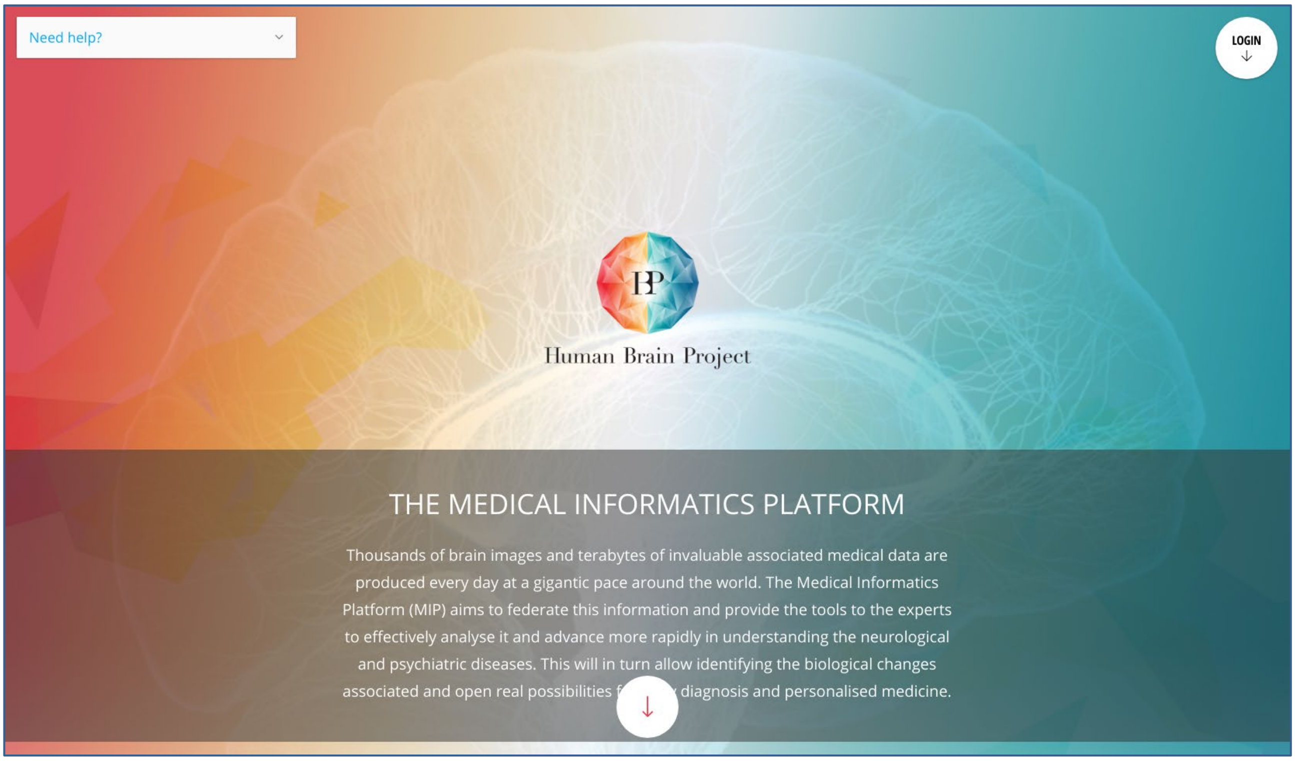


THE MEDICAL INFORMATICS PLATFORM

Thousands of brain images and terabytes of invaluable associated medical data are produced every day at a gigantic pace around the world. The Medical Informatics Platform (MIP) aims to federate this information and provide the tools to the experts to effectively analyse it and advance more rapidly in understanding the neurological and psychiatric diseases. This will in turn allow identifying the biological changes associated and open real possibilities f[illegible] diagnosis and personalised medicine.

Figure 1: Medical Informatics Platform M12 Release

Project Number:	785907	Project Title:	Human Brain Project SGA2

Document Title:	Medical Informatics Platform Releases – Software & Report
Document Filename:	D8.5.2 (D52.2 D8) SGA2 M12 ACCEPTED 190722
Deliverable Number:	SGA2 D8.5.2 (D52.2, D8)
Deliverable Type:	Report
Work Package(s):	WP8.5
Dissemination Level:	PU (= Public)
Planned Delivery Date:	SGA2 M12 / 30 Mar 2018
Actual Delivery Date:	SGA1 M14 / 22 May 2019; ACCEPTED 22 Jul 2019
Authors:	Evdokia MAILLI, UoA (P43), Giorgos PAPANIKOS, UoA (P43)
Compiling Editors:	Evdokia MAILLI (P43)
Contributors:	Ludovic CLAUDE, CHUV (P27), Thanasis KARABATSIS, UoA (P43) Kostis KAROZOS, UoA (P4), Jacek MANTHEY, CHUV (P27), Jason SAKELLARIOU, UoA (P43) Eleni ZACHARIA, UoA (P43
SciTechCoord Review:	Revised version not reviewed
Editorial Review:	Revised version not reviewed
Description in GA	SOFTWARE & REPORT
Abstract:	This document describes the Medical Informatics Platform (MIP) for the M12 release in SGA2. The document focuses on major new developments for the MIP according to the SGA2 Grant Agreement and the Deliverable D8.5.1 “Software Requirements and Specifications”, 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

History of changes

date	action
22 Mar 2019	Draft sent to PCO
25 Mar 2019	STC review: major changes requested, including the following: • A clear discussion of what the substantial new features are: What is their function/role, are they a new version of a previously integrated component or a new component altogether, what is this component's level of maturity, and how much effort did its development and/or integration entail. In short, to structure in a more systematic way, and expand upon (to the point of becoming exhaustive), the discussion in section 4.1. • A platform-wide overview, providing a short discussion of its architecture (a functional block diagram would be helpful) and feature set (previous, current, future). • A synthetic discussion of the implications and impact of the release on the platform's (feature set) roadmap.
21 May 2019	Revised version sent to PCO main changes : • Added input from Jacek MANTHEY regarding Quality Control tools • Added input from Ludovic CLAUDE regarding components • Added architecture diagram and explanation • Expanded description of new additions/developments • Added high-level release roadmap
22 May 2019	Submission to EC, no PCO review (STC and editorial) of the revised version

Table of Contents

Table of Figures

Table of Tables

Summary

This deliverable describes the Medical Informatics Platform (MIP) for the M12 release in SGA2. First, we indicate where deployment and documentation for the MIP is available. Then, we describe the major new developments for the M12 release, according to the SGA2 grant agreement and the deliverable D8.5.1 Software Requirements and specifications, and present releases for the major software components of the MIP. Finally, as privacy and privacy-compliant algorithms are the top priority requirements for the platform, we describe how privacy is enforced on the federated analytical capabilities of the MIP.

1. Introduction

During this first phase of SGA2, the focus of the development effort was directed into stabilizing the SGA1 MIP product. Several components were put under thorough testing and needed refactoring to produce a more robust bases for the subsequent development of new features as well as to stabilize and secure the hospital deployment process. Parallel to these actions, a number of new features and value adding components were developed to continue building and fulfilling the SGA2 requirements and objectives. This document focuses mostly on new developments.

2. Deployment

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

https://github.com/HBPMedical/mip-microservices-infrastructure. This is an open access repository.

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 features / components that are entirely new for the MIP, compared to the end of SGA1 MIP [1]release. According to the Software Requirements and Specifications deliverable D8.5.1, top priority functional requirements included full anonymisation of clinical data for federation usage, new statistical models and algorithms, scientific workflow engine, and organising data with the metadata catalogue. During the first year of SGA2 we focused on these top priority requirements, while in parallel we robustized the MIP where needed.

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

[1] MIP – Medical Informatics Platform

can be found 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*

The implementation of privacy compliant federated machine learning algorithms presents important challenges due to the constraints that need to be enforced. In our implementation we handle privacy at two levels. First, the full anonymisation of the local datasets, a more detailed explanation of which can be found in paragraph *4.1.3* and D.12.4.8. Second, the design of the algorithms themselves. We have redesigned a number of highly used statistical analysis and machine learning algorithms in order to render them privacy-compliant. Redesign includes that they are split in local and federation steps. Further, each algorithm must be integrated with the Federated Data Processing Engine EXAREME[2], and for that purpose we may have to enhance the engine with new user defined functions. Finally, a set of visualizations needed to support the algorithm's output to the MIP web portal is implemented, and the algorithm is integrated to the web portal.

More precisely, we have implemented, integrated into the federated data processing engine, and tested the following algorithms, according to the priorities set by Deliverable D8.5.1 (Software Requirements and Specifications):

1. Naïve Bayes Classifier (training and prediction procedures)
2. K-means. Previously, we had implemented an approximate version of k-means. We replaced it with a new version which is equivalent to standard non-federated k-means.
3. ID3 decision tree algorithm.
4. Pearson correlation coefficient.

Although KNN algorithm was part of the top priority requirements, after following the approach described in Appendix A, we concluded that this algorithm cannot be implemented in a privacy aware manner, hence it is removed from the list of available algorithms.

Work in progress includes logistic regression (train and predict procedures) and Principal Component Analysis (PCA).

We have also developed a process that will enable the user to execute k-fold cross validation and hold out cross validation. This process is based on the experimental integration with the workflow engine of Galaxy[3].

The above can be found in https://github.com/madgik/mip-algorithms.

4.1.3 Anonymisation component / tool

MIP has access to a dual MIP-local database. The first database contains the pseudonymised data (allowing regular updates of the dataset), while the second database, created from the former, will be completely anonymised with no associated lookup table and no possibility to link it back to the pseudonymised database. Analyses performed through MIP-federated have access only to the fully anonymised data in the second database. More details can be found in the deliverable D12.4.8 Anonymisation Process in Local/Federated MIP.

