Medical Informatics Platform Releases - SOFTWARE & REPORT: D8.5.3 (D52.3, D9)

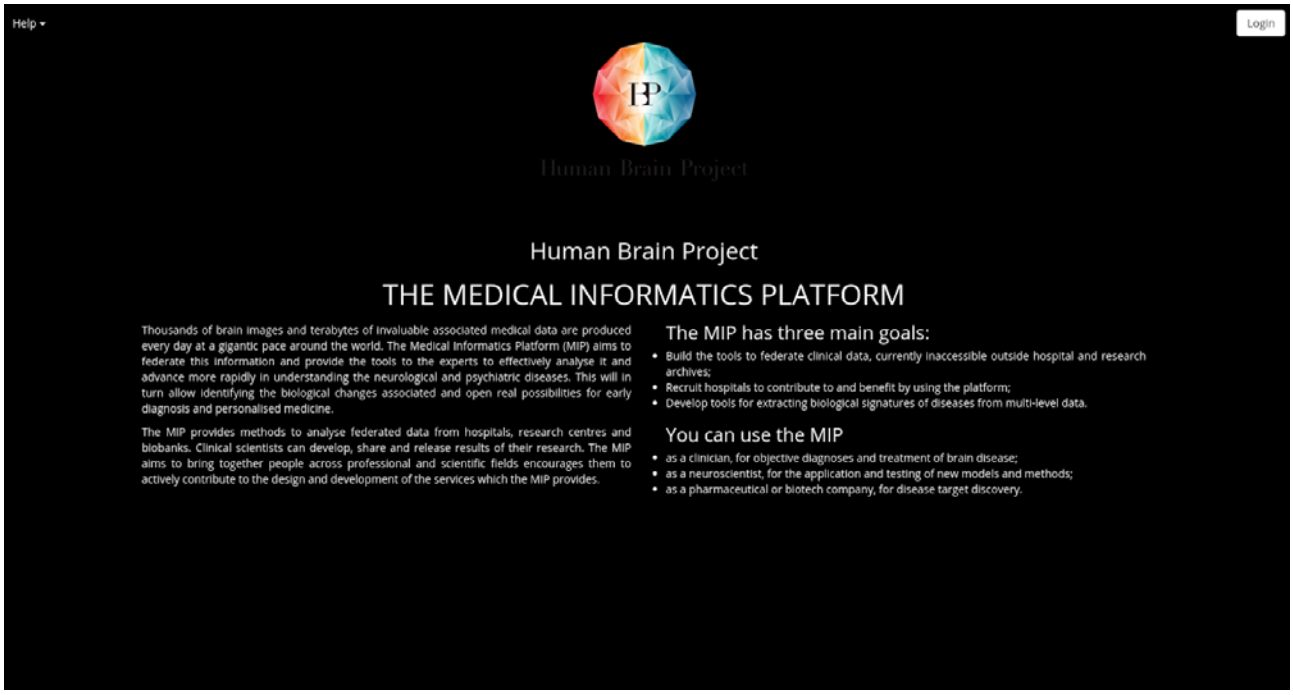

Figure 1: Medical Informatics Platform Release 5.0

Project Number:	785907	Project Title:	Human Brain Project SGA2

Document Title:	Medical Informatics Platform Releases – SOFTWARE & REPORT
Document Filename:	D8.5.3 (D52.3 D9) SGA2 M20 ACCEPTED 200731.docx
Deliverable Number:	SGA2 D8.5.3 (D52.3 D9)
Deliverable Type:	Report
Work Packages:	WP8.5
Dissemination Level:	PU = Public
Planned Delivery Date:	SGA2 M20 / 30 Nov 2019
Actual Delivery Date:	SGA2 M21 / 19 Dec 2019; Accepted 31 Jul 2020
Author(s):	Evita MAILLI, UoA (P43), Giorgos PAPANIKOS, UoA (P43), Kostis KAROZOS, AUEB (P4)
Compiled by:	Evita MAILLI, UoA (P43)
Contributor(s):	Christian DHONDT, CHUV (P27), Thanasis KARABATSIS, UoA (P43), Sofia KARVOUNARI, UoA (P43), Jacek MANTHEY, CHUV (P27), Jason SAKELLARIOU, UoA (P43), Manuel SPUHLER, CHUV (P27), Eleni ZACHARIA, UoA (P43)
SciTechCoord Review:	Mehdi SNENE, EPFL (P1)
Editorial Review:	Guy WILLIS, EPFL (P1)
Description in GA:	Medical Informatics Platform Releases - SOFTWARE & REPORT
Abstract:	This document describes the Medical Informatics Platform (MIP) for the M20 release 5.0 in SGA2. The document focuses on major new developments for the MIP according to the SGA2 Grant Agreement and the product roadmap as it evolved in SGA2 after M12 release, and presents releases for all the major software components of the MIP.
Keywords:	Medical Informatics Platform, Release, Software, Data Processing Engine, Data Catalogue, Anonymisation, Metadata, Scientific Workflow Engine, Galaxy, Quality Control Tools, Deployment, Documentation
Target Users/Readers:	Researchers, Policy Makers

Table of Contents

Table of Tables

Table of Figures

Summary

This Deliverable describes the SGA2 M20 release of the Medical Informatics Platform (MIP). Firstly, we indicate where the MIP and its documentation are available. Then, we describe the major new developments in the M20 release, according to the SGA2 Grant Agreement, the Deliverable D8.5.1 Software Requirements and specifications, and the evolution of the product roadmap to cover emerging requirements. The Deliverable also documents releases for the major software components of the MIP. Appendix A focuses on privacy and privacy-compliant algorithms and describes how privacy is enforced in the federated analytical capabilities of the MIP. Finally, the evolution of the MIP architecture is described in Appendix B.

1. Introduction

During the M12 – M20 stage of SGA2, the focus of the development effort was on securing a more robust and easily deployable MIP that would contain a new data model, while strengthening user experience and expanding analytical capabilities. Release 5.0 is far more stable and easier to deploy than M12 release 4.0.

One of the key changes that contributed to this enhancement was the fact that the Woken component was replaced by EXAREME - Federated Data Processing Engine component. The removed Woken component is an orchestrator platform for Docker containers relying on a runtime environment supported be Mesos[1] and Chronos[2] to control and execute the Docker containers over a cluster. A set of web interfaces exposes needed functionality to interact with the underpinning Docker images that provide the required analytical capabilities. The deployment process of this component added a large degree of complexity both at service configuration time as well as at data ingestion time in order to generate a series of feature tables containing the needed clinical data over which the analytical components operated. Furthermore, observed runtime instability was often hard to trace back to the faulty components. Additionally, the cluster based analysis approach could not be utilised for needed federated analysis as the two approaches have fundamental semantic differences with respect to data boundaries. For these reasons, the EXAREME component was favoured to undertake local analysis in the same way it was already used for federated analysis, restricting its processing pattern to single node execution. EXAREME is a query processing engine optimised for execution of federated database queries extended with user-defined functions processing data and exchanges aggregated results when crossing data ownership boundaries.

Another key new feature that stands out is the introduction of selection of datasets per Medical Condition (front end component, data factory component), a requirement that came from the need to deploy the MIP into networks of hospitals specializing in different pathologies (i.e. Dementia, Traumatic Brain Injury, Epilepsy). This document focuses solely on new developments since the previous release 4.0 (described in D8.5.2).

2. Deployment

A high level design of the deployed service ecosystem is presented in the following diagram. The diagram does not aim to detail each and all the comprising components but rather give an overview of the landscape.

[1] http://mesos.apache.org/
[2] https://mesos.github.io/chronos/

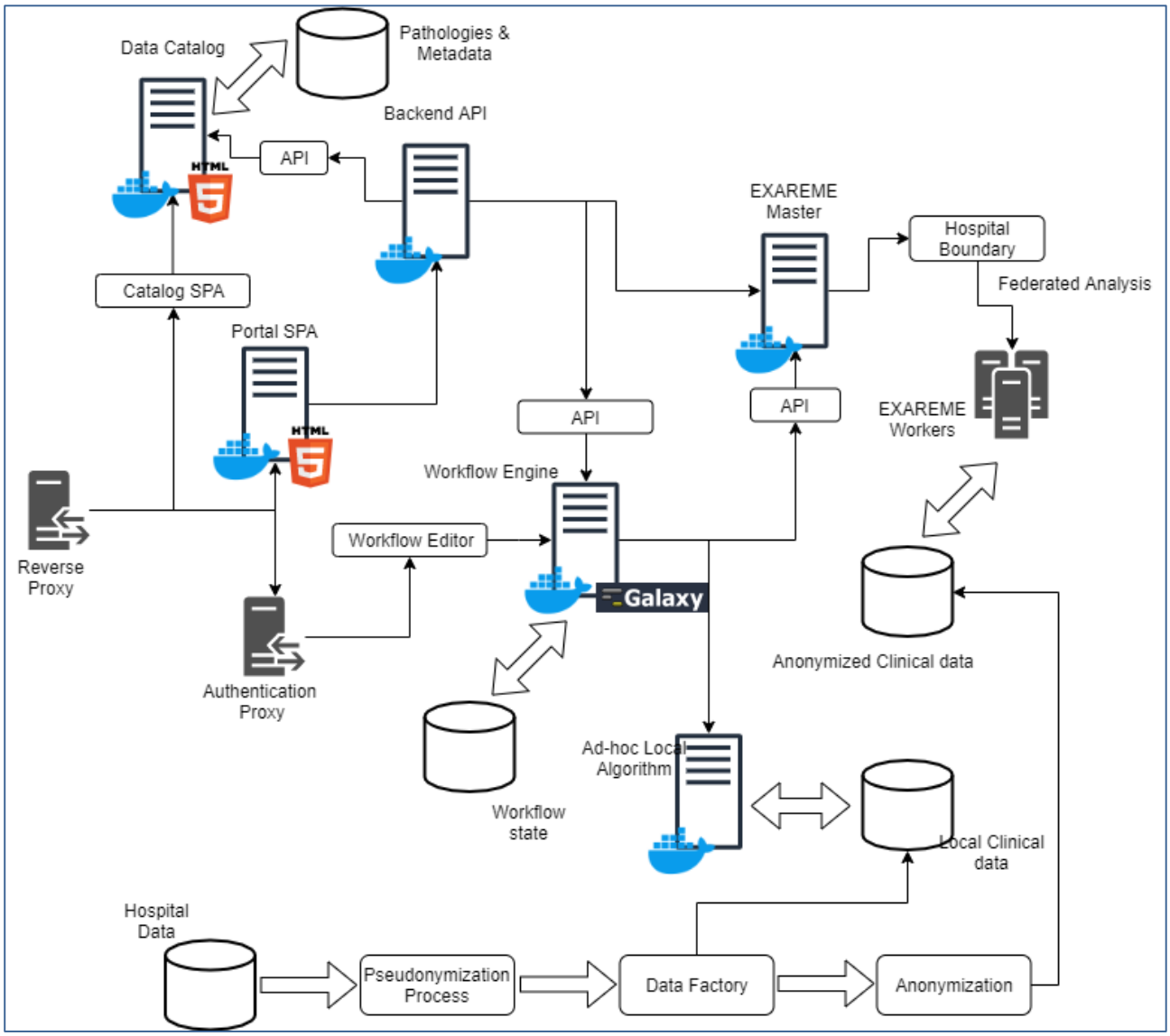

Figure 2: Deployment architecture overview

Key services and characteristics of the service deployment presented in Figure 2 include:

- The front most component is a Reverse Proxy configured to receive all incoming requests and forward them respectively, handling also secure transfer over https bindings
- A web server serving the Portal Single Page Application (SPA) that facilitates user interaction through the MIP Portal UI
- A web server serving the Data Catalogue Single Page Application (SPA) that facilitates the definition of pathologies and respective metadata along with a set of APIs that enable information retrieval
- An internal authentication proxy that serves to properly authenticate and authorize access to the visual Galaxy Workflow Editor
- The Galaxy Workflow Engine that allows both visual workflow editing as well as API based invocation of defined workflows. Such workflows may in turn contact the EXAREME federated analysis API or even orchestrate local component execution
- The EXAREME master node that orchestrates the federated analysis through the registered worker nodes that may span multiple hospital domains
- The Backend API that acts as the gateway for most of the underpinning modules and backs the operations available through the MIP Portal UI
- The Data Factory pipeline that handles the ETL process for data ingestion

- Most of the components available for the deployment and composition of the MIP platform are made available as Docker images to allow easier deployment and configuration

Deployment scripts for the MIP can be found in the following addresses:

All repositories listed are open access repositories.

https://github.com/HBPMedical/mip-deployment-infrastructure.

EXAREME:

https://github.com/madgik/exareme

Demo Installation of MIP:

https://github.com/HBPMedical/mip-deployment-infrastructure/blob/master/MIP-LOCAL-DEPLOYMENT.md

Whole Installation pack:

https://github.com/HBPMedical/mip-deployment-infrastructure/blob/master/README.md

Local Installation of EXAREME:

https://github.com/madgik/exareme/blob/master/Local-Deployment/README.md

Federated Installation of EXAREME:

https://github.com/madgik/exareme/tree/master/Federated-Deployment

Galaxy / Reverse proxy:

https://github.com/madgik/galaxy/blob/master/README.md Galaxy Middleware API:

https://github.com/madgik/Galaxy_Middleware_API/blob/master/README.md

3. Documentation

Software code and documentation for the MIP can be found in the following address:

https://github.com/HBPMedical. This is an open access repository.

4. Release Overview

4.1 New additions / developments

This section describes the evolution of features / components for the MIP, compared to the SGA2 M12 MIP release (Release 4.0). These include the addition of new algorithms, introduction of multi - pathologies to the data model, and integration of Scientific Workflow Engine[3] with MIP portal. In parallel, we simplified new algorithm addition and testing, data factory packaged installation and expanded documentation.

4.1.1 Privacy compliant analytical capabilities and anonymisation

One of the main strengths of the MIP lies in the existence of federated, privacy-compliant statistical analysis and machine learning algorithms. Details on the privacy-compliant approach can be found

[3] https://galaxyproject.org/

in D8.5.1 "Software Requirements and Specifications". For the scope of this project, a detailed description of the anonymisation approach guidelines for the MIP can be found in D12.4.8 "Anonymisation Process in Local/Federated MIP".

4.1.2 Analytical capabilities

For Release 5.0, work continued with improving federated algorithms input / output management, and implementation, integration and testing of the following algorithms (according to the priorities set by Deliverable D8.5.1 (Software Requirements and Specifications, and ongoing discussions with the product owner and stakeholders):

1) ANOVA. Procedure used to analyse the differences among group means in a sample.
2) Cross Validation. Procedure used to validate the training phase of supervised learning algorithms (Naive Bayes for the moment).
3) Descriptive Statistics. Simple algorithm for returning to the user a basic statistical description of the selected variables.
4) Histograms. Procedure used to create histograms of the selected variables, to be displayed graphically to the user.
5) Logistic Regression. Procedure used to train a logistic model. This is a model for the probability distribution of a binary dependent variable.
6) Paired T-test, Simple T-test. Procedures used, among other things, to determine if the means of two sets of data are significantly different from each other.

All of the above developments can be found in:

https://github.com/madgik/exareme/tree/master/Exareme-Docker/src/mip-algorithms.

4.1.3 Data Catalogue

The main difference of the latest Data Catalogue release to the one in the SGA2 M12 Deliverable is the notion of the multiple medical conditions (pathologies). Until M12, the idea was to have a single common data model that would include all pathologies and variables. When new hospitals with new types of data were being added to the MIP federation, this was proven not to be optimal nor convenient for analysis purposes. Therefore, there has been a decision to keep separate different data models depending on the medical condition each hospital is interested in. As a consequence, there are more than one federations of hospitals each one using its own global data model and includes data from the hospitals that actually focus on the related medical condition.

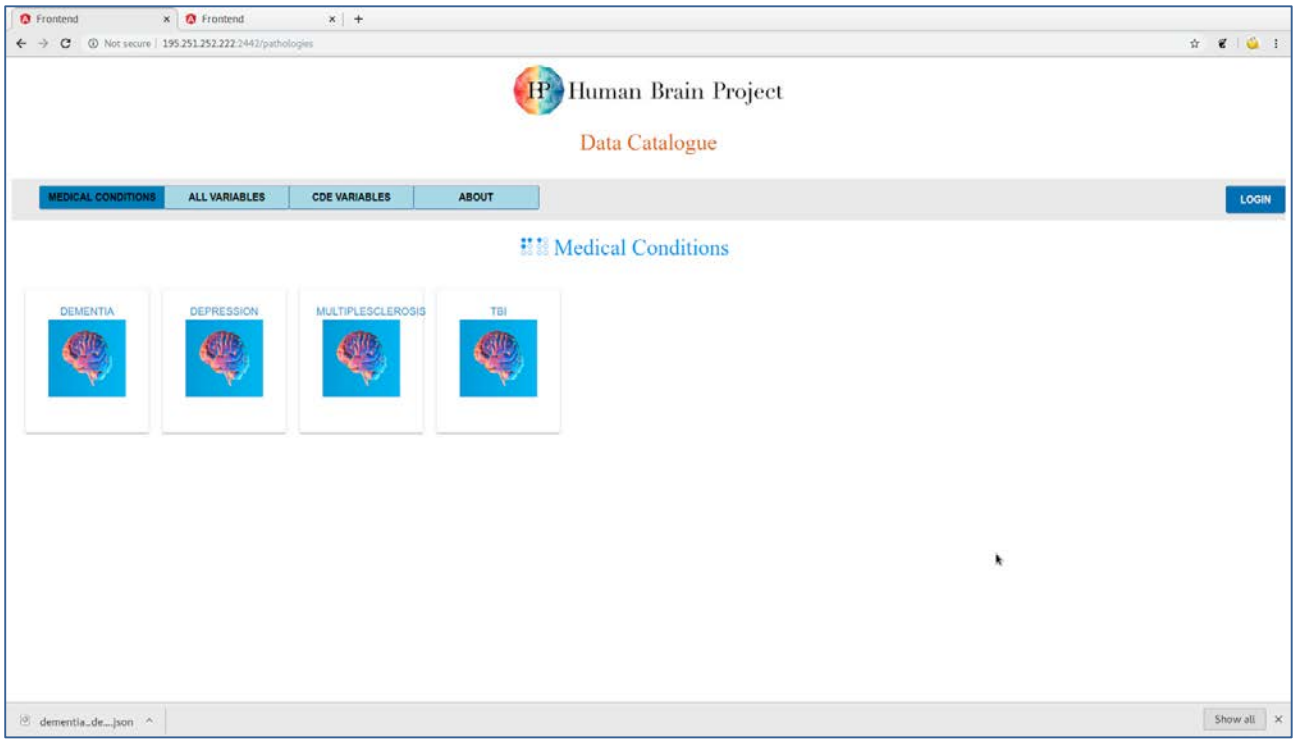

Figure 3: Medical Conditions in Data Catalogue

The required changes on the Data Catalogue front-end, back-end and databases have been designed and developed to support this.

