|  |  |
| --- | --- |
|  | Moving Picture, Audio and Data Coding by Artificial Intelligence  www.mpai.community |

**Public document**

|  |  |
| --- | --- |
| **N914** | 2022/10/26 |
| **Source** | Requirements (AIH) |
| **Title** | The MPAI-AIH Architecture WD0.3 |
| **Target** | MPAI Members |

Table of Contents

[1 Introduction 2](#_Toc117280304)

[2 Definition of Terms 2](#_Toc117280305)

[3 AIM Health Secure Platform 2](#_Toc117280306)

[3.1 Actors 3](#_Toc117280307)

[3.2 Services 4](#_Toc117280308)

[4 API Specification 5](#_Toc117280309)

[4.1.1 API: Mobile App <-> Back-end System 5](#_Toc117280310)

[4.1.2 API: Back-end System <-> Third Parties 5](#_Toc117280311)

[4.1.3 API: Back-end System <-> Blockchain 5](#_Toc117280312)

[5 Healthcare use case 5](#_Toc117280313)

[5.1 User health data collection 5](#_Toc117280314)

[5.2 Background process 6](#_Toc117280315)

[5.3 Access to healthcare data 6](#_Toc117280316)

[5.4 Intelligent processing of healthcare data 6](#_Toc117280317)

[5.5 Auditing healthcare data access 7](#_Toc117280318)

[6 7](#_Toc117280319)

[7 Intelligent Computational Service Organization 7](#_Toc117280320)

[8 Distributed Services 8](#_Toc117280321)

[8.1 Federated Learning Mobile applications 8](#_Toc117280322)

[8.1.1 Federated Learning: Healthcare Application Use Case 8](#_Toc117280323)

[8.1.2 Federated Learning Processes 8](#_Toc117280324)

[8.1.3 Training process 8](#_Toc117280325)

[8.1.4 Inference Process 9](#_Toc117280326)

[8.2 MPAI-AIH Federated Components 9](#_Toc117280327)

[8.3 MPAI-AIH Requirements 10](#_Toc117280328)

[9 Legal Requirements 10](#_Toc117280329)

[10 Functional requirements 12](#_Toc117280330)

[10.1 AIH-oriented requirements 12](#_Toc117280331)

[10.2 User-oriented requirements 12](#_Toc117280332)

[11 References 12](#_Toc117280333)

[12 Annex A: API Description 13](#_Toc117280334)

[12.1 API: Mobile App <-> Back-end System 13](#_Toc117280335)

[12.2 API: Back-end System <-> Third Parties 13](#_Toc117280336)

[12.3 API: Back-end System <-> B&DLT 13](#_Toc117280337)

# Introduction

Artificial Intelligence for Health data (MPAI-AIH) is an MPAI project addressing secure collection, AI-based processing, and secure access to Health data. The motivation for the development of this standardization comes from the AIM Health Portuguese project ([[[1]](#footnote-1)https://istar.iscte-iul.pt/aimhealth-en/)[[2]](#footnote-2)](https://istar.iscte-iul.pt/aimhealth-en/)%05%20The%20AIM%20Health%20project%20is%20funded%20by%20the%20Portuguese%20government%20and%20aims%20an%20early%20identification%20of%20symptoms%20that%20might%20be%20related%20to%20a%20COVID-19%20infection,%20especially%20for%20patients%20suffering%20from%20heart%20and%20cardiac%20problems.%20The%20AIM%20Health%20project%20is%20funded%20by%20the%20Portuguese%20government%20and%20aims%20an%20early%20identification%20of%20symptoms%20that%20might%20be%20related%20to%20a%20COVID-19%20infection,%20especially%20for%20patients%20suffering%20from%20heart%20and%20cardiac%20problems.%20This%20should%20be%20considered%20as%20just%20a%20simple%20possible%20use%20case.%20Multiple%20usage%20scenarios%20may%20coexist%20in%20the%20system,%20but%20for%20simplicity%20purposes,%20we%20have%20just%20described%20a%20possible%20one.). This document addresses the technical as well as legal specifications of an AIH System