To materialize this approach in a component, we have integrated into the data ingestion scripts a final step for anonymizing hospital data to be exported to each hospital's federation node. Our method deletes the id's of a relational i2b2[4]-schema database while preserving the entities'

[2] http://madgik.github.io/exareme
[3] https://galaxyproject.org/
[4] https://www.i2b2.org/software/projects/datarepo/CRC_Design_Doc_13.pdf

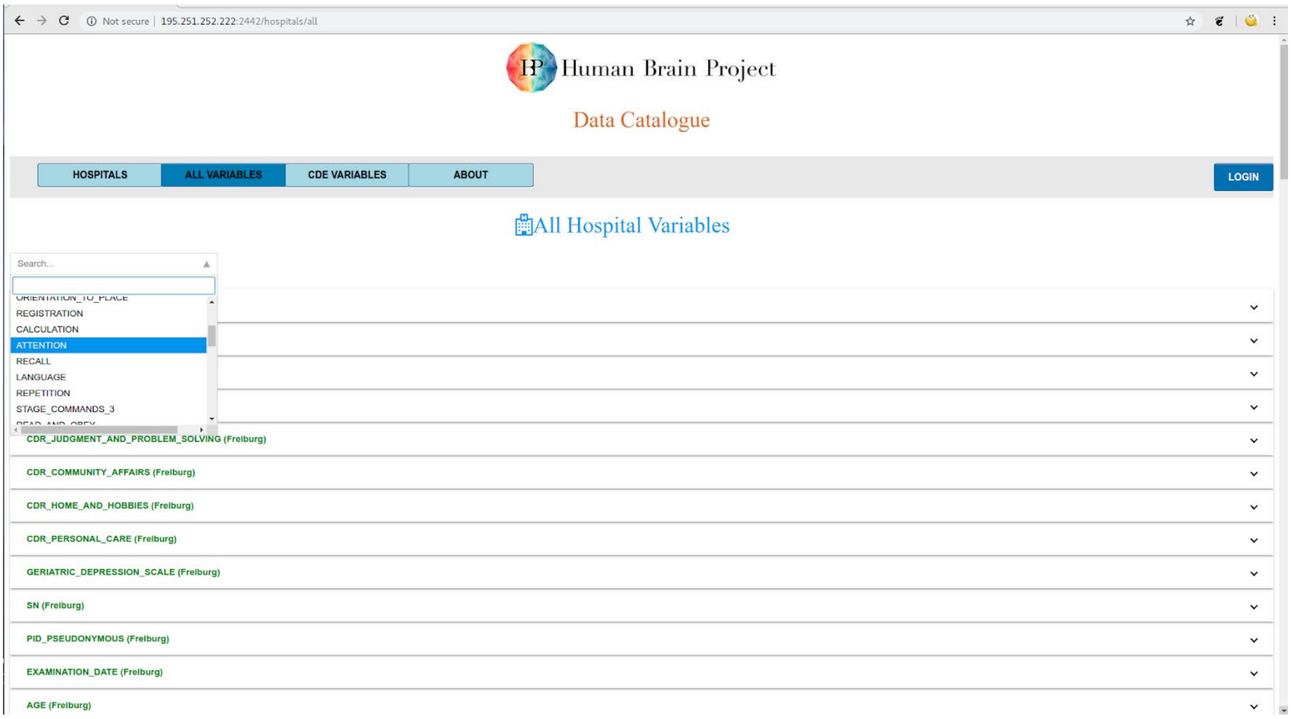
resolution and relationships among them by replacing already pseudonymized id's with the result of a hash function on them along with a random number. Having as input an i2b2 database, the output is a replicated database with randomly generated Patient and Visit id's which are not possible to be linked to the input database's pseudo-id's. This applies in both ways, meaning the output id's cannot be re-generated in the same way therefore cannot be linked to those of an already federation-anonymized database.

We have set up our scripts to provide an option between MD5 and SHA1 hashing.

Our component can be found in https://github.com/aueb-wim/anonymization-4-federation

4.1.4 *Data Catalogue*

Data Catalogue is an informative portal providing descriptive information about the data that reside in HBP hospitals that also offers metadata management capabilities to MIP users which have the proper rights. The need for such a centralized point of truth has been realized from SGA1 since metadata information was scattered around several hospitals and documents.



CDR_JUDGMENT_AND_PROBLEM_SOLVING (Freiburg)
CDR_COMMUNITY_AFFAIRS (Freiburg)
CDR_HOME_AND_HOBBIES (Freiburg)
CDR_PERSONAL_CARE (Freiburg)
GERIATRIC_DEPRESSION_SCALE (Freiburg)
SN (Freiburg)
PID_PSEUDONYMOUS (Freiburg)
EXAMINATION_DATE (Freiburg)
AGE (Freiburg)

Figure 2: Searching all hospitals' variables

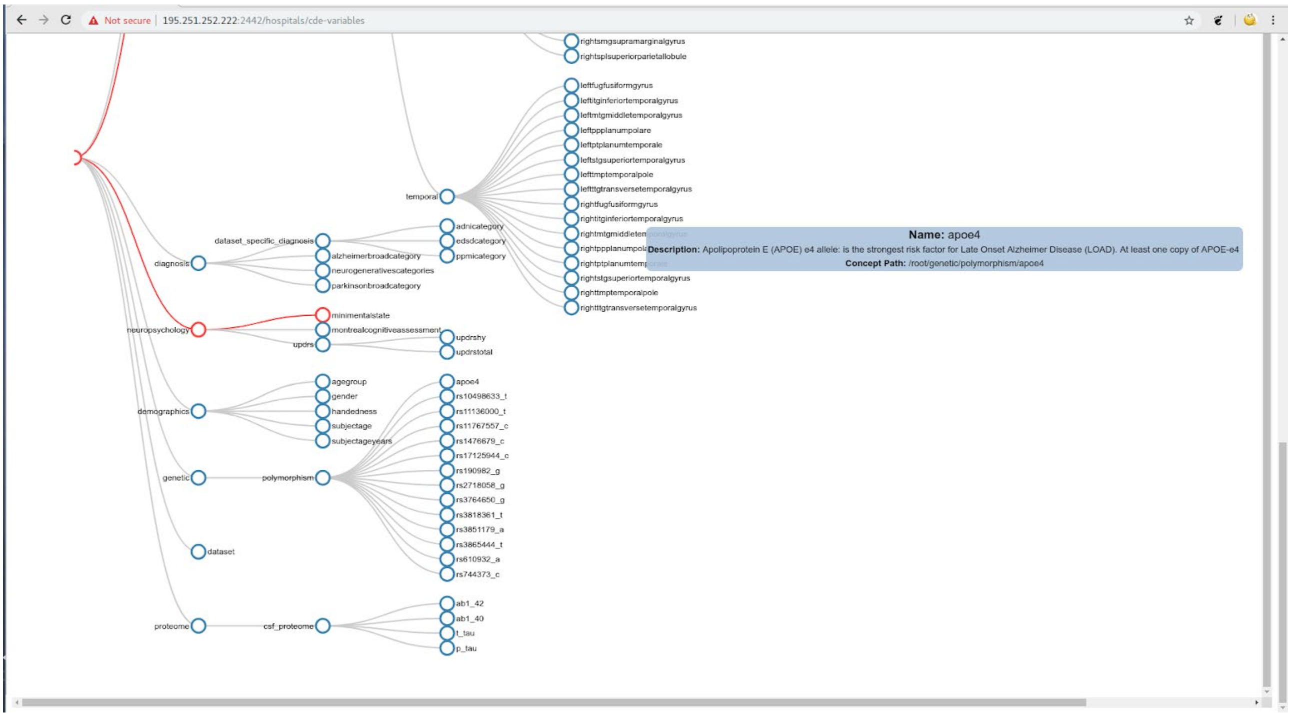
It presents metadata like Description, Concept Path, Measure Unit etc. for all hospital variables as well as CDE's[5]. It also depicts the way, if any, the hospital variables correspond to the CDE's. This is a new component that did not exist in SGA1. Its main purpose is to inform researchers and clinicians about the variables' metadata. Before executing an experiment on the MIP they will be able to search for the variables of their interest in all hospitals or in each one separately. Searching variables in all hospitals is done in the "ALL VARIABLES" tab while in "HOSPITALS" tab we restrict our search to the hospital of our choice. Each hospital can have several versions of its metadata schema for each one of which Data Catalogue presents not only the actual metadata but also an imported statistical report generated by our Quality Control tools.

Data Catalogue has the following information for every clinical variable:

[5] CDE – Common Data Element (https://github.com/HBPMedical/mip-cde-meta-db-setup/blob/master/variables.json)

1. csvFile: The name of the dataset file that contains the variable
2. name: The name of the variable
3. code: The variable’s code
4. type: The variable’s type
5. values: The variable’s values. It may have an enumeration or a range of values
6. unit: The variable’s measurement unit
7. canBeNull: Whether the variable is allowed to be null or not
8. description: The variable’s description
9. comments: Comments about the variable’s semantics
10. conceptPath: The variable’s concept path
11. methodology: The methodology the variable has come from
12. mapFunction: The function that transforms the variable’s value into the value of its corresponding CDE

In addition to presenting the actual values of the metadata columns, the taxonomy of the variables is recognized and depicted in a searchable graph. This visualisation is done for every hospitals’ schema version as well as every CDEs version.

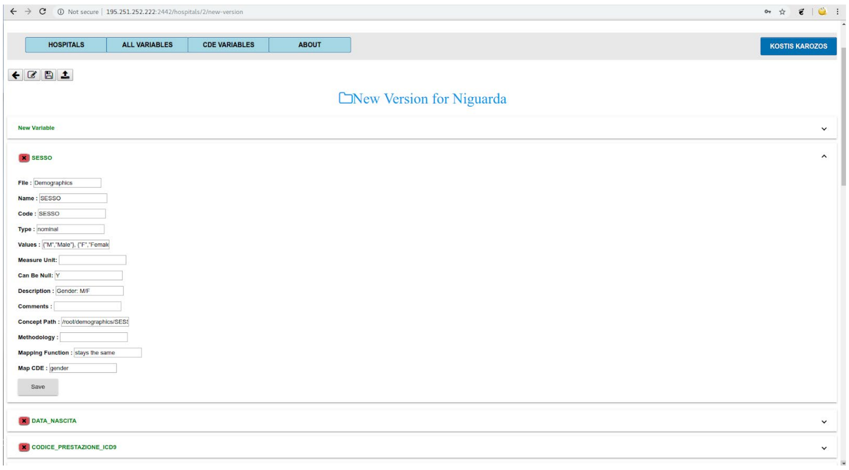


Name: apoe4
Description: Apolipoprotein E (APOE) e4 allele: is the strongest risk factor for Late Onset Alzheimer Disease (LOAD). At least one copy of APOE-e4
Concept Path: /root/genetic/polymorphism/apoe4

Figure 3 CDEs’ taxonomy

Another feature is the ability of the user to download every metadata version in a JSON file. This JSON file is used when importing a dataset into the MIP.

The aforementioned functionalities are available to anyone without having to login (guest user). In order to login the user must provide her HBP credentials. While logged in, she is authorized to manage metadata as well, in terms of creating a new metadata version either via the GUI’s text fields or by uploading and importing an xlsx file having information about the 13 metadata variables.



