14). On top of this, unstructured data can be defined as data that do not comfort in predefined data models and traditional structures that can be stored in relational databases. Data generated by medical reports, medical advice, texts in questionnaires or even posts on modern social networks are such types of unstructured data and their main characteristic is that they include information that is not arranged according to a predefined data model or schema. Therefore, these types of data are usually difficult to be managed, and as a result analysing, aggregating, and correlating them in order to extract valuable information and knowledge is a challenging task. Hence, deriving value and knowledge from this type of data based on the analysis of their semantics, meanings and syntactic is of major importance (M., B. + 10). The latter demonstrates the need for the modern stakeholders in the healthcare domain to implement techniques, mechanisms, and applications that focus their operations on the concept of providing cleaned, qualified and interoperable data, for offering more precise and personalised prevention & intervention measures, higher experience for patients' health monitoring, risk assessment and personalised decision support, hence actionable and valuable knowledge as depicted in Figure 2.

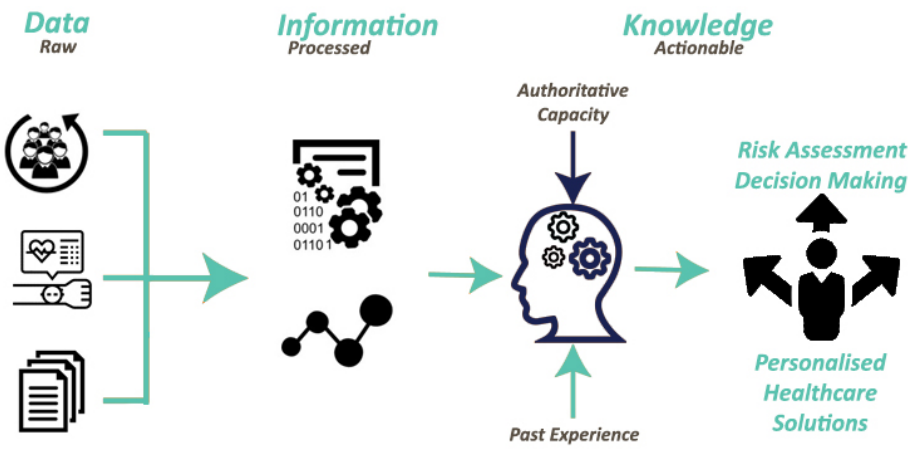

Figure 2: Data to Knowledge path.

To this end, Standardisation & Quality Assurance mechanism is the key mechanism for addressing all the above issues and challenges and for deriving actionable knowledge from the raw data. The initial design of this holistic mechanism is based on the following basic objectives, to be addressed through the implementation of the different sub-functions that have been introduced in previous sub-sections.

- *Data Cleaner* aims to assure the incoming data's accuracy, integrity, and quality.
- *Data Qualifier* aims to provide a decision whether a connected data source will be considered as reliable or not.
- *Data Harmonizer* aims to support data coming from divergent sources in order to deal with different formats and to enhance the interoperability of data.
- *Data Harmonizer* aims to provide automated health data transformation to the identified HHR format.
- *Data Harmonizer* aims to consolidate data physically and virtually into knowledge graphs.
- *Data Harmonizer* aims to provide automated translation mechanisms in order to deal with different languages.
- *Primary Data Mapper* aims to provide a structure mapping mechanism between primary data resources and HHR data format.
- *Secondary Data Mapper* aims to provide a structure mapping mechanism between secondary data resources and HHR data format.

2.4 Positioning and Relations with other components

As presented in Section 6 of the D2.4 "Conceptual model and reference architecture I", the iHelp platform consists of four main big blocks that represent core functionalities and solutions that the iHelp project aims to provide. One of these main blocks is the Data Ingestion block, which represents all those components that are used to fetch, process and store data derived from heterogeneous sources. Processed data will comply with the common HHR data model and will be stored into the project's central Data Storage.

Moreover, as it has been introduced by the above sub-sections of this deliverable, the Standardisation and Quality Assurance Mechanism has strong integration and dependencies with various components that are part of the overall data ingestion pipeline, which is presented in Figure 3 below. At first, raw data from heterogeneous data sources should be fetched into the iHelp platform. Once the data are profiled and

encoded into the appropriate data schema, they are ingested either as batches or through a streaming process. The overall ingestion process is being accomplished through the utilization of Data Connectors and the Data Gateway of the project that will be designed and implemented under the scopes of task T3.2 "Primary Data Capture and Ingestion". Once data are extracted from source systems, their structure or format need to be adjusted, therefore these raw data should be further cleaned, processed and harmonized. To this end, the Standardisation and Quality Assurance Mechanism and its sub-components will be utilized. Then, quality assured, cleaned and harmonized data should be mapped and transformed into the project's common format and model, the HHR. Hence, the Primary and Secondary Mapper sub-functions that will, also, be implemented under the scopes of this task, will rely on the specification of the common data model, the HHR, that will be defined under the scopes of T3.1 ("Data Modelling and Integrated Health Records"). Finally, the HHR Importer mechanism, which will be provided by the T4.4 ("Big Data Platform and Knowledge Management System"), is responsible to store data into the relational data schema of the iHelp's Big Data Platform.

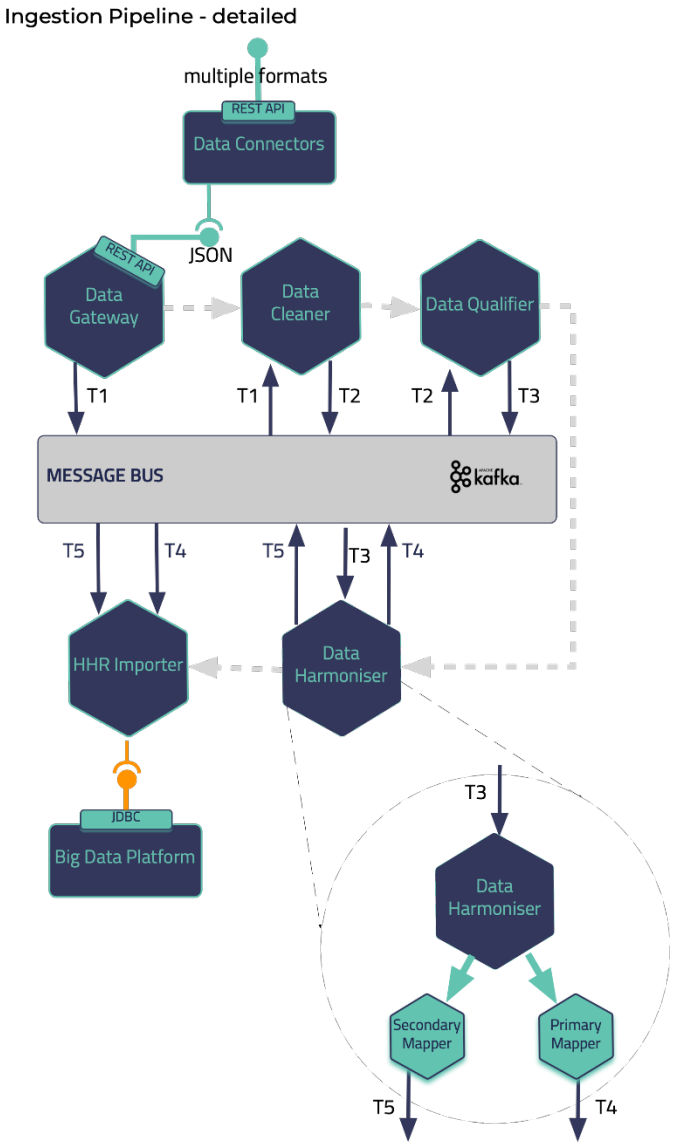

Figure 3: Details of the iHelp ingestion pipeline.