# Definition of Terms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| AI Module |  |
| AI Workflow |  |
| AIH Platform |  |
| AIH Platform Back-end |  |
| AIH Platform Front-end |  |
| End User |  |
| Smart Contract |  |
| Third Party |  |
| External Source |  |

# AIM Health Secure Platform

The overall architecture for MPAI-AIH (Figure 1) comprises a set of different actors, specific distributed services and APIs that will be described in the following sections.



Figure 1 - Reference Model of MPAI-AIH Secure Platform

The platform is divided into front-end and back-end components. The front-end architecture is detailed in the following image (Figure 2).



Figure – Front-end architecture of the MPAI-AIH Secure Platform

## Actors

The AI for Health data system identifies and recognizes the following different actors/entities.

1. *End-user*: a user for which his/her health-related data is going to be collected by the system. This end-user will control and audit the access of any third party to his/her health-related data according to the terms of a smart contract issued at the time a “third party” entity requires access and gets user approval to do so.
2. *Third-party*: any third-party entity requiring access to the data on the system or to process that data and extract knowledge through the usage of some artificial intelligence mechanism (or through the orchestration of multiple intelligence mechanisms). This includes hospitals, clinics, research centers, caretakers, and others. Access is granted according to the terms of the smart contract between that third party and the end-user.
3. *External Data Sources*: represent platforms from which the system may collect subsidiary data for the integration of relevant information for health-related predictions. Access and provenance of External Data Sources are regulated via smart contracts.

## Services

The AI Health data system is composed of a set of distributed group components and services that are depicted in Figure 1. These are:

1. The *front-end system*, a smart device (smartphone or other similar) application (AI-Health app) that is capable of capturing, storing, and processing health-related data from the end-user. The user smart device represents the personal gateway to the user-data. This data can be collected directly using the device sensors and applications installed in the smartphone, such as Google Fit and Apple Health. The smartphone can also act as communication gateway for any external biometric sensors that capture health-related data events from the user. All the collected data will be securely stored in the user’s smartphone using a “Secure Data Vault” whose access is controlled by the end-user (Figure 2). The smartphone will also alert the end-user about any deviation that may be associated to a disease or symptomatic abnormality that can be inferred from a machine learning data analysis.
2. The *back-end system*, composed by a set of tools that implement the necessary services to securely store data, de-identify and anonymize data, control entity authentication and access control to data, and to license and audit the access to health-related user data on the backend system. This system gathers anonymized data from all the various sources (the users that allow it and other important sources) and acts as a broker gateway between the entities that would like to access data and those who will supply it.

There are two types of smart contracts (SC): between the user and the backend, and between the backend and the entity.

Elements of the SC are:

1. Parties.
2. Data.
3. Rights granted
   1. Processing to be made on the data (need a taxonomy).
   2. Auditing of the processing done by the backend and by the third-party entity using the data.

The backend grants the rights without making reference to the end users identification who have provided the data. Of course, the backend may not grant the entity the right to make processing that the user did not grant to the backend.

The system takes advantage of Blockchain and Distributed Ledger technologies (B&DLT). The objective of this B&DLT system is to enable the transparency and auditability of the system. Every health-related data access request will require the emission of a license (smart contract) that will be stored on the B&DLT. This smart contract will contain information about the requesting entities (someone requesting access to data), the data requested, the access conditions (e.g., timeframe and the user’s permission). This information can be audited by the system and by the user to be sure that the data is being used according to what was established in the smart contract.

The system will also display a set of AI services that can be used to directly treat and process the data on the device for the extraction of specific knowledge of interest to the end-user or for third parties with contracted rights. These services may be selected from the ones available from the MPAI Store and may be orchestrated to produce specific analyses for the entities that request access to health-related data. AI services through data processing enable specific and customized training of machine learning models to identify and assist in the identification of medical diagnosis and prognosis.

