D5.17

Data integration platform for healthcare professionals – v2

ABSTRACT

One of the main objectives of the InteropEHRate project is to offer the interoperability of personal healthcare data throughout Europe. This is achieved through combining knowledge extraction, adaptive data integration, conversion of data to a common representation, as well as its translation into the natural language(s) of its consumers. The back-end platform (infrastructure and fundamental services) that supports the integration and interoperability methods includes healthcare knowledge and related tools organised into linguistic (NLP), terminological, and ontological layers, formally representing lexical units, schemas and encoding standards used by member countries, as well as mappings between these wherever applicable. This document accompanies the release of the software components of the Platform.

Delivery Date	June 30th 2021
Work Package	WP5
Task	T5.5
Dissemination Level	Public
Type of Deliverable	Demonstrator
Lead partner	UNITN

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106.

This document has been produced in the context of the InteropEHRate Project which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826106. All information provided in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose.

CONTRIBUTORS

	Name	Partner
Contributors	Gábor Bella	UNITN
Contributors	Simone Bocca	UNITN

LOGTABLE

Version	Date	Change	Author	Partner
0.1	20/08/2020	Document created from V1, updated with new libraries	Simone Bocca	UNITN
0.2	16/06/2021	Started preparing V2	Gábor Bella	UNITN
0.3	23/06/2021	Extended and updated contents	Simone Bocca	UNITN
0.9	25/06/2021	Version ready for quality check	Gábor Bella	UNITN
Vfinal	30/06/2021	Quality check and submission	Laura Pucci	ENG

ACRONYMS

Acronym	Term and definition
EHR	Electronic Health Record information system (e.g., as provided by a hospital).
JDK	Java Development Kit
HCP	Health Care Professional
REST	REpresentational State Transfer
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
SEHR	Smart Electronic Health Record
FHIR	A standard providing data schemas for healthcare data exchange.
ICD	International Classification of Diseases, an international standard.
LOINC	An international standard (set of identifiers, names, and codes) for identifying health measurements, observations, and documents.
SNOMED CT	SNOMED Clinical Terms is a systematically organized computer processable collection of medical terms providing codes, terms, synonyms and definitions used in clinical documentation and reporting.
WHO ATC	The World Health Organization’s classification system for active substances of pharmaceutical products. The active substances are classified in a hierarchy with five different levels.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Scope of the document

This document provides details about the second version of the Data Integration (DI) Platform for healthcare professionals and the component which allows the exploitation of the Platform, calle IHS Controller. The details refer to:

- a general overview of the components that compose the Platform and IHS Controller;
- how the different software components are built and installed properly;
- how the user can access and interact with the components;
- release modalities.

1.2. Intended audience

The document is intended for all people interested to have an overview of how the Data Integration Platform and the IHS Controller can be implemented and/or can be used.

1.3. Structure of the document

Section 2 reports a general overview of the two main components considered in this deliverable (DI Platform and IHS Controller) together with succinct installation and user guides. Moreover, a description of the internal components supporting the two main ones is included. Section 3 provides the synthetic description of the software components considered in this deliverable.

1.4. Updates with respect to previous version

This is the second version of the deliverable. The current version has been updated adding the descriptions and overview of the IHS Controller, as well as the description of all the internal software components for both the DI Platform and IHS Controller. With respect to the first version, the updates have been performed on:

- section 2: added the description of the IHS Controller software component and its internal components, as well as the description for all the components composing the DI Platform environment;
- section 3: added the information about EHR Data Cache database, User database and Platform API layer.

2. OVERVIEW

The role of the Platform, as the core component of *InteropEHRate Health Services* (IHS), is to provide fundamental syntactic and semantic data integration and query functionalities to the local healthcare institution where it is deployed. Figure 1 below shows the high-level architecture of the Platform.