What is more, as depicted in Figure 3 above, all the components that are being incorporated into the data ingestion pipeline of the iHelp platform will inter-exchange their input and output data through the utilization of specific Kafka queues. Kafka has been identified as the project's internal message bus and data inter-exchange system. The latter implies that each of these sub-components involved into the data pipeline will consume data from one Kafka topic and will produce the output into another Kafka topic, so that the next involved sub-component can retrieve the processed data.

At this point, it should be noted that as the implementation of the Standardisation and Quality Assurance Mechanism and its sub-components has started recently and it is yet under development, while its functionality and performance have not been validated by an end-user, the input and output parameters, the internal workflows and functionalities will be further revised, updated, and extended and will be reported at the next versions of this deliverable. Finally, all the above introduced sub-components will offer the ability to be utilized either asynchronously or synchronously depending on the deployment status. To this end, corresponding REST APIs interfaces and endpoints for the majority of these sub-components will be introduced, while also their integration with the iHelp's Kafka message bus will be outlined.

3 Data Cleaner

The Data Cleaner sub-component will be utilized as an integrated microservice in the overall iHelp project, and its main objective is to deliver the software implementation that will provide the assurance that the provided data coming from several heterogeneous data sources will be clean and complete, to the extent possible. This microservice will be designed to minimize and filter the non-important data, thus improving the data quality and importance. To address a portion of these challenges, referring mainly to reducing the complexity and facilitating the analysis of large datasets, data cleaning procedures can improve the data quality and lead to better analysis outcomes, since wrong data can drive an organization to wrong decisions, and poor conclusions.

To this end, this sub-component seeks to assure the incoming data's accuracy, integrity, and quality. The Data Cleaner microservice will be utilized for every new incoming dataset in the platform since it seeks to detect and correct (or remove) inaccurate or corrupted data from the datasets. The input to this microservice will be provided by the shared message bus which will be utilized in the scopes of the iHelp project (e.g. Kafka). The topic from which data in the format of messages will be consumed can be set either dynamically, as a parameter whenever the microservice is called, or statically, based on an agreement in case that it is preferred. The input message will be the whole dataset in order to allow the microservice to provide all the necessary cleaning actions in order to produce consistent and cleaned data and datasets. Finally, the Data Cleaner will act also as a producer providing cleaned data to another Kafka topic, in order cleaned data to be passed to the rest sub-components of the data ingestion pipeline.

Specifically, the Data Cleaner workflow comprises of three discrete, but integrated steps, each one of them being provided as an individual sub-function. These sub-functions are being depicted in Figure 4 below and are further presented in the following sub-sections.

Figure 4: Data Cleaner internal workflow.

The architecture and design of the Data Cleaner sub-component seeks to address the volatility of the incoming data information towards the aim of providing data accuracy, consistency, and completeness to the iHelp platform. Thus, the Data Cleaner sub-component implements all the processes that identify inaccurate or corrupted datasets that may contain incorrect, incomplete or irrelevant data elements and consequently replace, modify or delete these data elements safeguarding the reliability and appropriateness of the incoming data information. The software prototypes of the Data Cleaner sub-component will be driven by this specification. Moreover, in order to facilitate the overall cleaning functions and procedures and to collect and identify specific pilot needs concerning the data schemas, data constraints and cleaning actions of the different pilots' datasets, a document has been circulated to each pilot. This document, which was introduced during one of the iHelp consortium meetings, is a live

document, and is being introduced in the Annex A – Cleaning Action and Constraints where a corresponding example template for cleaning actions and constraints concerning Pilot#1 of the iHelp project in being presented.

To this context, the Data Cleaning component is composed by one main service, namely the *DataCleaningService*, which in turn consists of three internal services: the *ValidationService*, the *CleaningService*, and the *VerificationService*. The main service, *DataCleaningService*, handles all the incoming and outgoing traffic of the Data Cleaner sub-component and is the only service exposed to the rest of the platform components. Contrary to the main service, the three internal services are not exposed to the rest of the platform components, whereas the *DataCleaningService* is interacting with these services internally to realize the overall data cleaning workflow. The main service of the Data Cleaner is introduced below, while the provided functionalities of the services are further depicted in the following sub-sections.

- ***DataCleaningService***: It is the main service of the Data Cleaner sub-function, which is in charge of executing the data cleaning workflow of iHelp platform. Since the data cleaning workflow comprises of several sequential steps, each one implemented by one of the internal services of the component, the *DataCleaningService* is responsible for the orchestration of these internal services as well as for monitoring their execution, and thus providing the execution results to the requestor. In addition to the data cleaning workflow execution, the *DataCleaningService* is responsible for the implementation of the single interface for data cleaning requests.

3.1 Data Validator

The Data Validator sub-function performs the data validation functionality with the purpose of identifying errors associated with the conformance to a specific set of constraints and schemas. This sub-function will ensure that the other iHelp microservices will operate on clean, correct and useful data. Therefore, the Data Validation service will perform data validation of the incoming information data with the purpose of identifying errors based on conformance to a specific set of constraints. As depicted in Figure 5 this sub-function incorporates two specific steps. At first, it receives as input the data along with the needed data schema (Data Load & Data Schema), in order then to be able to review the conformance of the provided data to their schema and provide the corresponding validation reports and identify possible errors (Review Data, Validation Reports). For implementing the above-mentioned steps two specific python tools and libraries, the PythonOnWheels[1] and the Cerberus[2], will be used.

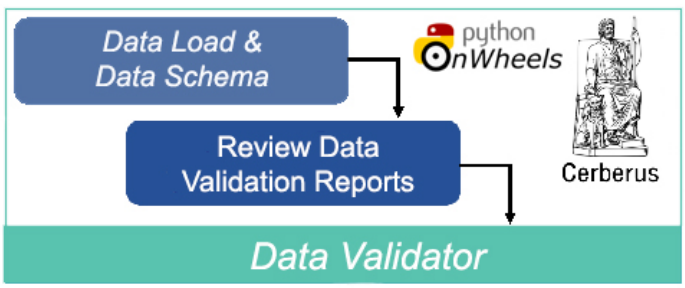

Figure 5: Data Validator Conceptual Diagram.

[1] https://www.pythononwheels.org/
[2] https://docs.python-cerberus.org/en/stable/

- ***ValidationService***: the internal service responsible for the data validation processing of the incoming data. The *ValidationService* performs a series of validation checks in order to evaluate the conformance to a set of constraints currently integrated in the business logic of the service. The current list of validation rules includes the following:
  - Conformance to specific data type.
  - Conformance to mandatory fields.
  - Conformance to specific value length.
  - Conformance to specific value representation.
  - Conformance to specific value range.
  - Identification of duplicate values for the data elements.
  - Identification of duplicate data elements.

At this point, it should be noted that the list of the validation rules will be furtherly enriched as the project evolves.

