Project title: European Federation for Cancer Images
Project acronym: EUCAIM
Grant Agreement: 101100633
Call identifier: DIGITAL-2022-CLOUD-AI-02

D5.5: The EUCAIM tools for Data / Metadata Management & interoperability

Responsible partner(s): MEDEX, group member of BC Platforms

Author(s): Laure Saint-Aubert (MEDEX, group member of BC Platforms), Victor Sóñora Pombo (BAHIA), Rafael Pallarés Palos (BAHIA), Thomas Trouillard (MEDEX, group member of BC Platforms)

Reviewer (s): Michał Kosno (GUMED), Maciej Bobowicz (GUMED), Celia Martin Vicario (QUIBIM)

Date of delivery: 31 January 2025
Version: 1.0

Link to demonstrable:

https://www.youtube.com/watch?v=j7oybC_j5bo

Table of contents

List of Abbreviations and acronyms

AI4HI : Artificial Intelligence for Health Imaging

CDM : Common Data Model

DH : Data Holder

ETL : Extraction-Transform-Load

1. Introduction

1.1. Aim and scope of the deliverable

This report completes the information presented in the demonstration video and jointly contributes to Deliverable D5.5 “The EUCAIM tools for Data and Metadata Management and interoperability”. These EUropean Federation for CAncer IMaging (EUCAIM) tools support the pipeline for the ingestion and transformation of clinical and imaging data from Data Holders (DH), to prepare their datasets to be accessible through the EUCAIM platform. The pipeline aims to adapt to any source of data, whether they come from the five Artificial Intelligence for Health Imaging (AI4HI) projects EUCAIM was originated from (ProCancer-I, ChAImeleon, EUCANImage, Incisive, and Primage), from other established repositories of data, or single research hospitals.

The whole data management and interoperability pipeline is fully aligned with the continuous effort on data preprocessing, for which a first presentation of the tools and services was demonstrated and described in D5.4 “Data pre-processing tools and services”.

In the present deliverable, we present a proof of concept of how Tier 2 and Tier 3 clinical data and imaging metadata are ingested, transformed, and released for their subsequent use by the EUCAIM ETL. In addition, we present a tool capable of extracting imaging metadata from DICOM imaging files, for their subsequent ingestion by the ETL. These tools are meant to be as flexible as possible and user friendly. They are presented here as a demonstrator video. The whole workflow of data preparation that these tools are part of is presented in deliverable D5.6 “Minimum Data Federation and Interoperability Framework”.

This document aims to add context to understand the demonstration, provide additional information about the tools, and expose the perspectives for future work. It is structured as follows: First a brief overview on when the tools intervene in the general workflow of data preparation; then a presentation of each tool, and the technical requirement to use them; finally, the link to the demonstration video is provided, and future work is discussed.

2. Data preparation workflow

As part of the whole data preparation workflow for DH before their data can be used in EUCAIM, two data ingestion tools are provided to ensure data interoperability: the EUCAIM ETL and the DICOM tag extractor. The latter extract the DICOM metadata from all series of an imaging dataset, while the ETL ingest, transform and map clinical data and imaging metadata to a standard. These tools are required for Tier 2 and 3 datasets, whether data are to remain in a federated node (Figure 1), or are to be transferred to any EUCAIM reference node. For Tier 2 datasets, only the variables to query must comply with the EUCAIM CDM, as for Tier 3 datasets, all variables have to comply with it. Further information on data preparation is provided in D5.6 “Minimum Data Federation and Interoperability Framework”.