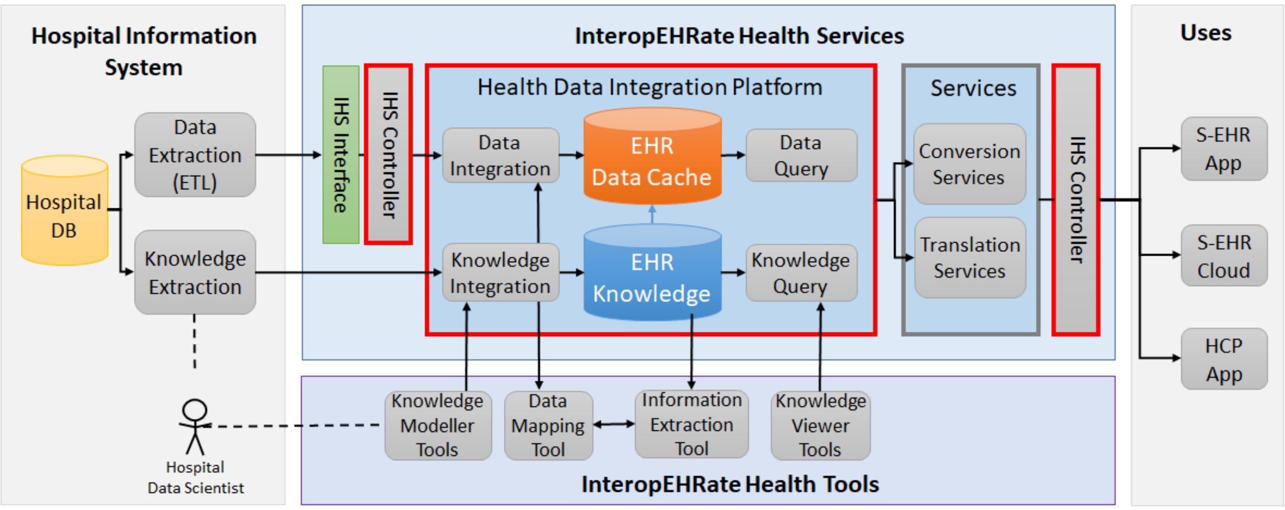


Figure 1 – Architecture of the Data Integration Platform within the InteropEHRate Health Services; the components covered by this deliverable are highlighted in red.

This document provides details on the use of the Platform for demonstration purposes, including the customisation of its setup with respect to local hospital requirements and infrastructure, through the configuration of the *IHS Controller* component. For more detailed specifications of the Platform, please consult the Platform specification deliverable [D5.8].

As shown in Figure 1, the components covered by this deliverable are:

- the *Data Integration Platform* that provides the semantic integration functionalities, transforming input health data and knowledge into a supra-lingual knowledge graph, searchable and exportable in a cross-border interoperable FHIR format;
- the *IHS Controller* that configures the connectivity between the Platform and the local hospital infrastructure.

Accordingly, the demonstrator presented here covers the deployment and setup of the Platform and the IHS Controller. In the context of the InteropEHRate project, however, the Platform should not be considered as a stand-alone demonstrator, as it needs to be prepared to execute data conversion, mapping, and translation functionalities according to the language, terminologies, and data schemas used by the hospital. Instead, the demonstrators of the Platform, of the data conversion services [D5.19], and of translation and information extraction [D5.20, D5.21] should be deployed and used together.

2.1. Data Integration Platform

The Data Integration (DI) Platform, detailed in [D5.8], provides the support for the integration, conversion and translation functionalities offered as parts of the InteropEHRate Health Services (IHS). It is typically used in a hospital setting in the context of higher-level services and wrapped into the configuration layer of the IHS Controller. The DI Platform supports the functionalities mentioned above through the following software components:

- a PostgreSQL database instance: a database instance that includes both the EHR data cache and the knowledge base that stores the knowledge resources required for the data integration, conversion and translation processes (for more details regarding these two storage components see [D5.8]).
- A MongoDB database instance: this database maintains the information of the users accessing the platform, and it is used to control the access to the EHR database, described above.
- A RESTful API layer: this software component allows the interactions between the EHR database and the other tools and services included in the DI Platform.
- The Knowledge Management Tools (KMT): a web application that includes both knowledge modeller and explorer tools. It allows the user (data scientist) to visualize, query and update the health knowledge information, as well as to check the health data stored temporarily in the EHR data cache (see [D5.19] for more details about the KMT's demonstrator). This tool is supported by the API layer and by the users database, described above.
- The Data Mapping Tool (DMT): a web application used to perform the syntactic and semantic mapping operation, in order to convert local health records into the interoperable FHIR format (see [D5.19] for more details about the KMT's demonstrator). This tool is supported by the API layer described above, in order to access the health knowledge information required during the data mapping process.

The DI Platform, including all the software components listed above, is deployed in the hospitals in order to interact with the EHR information system and support the conversion and translation services accessed by the IHS Controller (see next section).