3.2 Data Cleanser

The Data Cleanser sub-function seeks to correct or remove all the data elements for which validation errors were raised, considering missing, irregular, unnecessary, and inconsistent data. This sub-function entails the main sub mechanism of the Data Cleaner sub-mechanism and its main goal is to correct or remove all the data elements for which validation errors were raised, considering missing, irregular, unnecessary, and inconsistent data. Thus, the Data Cleanser sub mechanism will perform the necessary corrections or removal of errors identified by the *ValidationService*. Under this scope, several steps of the overall cleaning process will be implemented, such as Parsing, Collection, Standardizing, Matching and Consolidation that are also depicted in Figure 6. For implementing the above-mentioned steps several python tools and libraries, such as Keras[3] and scikit-learn[4], will be used.

Figure 6: Data Cleanser Conceptual Diagram.

- ***CleaningService***: the internal service responsible for the cleaning of the incoming data. The *CleaningService* eliminates the list of errors identified by the *ValidationService* by applying all the necessary corrective actions on the data elements marked with errors. Cleaning is performed in an automated way based on a set of actions currently integrated in the business logic of the component. The current list of cleaning actions includes the following:
  - Deletion (drop) of the complete record (row).
  - Replacement of data element's value with the mean value.

[3] https://keras.io/
[4] https://scikit-learn.org/stable/

- Replacement of data element's value with the maximum value.
- Replacement of data element's value with the minimum value.
- Replacement of data element's value with the most frequent value.

At this point, it should be noted that the list of the cleaning actions will be further enriched as the project evolves.

3.3 Data Verifier

The Data Verifier sub-function aims to check the data elements of a dataset for accuracy and inconsistencies and to verify the compliance to the identified iHelp's HHR data models. The main objective of this sub-function is to check the data elements of a dataset for accuracy and inconsistencies after the steps of data validation and cleaning are performed. To this end, it will ensure that all the corrective actions performed by the *CleaningService* will be executed in compliance with the data models design of the iHelp platform. To this end, this service - through the Accuracy and Consistency step as depicted in Figure 7 - seeks to ensure that data will accurately be corrected or completed, and that the dataset will eventually be error free.

Figure 7: Data Verifier Conceptual Diagram.

- ***VerificationService***: the internal service responsible for the verification and evaluation of the corrective actions undertaken by the *CleaningService*, aiming to ensure the accuracy and the consistency of the updated incoming data according to the iHelp platform requirements.

Based on the aforementioned, the *VerificationService* checks and confirms that the *CleaningService* has successfully performed all the needed cleaning actions, returning a *null* list since no further corrective actions are needed to take place.

3.4 Interface

Besides the overall asynchronous integration with the Kafka message bus, the Data Cleaner component can be utilized through a provided API endpoint. The incoming HTTP requests are handled by the *DataCleanserController* while the execution of the workflow is performed by the *DataCleaningService*. Through the provided interface the following endpoint is offered: *Data Cleaner endpoint*. This endpoint is responsible for handling the data cleaning workflow execution through a POST request and providing the cleaned data as a result of the execution. The Data Cleaner endpoint expects the dataset for which the data cleaning workflow will be executed in CSV format, accompanied by its corresponding data schema. The Data Cleaner endpoint is documented below:

```
{
  "dataset": "DatasetXYZ.csv",
  "topic": "topic name, or can be ommited in case of the default",
  "schema": "the schema of the dataset as a JSON object and as provided by the Data Gateways",
```

```
  "mandatory": "list the mandatory fields/attributes as a JSON object",
  "allow_empty": "true/false",
  "cleaning_rules": "cleaning rules as provided by the pilot partners as a JSON object ",
}
```

3.5 Baseline Technologies

The Data Cleaner sub-component has started to being developed based on the utilization of Python 3.7 and the Flask[5] python micro web framework. Flask is a powerful framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine that is independent from particular libraries or tools and that supports a large list of extensions for application features. Besides the Flask framework, a list of libraries and tools has been used in the context of the Data Cleaner to support several functionalities of the component. On top of this, for the data structure handling Pandas[6] library has been selected, while NumPy[7] library is used for all the numerical computations.

[5] https://flask.palletsprojects.com/en/2.0.x/
[6] https://pandas.pydata.org/
[7] https://numpy.org/

4 Data Qualifier

The goal of the Data Qualifier sub-component is to automatically categorize both known and unknown data sources to specific reliability levels. To this end, the provided microservice seeks to provide a predictive selection mechanism for achieving data source's reliability during runtime providing a trustfulness of the connected data sources.

The Data Qualifier sub-component classifies data sources as reliable or non-reliable both during the primary and secondary data injection. This sub-component receives data by subscribing to a Kafka topic, specifically it acquires the file names of the cleaned and faulty data produced by the Data Cleaner sub-component. The cleaned data is the dataset with the appropriate changes applied by the *CleaningService* of the Data Cleaner sub-component. The faulty data file informs about the values that have been *cleaned* by the *CleaningService*. For instance, if one or more attributes (i.e, column) in a record (row) is cleaned in a data set, the faulty data file will describe these changes

The Data Qualifier sub-component is divided into different sub-services shown in Figure 8 below. The Dataset Qualifier sub-service processes the cleaned dataset and the faulty file to evaluate the dataset reliability. The reliability is calculated over the whole data set if the dataset is in a file.

The reliability of static data (data in a file) is calculated by the Dataset Qualifier sub-service. For that purpose, it calculates the size of the data set and takes into account the amount of cleaned data in that data set. The reliability is provided for each value (e.g., a column) and for the whole data set. These values range from 0-1, where 1 is the highest reliability and 0 the lowest. The reliability of a column is one minus the total number of faulty values divided by the total number or occurrences of the column in the dataset. If the data is streamed, the reliability is calculated in the same way over a period of time we call the *reliability window* (every five minutes). The Data source Qualifier sub-service calculates the reliability of a specific device that produces the data. For instance, a smart watch monitors the heartbeats, sleeping time, number of steps, blood pressure among other metrics. If the heartbeat values are considered faulty for a batch of data or period of time, the heartbeat sensor is considered not reliable. Finally, the Data Qualifier sub-component output is published to a Kafka topic. The output is composed by the reliability data sources values and the cleaned data file name or the stored cleaned data file names in case of streamed data.

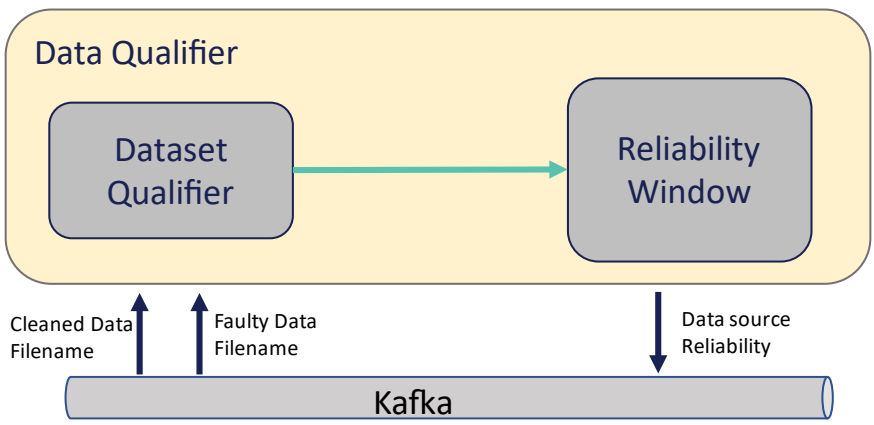

Figure 8: Data Qualifier internal workflow.