New Version for Niguarda

New Variable

SESSO

File : Demographics

Name : SESSO

Code : SESSO

Type : nominal

Values : {"M","Male"}, {"F","Femal

Measure Unit:

Can Be Null: Y

Description : Gender: M/F

Comments :

Concept Path : /root/demographics/SESS

Methodology :

Mapping Function : stays the same

Map CDE : gender

DATA_NASCITA

CODICE_PRESTAZIONE_ICD9

Figure 4: Creating a new metadata version

Data Catalogue's current version is on https://github.com/HBPMedical/DataCatalogue and is served temporarily in http://195.251.252.222:2442 where it is accessible by MIP users and non-users.

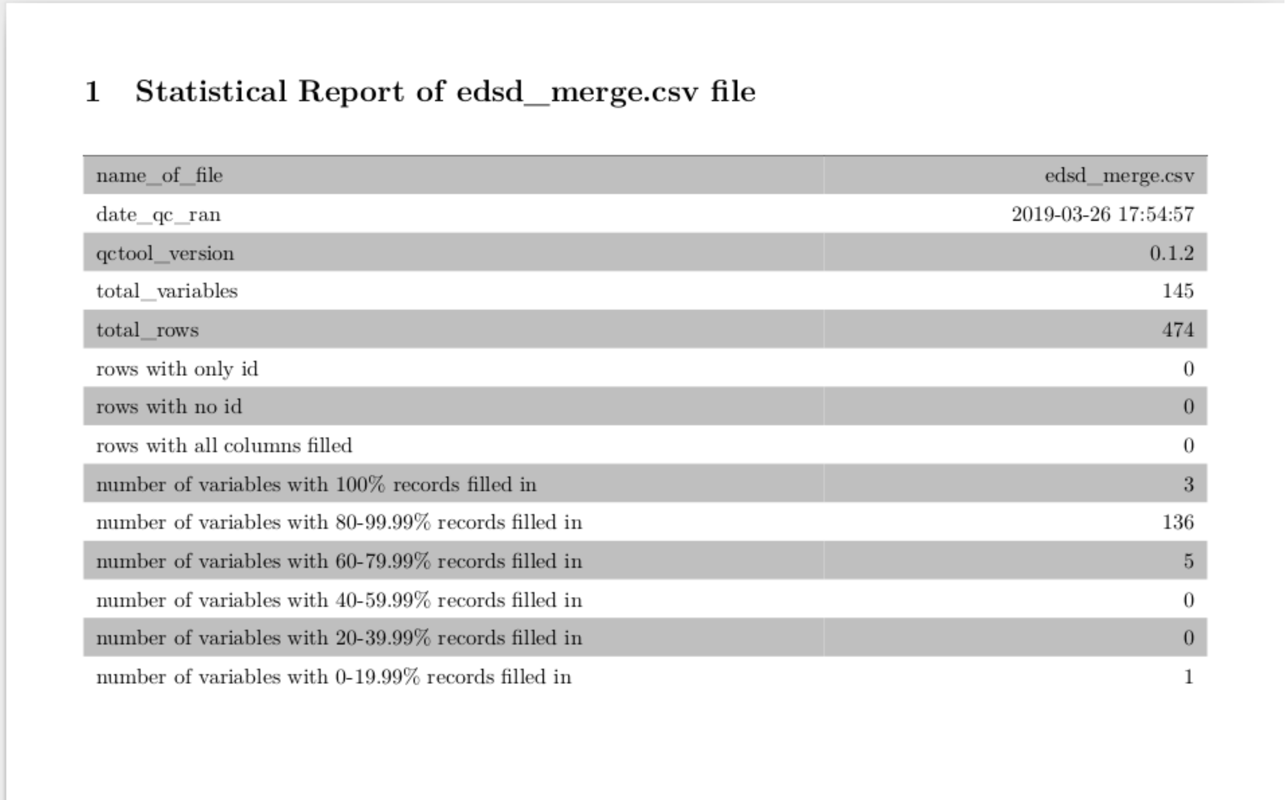
4.1.5 *Quality Control tools*

For checking the quality of the hospital data, we have been working on Quality Control (QC) tools for outliers and error detection. Our QC tools are divided in two categories:

1. The **QC data profiling tool** that generates statistical reports for the input datasets. The first module of the QC data profiling tool analyses tabular data (CSV[6] files) and produces reports with statistical information on the data, as well as the identification of missing data. The reports provide the ranges and distribution of numerical variables, as well as the values and frequency of categorical variables, thus allowing comparison with expected values and identification of unusual data, therefore contributing to increasing the quality of the incoming data. The reports are available in tabular (CSV file) form for use in automated analysis, as well as in easily interpretable and explanatory PDF format.

   We have run a first version of our profiling tool in some hospitals and generated reports. These reports consist of two parts. In the first one there are some statistics generated for all variables together.

[6] CSV: Comma Separated Value file

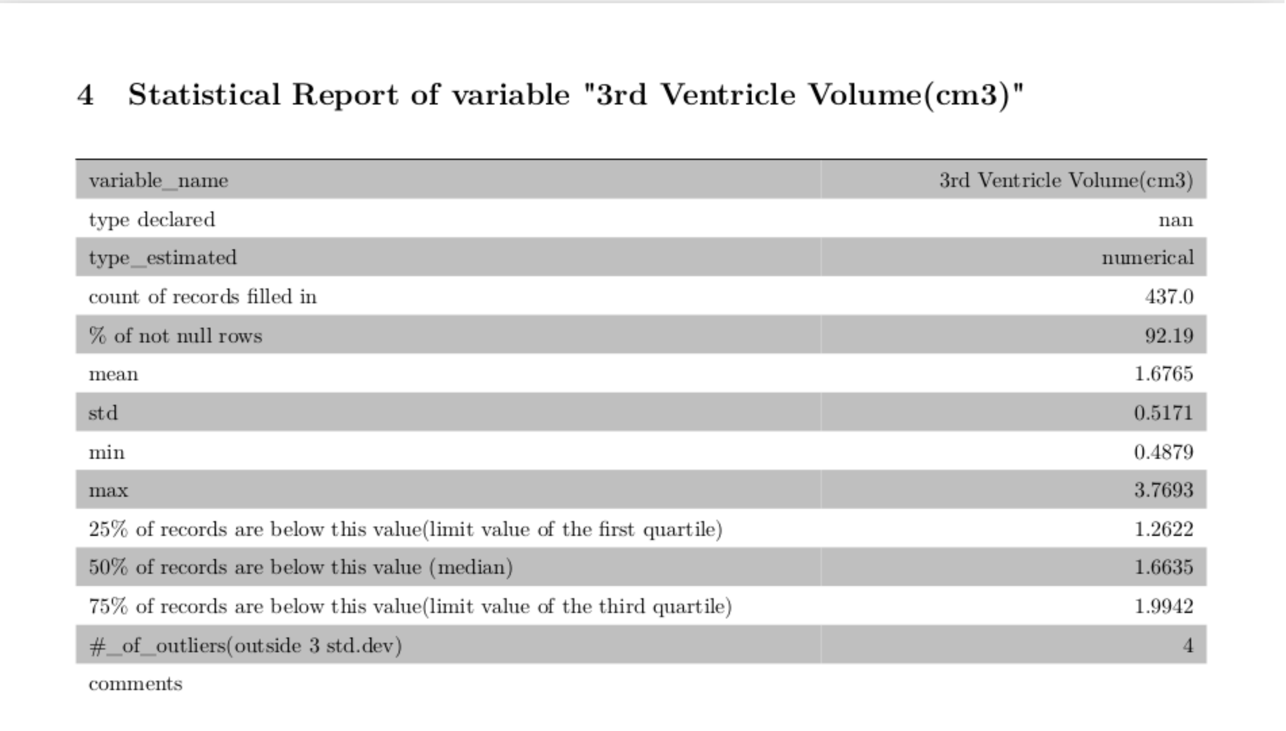


1 Statistical Report of edsd_merge.csv file

name_of_file	edsd_merge.csv
date_qc_ran	2019-03-26 17:54:57
qctool_version	0.1.2
total_variables	145
total_rows	474
rows with only id	0
rows with no id	0
rows with all columns filled	0
number of variables with 100% records filled in	3
number of variables with 80-99.99% records filled in	136
number of variables with 60-79.99% records filled in	5
number of variables with 40-59.99% records filled in	0
number of variables with 20-39.99% records filled in	0
number of variables with 0-19.99% records filled in	1

Figure 5: Statistical Report for all EDSD variables

In the second part the tool generates statistics for each dataset variable separately. For the EDSD dataset which has 145 variables, a report with statistical measurements is depicted in Figure 5.



4 Statistical Report of variable "3rd Ventricle Volume(cm3)"

variable_name	3rd Ventricle Volume(cm3)
type declared	nan
type_estimated	numerical
count of records filled in	437.0
% of not null rows	92.19
mean	1.6765
std	0.5171
min	0.4879
max	3.7693
25% of records are below this value(limit value of the first quartile)	1.2622
50% of records are below this value (median)	1.6635
75% of records are below this value(limit value of the third quartile)	1.9942
#_of_outliers(outside 3 std.dev)	4
comments	

Figure 6: Statistical Report for EDSD's 3rd Ventricle Volume variable

The second module of the same tool takes as input DICOM [7]brain scans and parses their attributes in order to check compliance with the MIP requirements for DICOM files defined

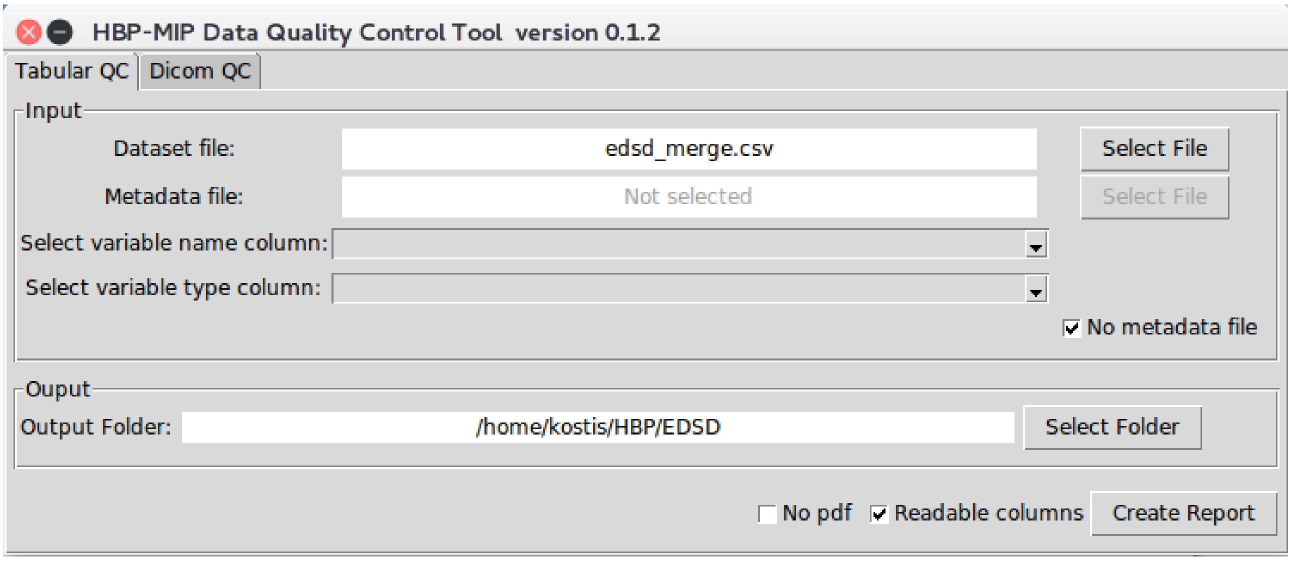
[7] https://www.dicomstandard.org/

in https://hbpmedical.github.io/deployment/data/. Each DICOM 3D image consists of several images slices which are grouped by the tool so as to produce aggregated attributes.