2.1.1. Building guide

The DI Platform is a set of software components running in a docker environment, each one having its own docker container, so an instance of docker is needed in order to create the containers following the instructions of the docker-compose file. Furthermore, in order to properly install the DI Platform, some directories are needed containing code and configurations for the tools and services provided by the Platform itself. For what concern the support directories, and configuration files, required for the KMT and DMP see [D5.19]. The API layer instead, requires a configuration file (named *override.properties*) used to specify the PostgreSQL database connection information (*username*, *password*, *host*, *port number*, *connection url*). In the end, the health information database as well as the users information database require the storage directories to save the data handled.

2.1.2. Installation guide

Having the configuration file and the required directories properly built, to install the DI Platform it is only necessary to execute the following command, in the path where the docker-compose-file is located:

```
docker-compose up
```

2.1.3. User guide

The KMT and DMT platform’s tools are provided as web applications, and the usage instructions are defined in [D5.19]. The remaining services included in the Platform are automatically integrated in the conversion and translation services provided to the hospital systems through the IHS Controller. Nevertheless the RESTful API layer can be used separately by calling all the web services offered. Here below is an example of one of such calls. The specific service call reported below returns a list of medical concepts identified by a word that includes a prefix passed as request input parameter.

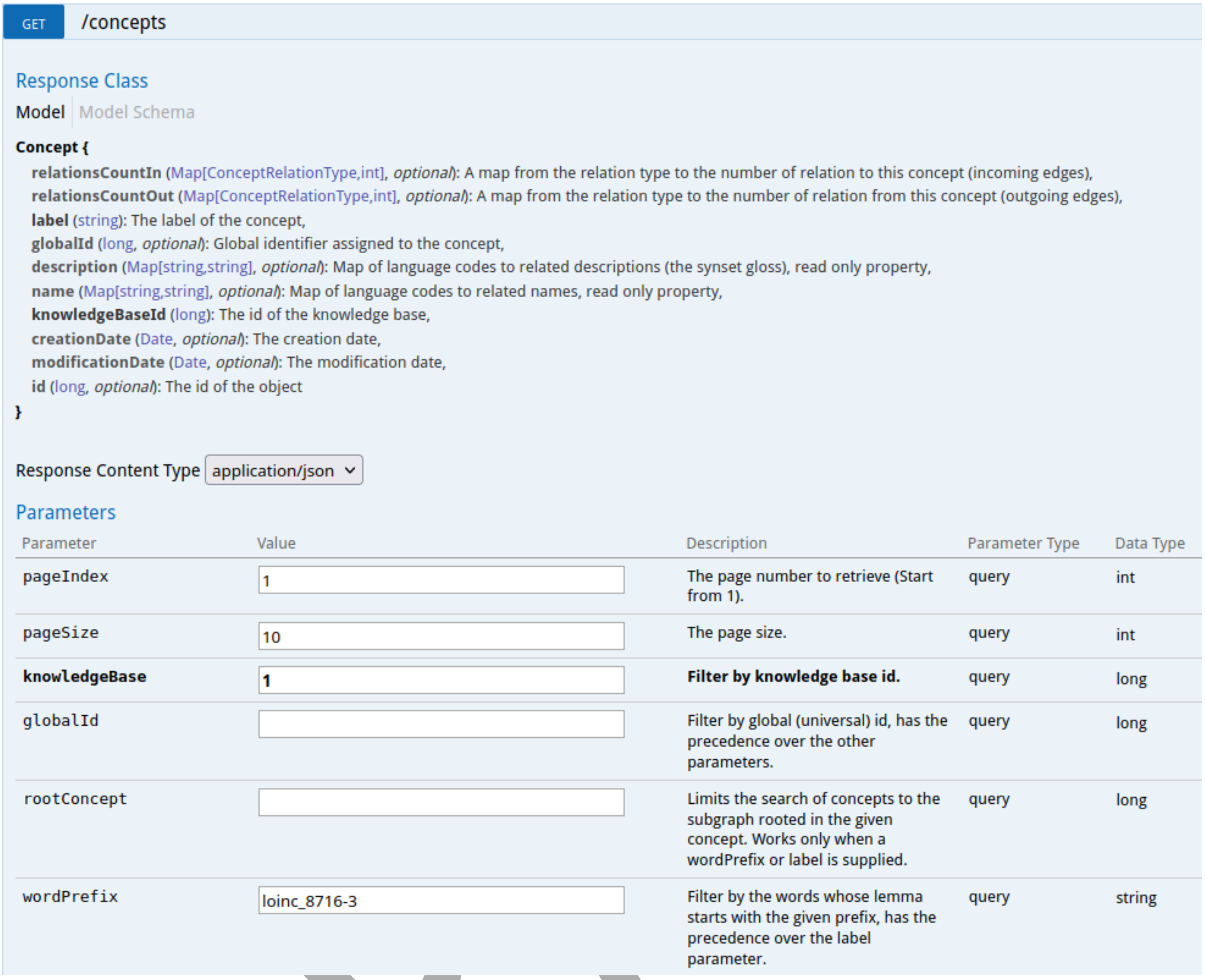
```
                                  curl
‘http://<host>:<portNumber>/concepts?pageIndex=1&pageSize=10&knowledgeBas
   e=1&wordPrefix=<prefix>’ --header 'content-type: application/json'
```