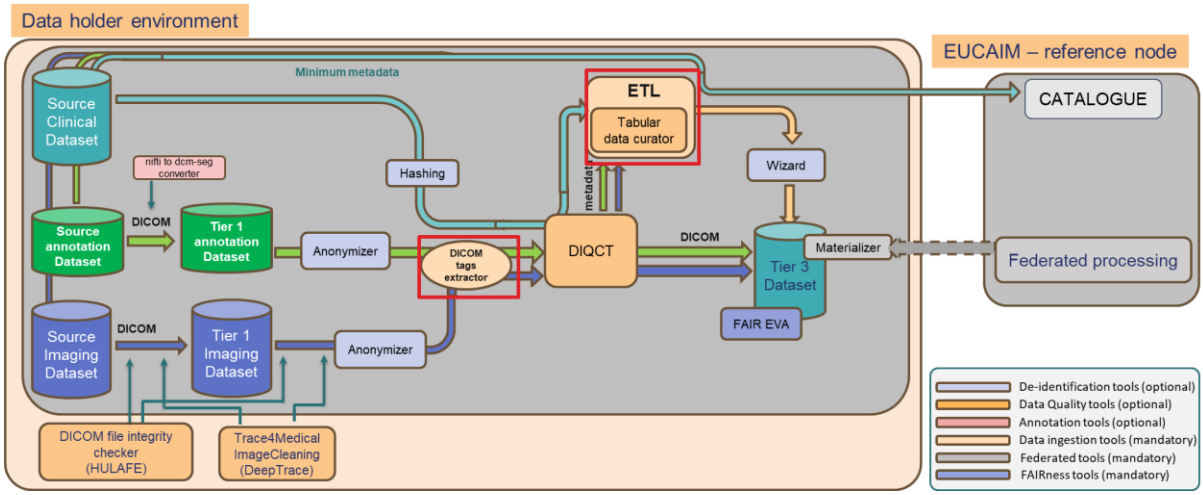

Figure 1: Inclusion of the tools in the Tier 3 data preparation workflow in a federated node. The DICOM tags extractor and the ETL are highlighted with a red frame.

The DICOM tags extractor applies to imaging datasets as well as annotation datasets, providing they are in DICOM format. Imaging metadata extraction is a prerequisite to map imaging metadata to the EUCAIM CDM using the EUCAIM ETL.

3. DICOM tags extractor

Extraction of metadata from DICOM imaging files is necessary to run various quality checks on the imaging data, and map these metadata to the EUCAIM CDM. While some basic information such as the mandatory metadata to collect for the EUCAIM catalogue may be collected manually, the mapping to EUCAIM CDM required automation, as the list of metadata in a DICOM file is extremely long. Such automatic extraction is possible using the DICOM tags extractor.

The tool scans the DICOM files of a defined directory at the series level, and produces as output one single JSON file containing all the DICOM tags for each detected series. DH only needs to provide the path to the folder to scan. The tool works recursively, so if that path is a collection of folders, it will scan all of them.

To use this tool, the user can download an executable file (.exe). It has two execution modes:

1. **manual execution** : the user double-clicks on the .exe, a console will show up asking to input the path to scan (Figure 2), after it finishes it will close and the output.json will appear in the same location as the .exe
2. **command line execution**
    - running without arguments: same as manual execution

    - running with arguments: 1st argument is the path to scan. A second argument, -o or --output can configure the name of the output file (default is output.json)

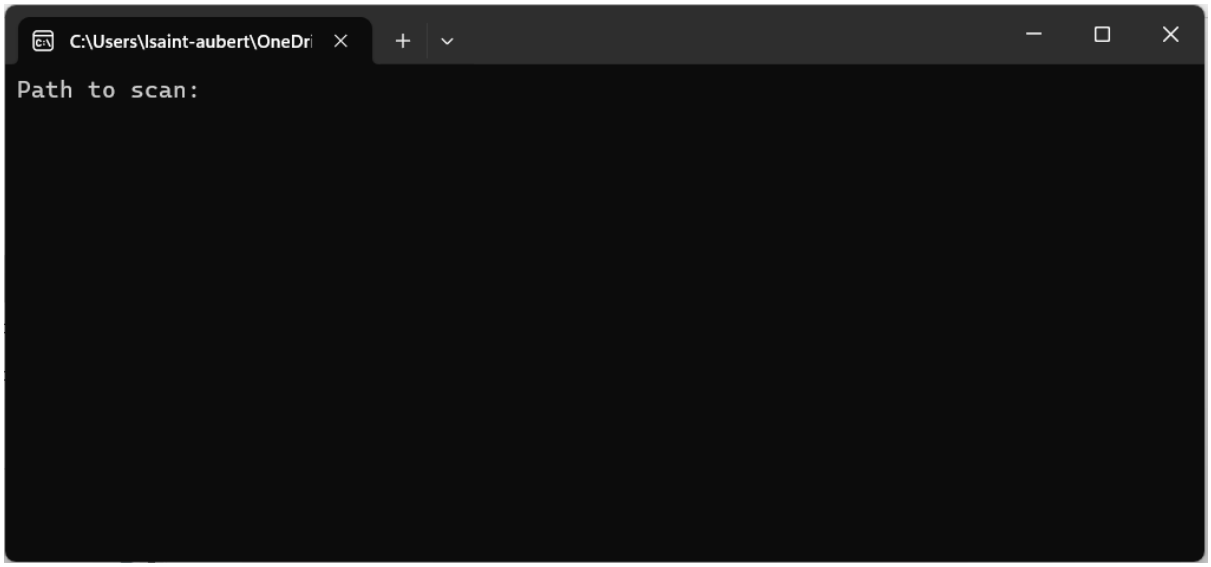

Figure 2: Windows console opened after launching the DICOM tags extractor executable