2. The **QC data cleansing tool** for tabular data that will be giving recommendations for value corrections that can be useful to hospitals' personnel. The implementation of this feature is planned to take place in the second half of SGA2.

All these tools are/will be integrated in the data ingestion process so as to guarantee quality of data at an early stage. In cases where the dataset exported by the hospital does not meet the specified quality standards, the hospital personnel will be informed and prompted to provide a new dataset having eliminated (or even better corrected) the problematic tuples and/or images.

The current version of the profiling tool which can be executed both via its GUI and in a terminal, is at https://github.com/HBPMedical/DataQualityControlTool.



HBP-MIP Data Quality Control Tool version 0.1.2

Tabular QC | Dicom QC

Input

Dataset file: edsd_merge.csv

Metadata file: Not selected

Select variable name column:

Select variable type column:

☑ No metadata file

Ouput

Output Folder: /home/kostis/HBP/EDSD

□ No pdf ☑ Readable columns

Figure 7: Quality Control tool's Graphical User Interface

4.1.6 *Scientific Workflow Engine*

Progress on MIP's Scientific Workflow Engine Component Galaxy [8]was focused on the experimental integration with the MIP Federate Network, as described in the deliverable D8.5.1. This included:

- An integration of the current Galaxy project workflow engine with the Distributed / Federated Query Execution Engine EXAREME (https://github.com/madgik/galaxy/tree/release_19.01/tools/exaremeTools)
- Deployment scripts that assist this process. (https://github.com/madgik/Galaxy-Ansible-Scripts)

In order to achieve the integration we had to modify some federated algorithms (https://github.com/madgik/mip-algorithms/tree/dev_exareme_v19). 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 [9] , tabular data resource format

[8] https://galaxyproject.org

[9] http://json.org

(https://github.com/frictionlessdata/specs/blob/master/specs/tabular-data-resource.md) or Highcharts [10]format.

Finally, we developed an example (Cross-Validation) in order to showcase a workflow execution through Galaxy's visual Workflow Editor with Federated MIP's engine. Once a predictive model is fitted to a dataset, it is natural to evaluate its performance. In case there are several alternative models, the model performance measure can be used for comparing between the alternatives and for selecting the best model (Model Selection). There are many ways to evaluate model performance. Different evaluation measures reflect different research goals, different statistical assumptions and even different statistical perspectives. One important family of model performance measures is Cross-Validation. MIP should allow users to evaluate their model performances. Therefore, we started separating the algorithms into distinct workflow components training component and predict component – see appendix A). Further, we have implemented K-fold and Hold out validation as two different workflow components. Finally, we have created workflow components that transform the output of the engine which is JSON format to other formats such as Highcharts [11]and tabular data resource format.

Through a graphical interface of the workflow engine, the user can create and execute workflows using the aforementioned components .

(https://github.com/madgik/galaxy/tree/release_19.01).

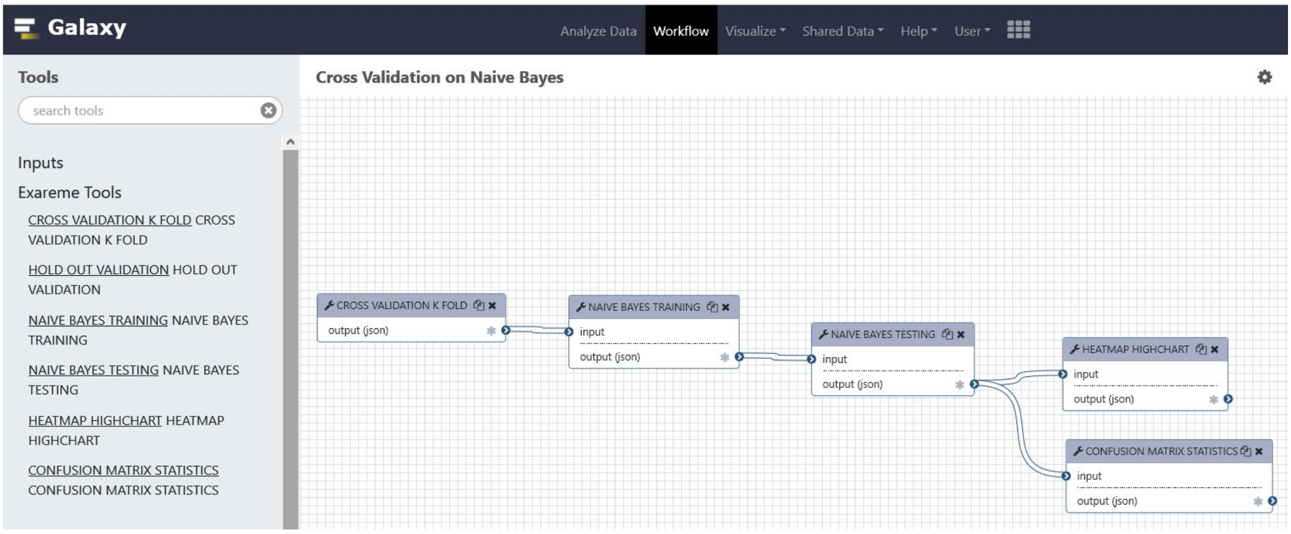

Galaxy
Analyze Data
Workflow
Visualize
Shared Data
Help
User
Tools
search tools
Inputs
Exareme Tools
CROSS VALIDATION K FOLD CROSS VALIDATION K FOLD
HOLD OUT VALIDATION HOLD OUT VALIDATION
NAIVE BAYES TRAINING NAIVE BAYES TRAINING
NAIVE BAYES TESTING NAIVE BAYES TESTING
HEATMAP HIGHCHART HEATMAP HIGHCHART
CONFUSION MATRIX STATISTICS CONFUSION MATRIX STATISTICS
Cross Validation on Naive Bayes
CROSS VALIDATION K FOLD
output (json)
NAIVE BAYES TRAINING
input
output (json)
NAIVE BAYES TESTING
input
output (json)
HEATMAP HIGHCHART
input
output (json)
CONFUSION MATRIX STATISTICS
input
output (json)

Figure 8: Galaxy graphical workflow engine interface

4.2 MIP Releases Component Roadmap

Table 1: Overview of new additions / developments roadmap

Component	MIP M12	MIP M20	MIP M24
Anonymisation component	Yes	Yes	Yes
Data Catalogue	Yes	Yes	Yes
Loris Integrator	No	Yes	Yes
QC Tools	Profiling Tools	Profiling Tools + Data cleansing	Profiling Tools + Data cleansing

[10] https://www.highcharts.com
[11] https://www.highcharts.com/demo

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	CART,Gradient Boosting, Random Forest	Stochastic Gradient Descent with Elastic Net Regularization, Mann–Whitney U test, Wilcoxon signed-rank test, Spearman correlation, Kendall correlation
Knowledge graph Proof of Concept	No	No	Yes
Automatic Data Extraction POC	No	No	Yes

4.3 MIP M12 Release Components

Table 2: Overview of major updates of MIP’s software components for M12 release

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: v.18 Release date: 02/15/2019 URL: https://github.com/madgik/exareme/releases/tag/v18 Execution 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)	Integration tests for the web portal, including end-to-end tests for algorithms powered by Exareme or Woken. The tests are for Input and output model and algorithm API test. This component will be integrated fully in Frontend test suite Release name : 2018.11.27 Release : 2018.11.27 Release date : 2018.11.27 URL : https://github.com/HBPMedical/mip-api-tests

New version of portal frontend	Software	M. SPUHLER (P27)	Release name : 2018.11.27;
			Release : 2018.11.27;
			Release date : 2018.11.27;
			URL : https://github.com/HBPMedical/mip-api-tests
			Integration tests for the web portal, including end-to-end tests for algorithms powered by Exareme or Woken.
			The tests are for Input and output model and algorithm API test. This component will be integrated fully in Frontend test suite
			Release name : 2019.02.07;
			Release : 2.15.0;
			Release date : 2019.02.07;
			URL : https://github.com/HBPMedical/portal-frontend/archive/2.15.0.zip
			Portal Frontend is the 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.
			Robustisation, error handling, normalized data structure. Migration performed in anticipation of complex visualization and third-party integrations

New release of Portal Backend	Software	Ludovic CLAUDE (P27)	Release name : 2019.02.07;
			Release : 2.15.0;
			Release date : 2019.02.07;
			URL : https://github.com/HBPMedical/portal-frontend/archive/2.15.0.zip
			Portal Frontend is the 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.
			Robustisation, error handling, normalized data structure. Migration performed in anticipation of complex visualization and third-party integrations

New release of Algorithm Factory (Woken)	Software	Ludovic CLAUDE (P27)	Release name : 2019.02.22 Release : 2.9.5 Release date : 2019.02.22 URL : https://github.com/HBPMedical/woken/releases/tag/2.9.5 https://github.com/LREN-CHUV/woken-messages/releases/tag/2.9.5 Woken is the execution and workflow engine for local or distributed algorithms. It uses a library of algorithms packaged in Docker containers for reproducibility and ease of deployment (the Algorithm repository). Algorithms are dispatched on a Mesos cluster for scalable execution and support for many users or complex experiments executing several algorithms in parallel.