4.1 Dataset Qualifier

The Dataset qualifier sub-function receives the file names of the cleaned and faulty data produced by the Data Cleaner sub-component and classifies the dataset. To do so, the Dataset qualifier calculates the percentage of attributes cleaned per column in relation to the total number of elements for that attribute in the data set. For instance, if 20% of the data in one column was cleaned by the Data Cleaner, the reliability of the column will be 8 (1 - 0.2). This is calculated by the Dataset Qualifier for every column. The Dataset reliability is calculated as the average of the reliability of each attribute. This component outputs this information together with the names of the corresponding files, i.e.: **`cleanedDataset1: 7.5`**.

4.2 Reliability Window

The Reliability Window service receives as input the information from the Dataset Qualifier and periodically calculates the reliability of a data source. The periodicity is a configuration parameter that depends on the application (i.e., how frequent the datasets are generated). The datasets from the same data source may be generated every 5 minutes or every hour or once per day. Based on this frequency the reliability of the data source is calculated for a time period (window) that must also be configured. For instance, the reliability of the data source may be calculated once per day, if the datasets are received every hour. The reliability of a data source can be calculated every hour, if datasets are received every five minutes. This component will output the name of the data source and the aggregation of the reliabilities of all data sets cleaned during that period (window), i.e.: **`fitbit-versa-3: 9`**.

4.3 Interface

The Data Qualifier interface can also be provided by a REST API interface. This interface will be documented with a Swagger UI[8]. Data will arrive to the Data Qualifier sub-component through a JSON object where the cleanedData field contains the file name of the dataset with the required modifications applied by the Data Cleaner sub-component and the faultyData field contains the file name where the attributes transformation are indicated.

```
{
  “cleanedData”: “CleanedData.csv”,
  “faultyData”: “faultyData.csv”
}
```

The output is a JSON object produced by the Reliability Window that contains the data source name from which the reliability has been calculated and the reliability.

```
{
  “dataSource”: “The name of the data source”,
  “reliability”: “The reliability of the data source”
}
```

4.4 Baseline Technologies

The Data Qualifier component will be implemented using Kafka Streams. Kafka Streams is a client API that allows you to create applications and microservices where data is stored in a Kafka Cluster. To this end, the

[8] https://swagger.io/tools/swagger-ui/

Data Qualifier consumes data asynchronously from a Kafka topic and directly produces data asynchronously to another Kafka topic. More specifically, this sub-component receives the input message to its system by subscribing to a Kafka topic the file names of the cleaned and faulty data produced by the Data Cleaner sub-component. Finally, the Data Qualifier sub-component output is published to a Kafka topic. The output is composed by the reliability data sources values and the cleaned data file name or the stored cleaned data file names in case of streamed data. As also presented in the Data Cleaner component, regarding automated deployments of the data ingestion pipelines, the name of the topic can be received during run-time and might be dynamically sent by the serverless platform.

5 Data Harmonizer

The Data Harmonizer sub-component will be utilized as an integrated microservice in the overall iHelp platform, and its main objective is to support and harmonize data coming from heterogeneous sources into a common format. To this end, it seeks to provide annotated / correlated health data & harmonize them with the HHR format. The microservice will use its own internal sub-components in order to correlate data resources to be compliant with the HHR model that will be defined in the scope of the Task 3.1 ("Data Modelling and Integrated Health Records").

The Data Harmonizer microservice will be utilized either asynchronous or synchronous depending on the deployment status that will be decided in collaboration with other technical partners. This microservice will be integrated with the provided message bus mechanism and it will consume and produce corresponding messages to the provided Kafka queues. The message to be consumed should be cleaned data, which will be further harmonized and transformed, hence an annotated, transformed and HHR format aligned message will be the output of this specific microservice. To this end, the Data Harmonizer microservice has been determined to integrate closely with the Data Cleaner microservice.

The Data Harmonizer sub-component incorporates the use of three integrated sub-functions, as shown in Figure 9 below.

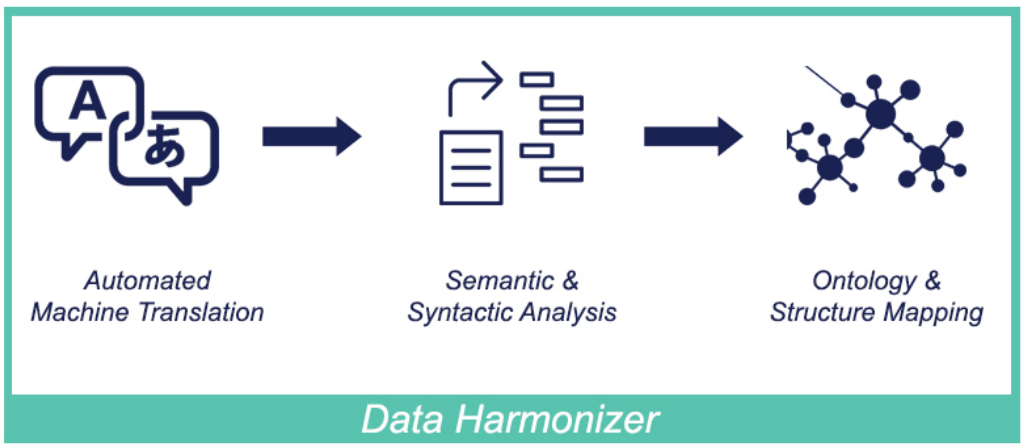

Figure 9: Data Harmonizer internal workflow.

This sub-component seeks to support data coming from divergent sources in order to deal with different formats, thus to enhance the interoperability of data. In recent years many approaches, standards, ontologies and vocabularies have been proposed as means of achieving various tasks of interoperability between heterogeneous and independent datasets. More specifically in the healthcare domain, an advanced Semantic Interoperability technique was introduced with emphasis on the utilization of Structural and Ontology Mapping services along with Terminology Linking services in order to transform the clinical information into interoperable and processable data using eHealth standards and terminologies (K., M., M. + 19).

The above introduced approach provides the means for common representation of domain specific datasets and the means for achieving interoperability across diverse databases and datasets. However, the above presented approach does not consider the emerging issue of multilingualism and language-independence. In modern multicultural and multilingual environments like European Union, deriving useful

knowledge is a complex and multilayered procedure of organizations, people, languages, information systems, information structures, rules, processes, and practices. Hence, the needs and trends in modern societies are increasingly demonstrating the need for creating multilingual and interoperable solutions and techniques that will operate in a wider and language-independent context. To this end, Neural Machine Translation (NMT) should obtain full and effective utilization in modern interoperability systems.

5.1 Automated Machine Translation

Nowadays, the overarching goal of Natural Language Processing (NLP) is to enable communication between humans and computers without resorting to memorable and complex processes. Modern chatbots, automatic translation engines, search engines and more are included in these applications (Bul, 2018). However, the needs and trends of modern intercultural societies are increasingly demonstrating the need for creating language-independent solutions and techniques that will operate in a wider context. Thus, the techniques of NMT will obtain full and effective utilization in the scopes of this proposed approach. Recent advances in the field of NMT have proven to be competitive with the encoder-decoder architecture based on the utilization of Recurrent Neural Networks (RNNs), which encode the length of the variable input into unstable dimensions vector and use its encoding to then decode the desired out-put sequence (see Figure 10). Hence, NMT models are often based on the seq2seq architecture (T., S., S. + 20), which is an encoder-decoder architecture and consists of two Deep Neural Networks: the encoder and the decoder (Y., L., C. + 20). The input to the encoder is the sentence in the original language, while the input to the decoder is the sentence in the translated language with a start-of-sentence token. The output is the actual target sentence with an end-of-sentence token.