The path to scan corresponds to the current location where the DICOM files are stored. It is recommended to give the absolute (i.e. complete) path (e.g.: C:\…).

As an output, the tool produces a JSON file with the following structure:

```
{
  'patient_id_1': {
      'study_uid_1': {
          'series_uid_1': {
              'tag_1': {
                "vr": value,
                "Value": value,
              },
              {
                "vr": value,
                "Value": value,
              },
```

```
            ...
          },
        },
        'study_uid_2': {
            ...
        }
    },
    'patient_id_2': {
        ...
    },
    ...
}
```

Where:

- the tag key is the group id and element id of the tag (eg '00101010' for Patient's Age)
- the "vr" key is the VR of the tag, and the "Value" key should be parsed according to the related VR.

For the tool to scan data successfully, it is important that :

- the extension ".dcm" is visible, as the tool only scans those files, and discard any other;
- each DICOM series has its own folder, as the tool gets a single .dcm file from each folder and extracts the tags.

Indeed, the way the tool works, it will provide all metadata at the series level, so it is necessary that the imaging dataset is organized as Patient>Study>Series, with a subfolder for each level. Such structuring of data is mandatory for Tier 2 and Tier 3 data. For Tier 1 DH who would like to use the DICOM tags extractor on their data, they first need to ensure that their imaging dataset follows the above-mentioned structuring. This may be achieved using the data quality tool "DICOM file integrity checker" (see Figure 1). More information on the latter is available in Deliverable D5.4 "Data pre-processing tools and services".

4. Extraction-Transform-Load (ETL) tool

4.1. ETL technology stack

The current version of the ETL is the result of a long research and various development attempts to find the suitable software architecture. Previous work was unsatisfying due to architecture that was lacking the necessary flexibility (implementation built on top of Trino plugins specifically developed for OMOP data). After discarding those, it was decided to refactor the ETL set of processes to an architecture based on Apache NiFi, a robust and powerful software to process and distribute data. This software is designed around the flow-based programming paradigm, allowing for the creation of dataflows as directed graphs, where each node is a "processor" that performs operations on input data, and each node connector defines a flow of data with its own queue.

NiFi includes its own embedded web server for HTTP based communications between different instances (if needed), that facilitates the user experience for designing, monitoring

and managing the dataflows. An extensive collection of processors already covers most common tasks, and extensions support custom processors, giving users the ability to implement any functionality that can be programmed in Java and run within the Java Virtual Machine (JVM).

4.2. ETL overall design

The primary goal of the ETL architecture is to meet the specific requirements of the EUCAIM workflow, including accommodating the expected wide variety (in terms of both format and standards) of datasets from DH, and minimizing manual data preparation efforts. The architecture is designed to be flexible, allowing for customization and integration with other EUCAIM components, such as data quality assurance tools.

The ETL pipeline has been modularized into two distinct phases (**Figure 3**). The initial phase focuses on :

1. data ingestion,
2. basic transformations that require custom definition of the source datasets mapping,
3. data quality checks,
4. to ultimately load the data into a relational data model.

The subsequent phase, which requires no customization, generates the final output in the correct format, according to the EUCAIM CDM.