# API Specification

The system will use REST API interfaces that will provide data access. A special API acts as the interface between the overall system’s backend data and the AI modules for using and processing data. Data may also be collected from other sources such as public services and third-party entities, using other specialized APIs.

### API: Mobile App <-> Back-end System

This describes the API that is exposed by the back-end system to the smart device app. This API will provide the necessary services to register, authenticate and control access of the user in the system. Moreover, this API will also provide the mechanisms for handling user health related data on the system – data storage, permissions for data usage, and data usage auditing.

This API is composed of the following API entry points. Further details of this API is provided in the Annex A: API Description.

### API: Back-end System <-> Third Parties

This API is composed of the following API entry points. Further details of this API is provided in the Annex A: API Description.

### API: Back-end System <-> Blockchain

This API is composed of the following API entry points. Further details of this API is provided in the Annex A: API Description.

TBC: is there any other API that is relevant to describe in the document?

# Healthcare use case

This section presents a simple use case[[3]](#footnote-3) that describes the system and its usage (Figure 1).

## User health data collection

* The user signs into the system, which is equipped with the necessary security features and initializes the end-user secure data vault.
* The user configures the smartphone app to connect to the different data sources (either external sensors or internal installed apps).
* The user configures needed personal or special data (age, weight, specific health conditions, etc.)
* The MPAI-AIH app starts collecting relevant data from the user and securely stores it locally on the smartphone data vault.
* The user is requested for permission to contribute specific subsets of their health data to the AIH Platform and potentially to other well-identified systems. The user analyses the request and may give permission for this contribution . If the user gives permission to access their data, the back-end will create a smart contract to be accepted by the user and data will be collected via an API.
* The data from all the users participating in the system that have given their permission is collected into the back-end of the AIH Platform where it will be de-identified and anonymized.

## Background process

Background processes may take place in the front-end and in the back-end. They include:

1. Housekeeping services: a set of ancillary processes tasked with the proper functioning of the system. Those include scheduling, communications (messaging, file transfer), security, hardware and software diagnosis, calibration, as well as resource allocation (memory, storage, data bandwidth allocation).
2. Inference services: a set of services processing data and using machine learning models producing results for use by External Parties.

## Access to healthcare data

This section deals with processing done by the back-end based on the AIH System policy and on-demand processing of health data requested by External Parties. A demand may involve a huge amount of data transfer and processing. However, the project does not currently consider the associated cost, latency, payment, etc.

* Any authorized and authenticated third-party entity may request data access. This External Party needs to be properly registered and authenticated on the system to be able to access the proper APIs to request access to data.
* The External Party requests access to the data catalogue existing on the system using a query interface. The catalogue provides the following metadata with the following level of detail.
  + TBD
* The External Part optionally selects the intelligent mechanisms that exist to process the data and extract some type of results and intelligence from the data.
* Data may be processed inside the front-end and there shall be no need for an SC to process the data inside the smart phone.
* There is a need for an SC to process the data in the back-end to serve a request by an External Party. What is written in the SC: the back-end grants the External Party the right to process the designated data with proc1, proc2,…
* This involves the selection of the specific services to process the data (one service, or multiple services properly orchestrated) that can access the system’s data and perform some treatment. The existing services to process data will be selected and instantiated for this processing. Access to the service is based on a choice made from a service taxonomy, likely to be compatible with the taxonomy used in SCs.
* The entity accepts a smart contract created by the B&DLT users. Access to data is permitted as long as the smart contract is valid.