Where:

- <host> is the address of the machine which has the DI Platform installed.
- <portNumber> is the port number used to access the Platform services.
- <prefix> is the word's prefix used to search for the medical concepts desired.

Moreover, once the Platform Is correctly deployed and installed, the RESTful API layer is accessible also through a dedicated Swagger user interface that includes all the services provided by the Platform. Here below an example reporting the call of the same service called before (through `curl` command line interface), used through the Swagger UI.

The figure below shows how to call the service specifying the word prefix desired in the parameter field `wordPrefix`.



/concepts

Response Class

Model | Model Schema

Concept {

relationsCountIn (Map[ConceptRelationType,int], *optional*): A map from the relation type to the number of relation to this concept (incoming edges),
relationsCountOut (Map[ConceptRelationType,int], *optional*): A map from the relation type to the number of relation from this concept (outgoing edges),
label (string): The label of the concept,
globalId (long, *optional*): Global identifier assigned to the concept,
description (Map[string,string], *optional*): Map of language codes to related descriptions (the synset gloss), read only property,
name (Map[string,string], *optional*): Map of language codes to related names, read only property,
knowledgeBaseId (long): The id of the knowledge base,
creationDate (Date, *optional*): The creation date,
modificationDate (Date, *optional*): The modification date,
id (long, *optional*): The id of the object

}

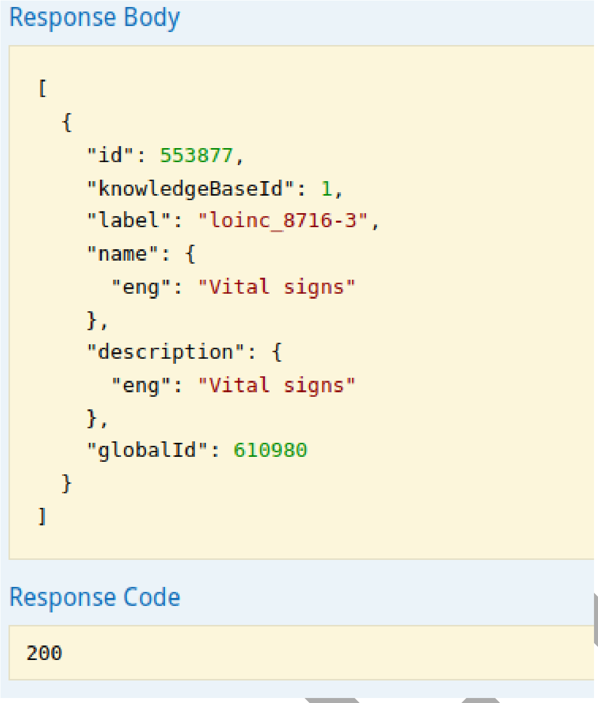
Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
pageIndex	1	The page number to retrieve (Start from 1).	query	int
pageSize	10	The page size.	query	int
knowledgeBase	**1**	**Filter by knowledge base id.**	query	long
globalId		Filter by global (universal) id, has the precedence over the other parameters.	query	long
rootConcept		Limits the search of concepts to the subgraph rooted in the given concept. Works only when a wordPrefix or label is supplied.	query	long
wordPrefix	loinc_8716-3	Filter by the words whose lemma starts with the given prefix, has the precedence over the label parameter.	query	string

Figure 2 – Call of DI Platform’s web service through Swagger user interface.

The second image below shows the results of the call, reporting the medical concept “vital signs” identified also by the Loinc standard code “loinc_8716-3”.