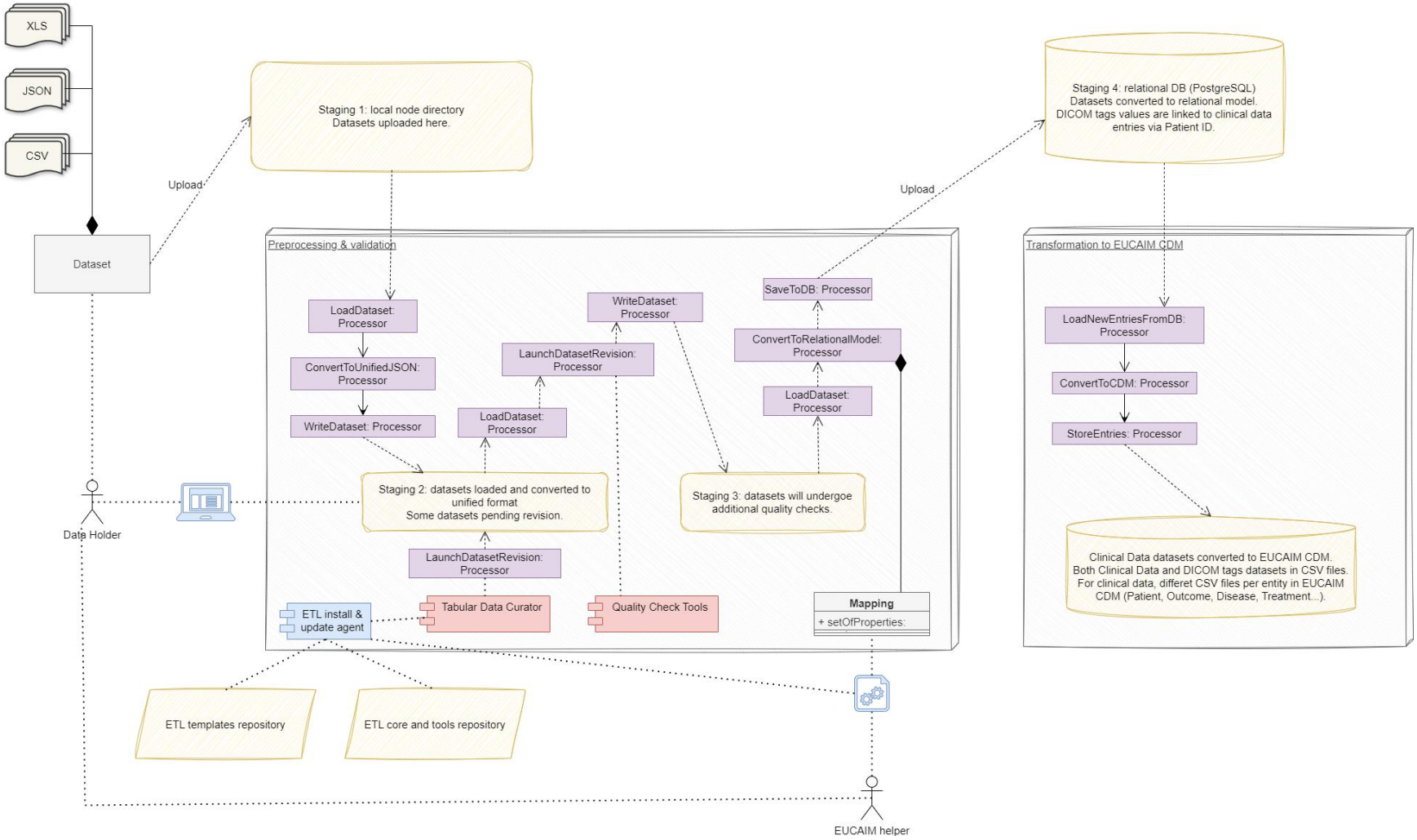

Figure 3: High-level ETL architecture including all the main staging areas, the main groups of processors (purple), external tools integrated in the ETL (red), dependencies with the original datasets (grey), and auxiliary ETL deployment and update subsystem (blue)

4.3. ETL design for the main data flows

To explain the main steps of the ETL, the metaphor of “loop” will be used. A loop refers to a dataflow that converts an input dataset (either clinical data or DICOM tags) from files or relational databases, into another file or relations tables as output, sometimes executing and integrating with external tools.

The first iteration (Loop 1) of the process deals with the ingestion of original datasets in diverse formats, and their harmonization into a unique standardized CSV format. Figure 4 shows an example of Loop 1 for three different clinical data source files of currently supported formats: a .csv, a .json, and an .xls. The same first loop is also illustrated in Figure 5 for imaging metadata ingestion, where the structured JSON file containing all DICOM tags is flattened to CSV format.