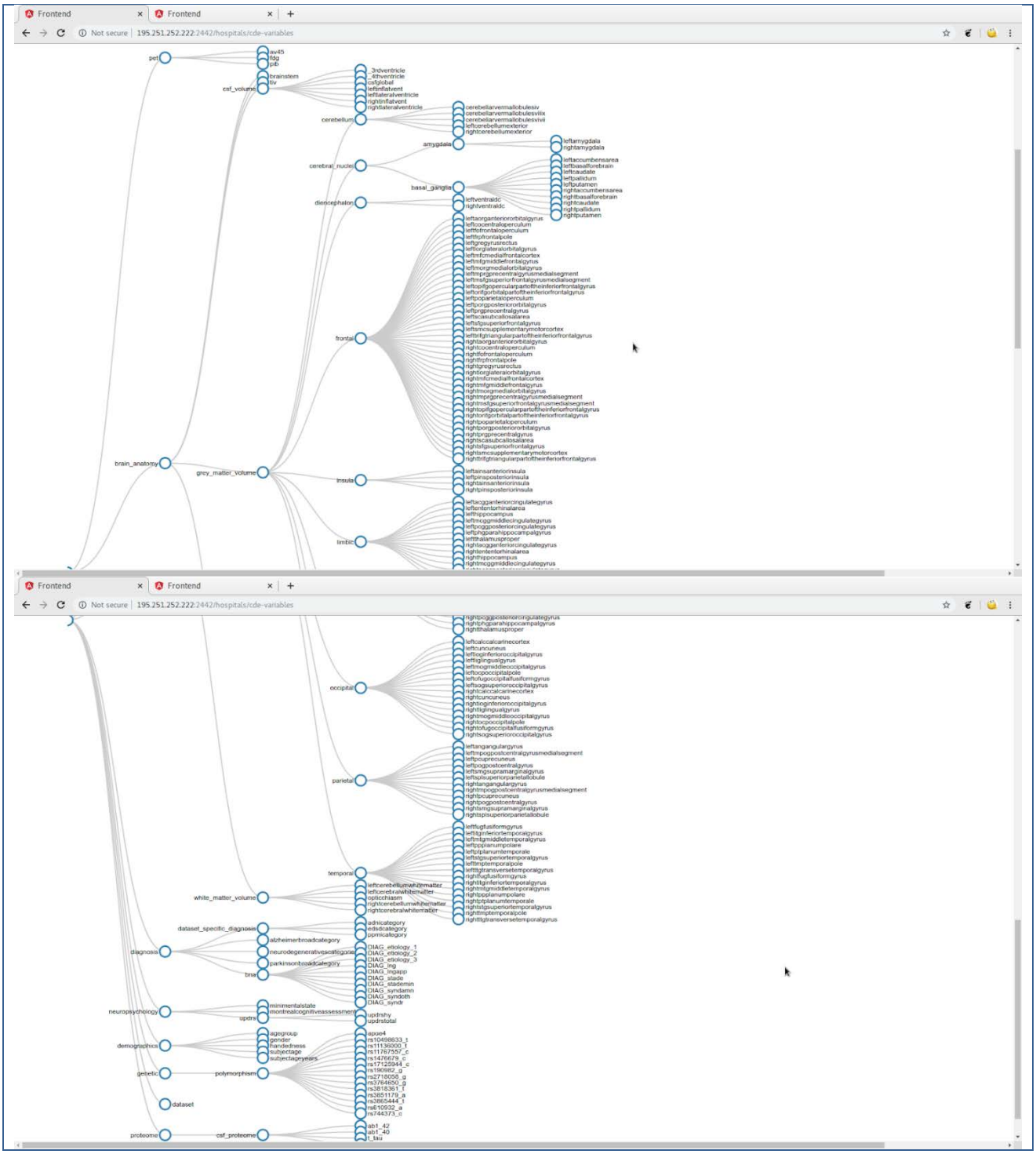

Figure 4: Dementia Data Model as depicted in the Data Catalogue

The data models of the Medical Conditions are presented and described by the Data Catalogue via its various pages among which there is the hierarchy tree. In Figure 4 we have the example of the dementia model which is also described below in a high level.

The dementia data model consists of 181 variables which are organised in 7 groups:

- PET - Scores referring to a Positron-Emission Tomography examination
- Brain Anatomy - The brain volumetric values generated by John Ashburner's feature extraction tool after processing MRI scan images
- Diagnosis - Diagnosis enumerated variables related to dementia, Alzheimer's and neurodegenerative diseases

- Neuropsychology - Scores for neuropsychology tests such as MMSE and MoCA
- Demographics - Basic demographic attributes of the patient such as age and gender
- Genetic - Genetic features (SNP's) of the patient's DNA
- Proteome - Proteins in the patient's DNA

4.1.4 Quality Control tools

We have been working on complementing the QC tools with data validation and data cleaning features. The QC tool for tabular data, on its latest version, is more than a plain profiling tool. It takes as input each dataset CSV file along with its corresponding metadata file to validate whether data values actually conform to their descriptions and rules. More specifically, it does the following:

- Checks column names and number
- Checks if values conform to their stated data type. In case of an inconsistency, it cleans the variable (replaces it with NULL value).
- For polynomial data type, if a data value is not listed in the enumeration list the QC tool uses Levenshtein [4] distance so as to propose the correction with a similar existing one.
- Checks dates and in case of wrong syntax it examines alternative regular expression patterns.

The aforementioned cleaning recommendations depend heavily on the metadata file that characterizes the dataset. We are also examining the idea of a dataset-based cleaning recommendation strategy, which will be based on the distribution of an existing dataset that will be considered as reference. When values largely diverge, there will be a strong possibility of an error. This mostly can be applied for the brain volumetric variables.

4.1.5 Scientific Workflow Engine

In order to create a federated algorithm a complex process is required so the end user will not be able to create one. In order to enable the user to have some "editing" capabilities the workflow engine was added. It allows the users to combine existing federated algorithms as building blocks in order to create more complex, composite algorithms.

In order to achieve that we are following these steps:

1) We integrate the federated algorithms in the Workflow Engine, in the form of a tool that it can recognise.
2) Visualisation techniques are added in the engine, for the specific algorithm that was added.

The user then will be able to combine algorithms and visualisation techniques and create custom workflows.

[4] https://en.wikipedia.org/wiki/Levenshtein_distance

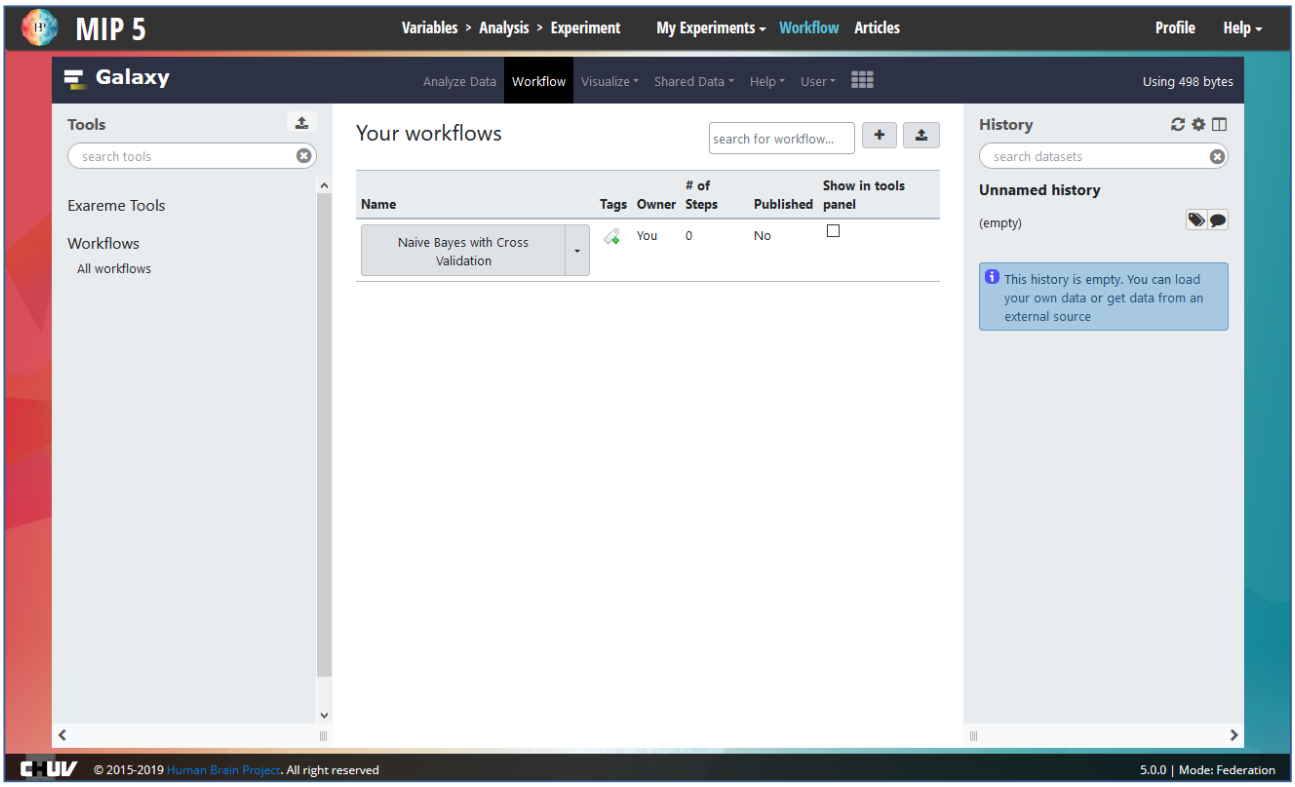

Figure 5: Workflows in Galaxy interface

Integration of the MIP Scientific Workflow Engine Component Galaxy with the MIP Federate Network was completed, as described in the Deliverable D8.5.1. This included:

Creation of scripts that convert Galaxy into a Docker image, as are the rest of the MIP components. (https://github.com/madgik/galaxy/tree/v1.2.2/Docker_Build_Scripts)

Improvements to the current Galaxy project workflow engine with the Distributed / Federated Query Execution Engine EXAREME https://github.com/madgik/galaxy/tree/v1.2.2/tools/exaremeTools)

In order to achieve the integration we had to modify some federated algorithms (https://github.com/madgik/exareme/tree/v21.2.0/Exareme-Docker/src/mip-algorithms). Specifically, we had to separate the algorithms into distinct workflow components, i.e. we separated the Naive Bayes algorithm into training component, testing component, and output components, each transforming the output of the engine into JSON, tabular data resource format (https://github.com/frictionlessdata/specs/blob/master/specs/tabular-data-resource.md) or Highcharts format.

On this release we developed the components needed to integrate the Galaxy Workflow Editor in MIP and make them look as a single service to the final user, hiding the separate authentication of Galaxy. We also improved the workflow execution process and integrated it completely with the MIP algorithm execution.

(https://github.com/madgik/galaxy/tree/v1.2.2).