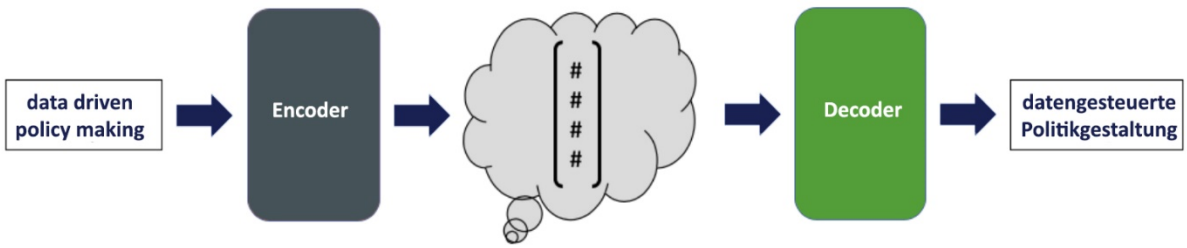

Figure 10: Encoder-decoder architecture in NMT.

Moreover, new advancements in the field of NMT introduce and propose the utilization of Transformers to solve the Machine Translation problem that relies mostly on the attention mechanism to draw the dependencies between the language models (B., M., N. + 20). The attention mechanism enables the decoder to look backward on the whole input sequence and selectively extract the information it needs during processing. Like RNNs, the Transformer is an architecture for transforming one sequence into another using the encoder-decoder mechanism, but it differs from the previous existing seq2seq models because it does not imply any Recurrent Network (GRUs, LSTMs, etc) (P., P., W. + 20). Yet, unlike the RNNs the Transformer stacks several identical self-attention based layers instead of recurrent units for better parallelization, while it also handles the entire input sequence at once and does not iterate word by word.

Both above introduced approaches and technologies will be utilized and their performance and overall functionality will be evaluated under the scopes of Standardisation and Quality Assurance hybrid mechanism and their implementation will be described in later versions of this deliverable.

5.2 Semantic & Syntactic Analysis

To exploit what the Data Harmonizer offers, translated data first needs to be structured and annotated. To this end, in the second subcomponent, the Semantic & Syntactic Analysis, translated data will be analyzed, transformed and annotated with appropriate URI metadata and controlled vocabularies will be identified and designed through the utilization of Semantic Web technologies coupled and enhanced by the utilization of NLP techniques, such as Named Entity Recognition (NER), Part-of-Speech Tagging etc, through the utilization of advanced and multilingual NLP tools such as spaCy[9], ntlk[10] and CoreNLP[11]. In next phases and steps, semantic and syntactic URI-annotated data (Unified Resource Identifier) will be interlinked through the task of Ontology Mapping. The main objectives of this second sub-function of the Data Harmonizer are the identification and recognition of entities, which will be further used for interconnection and interlinking with the HHR resources and model that have been identified in the context of the T3.1 "Data Modelling and Integrated Health Records". Moreover, classifying named entities found in translated data into pre-defined categories, such as persons, places, organizations, dates etc, will make it possible to identify, design and use proper widely used and controlled vocabularies and standards. The overall Data Harmonizer sub-function will be further enhanced and completed in the next step by the utilization of Ontology Mapping subcomponent, where an Ontology and Structuring Mapping service will be utilized in order to interlink not only URI-annotated data with proper ontologies, but also to interlink and correlate datasets among them.

5.3 Ontology & Structure Mapping

The Structural Mapping sub-function will take advantage of well-established ontology alignment approaches to perform the mapping between the schema/model of incoming document with the use-case specific target schema/model in the iHelp platform.

Ontology alignment approaches can be utilized for finding structural mappings between two different data models. A number of tools and frameworks have been developed for aligning Ontologies, which vary in the degree of user intervention required to produce accurate mappings. In typical Ontology alignment approaches, data models or Ontologies are usually converted to a graph representation before being matched. Such graphs can be represented in the Resource Description Framework (RDF) line of languages by triples of the form *<subject, predicate, object>*. In this context, aligning ontologies is sometimes referred to as "ontology matching".

Successful annotation, transformation and mapping of data and corresponding ontologies in terms of semantic and syntactic interoperability of data is one of the key elements of the Data Harmonizer sub-function. To this end, one of the main objectives of the Ontology Mapping subcomponent is to save correlated, annotated and interoperable data in JSON-LD format and as linked ontologies. Hence, it will be feasible to store semantic facts and the support of the corresponding data schema models. Moreover, this subcomponent seeks to map concepts, classes, and semantics defined in different ontologies and datasets and to achieve transformation compatibility through extracted metadata. In addition, a data modelling subtask by standard metadata schemas will be defined in order to specify the metadata elements that should accompany a dataset within a domain. To this end, semantic models for physical entities (e.g.

[9] https://spacy.io/
[10] https://www.nltk.org/
[11] https://stanfordnlp.github.io/CoreNLP/

specific grading features of Pancreatic Cancer) and measures (e.g. specific grading features of Pancreatic Cancer) will be identified. These models will be based on a set of transversal and domain-specific ontologies and could provide a foundation for high-level interoperability and rich semantic annotations across the healthcare ecosystem. As shown in Figure 11 below, there are several levels of structuring before reaching proper ontologies. At the beginning, the annotation and creation of metadata representations through the utilization of JSON-LD technology is a key point. Afterwards, vocabularies and taxonomies expressed by RDFs are created and in the final step they are correlated and interlinked into ontologies with high semantic expressivity through the utilization of OWL technology.

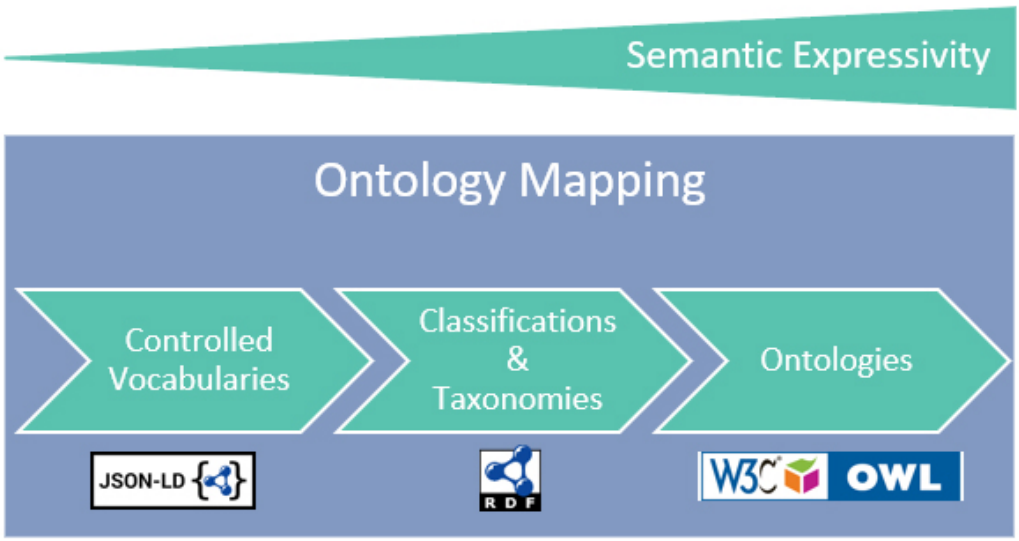

Figure 11: Ontology Mapping Steps.