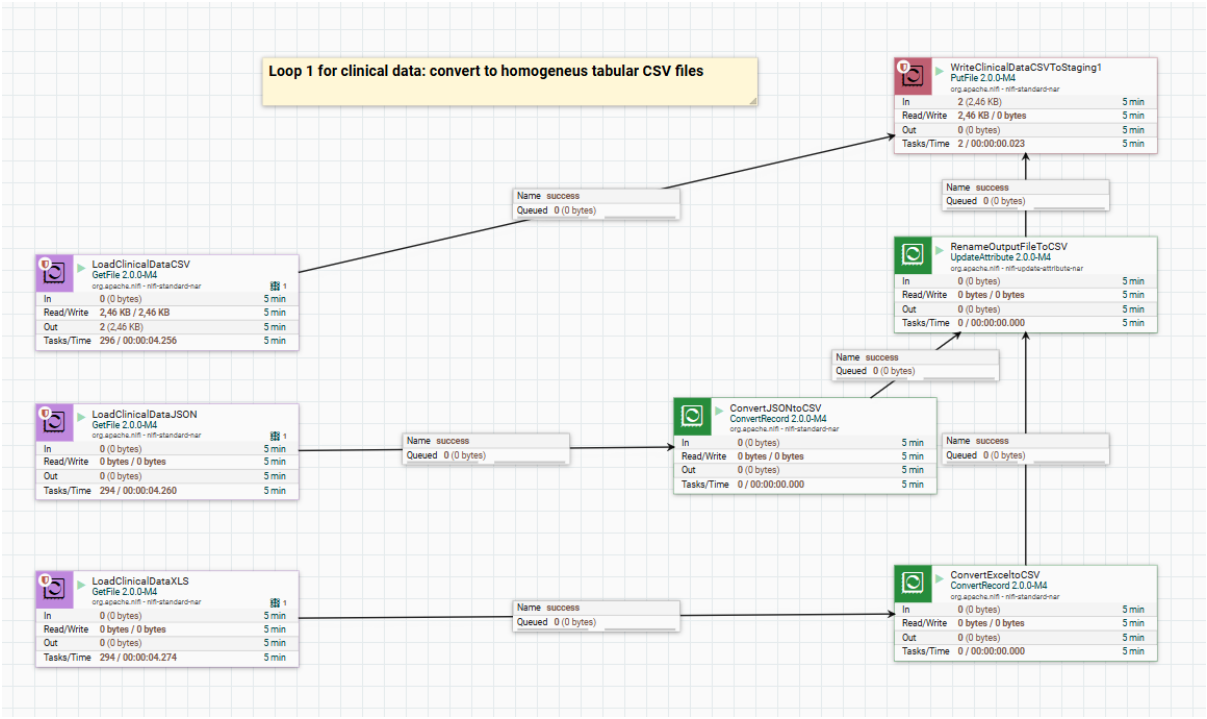

Figure 4: Ingestion of original clinical datasets of various formats and their unification into a standardized CSV format