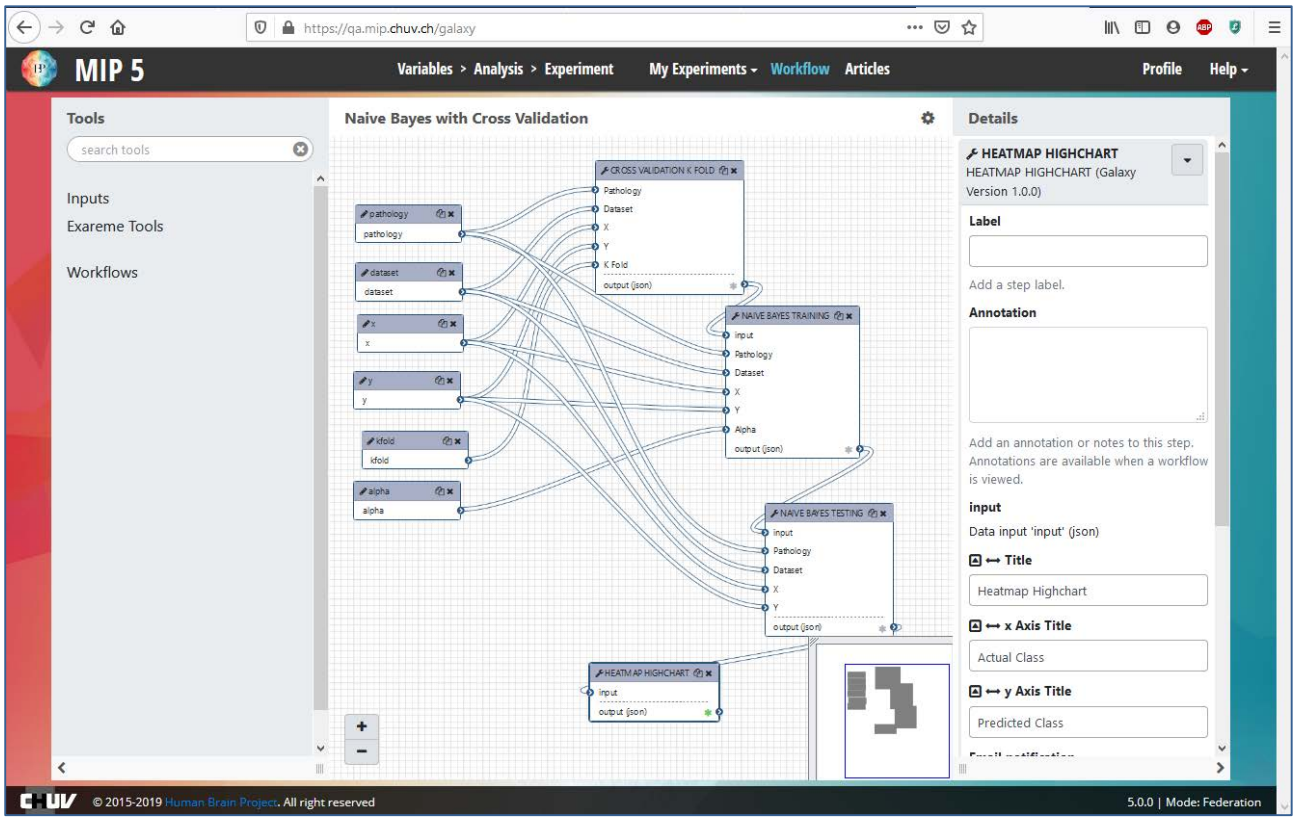

Figure 6: Galaxy graphical workflow engine interface

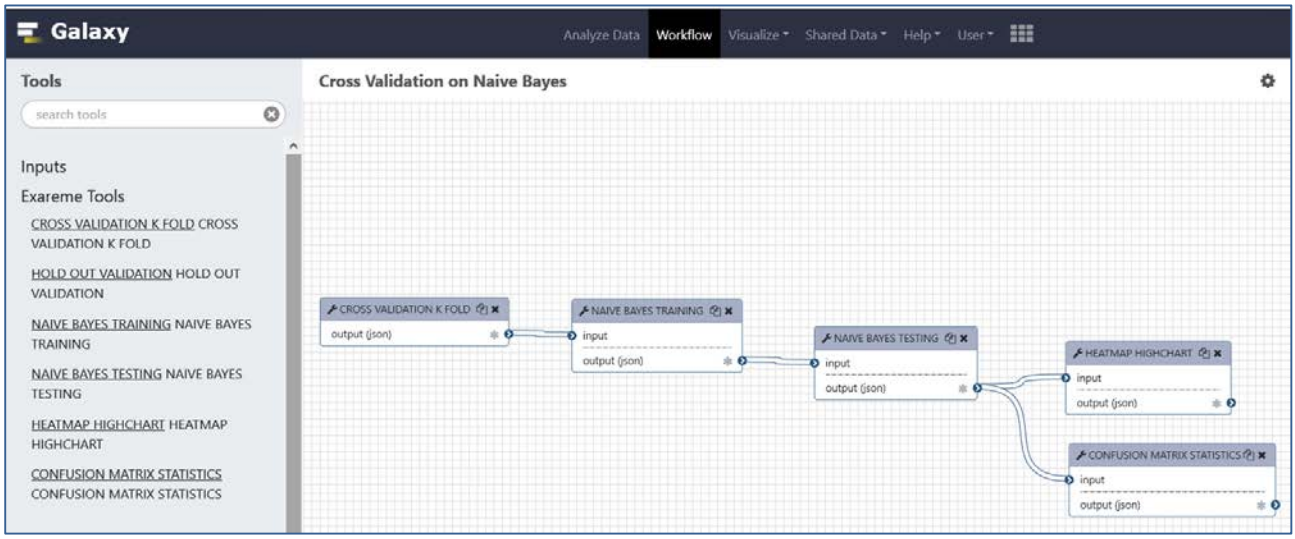

Figure 7: Galaxy graphical workflow engine interface

4.2 MIP Releases Component Roadmap

Table 1: Overview of new additions / developments roadmap

Component	MIP M12 4.0	MIP M20 5.0	MIP M24
Anonymisation component	Yes	Yes	Yes
Data Catalogue	Yes	Yes + Enhancement for multi pathologies	Yes
Loris Integrator	No	Partial integration	Yes
QC Tools	Profiling Tools	Profiling Tools + Data cleansing	Profiling Tools + Data cleansing

Scientific Workflow Engine	Integration with Federated Data Processing Engine	Integration with MIP Portal	Integration with portal enhancements
Analytical Capabilities	Naïve Bayes, K-means, ID3, Pearson Correlation, Logistic Regression, T-test, Paired t-test	ANOVA, Cross Validation, Descriptive Statistics, Histograms, Logistic Regression, Paired T-test, Simple T-test	Calibration Belt, Linear Mixed Models, Cart, Random Forest, Gradient Boosting, PCA, Kaplan -Meier, Stochastic Gradient Descent with L1+L2 Regularization
Knowledge graph Proof of Concept	No	No	Yes
Automatic Data Extraction POC	No	No	Yes

4.3 MIP Release 5.0 Components

Table 2: Overview of major updates of MIP's software components for release 5.0

Component Name	Type	Contact	Release
Anonymisation module	Software	Vasileios VASSALOS (P4)	Release Name: Data anonymisation Release date: 03/31/2019 URL: https://github.com/aueb-wim/anonymization-4-federation Anonymizing data so as to be imported into the hospital's federation node
Federated / distributed data processing engine	Software	Giorgos PAPANIKOS (P43)	Release Name v21.2.0 Release date: 30/10/19 URL: https://github.com/madgik/exareme/tree/v21.2.0 Changelog: https://github.com/madgik/exareme/releases/tag/v21.2.0Execution of requests received by the MIP web portal in the form of parameterized templates. Templates supported are: federation of a single algorithm, and execution of incremental learning algorithms running across hospitals.
API test scripts	Software	M. SPUHLER (P27)	This component was merged into the Frontend latest release (>5.0)
New version of portal frontend	Software	M. SPUHLER (P27)	Release name: v5; Release date :2019.11.12; URL : https://github.com/HBPMedical/portal-frontend Portal Frontend is a collection of web pages and JavaScript code powering the MIP Portal. It is packaged in a Docker container with a Nginx server serving the pages and other content. From the 5.0 version, the frontend was migrated fully to React 16. Integration tests were also fully merged in the 5.0 Robustisation, error handling, normalized data structure. Migration performed in anticipation of complex visualization and third-party integrations

New release of Portal Backend	Software	Manuel SPUHLER (P27)	Release name: v5; Release: 5.0.4 Release date: 2019.11.12; URL : https://github.com/HBPMedical/portal-frontend/archive/2.15.0.zip
Integration of t-SNE algo visualization on frontend	Software	M. SPUHLER (P27)	Release name: 2018.12.13; Release: 0.4.3; Release date: 2018.12.13; URL : https://github.com/LREN-CHUV/algorithm-repository/tree/master/python-tsne Packaged t-SNE. Algorithm rules in Woken. Frontend integration, input form, visualization output.;
Clinical Data Catalogue	Software	Vasileios VASSALOS (P4)	Release Name: Data Catalogue Release date: 02/22/2019 URL: https://github.com/HBPMedical/DataCatalogue A catalogue presenting the structure and the semantics of the HBP-official latest set of the Common Data Elements
Data cleansing component	Software	Vasileios VASSALOS (P4)	Release Name: QC tools Release date: 02/22/2019 URL: https://github.com/HBPMedical/DataQualityControlTool This software Component is used in the first steps of the Data Factory so as to ensure good quality of EHR and imaging data.
New release of Deployment scripts for MIP platform	Software	François TCHAMABÉ(P27)	Release name: v5 Release: 5.1 Release date :2019.11.12URL: https://github.com/HBPMedical/mip-deployment-infrastructure
New release of the hierarchizer	Software	Mirco NASUTI (P27)	Support new data formats Release name: 07.08.2018 Release: 1.3.8 Release date: 07.08.2018 URL: https://github.com/HBPMedical/hierarchizer
Scientific Workflow Engine	Software	Thanassis Karabatsis (P42)	Release name: v1.2.2 Release date: 18/10/19 URL: https://github.com/madgik/galaxy/tree/v1.2.2 Changelog: https://github.com/madgik/galaxy/releases It will allow the users to define and execute workflows of algorithms. According to the defined workflows it will submit jobs to the Rest APIs of Local and Federated execution engines.