Response Body

```
[
  {
    "id": 553877,
    "knowledgeBaseId": 1,
    "label": "loinc_8716-3",
    "name": {
      "eng": "Vital signs"
    },
    "description": {
      "eng": "Vital signs"
    },
    "globalId": 610980
  }
]
```

Response Code

```
200
```

Figure 3 –Response of DI Platform’s web service through Swagger user interface.

2.2. IHS Controller

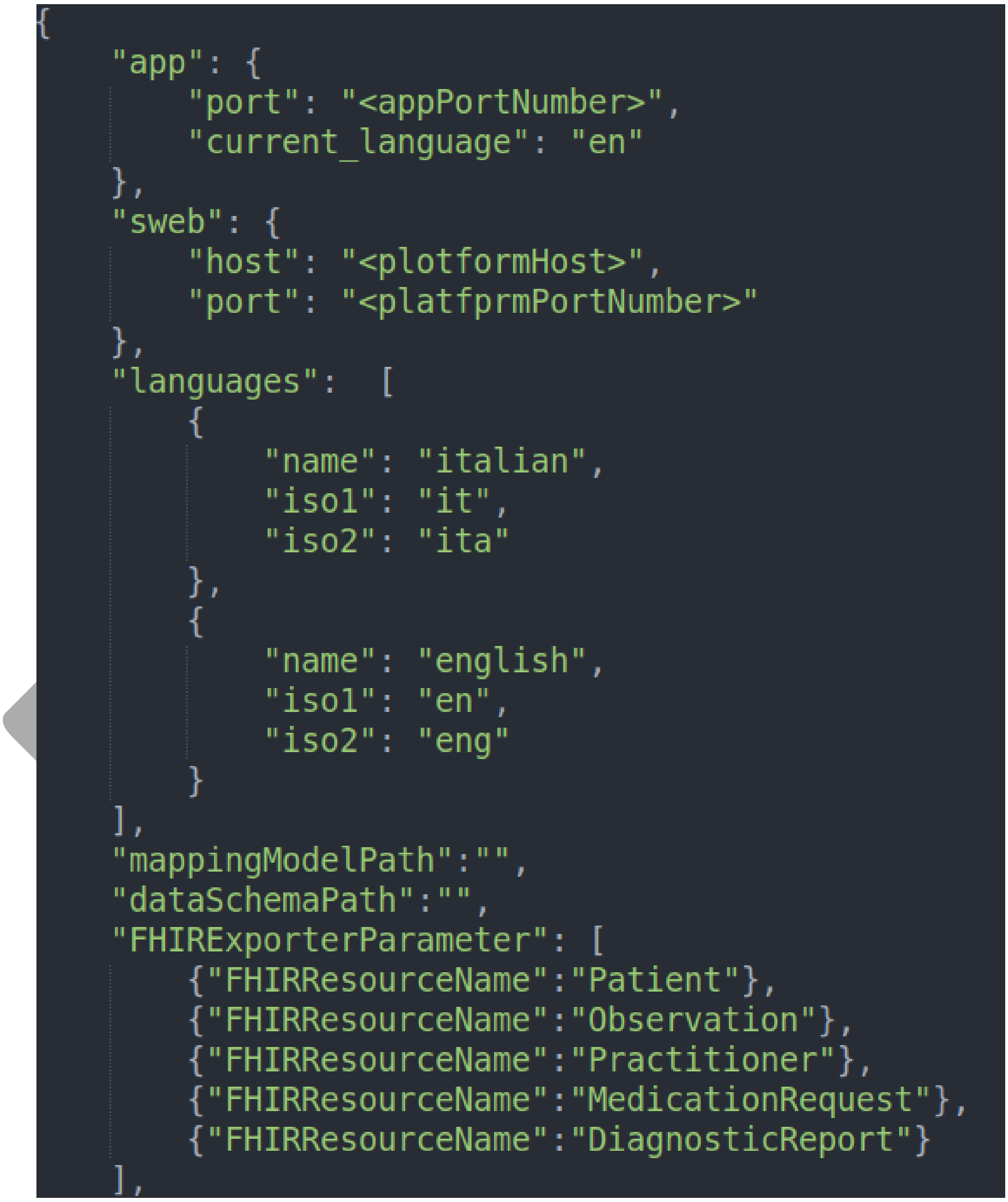
The IHS Controller (see Figure 1), detailed in [D5.8], acts as a middleware between calls from external applications (such as the HCP App) and health data conversion and translation functionalities supported by the DI Platform. Its role is to adapt the latter operations to the specific requirements of the local hospital and/or citizen context: the local language, the local standards used, the local IT system, etc. The IHS controller is composed by two principal software components, provided as a RESTful web services, listed below:

- the *Translation service*: provided to the HCP applications to translate the content of the FHIR health records in a target language specified. More details about the Translation service design are available in [D5.20].
- The *Conversion service*: provided to the EHR information system as a RESTful service used to convert local health records into FHIR health records, as well as to convert medical standard codes used locally, within the health records, by the hospitals, into international medical standards codes. More details about the Conversion service design are available in [D5.10]. The Conversion service exploits another internal software component provided by the IHS Controller, called *FHIR exporter*. This component is in charge of extracting the FHIR version of the health records temporarily stored in the EHR Data cache, and producing the FHIR Bundle that contains such information.

Moreover both the services listed above are supported by another internal software component provided by the IHS Controller, called *Platform Hub*. This component is a lower level API layer that allows the IHS to exploit the RESTful API offered by the DI Platform (see section 3.1).

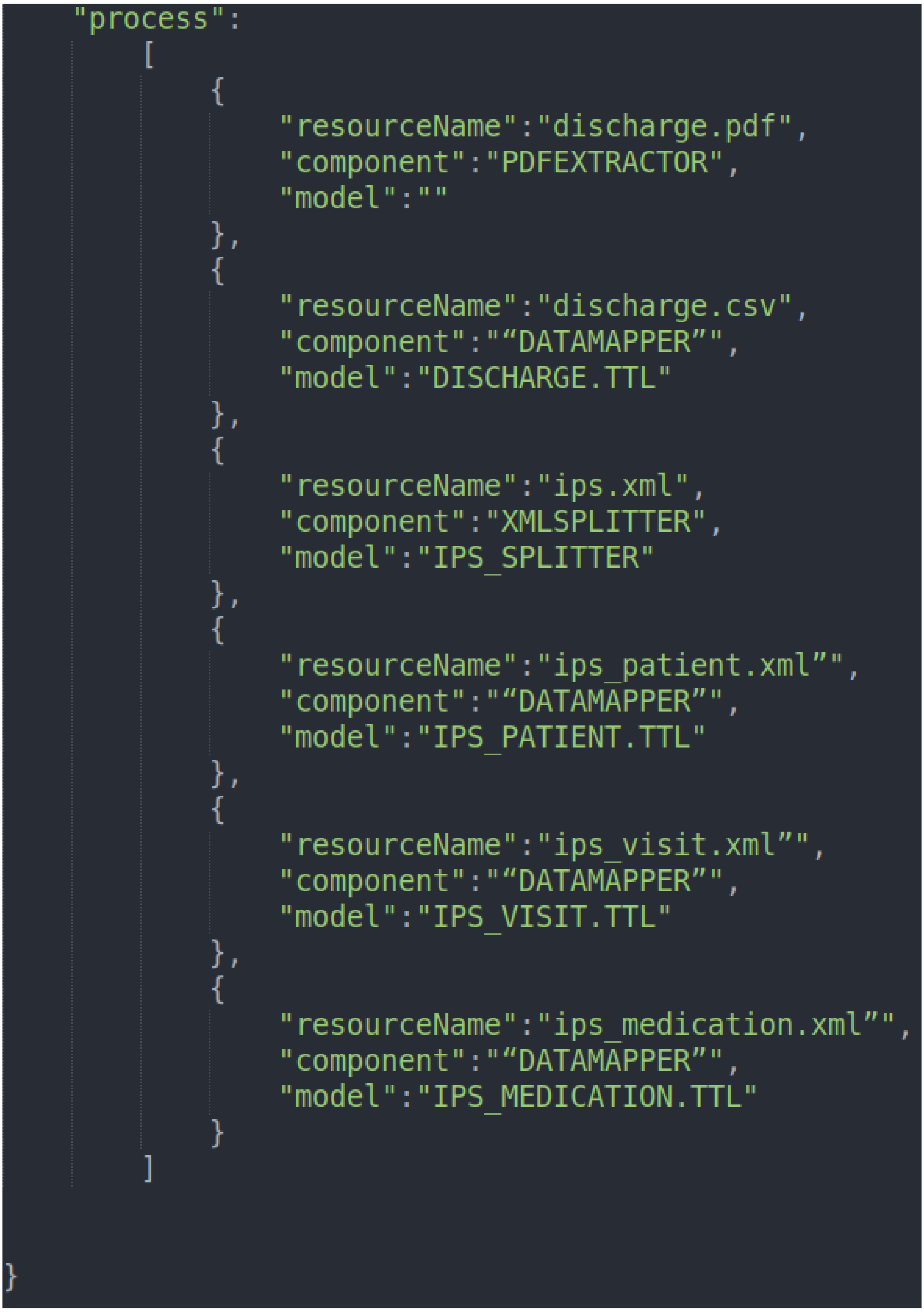
2.2.1. Building and installation guide

The IHS Controller is developed in NodeJs, due to that a NodeJs environment has to be available and properly configured. The IHS Controller requires a configuration file, called *default.json* (see [D5.8] for all the details about the configuration information needed for the IHS Controller) located in the *./config* directory that can be found in the IHS main directory. The figures below show an example of the parameters in the IHS Controller config file. The first figure shows a portion of the config file including most of the required configuration parameters, while the second figure shows the remaining part of the same configuration file where the *process* parameter is included.



```
{
    "app": {
        "port": "<appPortNumber>",
        "current_language": "en"
    },
    "sweb": {
        "host": "<plotformHost>",
        "port": "<platfprmPortNumber>"
    },
    "languages":  [
        {
            "name": "italian",
            "iso1": "it",
            "iso2": "ita"
        },
        {
            "name": "english",
            "iso1": "en",
            "iso2": "eng"
        }
    ],
    "mappingModelPath":"",
    "dataSchemaPath":"",
    "FHIRExporterParameter": [
        {"FHIRResourceName":"Patient"},
        {"FHIRResourceName":"Observation"},
        {"FHIRResourceName":"Practitioner"},