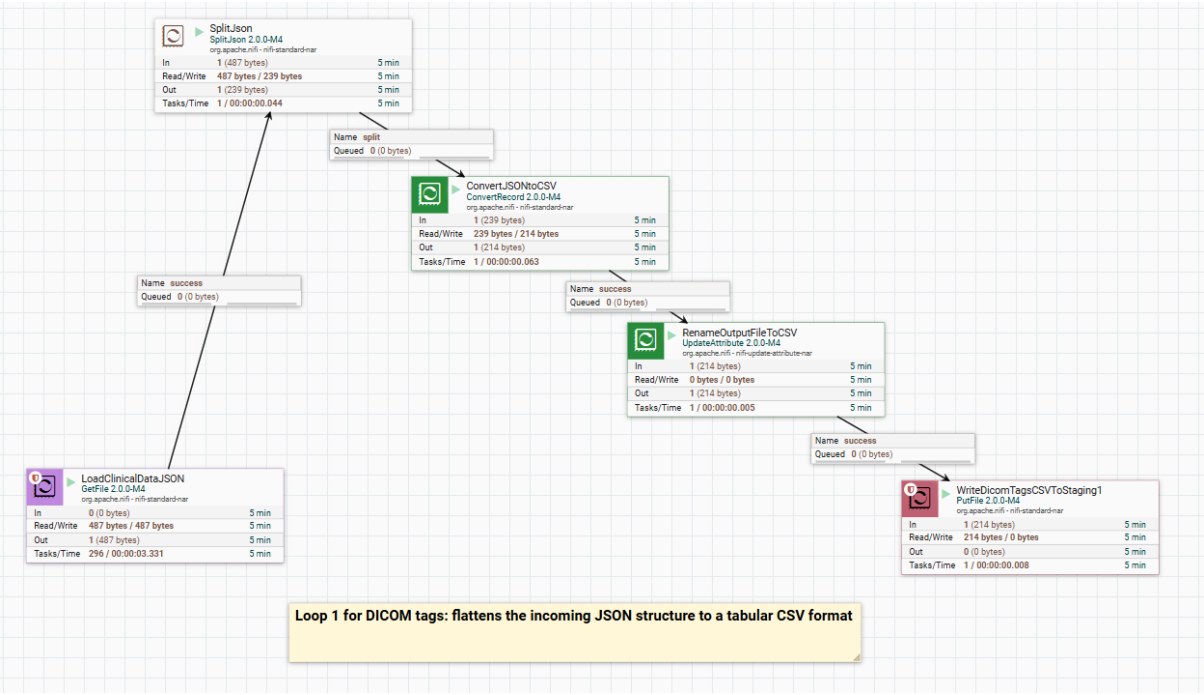

Figure 5: Ingestion of imaging metadata json source file and its flattening to csv format

The second loop, not depicted in this deliverable, involves the execution of the **Tabular Data Curator** on the harmonized CSV files for clinical data. This tool, already presented in D5.4 “Data pre-processing tools and services”, has the capability to generate a curated version of a csv dataset, applying various curation preprocesses including outliers and missing values management, and ensuring the data is correctly structured for the subsequent analyses. It generates a data quality evaluation report, a curated dataset highlighting the problematic fields, and another curated dataset removing the features that did not pass the quality rules. Originally, this tool was presented as a stand-alone tool, but we now plan on having it embedded into the ETL pipeline. A web-based user interface is being developed to integrate these outputs, centralizing this interaction to offer the DH user the choice to either use the curated clean dataset, to review and edit the problematic fields, or to reject the automatic corrections and manually review the dataset (e.g. missing values that were imputed).

The third loop in the ETL involves more complex transformations to map the source data to the intermediate relational model. This includes several sub-loops that will be detailed later in this document, as well as in the video demonstration. A simplified schema used in early ETL versions with specific mappings is presented in Figure 6. It highlights a key design goal of the ETL architecture: the dependency on the original datasets definition is encapsulated within a single processor. In this example, also illustrated in the demo video, each patient record in the source file contains three variables: Drug_Treatment_1, Drug_Treatment_2, and Drug_Treatment_3. The ETL was adapted so that these are mapped to separate Treatment records in the relational model. For less complex cases, mappings can be defined using a simple SQL-based expression language, and these mappings are configured once during the ETL deployment for a specific DH.