4.4 MIP Work in Progress

This section describes developments that are almost complete but could not be part of release 5.0.

4.4.1 LORIS

LORIS [5] Is a web application for managing neuroimaging data with data acquisition and quality control features. We are currently working on integrating LORIS in the Data Factory pipeline so as to take advantage of its features and control the quality of the brain scans.

We have created a LORIS installation tailored for the MIP in the sense that we eliminated some features that are out of our context while focusing on the brain scans' management. To support our Use Case scenario, we developed some complimentary scripts for importing and managing the brain scan images.

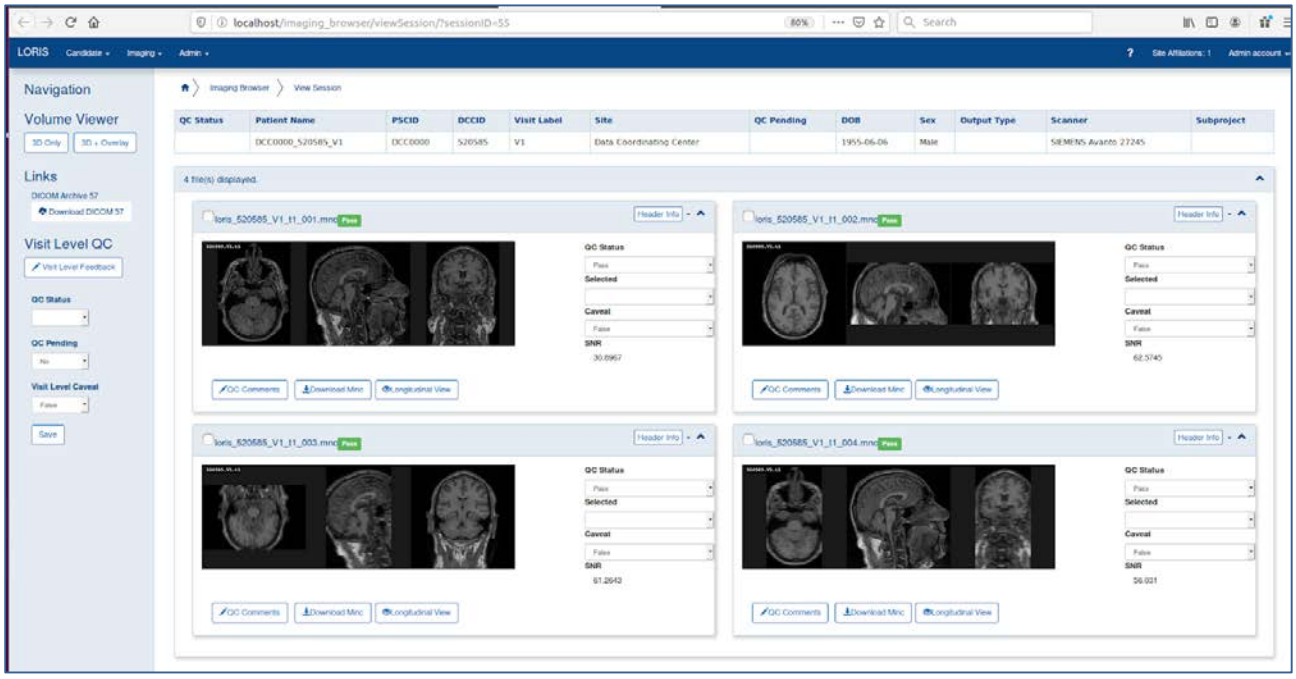

Figure 8: Browsing MRIs, quality commenting and annotating

On the deployed Local MIP of each hospital, the first step will be to import the MRI's, which are in DICOM format, to LORIS. Unless they do not comply with the minimum quality constraints (ex. poor image resolution, not enough image slices), they are imported to LORIS. Once data import is done, the MIP user is able to view the brain scans in 3D via the BrainBrowser [6]which LORIS uses. She is also able to comment and annotate (Pass/Fail) regarding the quality of the image. The images are all initiated with a 'Pass' label so as the clinician/data manager to select the ones to be excluded from the MIP. The DICOM images that finally have the 'Pass' label are transformed to NIFTI [7]format and are forwarded to the next step of brain image feature extraction. In addition to these 2 formats the images are also transformed into MINC files which are the ones loaded and visualised by the BrainBrowser viewer.

[5] Longitudinal Online Research and Imaging System https://github.com/aces/Loris/tree/master
[6] https://brainbrowser.cbrain.mcgill.ca/
[7] https://nifti.nimh.nih.gov/nifti-1

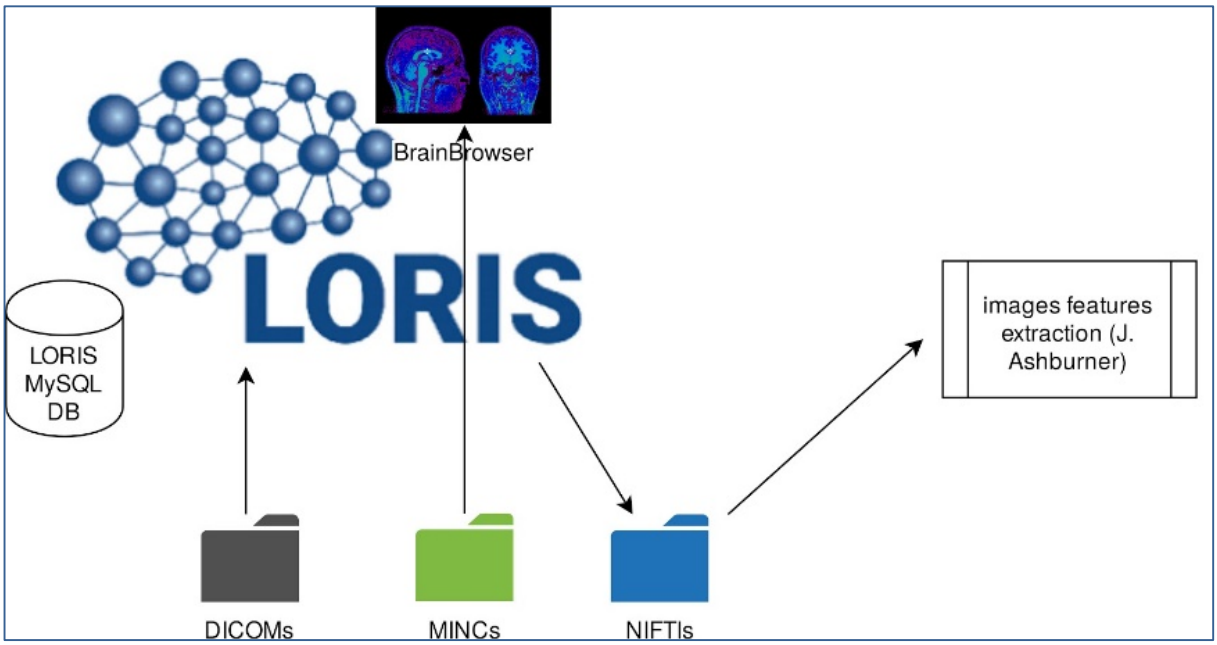

Figure 9: LORIS in the MIP

Currently, this is a work in progress since we encounter a few cases where brain images are not visualised properly by the BrainBrowser even though having been imported according to LORIS' requirements. We are working on specifying the conditions in which this is happening while being in contact with the LORIS team. In addition to that, to finalize the integration of LORIS in to the MIP we will create a setup in Docker containers for deploying purposes.

4.4.2 Analytical capabilities

Work in progress for next releases and final release (M24) includes Calibration Belt, an assessment of quality of care algorithm from Bergamo hospital, initially written in R. Work included analysis and breakdown of the algorithm into global and local components.

Appendix A: Federated privacy-complying algorithms

A. Notation

The complete dataset is composed of M local datasets, one for each hospital

$$\mathcal{X} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(L)}\}. \quad (1)$$

Each local dataset is represented as a matrix of size $n \times p$, where n is the number of points (patients) and p is the number of attributes. *E.g.* $x_{ij}^{(l)}$ is the value of the j^{th} attribute of the i^{th} patient in the l^{th} hospital. We will also use the notation $\mathbf{x}_i^{(l)}$ for the i^{th} patient's vector of attributes. The elements of the above matrices can either be continuous or discrete (categorical). When needed, we transform the categorical variables to dummy Boolean variables as a pre-processing step. Moreover, in Linear and Logistic Regression we add a column of 1's to account for the intercept term.

For *supervised* models, such as Linear Regression, Logistic Regression, Naive Bayes *etc.* we add a dependent variable (selected from the attributes by the user)

$$\mathcal{Y} = \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(L)}\}, \quad (2)$$

where each $\mathbf{y}^{(l)}$ is a vector of size $n \times 1$. The dependent variable can be continuous (*e.g.* Linear Regression) or categorical (*e.g.* Logistic Regression).

For classification tasks (Logistic Regression, etc.) we use the notation $C_m \in \{C_1, C_2, \ldots, C_M\}$ for the domain of the corresponding variable.

B. Privacy

Privacy in the MIP's Federated Processing Engine analytical capabilities is enforced in three stages.

- *Full anonymization of the local database*. MIP has access to a dual MIP-local database. The first database contains the pseudo-anonymized data (allowing regular updates of the dataset), while the second database, created from the former, will be completely anonymized with no associated lookup table and no possibility to link it back to the pseudo-anonymised database. Analyses performed through MIP-federated have access only to fully anonymized data from the second database.
- *Aggregate results.* Whenever some result, computed locally on a local database, is broadcasted to the central node, this result is always an aggregate (sum, sum of squares, count, *etc.*) computed on groups of size at least k. In practice we use $k = 5$.
- *No straightforward data reconstruction.* Finally, we develop case-by-case arguments showing that there is no straightforward way to reconstruct the original data from the aggregated quantities. We develop these arguments in the following paragraphs where we also describe in detail the algorithms' steps.