On top of this, ontologies are central to the Data Harmonizer as they allow applications to agree on the terms that they use when communicating and they enable the correlation of divergent data and datasets from various sources. To this end, the utilization of ontologies under the scope of Data Harmonizer facilitates communication by providing precise notions that can be used to compose messages (queries, statements) about the healthcare domain. In stakeholders and user level, the ontology helps to understand messages by providing the correct interpretation context. Thus, ontologies, if shared among stakeholders, may improve system interoperability in the healthcare ecosystem. The overall approach that will be followed brings together techniques in modeling, computation linguistics, and information retrieval in order to provide a semi-automatic mapping method and a prototype mapping system that support the process of Ontology Mapping for the purpose of improving and enhancing interoperability and usage of data during the whole data lifecycle.

The novelty of the proposed Ontology Mapping sub-function is not solely the use of formal application ontologies as an initial mechanism to achieve meaningful interoperability, but moreover the utilization of divergent ontologies to support the formal application ontologies mapping process, integrated into an architectural framework.

5.4 Interface

At the current time the deployment of the Data Harmonizer sub-function has been focused on the implementation of the Semantic & Syntactic Analysis sub-mechanism. Under the scope of this Deliverable,

the latter is provided through a IPython Notebook (.ipynb file extension) in order for any stakeholder or user to be able to execute the code and identify the outcomes in every step and sub-task of this specific sub-mechanism. Moreover, the Data Harmonizer sub-mechanism will be utilized in an asynchronous way. Hence, no endpoint or API call will be available in order to trigger the execution of this specific functionality. At later steps, a docker installation with a corresponding Flask application coupled with the utilization of Swagger UI will be implemented in order to provide a REST application interface following the OpenAPI specification. To this end, it will be easier for the end user to discover the capabilities of this sub-component and to provide well-structured documentation for each of the component's services.

5.5 Baseline Technologies

As noted in the previous sub-section, at the current time the implementation of the Data Harmonizer has focused on the Semantic & Syntactic Analysis sub-mechanism. The overall code has started to be implemented based on Python3.7 programming language. Moreover, SPARQL, a widely used RDF query language, is being utilized in order to identify and interlink standards and resources that have been identified in the context of D3.1 "Data Modelling and Integrated Health Records: Design and open specification I" with the provided incoming data. The latter is being used to perform queries on Knowledge bases to identify and interlink appropriate entities based on the ones that have already been recognized from the raw data.

6 Primary Data Mapper

The Primary Data Mapper sub-component seeks to enable the mapping of primary data, that is the clinical data from the Electronic Health Records (EHRs), into the common data format that will be used in the iHelp platform (HHRs). To this end, this specific component will provide the necessary transformation functions that are required to map the primary data to the HHRs stored in the iHelp platform. Initial information about the HHRs and the mapping process can be found at Sections 4 and 5 of D3.1 (K., D., P. + 21).

The Primary Data Mapper receives cleaned, qualified, and harmonized data as part of the overall ingestion pipeline. Sequentially, the harmonized health records that are being received from the Data Harmonizer are mapped to the HHR ontology-based records, through semantic matching and harmonization techniques. Finally, after the necessary mapping/transformation process, the Primary Data Mapper produces and sends the new HHR aligned record, through a Kafka topic to the HHR Importer component.

7 Secondary Data Mapper

The Secondary Mapper sub-component will enable the mapping of secondary data (e.g. from mobile, wearable and social-media platforms) into the standard data format (HHR schema, model) used in the iHelp platform. The component will provide the necessary transformation functions that are required to map the secondary data from heterogeneous sources to the holistic health records stored in the iHelp platform. The Secondary Mapper will be an integral part of the iHelp platform as it supports the enrichment of typical health records with the secondary (e.g. lifestyle, social etc.) data of the individuals.

Based on the adoption of microservices architecture model in the iHelp project, the Secondary Data Mapper component will be exposed as a microservice in the iHelp platform. As shown in Figure 12, it will offer dedicated converter/mapper functions that will handle data from different secondary data devices or interfaces. Since, during the project, iHelp platform will support secondary data ingestion from a specific number or type of devices (e.g., the mobile application or Fitbit activity tracker), thus the mapper functions can be designed and configured to deal with the data (interfaces, models, formats, syntax) associated with those devices. However, the secondary data mapper services will be extensible in nature, thus allowing the development and integration of mapping functions that deal with other types of devices or data models.

A controller mechanism with the secondary data mapper will be responsible for assigned the incoming data to the relevant mapper functions. The controller will work on the basis of interpreting meta data associated with the incoming data (batch or stream) and forwarding the data to the relevant mapper function.

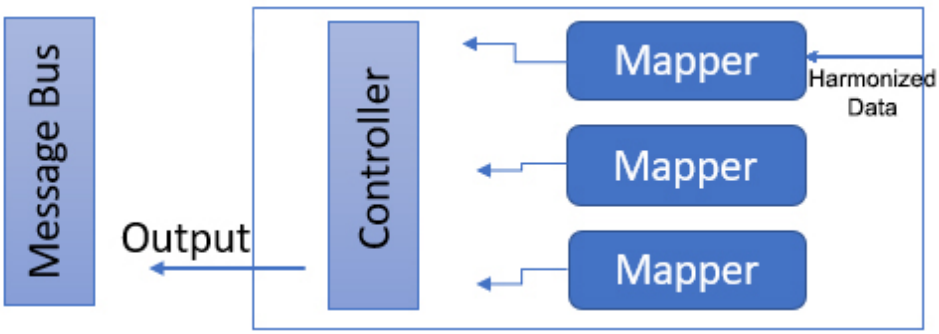

Figure 12: High-level architecture of Secondary Data Mapper.

The Secondary Data Mapper microservice will be utilized either asynchronous or synchronous depending on the deployment status; and it will be integrated with the iHelp message bus mechanism. The interaction with the message bus will allow the microservice to produce secondary data derived from different mobile and wearable devices in the standardised HHR format. To this context, this microservice seeks to apply the relevant mapping function to convert the harmonized data, which are derived from the Data Harmonizer and are the input to its system, according to the standard HHR format that will be defined under the scope of T3.1 (“Data Modelling and Integrated Health Records”) and expose the converted data or the outcome of the mapping function to the provided message bus (Kafka) topic. To this end, the secondary data mapper microservice will serve as the final stage of the data cleaning and standardisation operation, before the data is passed on to the iHelp’s Big Data Platform through the HHR Importer mechanism, which will be provided by T4.4 (“Big Data Platform and Knowledge Management System”).