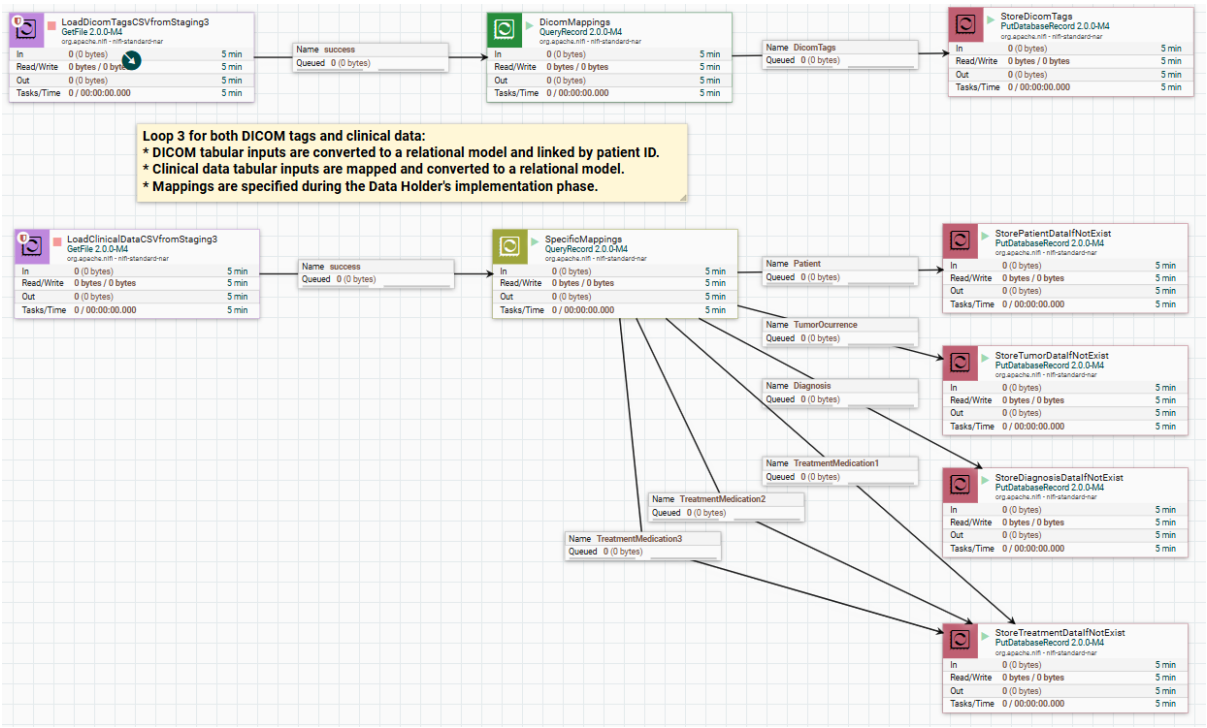

Figure 6: Transformations into intermediate relational model for clinical data and DICOM metadata

However, this loop is still missing extension points to review the results of the Tabular Data Curator quality tool. The user should be able to decide whether to automatically accept the curated dataset or manually review it before re-running the ETL. To facilitate this process, a web-based user interface is under development, to provide easy access to these reports and record Data Holder decisions.

Finally, the last transformation block in the ETL process depicted in Figure 7 requires no specific adaptations. It retrieves recent entries from the relational model tables for both DICOM tags and clinical data, and translates these into valid CSV files according to the EUCAIM CDM, generating separate CSVs for each corresponding CDM entity.

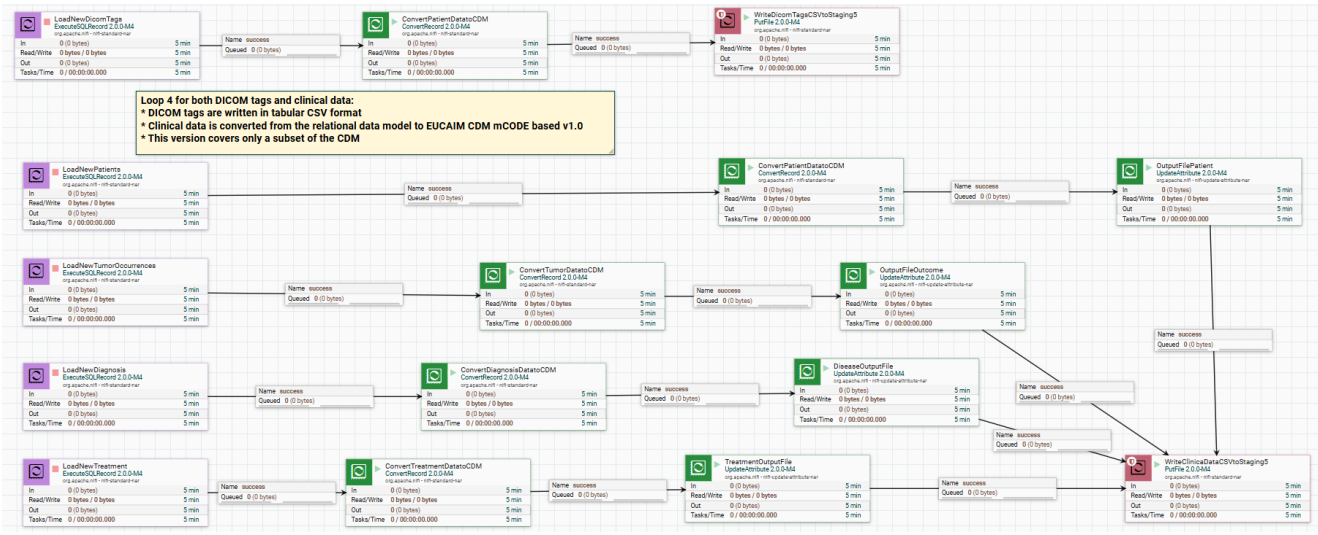

Figure 7: Final transformations, exporting datasets currently as CSV files following EUCAIM CDM

A few remarks :

- For simplicity, most “failure” dataflows were omitted in the version depicted here. For each data processor, there are different flows for the successfully processed records (“success” edges) and also for the ones that failed to be processed (“failure” edges). At various stages, particularly in the third loop (dataset harmonized and curated, passing quality checks and being mapped to the intermediate relational data model), a set of rules should check conditions like the completeness of the mandatory properties, generating alerts and storing in a different CSV the set of records that could not pass to the next steps.
- At any point of the ETL, this technology provides full data provenance and traceability support. Each processor contains the full log of operations that were executed, storing both the input and output data.
- The full design and configuration of the ETL can be exported and imported again, in different nodes, as a simple JSON descriptor file.
- The dependencies on a local specific installation are minimized through the use of sets of environment variables, requiring minimal configuration during setup phase.