			Release highlights: new distributed algorithms (PCA, k-means), updates to existing algorithms. New features include support for multiple tables, websockets for efficient distributed algorithms. Fixes on k-fold validation, numerous bug fixes and stabilisation, error recovery and internal health checks reporting errors with Bugnag and automated creation of bug reports in JIRA. Full list of algorithms defined in the repository: https://github.com/LREN-CHUV/algorithm-repository/blob/master/README.md Detailed release notes: https://github.com/LREN-CHUV/woken/blob/master/CHANGELOG.md

New release of Algorithm Factory (Woken validation part)	Software	Ludovic CLAUDE (P27)	Release name : 2019.02.22;
			Release : 2.6.4;
			Release date : 2019.02.22;
			URL : https://github.com/HBPMedical/woken-validation/releases/tag/2.6.4
			Detailed release notes: https://github.com/HBPMedical/woken-validation/blob/master/CHANGELOG.md;
			Stabilisation, error recovery and internal health checks

Reference data and metadata	Software / data	L. CLAUDE (P27)	Generic tool to install database tables and fill them with data https://github.com/LREN-CHUV/data-db-setup/tree/2.6.1 Deployment of the table used to store MIP CDE variables in the features database https://github.com/LREN-CHUV/mip-cde-data-db-setup/tree/1.5.1 Generic tool https://github.com/LREN-CHUV/meta-db-setup/tree/2.5.0 https://github.com/LREN-CHUV/mip-cde-meta-db-setup/tree/1.3.5 Reference data and metadata, including the definition of MIP Common Data Elements (CDE) and pre-processed research datasets (ADNI, PPMI, EDSD).

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 will detect outliers, corrupted records and also offer the user filtering and partitioning abilities.
New release of Deployment scripts for MIP platform	Software	Ludovic CLAUDE (P27)	Release name : 2005.23 Release : 2.8.5 Release date :05.23.2018 URL: https://github.com/LREN-CHUV/mip-microservices-infrastructure/releases/tag/2.8.5
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: Relase date: 03/01/2019 URL: https://github.com/madgik/galaxy/releases/tag/v19.01 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.

Appendix A Federated privacy-complying algorithms

A. Notation

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

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

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 preprocessing 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)

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

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 $C_m \in \{C_1, C_2, \dots, C_M\}$ for the domain of the corresponding variable.

B. Privacy

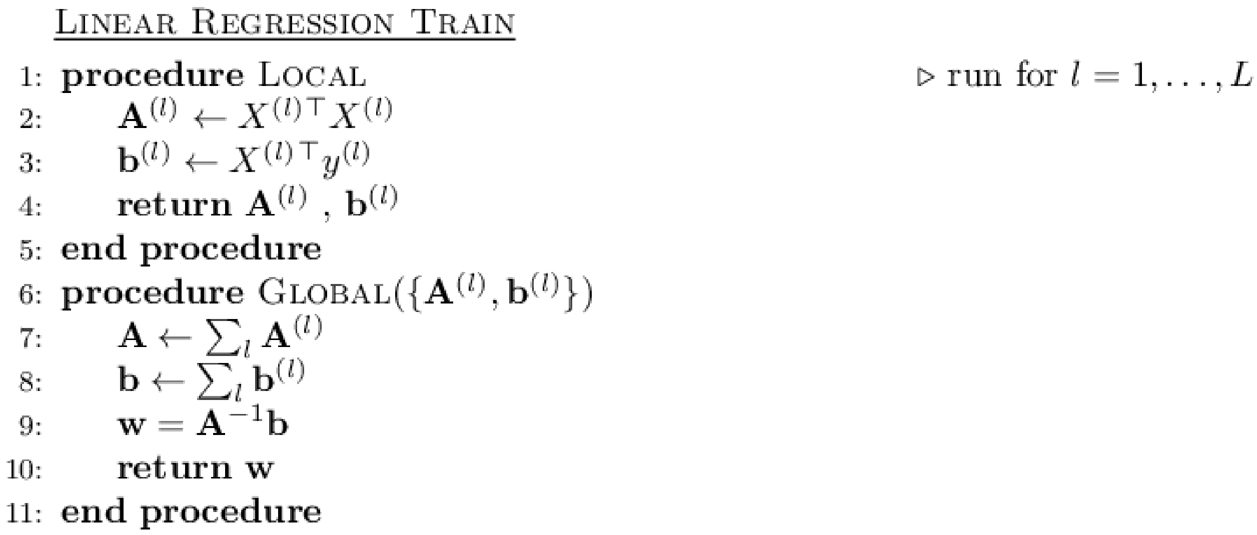
PriPrivacy in the MIP's Federated Processing Engine analytical capabilities is enforced in three stages.

- *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.Linear Regression

The current algorithm asks for the local datasets to return a matrix A and a vector b to the central node. For continuous variables, as long as $n > p$ there is no straightforward way to reconstruct the original data from these quantities, as the system of equations has infinite solutions. We explicitly enforce that $n > p$.

Below we provide the pseudocode of the training procedure for Linear Regression.



```
LINEAR REGRESSION TRAIN
procedure LOCAL                                   ▷ run for $l = 1, \ldots, L$
    $\mathbf{A}^{(l)} \leftarrow X^{(l)\top} X^{(l)}$
    $\mathbf{b}^{(l)} \leftarrow X^{(l)\top} y^{(l)}$
    return $\mathbf{A}^{(l)}$ , $\mathbf{b}^{(l)}$
end procedure
procedure GLOBAL($\{\mathbf{A}^{(l)}, \mathbf{b}^{(l)}\}$)
    $\mathbf{A} \leftarrow \sum_l \mathbf{A}^{(l)}$
    $\mathbf{b} \leftarrow \sum_l \mathbf{b}^{(l)}$
    $\mathbf{w} = \mathbf{A}^{-1}\mathbf{b}$
    return $\mathbf{w}$
end procedure
```

Once the process has been completed we compute the usual diagnostics as follows.

The local nodes compute and broadcast to the central node the quantities $\min(\{\epsilon_i\})$, $\max(\{\epsilon_i\})$, $\sum_i \epsilon_i$, $\sum_i \epsilon_i^2$, where ϵ_i are the residuals, as well as the partial SST and SSE. The central node then integrates these values to compute the corresponding global ones.

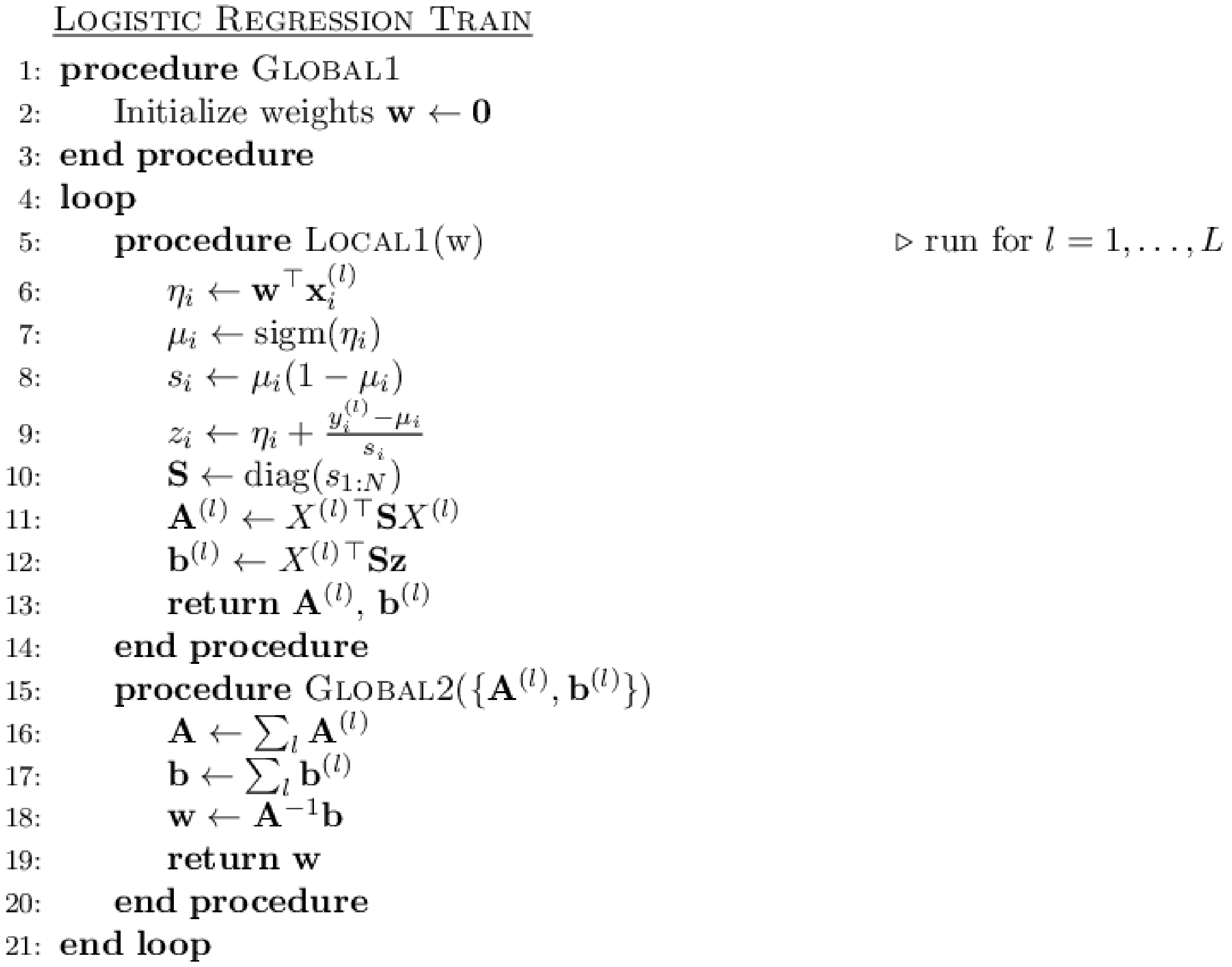
From these quantities the central node then computes the following diagnostic quantities:

- For each coefficient β_k the SE, t-statistic and $\Pr(> |t|)$
- $\min$, $\max$, and SE of residuals ϵ_i and the degrees of freedom
- R^2 and Adjusted R^2
- F-statistic and p-value

D. Binary Logistic Regression

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.