## Intelligent processing of healthcare data

There are two categories of intelligent processing of healthcare data: a processing, which happens in the front-end, and in the back-end at the request of and External Party. Intelligent processing of healthcare data follows best practices and of state-of-the-art machine learning techniques. It must be technically and socially robust, that is, accurate and reproducible, and able to deal with and inform about possible failures, inaccuracies and errors, aware of the potential repercussions of false positive (resp. negative) responses, and adopting privacy and security-preserving techniques, and allow for adequate knowledge sharing. In summary, it must comply with the seven principles and requirements for trustworthy AI: respect for human agency; privacy, personal data protection and data governance; fairness; individual, social, and environmental well-being; transparency; accountability and oversight.

* State of the art machine learning: This includes the dimensions of efficient computational processing with the proper trade-offs between computational cost and accuracy; the adoption and identification of techniques to identify bias-free and representative datasets, use of algorithmically unbiased models.
* Efficient Implementation Architecture: This addresses the search for computational organization that is best adapted to the task at hand in terms of resource utilization, namely execution hardware resources, and energy and computational requirements. The alternatives include the adoption of centralized server organizations that concentrate processing and distribute global knowledge, as well as distributed and continuously-learning models.
* Explainable Artificial Intelligence: The communication of the results of intelligent processing of healthcare data must strive to adopt techniques that provide a rationale for computer-generated decisions, along with reasonable estimates of the accuracy of a particular response, as well as a relative ranking of plausible alternatives.
* Security and Privacy preservation: The intelligent of large health-related datasets is done in such a way as to preserve the privacy of individuals, namely via the adoption of anonymization techniques and the use of security-preserving communication and storage methods.
* Knowledge Sharing: This point focuses on how learning from one user can be transferred and aggregated with learning from another user, while maintaining user’s privacy.

In practical terms, the processing of healthcare data includes:

1. an AIW (AI Workflow) is selected to process the data and the AIMs (AI Modules) load the data as needed and may store the data in the “AIM Storage”.
2. The AIW orchestrates the execution of the AIMs, which operate over the data. All of these can be downloaded or updated from the “MPAI Store”, if necessary.
3. The data is processed based on the AIW and stored in the secure data vault.

## Auditing healthcare data access

This is a mechanism in the system that allows users to audit the usage of their healthcare data. To accomplish this, the users require auditing through the backend API.

# 

# Intelligent Computational Service Organization

This section discusses the potential organization of intelligent computational services, which are classified as Centralized server-slave architectures, or based on Federated Learning.

**Centralized Services**

* This section discusses the centralized learning processes and services offered by back-end. Chief among those are centralized master-slave architectures driven by a high-resource master in charge or controlling and distributing intelligent models to slave processing.

# Distributed Services

This section discusses the Distributed intelligent services organization. Chief among those are Federated Learning processes and services offered by MPAI-AIH.

## Federated Learning Mobile applications

Federated Learning (also known as collaborative learning) is a technique that enables machine learning algorithms deployed across multiple decentralized edge devices or servers holding local data samples to collaboratively train a global model. This is done without exchanging user data: only (incremental) changes from the local machine learning model are uploaded to the global machine learning model and, eventually, upon authorization, a new (updated) model is downloaded into the edge devices, in an exchange between the decentralized devices and a server.

### Federated Learning: Healthcare Application Use Case

Healthcare and the health insurance industry may leverage federated learning systems since it allows for the protection of sensitive user data in the original edge device. Because data is kept locally, Federated Learning may be used to build AI models on user’s health information from a data pool of smartphones without leaking personal data. Federated learning models can provide for improved data diversity by gathering data from various locations and use cases (e.g., hospitals, electronic health record databases), for instance, to diagnose rare diseases.

The article in Reference [2] by Rieke et. Al. illustrates how federated learning can help solve challenges about data privacy and data governance by enabling machine learning models from distributed data sources.

### Federated Learning Processes

Federated learning is composed of two processes: training and inference.