C. ANOVA

There are three different classical approaches for computing sums of squares (SS) and testing hypotheses in ANOVA for unbalanced data commonly called Type I, II, and III sums of squares.

Consider a model that includes two factors A and B; there are therefore two main effects, and an interaction, AB. The full model is represented by $SS(A, B, AB)$. Other models are represented similarly: $SS(A, B)$ indicates the model with no interaction, $SS(B, AB)$ indicates the model that does not account for main effects from factor A, and so on.

The influence of particular factors (including interactions) can be tested by examining the differences between models. For example, to determine the presence of an interaction effect, an F-test of the models $SS(A, B, AB)$ and the no-interaction model $SS(A, B)$ would be carried out.

It is handy to define incremental sums of squares to represent these differences. Let

$$SS(AB|A, B) = SS(A, B, AB) - SS(A, B)$$

$$SS(A|B, AB) = SS(A, B, AB) - SS(B, AB)$$

$$SS(B|A, AB) = SS(A, B, AB) - SS(A, AB)$$

$$SS(A|B) = SS(A, B) - SS(B)$$

$$SS(B|A) = SS(A, B) - SS(A)$$

The notation shows the incremental differences in sums of squares, for example $SS(AB|A, B)$ represents the sum of squares for interaction after the main effects, and $SS(A|B)$ is the sum of squares for the A main effect after the B main effect and ignoring interactions .

The different types of sums of squares then arise depending on the stage of model reduction at which they are carried out. In particular:

Type I, also called sequential sum of squares

The Type I analysis corresponds to adding each effect sequentially to the model and it depends on how the model terms are ordered. Different orders may give quite different results.

Let $SS(A, B, AB)$ be the full model. We test:

$SS(A)$ for factor A.

$SS(B|A)$ for factor B.

$SS(AB|B, A)$ for interaction AB.

Type II: hierarchical or partially sequential

Type II computes SS for all effects in the model that are at the same or lower level. For example, SS for the main effects take account of all other main effects, rather than simply accounting for those entered earlier in the model. Interaction effects take account of all main effects and all other interaction effects at the same level.

Let $SS(A, B, AB)$ be the full model. We test:

$SS(A|B)$ for factor A.

$SS(B|A)$ for factor B.

$SS(AB|B, A)$ for interaction

Type III: marginal or orthogonal

SS gives the sum of squares that would be obtained for each variable if it were entered last into the model. That is, the effect of each variable is evaluated after all other factors have been accounted for. Therefore, the result for each term is equivalent to what is obtained with Type I analysis when the term enters the model as the last one in the ordering.

Let $SS(A, B, AB)$ be the full model. We test:

$SS(A|B, AB)$ for factor A.

$SS(B|A, AB)$ for factor B.

Federated ANOVA

Based on the type of sum of squares selected by the user as well as the full model, a set of models is defined as described above. Federated linear regressions are executed for each of these models. Once the federated linear regressions are computed, we compute and output all the relevant statistics and p-values of standard ANOVA.

D. Independent Samples T-Test

Binary Logistic Regression training is done by Maximum Likelihood Estimation (MLE) using Newton's method. Applying Newton's method leads to the following algorithm, called *Iteratively Reweighted Least Squares* (IRLS). Here the dependent variable y has to be binary.

Concerning privacy, the same arguments as for Linear Regression apply. The quantities $\mathbf{A}$ and $\mathbf{b}$ do not allow reconstruction of the data in a straightforward way, as long as $n > p$. Moreover, additional nonlinear terms, make the task even more challenging than in the Linear Regression case.

T-TEST

procedure LOCAL1(x, y, `hypothesis`)
  for each variable y_m **do**
    Compute the following quantities
    $\text{sum}_{y_m|x_1}$
    $\text{sum}_{y_m|x_2}$
    $\text{sum}_{y_m^2|x_1}$
    $\text{sum}_{y_m^2|x_2}$
    $n_{y_m|x_1}$
    $n_{y_m|x_1}$
  end for
end procedure
procedure GLOBAL1(all sums and counts from local procedure)
  for each variable y_m **do**
    Sum local outputs and compute the following quantities
    $\text{mean}_{y_m|x_1}$
    $\text{mean}_{y_m|x_2}$
    $\text{SE}_{y_m|x_1}$
    $\text{SE}_{y_m|x_2}$
    $n^{\text{Total}}_{y_m|x_1}$
    $n^{\text{Total}}_{y_m|x_2}$
  end for
end procedure
procedure LOCAL2($\{\text{mean}_{(.)}\}$)
  for each variable y_m **do**
    Compute $\text{sse}_{y_m|x}, k = 1, 2$, where $\text{sse}_y = \text{sum}(y - \text{mean})^2$
  end for
end procedure
procedure GLOBAL2($\{\text{sse}_{(.)}\}$)
  for each variable y_m **do**
    Compute and output corresponding statistics and p-values
  end for
end procedure

LOGISTIC REGRESSION TRAIN

procedure GLOBAL1
  Initialize weights $\mathbf{w} \leftarrow \mathbf{0}$
end procedure
loop
  procedure LOCAL1(w) ▷ run for $l = 1, \ldots, L$
    $\eta_i \leftarrow \mathbf{w}^\top \mathbf{x}_i^{(l)}$
    $\mu_i \leftarrow \mathrm{sigm}(\eta_i)$
    $s_i \leftarrow \mu_i(1 - \mu_i)$
    $z_i \leftarrow \eta_i + \frac{y_i^{(l)} - \mu_i}{s_i}$
    $\mathbf{S} \leftarrow \mathrm{diag}(s_{1:N})$
    $\mathbf{A}^{(l)} \leftarrow X^{(l)\top} \mathbf{S} X^{(l)}$
    $\mathbf{b}^{(l)} \leftarrow X^{(l)\top} \mathbf{S}\mathbf{z}$
    return $\mathbf{A}^{(l)}$, $\mathbf{b}^{(l)}$
  end procedure
  procedure GLOBAL2($\{\mathbf{A}^{(l)}, \mathbf{b}^{(l)}\}$)
    $\mathbf{A} \leftarrow \sum_l \mathbf{A}^{(l)}$
    $\mathbf{b} \leftarrow \sum_l \mathbf{b}^{(l)}$
    $\mathbf{w} \leftarrow \mathbf{A}^{-1}\mathbf{b}$
    return w
  end procedure
end loop

E. One Sample T-Test

The Student's One-sample t-test is used to test the null hypothesis that the true mean is equal to a particular value (typically zero). A low p-value suggests that the null hypothesis is not true, and therefore the true mean must be different from the test value.

Input: y = variables of interest, testvalue, hypothesis = ('different', 'lessthan', 'greaterthan')

Local Step 1: For each variable y compute: sum(y), sum(y^2), N(y)

Global Step 2: For each variable y compute: mean(y), std(y), Ntotal(y). Then for each variable y compute: statistics, df, Hypothesis, Cohens_d. See lines 55-114 of the following link: https://github.com/madgik/exareme/blob/master/Exareme-Docker/src/exareme/exareme-tools/madis/src/functionslocal/vtable/ttest_onesample.py

F. Paired Samples T-Test

In k-means the learning is *unsupervised* so we only need the matrix X at each local database. Here we consider only continuous variables and we use the Euclidean distance as our metric.

The Student's paired samples t-test (sometimes called a dependent-samples t-test) is used to test the null hypothesis that the difference between pairs of measurements is equal to zero. A low p-value suggests that the null hypothesis is not true, and that the difference between the measurement pairs is not zero.

Input: y = A vector of strings (i.e. 'y1a-y1b,y2a-y2b' naming the pairs of interest in data. Here y1a,y1b is first pair, and y2a,y2b is the second pair." i.e. ""'righthippocampus-lefthippocampus"", hypothesis = ('different', 'lessthan', 'greaterthan')

Local Step 1: For each variable y compute: sum(y1a-y1b), sum((y1a-y1b)^2), N(y1a-y1b)

Global Step 2: For each variable y compute: mean(y1a-y1b), std(y1a-y1b), Ntotal(y1a-y1b). Then for each variable y compute: statistics, df, Hypothesis, Cohens_d. See lines 55-114 of the following link: Here, **testvalue = 0**

https://github.com/madgik/exareme/blob/master/Exareme-Docker/src/exareme/exareme-tools/madis/src/functionslocal/vtable/ttest_onesample.py

Appendix B: MIP Architecture

Compared to previous versions of this deliverable, the architecture described here mirrors a fundamental enhancement that has been implemented towards product robustisation. The "Woken" based analysis engine that was previously utilized for local analysis has been substituted by an augmented version of the "EXAREME" based analysis engine. The impact of this change has been impressive both in terms of ease of deployment as well as system stability.

The Medical Informatics Platform is a complex information system comprising numerous software components designed and integrated by different SP8 partners. In this chapter we present the logical architecture of the MIP in SGA2, depicting some of its key characteristics and major building blocks. This description does not aim to be a detailed listing of all the software components produced to compose the MIP. It rather aims to assist in understanding the overall structure and interdependencies between the major services that comprise the MIP.

The following diagram Figure 1 - Overall Architecture, sketches at a very high level the overall architecture of the MIP.