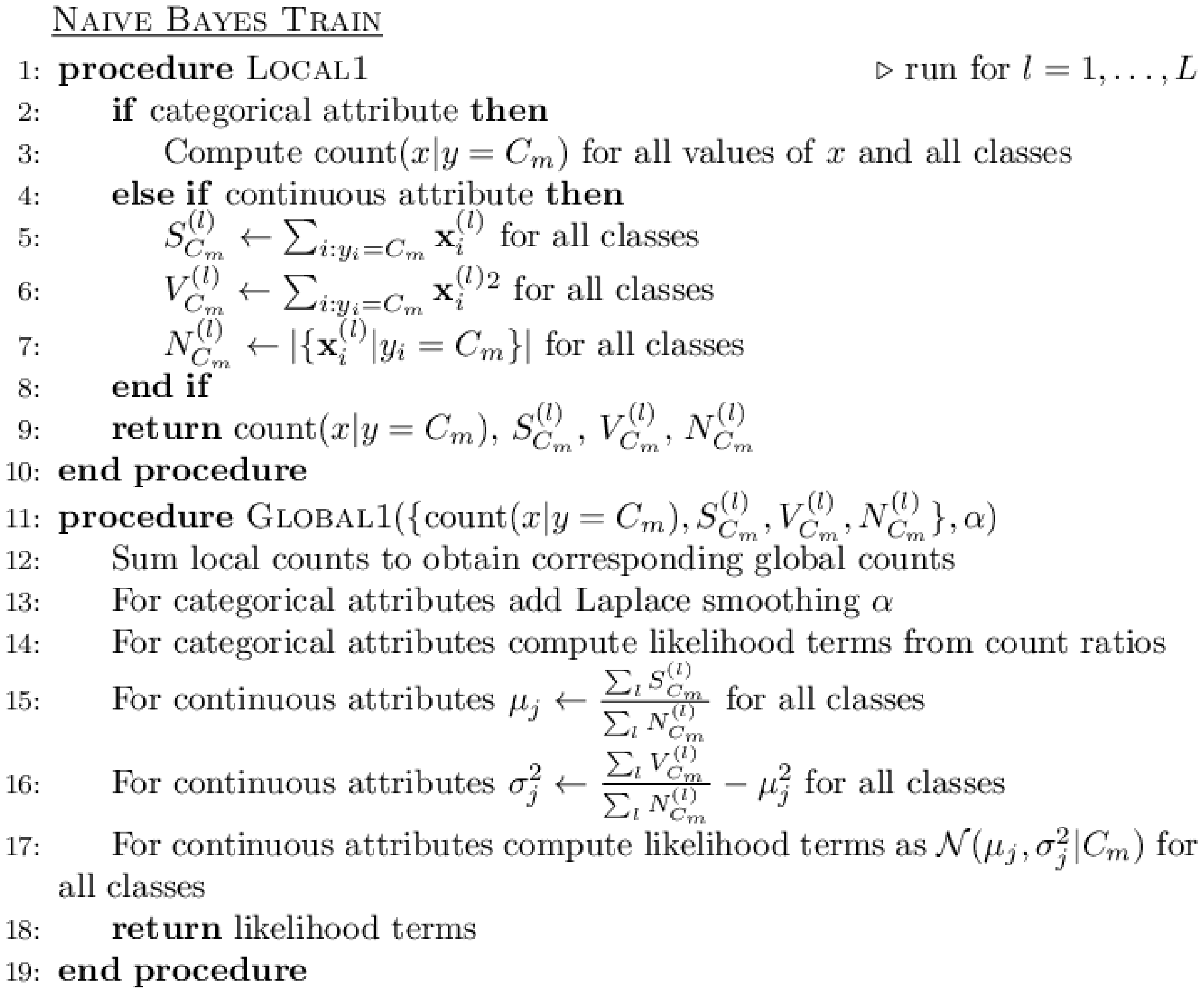


```
LOGISTIC REGRESSION TRAIN
procedure GLOBAL1
    Initialize weights $\mathbf{w} \leftarrow \mathbf{0}$
end procedure
loop
    procedure LOCAL1(w)                                   ▷ run for $l = 1, \dots, L$
        $\eta_i \leftarrow \mathbf{w}^\top \mathbf{x}_i^{(l)}$
        $\mu_i \leftarrow \mathrm{sigm}(\eta_i)$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

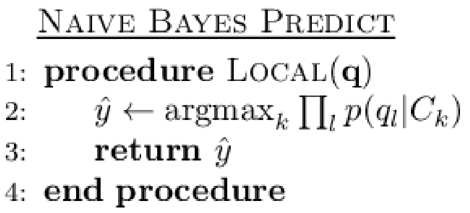
For the Naïve Bayes algorithm, the values of the various columns of $\mathbf{X}$ can be both categorical and continuous, while the y is always categorical and takes values $y \in \{C_1, C_2, \dots, C_M\}$ where M is the total number of classes. The training procedure is different for the categorical and continuous cases.

As we mentioned earlier, aggregate quantities, *i.e.* counts and sums, are always computed on subsets of size at least k. This, together with the fact that all sums and counts are computed on disjoint subsets in the present algorithm, ensures that full data reconstruction cannot be done straightforwardly.



```
NAIVE BAYES TRAIN
procedure LOCAL1                                                ▷ run for $l = 1, \ldots, L$
    if categorical attribute then
        Compute count$(x|y = C_m)$ for all values of $x$ and all classes
    else if continuous attribute then
        $S^{(l)}_{C_m} \leftarrow \sum_{i:y_i=C_m} \mathbf{x}^{(l)}_i$ for all classes
        $V^{(l)}_{C_m} \leftarrow \sum_{i:y_i=C_m} \mathbf{x}^{(l)2}_i$ for all classes
        $N^{(l)}_{C_m} \leftarrow |\{\mathbf{x}^{(l)}_i | y_i = C_m\}|$ for all classes
    end if
    return count$(x|y = C_m)$, $S^{(l)}_{C_m}$, $V^{(l)}_{C_m}$, $N^{(l)}_{C_m}$
end procedure
procedure GLOBAL1$(\{\text{count}(x|y = C_m), S^{(l)}_{C_m}, V^{(l)}_{C_m}, N^{(l)}_{C_m}\}, \alpha)$
    Sum local counts to obtain corresponding global counts
    For categorical attributes add Laplace smoothing $\alpha$
    For categorical attributes compute likelihood terms from count ratios
    For continuous attributes $\mu_j \leftarrow \frac{\sum_l S^{(l)}_{C_m}}{\sum_l N^{(l)}_{C_m}}$ for all classes
    For continuous attributes $\sigma^2_j \leftarrow \frac{\sum_l V^{(l)}_{C_m}}{\sum_l N^{(l)}_{C_m}} - \mu^2_j$ for all classes
    For continuous attributes compute likelihood terms as $\mathcal{N}(\mu_j, \sigma^2_j | C_m)$ for
all classes
    return likelihood terms
end procedure
```

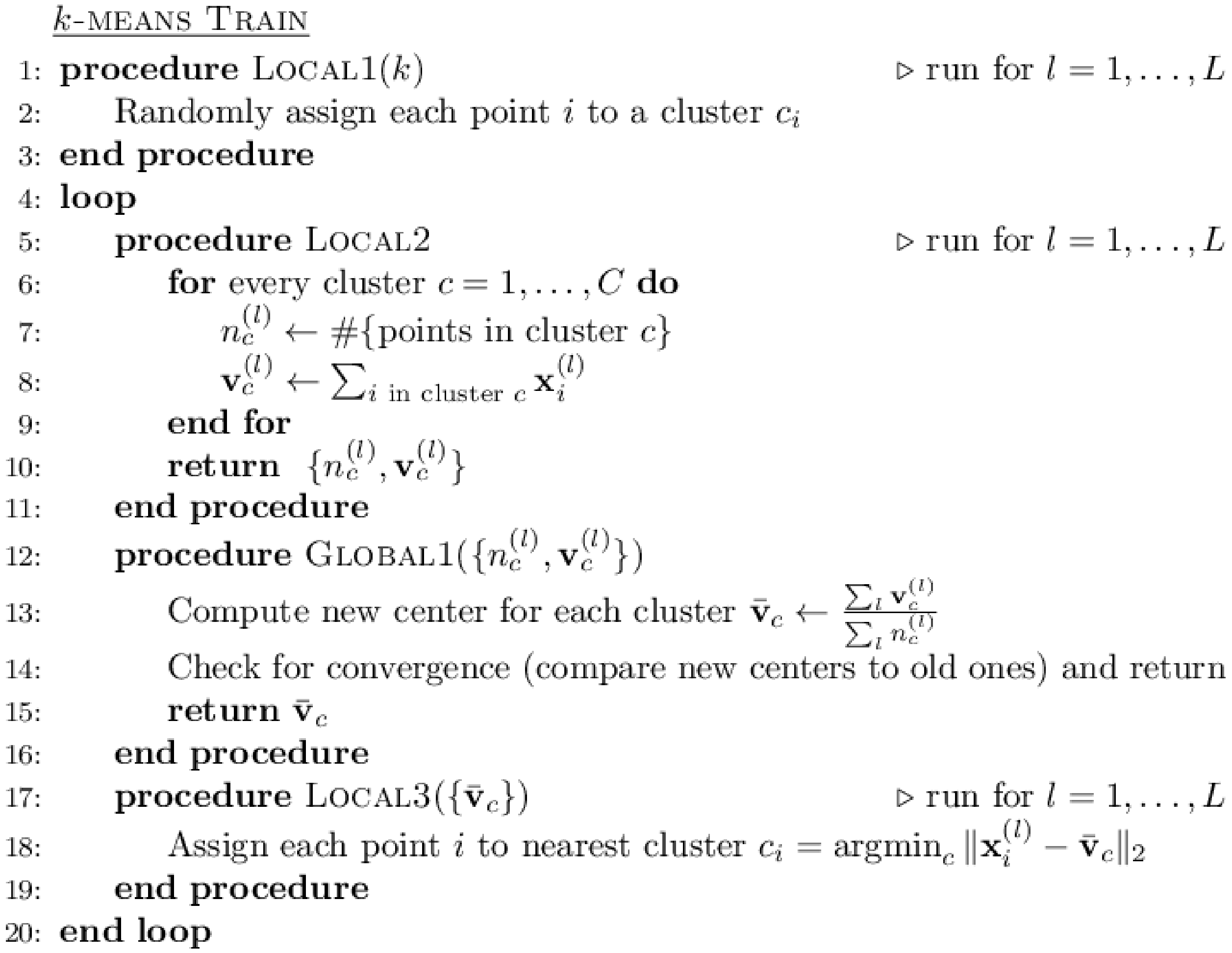
Once we have the likelihood terms we can compute the maximum posterior probability for the class of a new query datapoint q with the following procedure