3. Technical requirements for the EUCAIM tools

3.1. DICOM tags extractor

As it currently stands, the tool is an executable to download locally, with no specific requirement for installation. Future work (see section 5) will aim at adapting the tool to be run on any operating system. Specific installation requirements may appear then.

3.2. ETL

The whole ETL pipeline for both DICOM tags and clinical data can be set up in a single node in Data Holder premises where the datasets can be uploaded, or it can be deployed in different colaborating nodes connected by HTTP. For simplicity, the technical requirements for a single node are outlined here. The requirements for a setup in multiple nodes will be mostly the same, but for each node.

Installation and software setup

The preferred method to deploy or install the ETL will be using a set of Docker images distributed using Docker Compose. These images will contain the ETL server hosted using Apache NiFi, and a relational database implemented in PostgreSQL. The only requirements for installation with this method will be:

- Computer with either Windows or Linux OS with support for Docker and Docker Compose, including privileges to run Docker Compose and, if necessary and allowed, to install it.
- Recommended 16 GB of RAM.
- Advanced GPU support is NOT needed.

ETL specific mappings setup

The mappings and transformations required to accommodate the peculiarities of the original dataset to the intermediate relational model are centralized in SpecificMappings processors. In the current prototype versions, these mappings are specified directly in NiFi using SQL syntax and Expression Language. The intention is to include mapping management in the web-based user interface.

The specification of these mappings for a specific ETL installation for a Data Holder must be done in a joint session with the collaboration and approval of the DH. This specification will be straightforward, and in principle, it should only need to be done once, unless there are changes in the structure of the original datasets.

4. First demonstrator of the preprocessing tools

The video demonstration of this deliverable is accessible here :
https://www.youtube.com/watch?v=j7oybC_j5bo

It displays how the ETL and the DICOM tags extractor work with Tier 3 clinical data and imaging metadata (as shown in Figure 1).

5. Future work

5.1. General

The work presented in this deliverable is the fruit of numerous discussions and interaction among WP5 partners, as well as with partners from other WPs. Continuous effort will be made in the next period of the project to keep aligning this work as well as all the work around tool development across WPs, as we see that what comes as an output for some is the input for others. Also, further work is warranted on workflow simplification across tools, to make the user experience as easy and smooth as possible. Additional work will also be led to test the ingestion of several types of source data.

5.2. DICOM tags extractor

As briefly mentioned earlier, future work will consist in adapting the tool to any operating system. Documentation will be provided as a “Read Me” to the users. Finally, improvement on the handling of errors and the related information displayed on the screen will be made.

5.3. EUCAIM ETL

The development of the ETL will be an on-going task in the coming period. We have already identified a series of adds-on and improvements to bring, that are as follow, in order of priority:

1. To develop and test prototype/mockup of web-based UI to centralize user management of datasets being preprocessed, curated and validated. This UI will also

allow easy access to global configuration (hashing enabled/disabled, etc), and will display logs and alerts.

2. To refine and refactor the rules for transformation to EUCAIM CDM (currently embedded in simple processors, one for type of entity).
3. To specify and develop the checks for mandatory properties and basic mandatory format restrictions (DICOM metadata structure).
4. To develop all the secondary dataflows to handle data that did not pass the checks for the different core processors.
5. To specify and develop an improved management of ETL definitions versioning. Since the entire ETL definition can be exported and imported using JSON files, a simple version control system will be implemented to receive any important updates added to already deployed ETLs.
6. To define a kubernetes oriented (Helm Charts) deployment of the ETL components.
7. To integrate the launcher processor for the Tabular Data Curator.
8. To test and develop the loops for quality checks, integrating logs and alert flows with the new web-based UI.
9. To add feedback to the user in case the source file is not valid (example: no “patientID” variable recognized as key variable in the file).