8 Conclusion

This document reported the work that has been currently done in the scope of T3.4 “Standardisation and Quality Assurance of Heterogeneous Data”, whose main objective it is to provide the initial design, specifications, and internal workflows of the project’s Standardisation and Quality Assurance Mechanism and its initial development and implementation. The specifications that have been introduced in this deliverable will be utilized for the realization and implementation of this holistic mechanism, encompassing its main functionalities regarding the assurance of the incoming data’s accuracy, integrity, and quality, the interoperability of data, the automated data transformation to the HHR model, and their aggregation into unique turn-key offerings. On top of this, this report describes the measures adopted and developed in the project to ensure effective contributions towards standardisation and quality assurance of healthcare data. The deliverable highlights the functionalities used for various purposes (e.g. for data management, data cleaning, data transformation, data mapping etc) and the approaches/techniques implemented to make sure that the data remains within the quality constraints while being used by different stakeholders and applications in the project. Moreover, as already stated, this deliverable includes an initial and brief outline concerning the Primary and Secondary Data Mapper sub-components, as these sub-components will start being designed and implemented after the final modelling of the common HHR model and format that will be followed across the iHelp project and platform and that will be introduced in the context of task T3.1 (“Data Modelling and Integrated Health Records”). In next deliverables, more details about the implementation and architecture of these two sub-components will be outlined.

To conclude, the current document is delivered in M10 (October 2021) and is just the first of a series of documents and reports that are planned to be released under the scopes of task T3.4 (“Standardisation and Quality Assurance of Heterogeneous Data”) and throughout the project’s lifetime. On M20 (August 2022), a second version will update the work and will revise the procedures and functionalities of the Standardisation and Quality Assurance Mechanism. Finally, a third version is planned to be delivered on M32 (August 2023), in order to cover remaining aspects and to correct potential erroneous decisions or unnecessary implementation that might have been identified earlier, so that it can drive the final definition and implementation of the Standardisation and Quality Assurance Mechanism.

Bibliography

P. Bahar, N. Makarov, and H. Ney, “Investigation of Transformer-based Latent Attention Models for Neural Machine Translation”, *In: Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (AMTA 2020)*, pp. 7-20, 2020.

Y.E. Bulut, “AI for data science: artificial intelligence frameworks and functionality for deep learning, optimization, and beyond”, *Technics Publications*, 2018.

V. Chavan, and R.N. Phursule, “Survey paper on big data”, *Int. J. Comput. Sci. Inf. Tech-nol*, vol. 5, no. 6, pp. 7932-7939, 2014.

P. P. Jayaraman, A. R. M. Forkan, A. Morshed, P. D. Haghighi, and Y. B. Kang, “Healthcare 4.0: A review of frontiers in digital health”, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 2, p. e1350, 2020.

M. Kalogerini, A. Dalianis, C. Pandolfo, F. Melillo, and George Manias, “D3.1 - Data Modelling and Integrated Health Records: Design and open specification I”, iHelp, 2021.

A. Kiourtis, A. Mavrogiorgou, A. Menychtas, I. Maglogiannis, and D. Kyriazis, “Structurally mapping healthcare data to HL7 FHIR through ontology alignment”, *Journal of medical systems*, vol. 43, no. 3, pp. 62, 2019.

M. Mosley, M. H. Brackett, S. Earley, and D. Henderson, “DAMA guide to the data management body of knowledge”, *Technics Publications*, 2010.

S. C. Pandey, “Data mining techniques for medical data: a review”, *In 2016 International Conference on Signal Processing, Communication, Power and Embedded System*, pp. 972-982, 2016.

A. Pramodya, R. Pushpananda, and R. Weerasinghe, “A Comparison of Transformer, Recurrent Neural Networks and SMT in Tamil to Sinhala MT”, *In: 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp. 155-160. IEEE, 2020.

G. Tiwari, A. Sharma, A. Sahotra, and R. Kapoor, “English-Hindi Neural Machine Transla-tion-LSTM Seq2Seq and ConvS2S”, *In: 2020 International Conference on Communication and Signal Processing (ICCSP)*, pp. 871-875. IEEE, 2020.

M. Yang, S. Liu, K. Chen, H. Zhang, E. Zhao, and T. Zhao, “A hierarchical clustering ap-proach to fuzzy semantic representation of rare words in neural machine translation”, *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 5, pp. 992-1002, 2020.

List of Acronyms

AI	Artificial Intelligence
API	Application Programming Interface
CA	Consortium Agreement
CSV	Comma Separated Values
D	Deliverable
DoA	Description of Action
EHRs	Electronic Health Records
EU	European Union
HHRs	Holistic Health Records
JSON-LD	JavaScript Object Notation – Linked Data
M	Month
NER	Named-Entity Recognition
NLP	Natural Language Processing
NMT	Neural Machine Translation
OWL	Web Ontology Language
R&D	Research and Development
RDF	Resource Description Framework
REST	Representational State Transfer
T	Task
URI	Uniform Resource Identifier

Annex A – Cleaning Action and Constraints

Pilot #1 - UNIMAN

The pilot focuses on Genomics and Epigenomics Markers for Early Risk Assessment of Pancreatic Cancer.

Sample Datasets

Risk_factors_1 Sample Dataset

	A	B	C	D	E	F	G	H	I	J	K
1	ID	Gender	Age group	Current weight	Height (m)	BMI	FH	Smoke	Diabetes	Chronic pancreatitis	FruitVeg
2	10001	0	3	80	1,8	24,7	0	1	0	0	1
3	10002	0	3	90	1,75	29,4	1	0	0	1	1
4	10003	0	2	50	1,6	19,5	1	1	1	1	1
5	10004	1	2	65	1,72	22,0	1	1	1	1	0
6	10005	0	2	55	1,63	20,7	1	1	1	0	0
7	10006	0	3	49	1,55	20,4	1	2	1	0	0
8	10007	1	3	45	1,53	19,2	1	1	1	0	0
9	10008	0	3	77	1,72	26,0	0	0	0	0	1
10	10009	0	4	88	1,81	26,9	1	1	0	1	1
11	10010	1	1	71	1,60	27,7	1	2	1	1	1
12	10011	1	2	89	1,90	24,7	0	0	1	1	0
13	10012	0	4	105	1,79	32,8	1	0	0	1	1

Figure 13: Snapshot of UNIMAN pilot Risk_factors_1 sample dataset.

Dictionary

Variable	Answer	Code
Gender	Female	1
	Male	2
Age group	40-50	1
	51-60	2
	61-70	3
	>=71	4
FH	Yes-Family history of pancreatic cancer (first degree relative)	0
	No-family history of pancreatic cancer in 1st degress relative	1
Smoke	Non-smoker	0
	Smoke >=25 cigarettes/day	1
	Smoke 15-24 cigarettes/day	2
Diabetes	No diabetes type II	0
	Yes-Diabetes type II	1
chronic pancreatitis	No	0
	Yes	1
FruitVeg	No- I do not have 5 portions of fruit and vegtables per day	0
	Yes- I have 5 portions of fruit and vegtables per day	1

Figure 14: Dictionary and description of Risk_factors_1 sample dataset.