        {"FHIRResourceName":"MedicationRequest"},
        {"FHIRResourceName":"DiagnosticReport"}
    ],
```

Figure 4 – Portion of IHS Controller configuration file.



```
    "process":
        [
            {
                "resourceName":"discharge.pdf",
                "component":"PDFEXTRACTOR",
                "model":""
            },
            {
                "resourceName":"discharge.csv",
                "component":"“DATAMAPPER”",
                "model":"DISCHARGE.TTL"
            },
            {
                "resourceName":"ips.xml",
                "component":"XMLSPLITTER",
                "model":"IPS_SPLITTER"
            },
            {
                "resourceName":"ips_patient.xml”",
                "component":"“DATAMAPPER”",
                "model":"IPS_PATIENT.TTL"
            },
            {
                "resourceName":"ips_visit.xml”",
                "component":"“DATAMAPPER”",
                "model":"IPS_VISIT.TTL"
            },
            {
                "resourceName":"ips_medication.xml”",
                "component":"“DATAMAPPER”",
                "model":"IPS_MEDICATION.TTL"
            }
        ]


}
```

Figure 5 – Portion of IHS Controller configuration file with the ‘process’ parameter.

In order to install the IHS Controller, it is only necessary to execute the following command, in the component main directory:

```
npm run start-ihs
```

2.2.2. User guide

The IHS Controller provides a set of RESTful web services that are automatically called by the HCP applications, as well as by the EHR information systems. The interaction between the user and the functionalities provided by the IHS Controller, appears through the Conversion and Translation services, demonstrated in [D5.19] and [D5.20] respectively. Nevertheless, all the IHS Controller services (listed and detailed in [D5.8]) can be called singularly as RESTful web services. Here below is an example of a call for a service used by the hospital EHR information system to convert health records, represented in local format, into FHIR health records.

```
          curl -X POST -F 'resource=@<healthRecordFilePath>'
    http://<host>:<portNumber>/requestConversion?requestID=<reqID>
```

Where:

- <healthRecordFilePath> is the file path for the local health record.
- <host> is the address of the machine where the IHS Controller is installed.
- <portNumber> is the port number used to access the IHS Controller services.
- <reqID> is an internal parameter (generated automatically by the EHR information system) used to return the service's results to the corresponding request.