```
NAIVE BAYES PREDICT
procedure LOCAL($\mathbf{q}$)
    $\hat{y} \leftarrow \text{argmax}_k \prod_l p(q_l|C_k)$
    return $\hat{y}$
end procedure
```

F. k-Means

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.



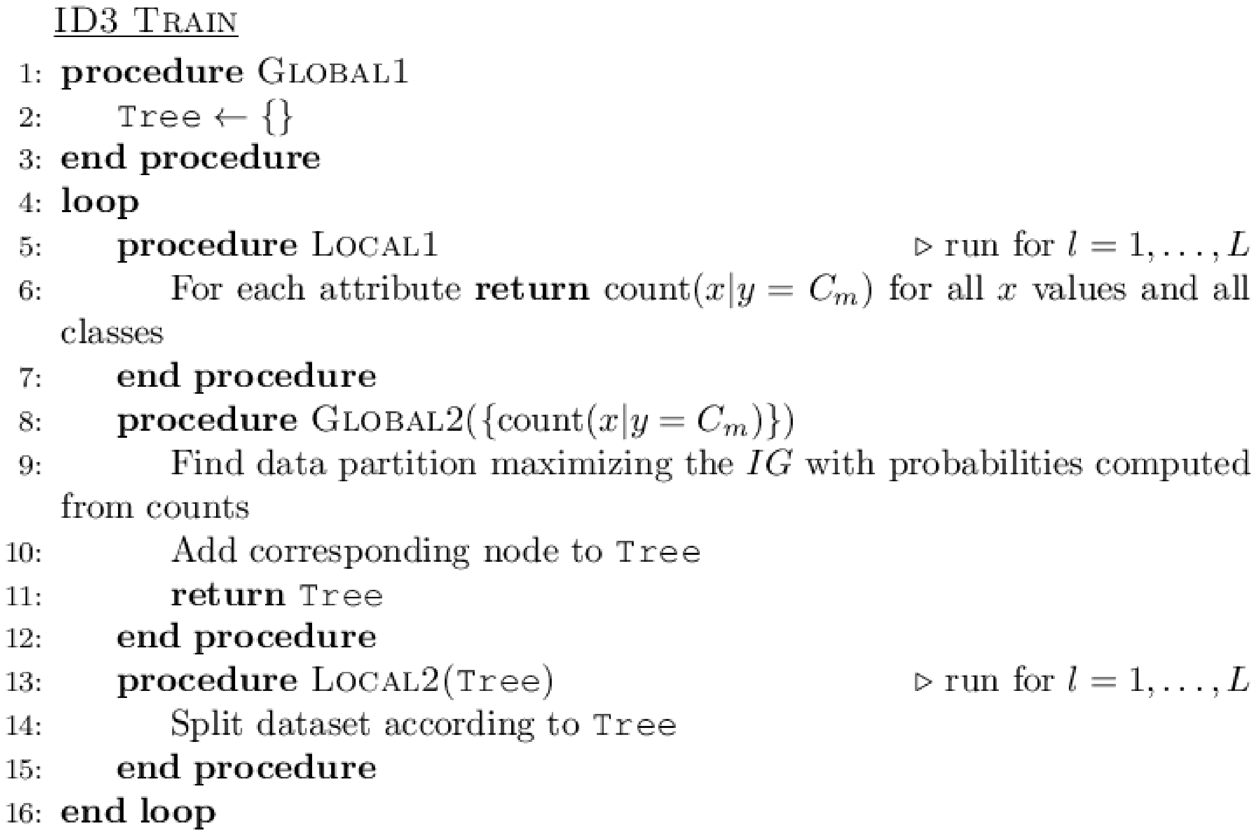
k-MEANS TRAIN

```
procedure LOCAL1($k$)                                  ▷ run for $l = 1, \ldots, L$
    Randomly assign each point $i$ to a cluster $c_i$
end procedure
loop
    procedure LOCAL2                                   ▷ run for $l = 1, \ldots, L$
        for every cluster $c = 1, \ldots, C$ do
            $n_c^{(l)} \leftarrow \#\{\text{points in cluster } c\}$
            $\mathbf{v}_c^{(l)} \leftarrow \sum_{i \text{ in cluster } c} \mathbf{x}_i^{(l)}$
        end for
        return $\{n_c^{(l)}, \mathbf{v}_c^{(l)}\}$
    end procedure
    procedure GLOBAL1($\{n_c^{(l)}, \mathbf{v}_c^{(l)}\}$)
        Compute new center for each cluster $\bar{\mathbf{v}}_c \leftarrow \frac{\sum_l \mathbf{v}_c^{(l)}}{\sum_l n_c^{(l)}}$
        Check for convergence (compare new centers to old ones) and return
        return $\bar{\mathbf{v}}_c$
    end procedure
    procedure LOCAL3($\{\bar{\mathbf{v}}_c\}$)                  ▷ run for $l = 1, \ldots, L$
        Assign each point $i$ to nearest cluster $c_i = \text{argmin}_c \|\mathbf{x}_i^{(l)} - \bar{\mathbf{v}}_c\|_2$
    end procedure
end loop
```

G. ID3

For the ID3 algorithm we will use the same notation for the local datasets $\mathbf{X}^{(j)}$ and $\mathbf{y}^{(j)}$ with the difference that here all values are necessarily categorical and the values of $y \in \{C_1, C_2, \ldots, C_M\}$ represent class membership, where M is the total number of classes. The purpose of the ID3 algorithm is to construct a decision tree for the given dataset.

The same privacy arguments as in Naive Bayes apply here. The local components of the algorithm always broadcast counts of disjoint subsets of size at least k.



```
ID3 Train
procedure Global1
    Tree ← {}
end procedure
loop
    procedure Local1                                              ▷ run for $l = 1, \ldots, L$
        For each attribute return count($x|y = C_m$) for all $x$ values and all classes
    end procedure
    procedure Global2({count($x|y = C_m$)})
        Find data partition maximizing the $IG$ with probabilities computed from counts
        Add corresponding node to Tree
        return Tree
    end procedure
    procedure Local2(Tree)                                        ▷ run for $l = 1, \ldots, L$
        Split dataset according to Tree
    end procedure
end loop
```

The information gain is defined as

$$IG = H(y) - \sum_{\text{values of } x} p(x) H(y|x) \tag{4}$$

Where

$$H(y|x) = - \sum_{\text{values of } y} p(y|x) \log p(y|x) \tag{5}$$

The entropy term $H(y)$ is constant so we only need to compute the second for the minimization.

H. Pearson Correlation

This algorithm computes the Pearson correlation coefficient between two vectors x and y using the equation

$$\mathrm{r}_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{6}$$

PEARSON CORRELATION COEFFICIENT