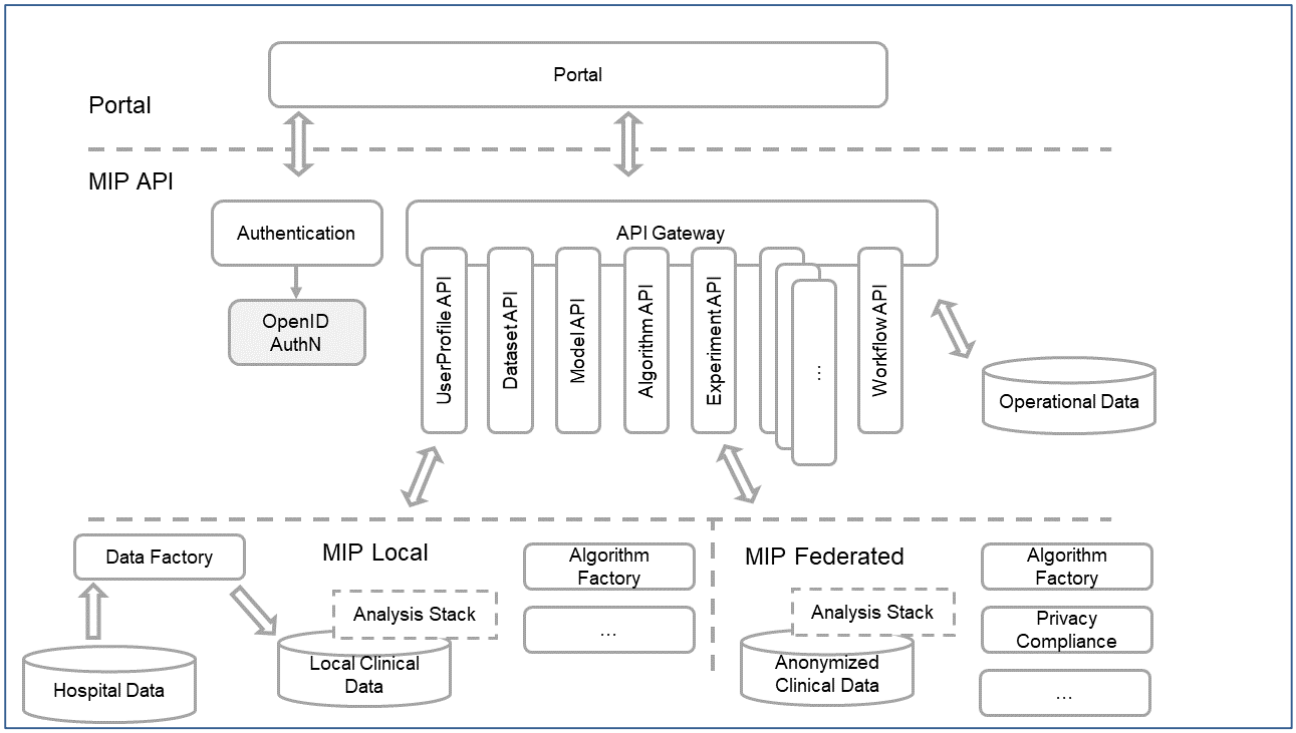

Figure 10: Overall Architecture

The MIP is architected following an N-Tier paradigm. Multiple tiers are identified within the platform to allow for a clear separation of domains and to provide reusability and separation of concerns between the business and technology layers of the overall platform. Additionally, the various service offerings of the platform are provided following the microservice paradigm, where applies and constitutes a value adding proposition.

- The **portal** acts as the main entry point to the business offerings of the MIP for researchers and clinicians
- The **API layer** offers a protected layer of functionality exposed in a uniform and interoperable manner
- The **API layer** acts as a gateway to the MIP offerings, providing horizontal reuse, vertical specialization and separation of concerns and technological restrictions through the employment of a microservices architecture
- For the **authentication** mechanisms proven standards are re-used to allow for a wide range of interoperability between the authenticated clients and the platform services

- A set of **operational data** assist in the streamlined communication and interaction between the clients and the MIP services
- The **Data Factory** set of services facilitate the ingestion of hospital data within the MIP
- The **deployment** stack of the MIP can be split to facilitate disjoint but complementary deployments
  - **MIP Local** offers enhanced services and analytical capabilities within the boundaries of each hospital
  - **MIP Federated** offers federated analysis over anonymized data, across multiple hospitals

In the following diagram Figure 2 - Data Factory, the data flow and logical architecture of the Data Factory pipeline is highlighted.

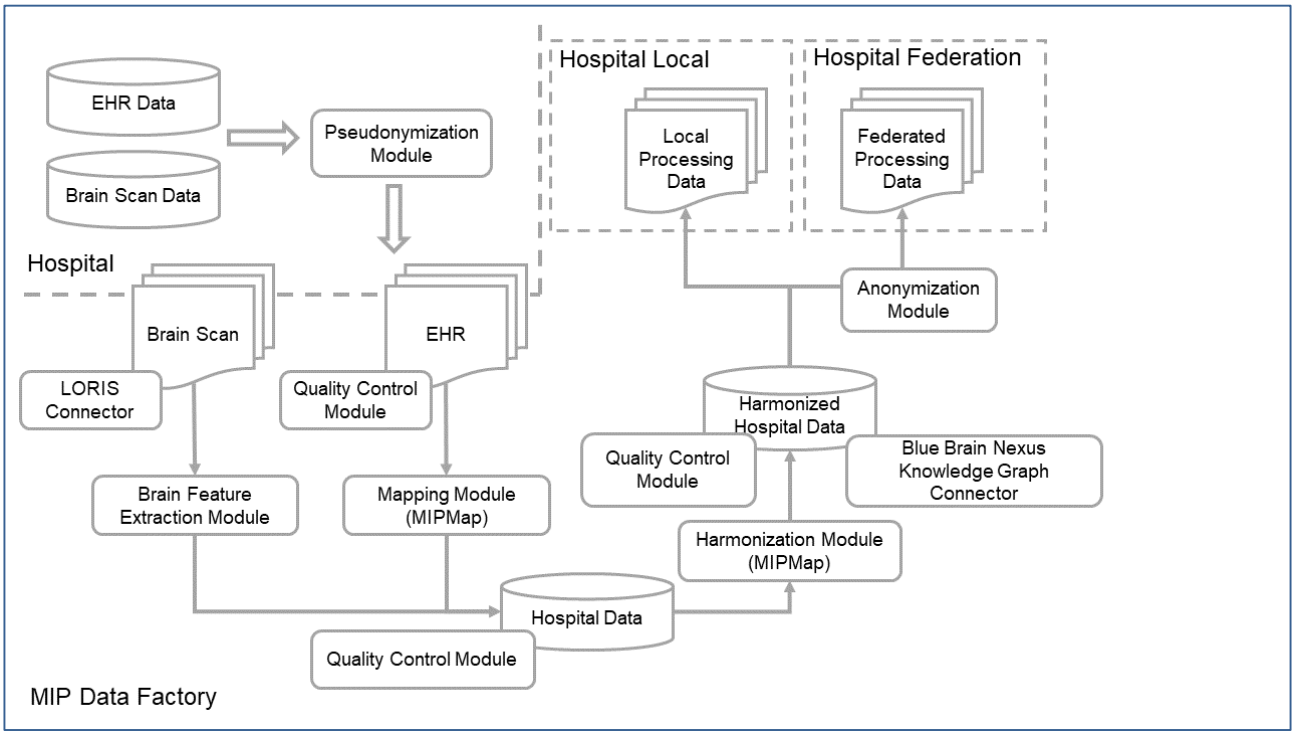

Figure 11: Data Factory

- The initial hospital data, including both electronic health records as well as brain scan data, go through a process of pseudonymisation and are injected into the MIP pipeline.
- A set of Quality Control tools are utilized throughout the process to ensure the validity and compliance of the data
- A number of extension points to external systems (Blue Brain Nexus Knowledge Graph) and value adding tooling within the MIP (LORIS) are identified and can be hooked in the pipeline through appropriate connector modules
- Underpinning tools are utilized for the extraction of brain features as well as the mapping of the ingested data to the MIP data model
- The harmonization process builds up the canonical model that subsequent MIP operate on
- The anonymization module makes sure that the data it processes cannot be linked back to its input and makes them available for federated processing
- The output of the process is propagated to the subsequent MIP platform components that exposes it for analysis, depending on the MIP mode of operation
  - Local analysis – Access and analysis over hospital local data
  - Federated analysis – Controlled access and federated analysis over multi-hospital data

To facilitate the dual mode of operation, the MIP offers a different deployment stack so that hospitals can opt-in to the federated processing capabilities offered. Figure 3 depicts the architecture and processing flow between the different modes of analysis offered.

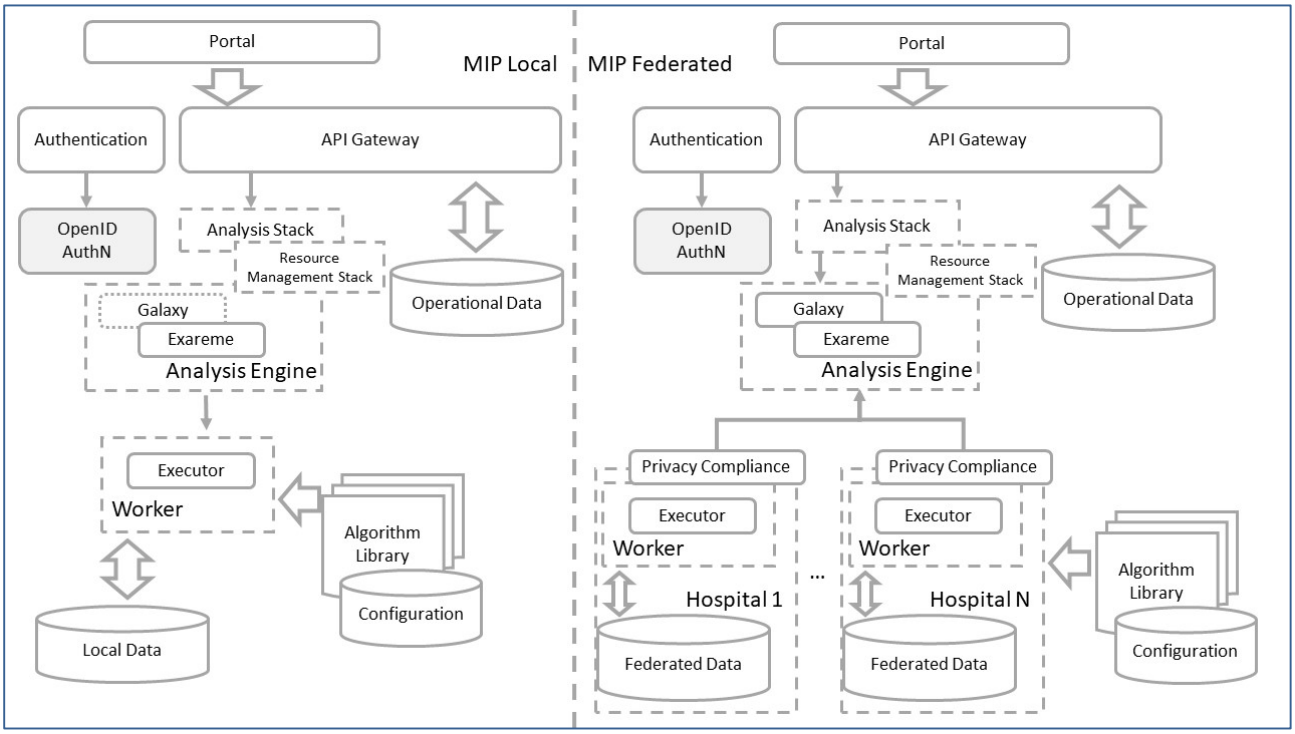

Figure 12: Local vs Federated Analysis

Depending on the mode of operation, different hospitals may operate on a single "Local analysis" mode, or they can also participate in federation, providing their data for federated analysis.