3. SW DESCRIPTION

3.1. Data Integration Platform environment

SW TITLE	Data Integration Platform
SW VERSION	1.0.0
PROGRAMMING LANGUAGES	Java, JavaScript, SQL
SUPPORTED PLATFORM(s)	Docker Cross-platform environment
SOURCE CODE	Tools and services set

Table 1 – Release information on Data Integration Platform

The DI Platform environment is a software component defined as a set of containers in a Docker environment. The Docker images required for each container, as well as the docker-compose file required to set up the Docker environment composed by the different container, can be provided on request by University of Trento as partner of InteropEHRate project.

3.2. EHR Data Cache database

SW TITLE	EHR Data Cache
SW VERSION	1.0.0
PROGRAMMING LANGUAGES	SQL
SUPPORTED PLATFORM(s)	PostgreSQL Cross-platform environment
SOURCE CODE	Database Docker image

Table 2 – Release information on EHR Data Cache

The Docker image for the EHR Data Cache can be provided on request by University of Trento.

3.3. User database

SW TITLE	User database
SW VERSION	1.0.0
PROGRAMMING LANGUAGES	MongoDB Query Language (MQL)
SUPPORTED PLATFORM(s)	MongoDB Cross-platform environment
SOURCE CODE	Database Docker image

Table 3 – Release information on User database

The Docker image for the User database can be provided on request by University of Trento.

3.4. Platform RESTful API layer

SW TITLE	Platform RESTful API
SW VERSION	1.0.0
LICENCES AND PATENTS	proprietary (UNITN)
PROGRAMMING LANGUAGES	Java
SUPPORTED PLATFORM(s)	Cross-platform
SOURCE CODE	Docker image

Table 4 – Release information on Platform RESTful API

The Docker image for the RESTful API layer can be provided on request by University of Trento.

3.5. Knowledge Management Tools

SW TITLE	Knowledge Management Tools
SW VERSION	1.0.0
LICENCES AND PATENTS	proprietary (UNITN)
PROGRAMMING LANGUAGES	Node.js
SUPPORTED PLATFORM(s)	Cross-platform
SOURCE CODE	Docker image

Table 5 – Release information on Knowledge Management Tools

The Docker image for the Knowledge Management Tools can be provided on request by University of Trento.

3.6. Data Mapper Tool

SW TITLE	Data Mapper Tool
SW VERSION	1.0.0
LICENCES AND PATENTS	Apache-2.0
PROGRAMMING LANGUAGES	Java
SUPPORTED PLATFORM(s)	Cross-platform
SOURCE CODE	https://github.com/usc-isi-i2/Web-Karma
EXECUTABLE	https://github.com/usc-isi-i2/Web-Karma

Table 6 – Release information on Data Mapper Tool

3.7. IHS Controller

SW TITLE	IHS Controller
SW VERSION	1.0.0
LICENCES AND PATENTS	Apache License
PROGRAMMING LANGUAGES	Node.js
SUPPORTED PLATFORM(s)	Cross Platform
SOURCE CODE	http://iehrgitlab.ds.unipi.gr/interopehrate/health-services

Table 7 – Release information on the IHS Controller

REFERENCES

[D2.1] InteropEHRate Consortium, User Requirements for cross-border HR integration - V1, 2019. https://www.interopehrate.eu/resources/

[D2.7] InteropEHRate Consortium, FHIR Profile for EHR interoperability - V1, 2019. https://www.interopehrate.eu/resources/

[D4.4] InteropEHRate Consortium, Design of libraries for remote D2D HR exchange - V1, 2019. https://www.interopehrate.eu/resources/

[D5.1] InteropEHRate Consortium, Software Requirements specification of an integrated EHR web app for HCP - V1, 2019. https://www.interopehrate.eu/resources/.

[D5.4] InteropEHRate Consortium, Design of an integrated EHR web app for HCP - V1, 2019. https://www.interopehrate.eu/resources/

[D5.8] InteropEHRate Consortium, Design of the Health Data Integration Platform - V2, 2021. https://www.interopehrate.eu/resources/

[D5.10] InteropEHRate Consortium, D5.10 - Design of the data mapper and FHIR converter - v2, 2021. https://www.interopehrate.eu/resources/

[D5.19] InteropEHRate Consortium, D5.19 - Library and tool for data mapping and conversion - v2, 2021. https://www.interopehrate.eu/resources/

[D5.20] InteropEHRate Consortium, D5.20 - Library for information extraction and translation - V1, 2021. https://www.interopehrate.eu/resources/

[D5.21] InteropEHRate Consortium, D5.21 - Library for information extraction and translation - V2, 2022. https://www.interopehrate.eu/resources/