```
procedure LOCAL                                        ▷ run for $l = 1, \dots, L$
    $n^{(l)} \leftarrow \#\{\text{datapoints in local db } l\}$
    $s_x^{(l)} \leftarrow \sum_i x_i$
    $s_y^{(l)} \leftarrow \sum_i y_i$
    $s_{xx}^{(l)} \leftarrow \sum_i x_i^2$
    $s_{xy}^{(l)} \leftarrow \sum_i x_i y_i$
    $s_{yy}^{(l)} \leftarrow \sum_i y_i^2$
    return $(n^{(l)}, s_x^{(l)}, s_y^{(l)}, s_{xx}^{(l)}, s_{xy}^{(l)}, s_{yy}^{(l)})$
end procedure
procedure GLOBAL($n^{(l)}, s_x^{(l)}, s_y^{(l)}, s_{xx}^{(l)}, s_{xy}^{(l)}, s_{yy}^{(l)}$)
    $n \leftarrow \sum_l n^{(l)}$
    $s_x \leftarrow \sum_l s_x^{(l)}$
    $s_y \leftarrow \sum_l s_y^{(l)}$
    $s_{xx} \leftarrow \sum_l s_{xx}^{(l)}$
    $s_{xy} \leftarrow \sum_l s_{xy}^{(l)}$
    $s_{yy} \leftarrow \sum_l s_{yy}^{(l)}$
    Compute $r_{xy}$ according to the above equation
    return $r_{xy}$
end procedure
```

Appendix B MIP Architecture

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.

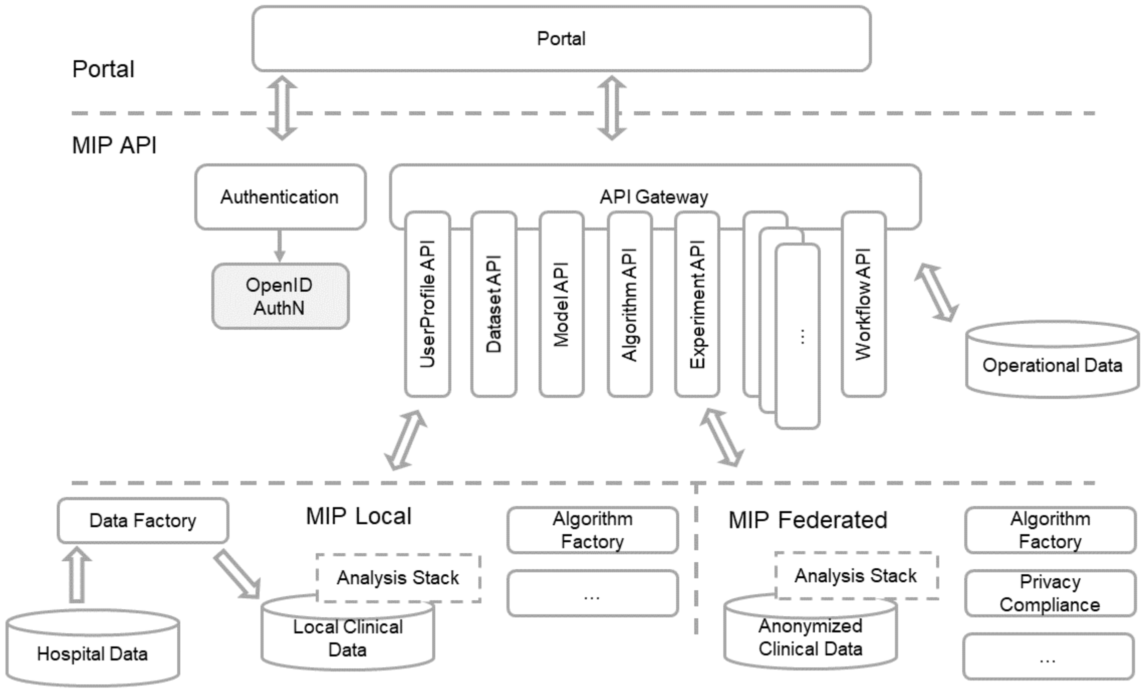


Figure 9: Overall Architecture

The MIP is architected following an N-Tier paradigm. Multiplied 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 reused 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.

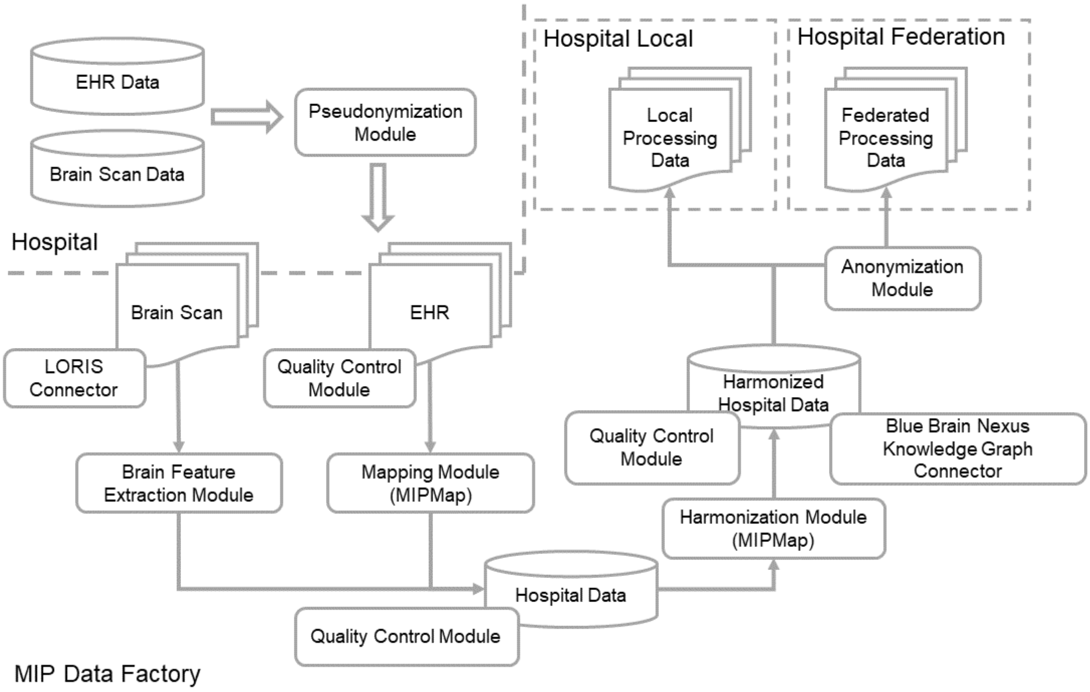


Figure 10: Data Factory

- The initial hospital data, including both electronic health records as well as brain scan data, go through a process of pseudonymization 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
- Federated analysis
- 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. The following diagram, Figure 3 - Local vs Federated Analysis, depicts the architecture and processing flow between the different modes of analysis offered.

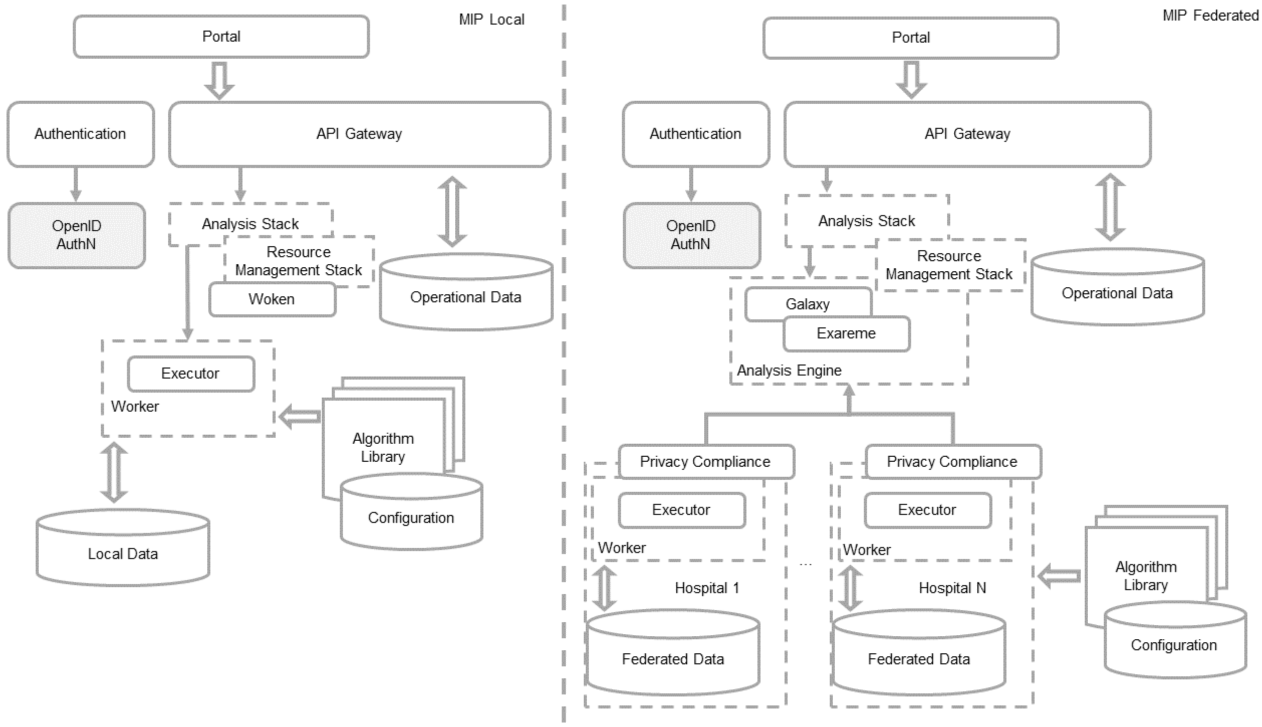


Figure 11: Local vs Federated Analysis

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.

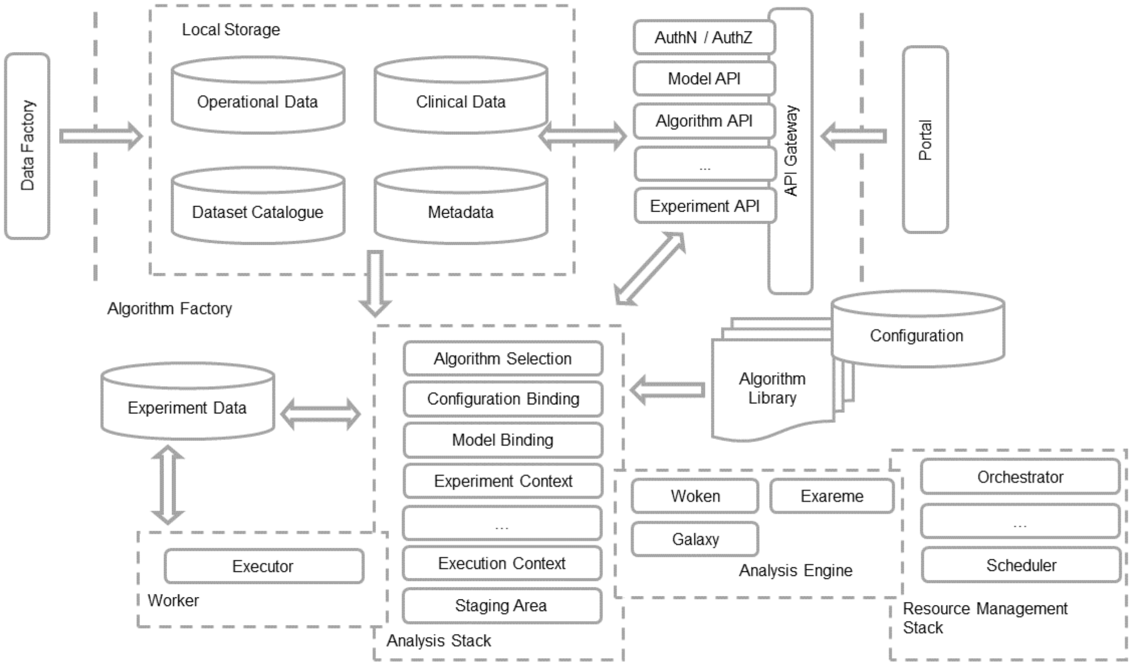


Figure 12: Algorithm Factory

Within each hospital, depending on the mode of analysis, local or federated, the set of available clinical data, pseudonymized or anonymized, 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 utilized, 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.

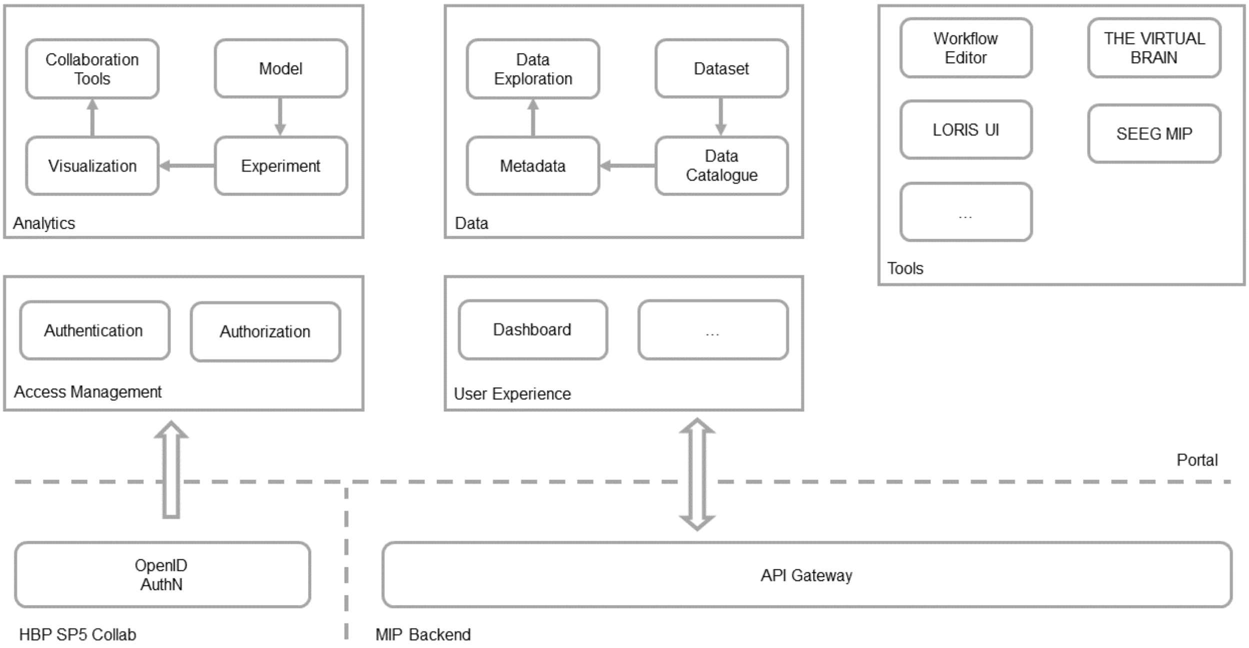


Figure 13: Portal

- 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
- The **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.