- The **analytical capabilities** are offered to authenticated and authorized users of the MIP
- The **API Gateway** offers a uniform and homogenized handling interface to the analytical capabilities
- The **Portal** presents and assists the user to perform the needed experiments
- The **Analysis Stack** along with the Resource Management Stack hide the complexity involved on performing the requested experiment
- The **Algorithm Library** offers the toolbox from which the users can select the required processing
- Depending on the mode of analysis, local or federated, the respective analysis engine will handle the appropriate communication and push the analysis within the needed boundaries
- In the case of federated analysis, the required privacy compliance will be applied at the boundaries of each contributing hospital

In the following diagram, Figure 4 - Algorithm Factory, some more details are given on the architecture and interactions between the components that underpin the analytical capabilities of the MIP.

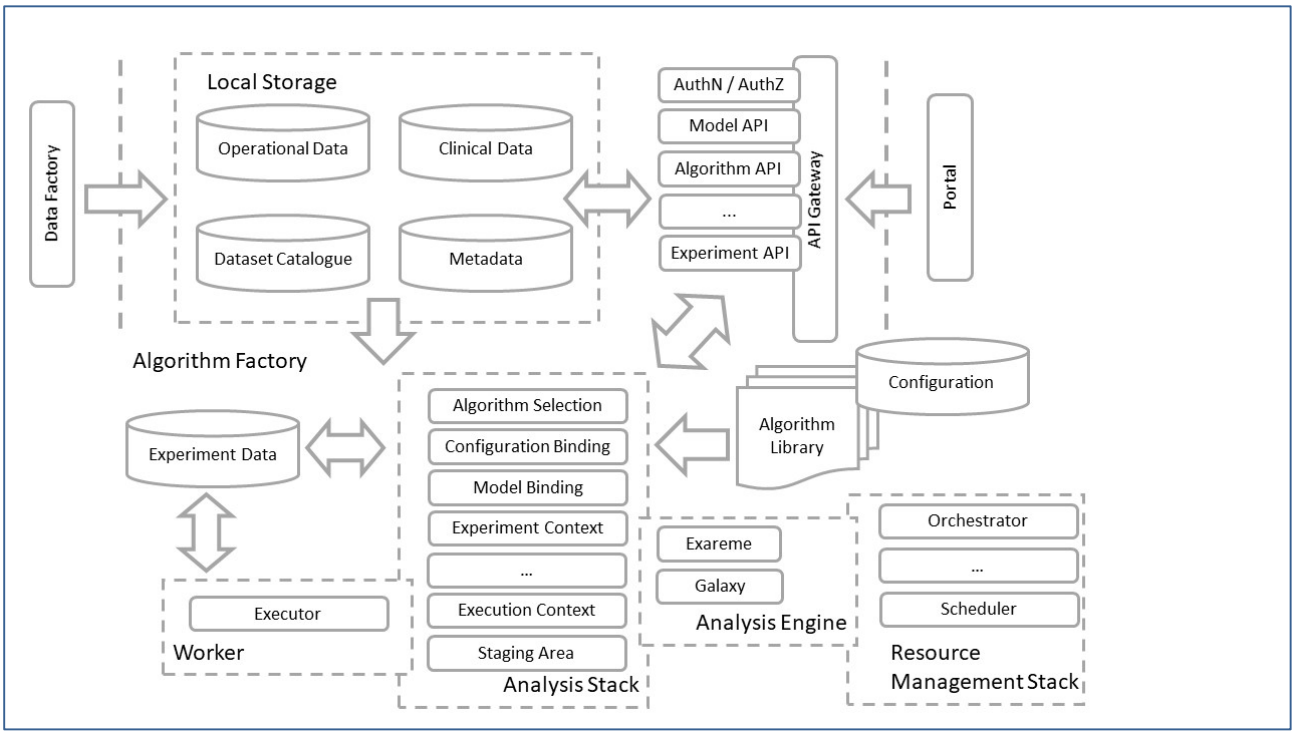

Figure 13: Algorithm Factory

Within each hospital, depending on the mode of analysis, local or federated, the set of available clinical data, pseudonymised or anonymised, are available for the analytical module to operate on:

- The metadata that describe the available dataset and exposed canonical model is used to build and evaluate the model requested
- The **Dataset catalogue** is used to drive the evaluation of the experiment within the appropriate boundaries
- The **API Gateway** interface exposes a uniform layer of interacting with the analytical flows
- The **Analysis Stack** contains all the required components that will assist in formulating the experiment, compose the runtime environment for its evaluation and stage its execution
- Depending on the semantics and requirements of the experiment evaluation, the appropriate **analysis engine** is utilised, and the needed resources are scheduled and employed
- The experiment configuration, runtime data, transient sets and results are stored within the context of the experiment

Exposing the functionality of the MIP but also enhancing it through the appropriate user experience and integrations with external services and tooling, the MIP Portal acts as the entry point to the MIP offerings. The following diagram, Figure 5 - Portal, depicts the main functional areas that the MIP Portal offers and its interactions within the MIP architecture layers.

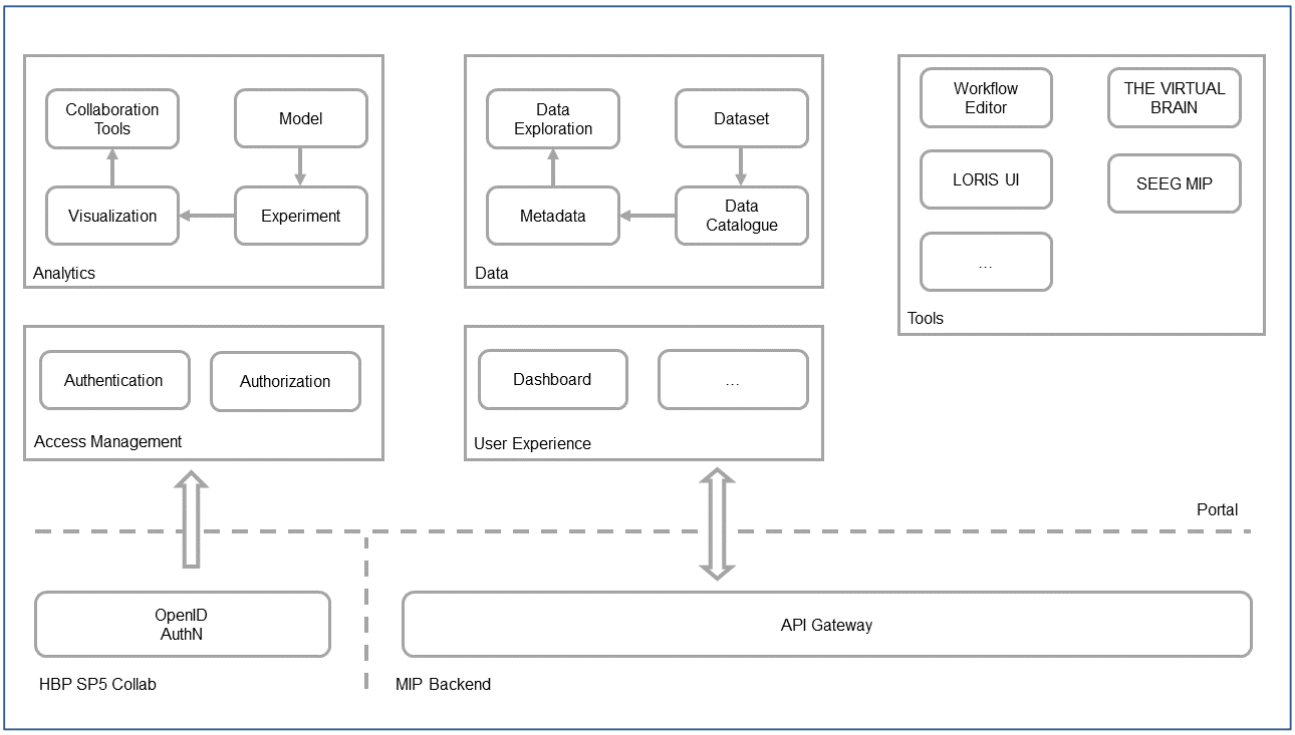

Figure 14: Portal

- The **Analytics** area is responsible to assist the user compose the model of the analysis, define the experiment to be evaluated, visualize the outcome of the analysis and define the means of collaboration through which this analysis can be further used by researchers and clinicians
- Through the **Data Exploration** area, the user can browse the available Datasets, define the model and schema of the data exposed by the respective datasets through the Data Catalogue, visualize the metadata available for the canonical model
- he **authentication** of the MIP user is performed through appropriate workflows assisted by the portal and the view presented to the user is tailored to the authorization the user is granted
- Several **additional tools** can be made available to the user depending on his roles and functions within the MIP and the available extensions offered through the Portal, such as Workflow Editor, a LORIS User Interface, integrations with The Virtual Brain and SEEG MIP.