### Training process

In the most common approach, presented by McMahan *et al.* [1], the training is done using a client-server architecture, as illustrated in Figure 3. A shared global model is defined by a central controller, also referred to as server (the backend system). Each client who participates in collaborative learning has a copy of the shared global model (their local machine learning model) and their private data set. The shared global model training is performed in rounds. At each round, the following steps are performed:

1. Groups of clients are selected in sequence by the server and sent a copy of the global model parameters (W).
2. The selected clients load the received parameters into their respective models and train them with their respective private datasets for a defined number of iterations/epochs. At the end, each client sends its parameters to the server (Δw).
3. Using an aggregation algorithm, the server combines the parameters received from the various clients and updates the global model.
4. The executions are then repeated until the model reaches convergence. In this way, sensitive data is not sent directly to the server, guaranteeing a certain level of privacy for clients

It is worth noticing that, even when the global shared model reaches convergence, it can be incrementally trained as the client’s data sets grow, in order to be more robust.

For example, end-users may have external devices connected to their smartphones which periodically exchange vital information to train their local machine learning models in order to alert of any anomaly. The model can be incrementally improved and extended by aggregation the knowledge of other models, from other users.

Diagram

Description automatically generated

Figure 3 - Federated Learning training process.

### Inference Process

For inference, each client simply uses the weights received from the global model and runs it on the desired data. Depending on the problem, sometimes the client may wish to perform a fine-tuning in their local model, in order to improve the accuracy and costumisation of the model for herself/himself.

## MPAI-AIH Federated Components

**Error! Reference source not found.**End-user devices and third-party entities communicate with the MPAI-AIH Secure Platform using secure APIs, to provide and request data.

The AIWs describes the process used to handle healthcare data. These AIWs orchestrate the usage of the AIMs, which are responsible for performing the computational operations, including transformations, training and inferences. For example, an AIW used to detect the presence of COVID-19 from a given user may have two AIMs, one for select, load and pre-process (e.g., apply normalization and cleaning) and another to load the deep learning model and perform the inference.

The MPAI Store is responsible for providing these AIWs and their AIMs. The AIMs are stored in the end-user devices, at “AIM Storage”. This process is orchestrated by the controller which may use the different components (e.g., “communication”, “global store” and “access”) to communicate with the backend and perform desired operations.

It is worth mentioning that multiple instances of AIWs can exist at the end-user device, and they can have the same or different objectives and may work with the same or different data, data sets or versions.

## MPAI-AIH Requirements

Requirements are categorized as: Legal Requirements, Functional Requirements and AIH-Related Requirements.

For the identification of the requirements, the following notation was used:

* **Requirement Identification Number (RIN)**: a unique identification number that identifies each of the requirements in the requirements identification process. This number will use the following format: [R-LEG|FUN|NFUN-XXXX]. LEG = Legal Requirement | FUN = Functional Requirement | NFUN = Non-Functional Requirement;
* **Requirement Title**: a title that resumes and identifies the requirement;
* **Requirement Description**: a detailed description of the requirement;
* **Requirement Type** (Optional, Mandatory): the type of requirement, including its obligation to be included or not. This is also related to the way the requirement should be expressed (using expressions such as “shall”, “should”, “may”, or others);
* **Requirement Dependencies or Relations**: identifies any relations or dependencies to other requirements. This contains the RINs of other existing requirements;
* **Requirement Notes**: Any other notes that might be relevant for the description of the requirement itself.

Each of the requirements is identified using the following table format.

|  |  |
| --- | --- |
| **RIN** | Requirement Title |
| Requirement Description | |
| Requirement Type | Requirement Dependencies or Relations |
| Requirement Notes | |

# Legal Requirements

This chapter lists the Rights and Duties of Users, Operators, and Participants in AIH:

|  |  |
| --- | --- |
| **R-LEG-0001** | Rights and Duties of Operators |
| The users that will use the AIH platform shall be informed, in a readable format of the following information:   * The jurisdiction of   + A AIH instance   + Multiple connected AIH instances. * Which data are collected:   + Biometric   + Behavioural   + Emotional * For what purposes   + Advertisements   + Diagnostics   + Statistics * By whom:   + The local platform   + Via access to Smart Contract. | |
| MANDATORY |  |
|  | |