Wellbeing Sample Dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	ID	I've been feeling optimistic about the future	I've been feeling useful	I've been feeling relaxed	I've been feeling interested in other people	I've had energy to spare	I've been dealing with problems well	I've been thinking clearly	I've been feeling good about myself	I've been feeling close to other people	I've been feeling confident	I've been able to make up my own mind about things	I've been feeling loved	I've been interested in new things	I've been feeling cheerful
2	10001	1	2	2	4	2	2	3	2	4	5	3	3	2	5
3	10002	3	2	4	5	3	5	2	5	3	4	2	5	2	5
4	10003	3	4	2	5	1	3	5	4	3	4	5	1	3	5
5	10004	2	4	5	3	2	4	1	1	2	3	3	2	1	4
6	10005	4	5	2	3	5	3	2	1	5	2	2	1	3	5
7	10006	5	3	4	1	2	2	2	4	2	1	3	1	5	5
8	10007	1	3	2	2	4	4	1	1	1	1	3	3	3	5
9	10008	5	3	4	5	2	3	5	4	3	5	5	1	5	2
10	10009	5	4	2	5	3	1	3	3	5	1	5	3	3	5
11	10010	3	5	5	1	5	2	3	3	4	5	1	3	2	5
12	10011	1	3	3	5	5	3	3	4	4	5	5	4	5	3
13	10012	3	5	1	2	1	3	2	4	2	5	3	2	4	1
14	10013	2	1	2	4	5	2	3	4	3	4	3	4	2	5
15	10014	1	5	2	4	2	5	3	3	3	1	5	3	2	2

Figure 15: Snapshot of UNIMAN pilot Wellbeing sample dataset.

Dictionary

Table 1: Dictionary and description of Wellbeing sample dataset.

Code	Description
1	None of the time
2	Rarely
3	Some of the time
4	Often
5	All of the time

Food group Sample Dataset

	A	B	C	D	E	F	G	H	I
1	ID	Cereals (grains, beans, legumes)	Vegetables	Fruits (sometimes grouped with vegetables)	White meat	Red meat	Processed meat	Dairy	Confectionery (aka sugary foods)
2	10001	6	8	4	2	1	1	2	7
3	10002	4	1	3	9	3	1	3	5
4	10003	3	4	8	6	6	8	2	5
5	10004	3	4	2	5	1	8	1	3
6	10005	6	3	4	5	3	4	5	5
7	10006	8	-9	4	8	6	6	9	5
8	10007	1	1	8	3	4	2	4	4
9	10008	6	3	1	9	4	7	1	2
10	10009	9	9	3	7	5	3	3	6
11	10010	4	6	3	1	2	6	6	8
12	10011	6	6	3	3	5	7	9	5
13	10012	1	6	8	1	3	9	4	3
14	10013	4	9	8	1	3	2	2	7
15	10014	6	6	7	2	5	9	1	7

Figure 16: Snapshot of UNIMAN pilot Food group sample dataset.

Dictionary

Table 2: Dictionary and description of Food group sample dataset.

Code frequency	Description of code
1	Never or less than once a month

2	1-3 times per month
3	once a week
4	2-4 times per week
5	5-6 times perweek
6	once a day
7	2-3 times per day
8	4-5 times per day
9	6+ times per day
-9	missing values

Physical Activity Sample Dataset

	A	B	C
1	ID	Moderate physical activit	Vigorus physical activity
2	10001	1	1
3	10002	0	0
4	10003	1	1
5	10004	1	1
6	10005	1	1
7	10006	0	0
8	10007	1	1
9	10008	0	0
10	10009	0	0
11	10010	1	1
12	10011	1	0
13	10012	0	1
14	10013	0	1
15	10014	0	0

Figure 17: Snapshot of UNIMAN pilot Physical Activity sample dataset.

Dictionary

Table 3: Dictionary and description of Physical Activity sample dataset.

Type of physical activity	Description	Code	Meaning
Moderate physical activity	On average have you undertaken at least 30 minutes of moderate physical activity per day – either at home or at work. (These activities can be made up of many components, for example, moving a table, pushing a vacuum cleaner, bowling or playing golf).	0	No
		1	Yes
Vigorous physical activity	On average have you undertaken 20 minutes or more of energetic activity at least 3 times per week whilst NOT at work. (These include, for example, keep fit, dancing or exercises, swimming or other brisk sport, long walks, jogging or running, hard work in a job at home or in the garden, cycling).	0	No
		1	Yes

Conceptual Diagram

Provide a simple UML (or any other graphic class) diagram representing the names of entities described in the dataset, their relationship and cardinality. Just for reference, the following figure provides an example of a class diagram to be replaced with the actual diagram of the dataset.

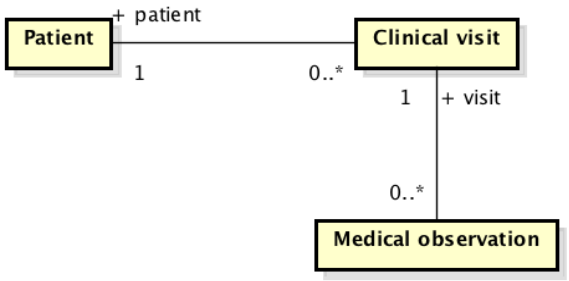

Figure 18: Example of dataset's entities UML Conceptual Diagram

List of Entities

List and describe the entities reported in the conceptual diagram of the previous section using a table as in the following example.

Table 4: Example table listing entities of a sample dataset.

#	Entity name	Description
1	Patient	Demographics and other administrative information about an individual receiving care or other health-related services
2	Clinical visit	An interaction between a patient and healthcare provider for the purpose of providing healthcare services or assessing the health status of a patient
3	Medical Observation	Measurements and simple assertions made about a patient

Constraints – Cleaning Actions

Mandatory Constraints that must be fulfilled for each unique attribute:

- **Specific data type** (e.g. Numeric, String)
- **Mandatory field**
- **Specific value length** (e.g. maximum 20 digits)

Optional Constraints that could be fulfilled for each unique attribute:

- **Specific coding standard** (e.g. LOINC, SNOMED, ICD10)
- **Value representation** (e.g. text formatting "123-45-67" or "1234567" or "123 45 67")
- **Value uniformity** (e.g. all times are provided in UTC, all weight values in KGs, etc.)
- **Value range constraints** (minimum and maximum values)
- **Pre-defined values** (e.g. values selected from a drop-down list)
- **Regular expression patterns** - data that has a certain pattern in the way it is displayed, such as phone numbers)
- **Separation of values** (e.g. complete address in free form field without any indication where street ends, and city begins)
- **Uniqueness** - data that cannot be repeated and require unique values (e.g. social security numbers)
- **Logical Error** (e.g. female individual with prostate cancer medications prescribed)
- …

For the different constraints described, the list of cleaning (corrective) actions should be documented in the table. The following list includes some examples that can be used or combined for the described constraints. Note: This is an indicative and not an exhaustive list. Additional cleaning actions can be introduced and described by the UC partner in case they are not covered in the list below.

- **Deletion of value** that does not conform to a constraint by:
  - Drop whole entity
  - Drop specific attribute
  - …
- **Replacement of value** that does not conform to a constraint through:
  - Transformation of wrong data type value
  - Prediction of erroneous/missing value
  - Prediction of erroneous/missing value based on similar values in the past
  - Creation of a list of features with high percentage of similarity with the same value
  - ……

Risk_factors_1 Sample Dataset

ID (example)

Table 5: Example table for specifying constraint rules and cleaning actions for the ID attribute.

	#	Constraint Type	Constraint Description	Cleaning Action
Mandatory	1	Specific data type	The expected value must be a Numeric value	Replacement of value through transformation of non-numeric value with a numeric one
	2	Mandatory field	The value is mandatory	Deletion of value by dropping the whole entity
	3	Specific value length	Positive integer max 5 digits	Deletion of value by dropping the whole entity

Optional	4	Uniqueness	All the values must be unique	Deletion of value by dropping the duplicate entries and keep only the first one
	5	Value representation	“1234567890”	Replacement of value through transformation to the expected format