1. Rights of
   1. User
      1. To be made aware (in a machine-readable form) of
         1. The jurisdiction of
            1. A AIH instance
            2. Multiple connected AIH instances.
         2. Which data are collected:
            1. Biometric
            2. Behavioural
            3. Emotional
         3. For what purposes
2. Advertisements
3. Diagnostics
4. Statistics
   * + 1. By whom:
          1. The local platform
          2. Via access to Smart Contract.
     1. To give informed consent to the applicable law.
     2. To be anonymous.
     3. To have privacy preserved
     4. To have properly represented the User Metadata items that the user wants to present.
     5. To register an objection to a User Metadata Item not properly represented in the AIH.
   1. Operator
      1. Manages privacy data of User, according to characteristics, e.g. nationality, maintaining compliance with their country’s laws.
5. Duties of:
   1. User
      1. To respect the laws of their jurisdiction.
      2. To respect the laws of the jurisdiction of the AIH.
      3. To accept the legal requirements of the AIH.
      4. To act in good faith in the AIH.
      5. To avoid harm to other Users.
      6. To declare whether their displayed identity corresponds to a real human or is fictitious.
   2. Operator

1. Governance.

# Functional requirements

## AIH-oriented requirements

1. A AIH instance shall be persistent: i.e., a data shall exist independently of time and place and run continuously.
2. A AIH shall offer synchronous experiences: i.e., Users shall be able to interact with the AIH in real-time.
3. A AIH shall be available: i.e., the Metaverse Manager shall be able to control the theoretical cap to the number of
   1. Users logging on simultaneously.

## User-oriented requirements

1. A User may obtain the right:
   1. To create an Identity and associated Environment within the AIH.
   2. To populate the Environment with Health data.
   3. To export their Data from the Metaverse into another AIH in the same or different AIH, or the real world

# References

[1] McMahan, Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. "Communication-efficient learning of deep networks from decentralized data." In *Artificial intelligence and statistics*, pp. 1273-1282. PMLR, 2017.

[2]Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H.R., Albarqouni, S., Bakas, S., Galtier, M.N., Landman, B.A., Maier-Hein, K. and Ourselin, S.; The future of digital health with federated learning; NPJ digital medicine, 3(1), pp.1-7; 2020.

TBD:

Study workflows for potential standardization. Data ingested, workflows, etc.

Identify data types acquired;

Identify processing on the data, either as single entity or workflow;

Identify new processing that can leverage the architecture.

New research are: “System aspects of federated learning implemented in the system and AIF in particular”

Potential requirement to have smart contracts executed in multiple BC.

“MPAI-AIH Use-cases and functional requirements” -> change the name of the document.

# Annex A: API Description

## API: Mobile App <-> Back-end System

|  |  |  |  |
| --- | --- | --- | --- |
| **API entry point** | **Input Data** | **Output Data** | **Description** |
| /register | (username,password,type,details) | [userid] | Registers a new user on the backend system |
| /signin | (username,password) | [jwt] | Signs in the user on the backend system and creates an authentication and authorization token (JWT) |
| /storeData | (userData) | [] | Store health data on the Global Secure DataVault |

## API: Back-end System <-> Third Parties

|  |  |  |  |
| --- | --- | --- | --- |
| **API entry point** | **Input Data** | **Output Data** | **Description** |
| /retrieveData | () | [userData] | Retrieve health data from the Global Secure DataVault |

## API: Back-end System <-> B&DLT

|  |  |  |  |
| --- | --- | --- | --- |
| **API entry point** | **Input Data** | **Output Data** | **Description** |
| /signSmartContract | (userid,thirdPartyid) | [] | Registration of a smart contract describing authorization by third-party entities towards user's data. |
| /checkValidity | (userid,thirdPartyid) | [] |  |

1. [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)
3. [↑](#footnote-ref-3)