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

**Public document**

|  |  |
| --- | --- |
| **N1003** | 2022/12/21 |
| **Source** | Requirements (AIH) |
| **Title** | MPAI-AIH Use Cases and Functional requirements WD0.0.1 |
| **Target** | MPAI-27 |

Table of Contents

[1 Introduction 2](#_Toc122358806)

[2 Scope 3](#_Toc122358807)

[3 Definition of Terms 3](#_Toc122358808)

[4 Architecture of the AI Health Secure Platform 4](#_Toc122358809)

[4.1 Actors 5](#_Toc122358810)

[4.2 Services 5](#_Toc122358811)

[5 Healthcare use case 6](#_Toc122358812)

[5.1 User health data collection 7](#_Toc122358813)

[5.2 Background process 7](#_Toc122358814)

[5.3 Access to healthcare data 7](#_Toc122358815)

[5.4 User stories 8](#_Toc122358816)

[5.5 Intelligent processing of healthcare data 9](#_Toc122358817)

[5.6 Verification of AIH data access 10](#_Toc122358818)

[6 Intelligent Computational Service Organization 10](#_Toc122358819)

[6.1 Centralized Services 10](#_Toc122358820)

[6.2 Federated Learning Mobile applications 10](#_Toc122358821)

[6.2.1 Federated Learning in the Healthcare Application Use Case 10](#_Toc122358822)

[6.2.2 Federated Learning Processes 10](#_Toc122358823)

[6.3 MPAI-AIH Federated Components 11](#_Toc122358824)

[7 MPAI-AIH Requirements 12](#_Toc122358825)

[7.1 Methodology 12](#_Toc122358826)

[7.2 Legal Requirements 12](#_Toc122358827)

[7.3 Functional requirements 14](#_Toc122358828)

[7.3.1 List of specific functional requirement areas 14](#_Toc122358829)

[7.3.2 AIH-oriented requirements 14](#_Toc122358830)

[7.3.3 User-oriented requirements 14](#_Toc122358831)

[7.4 API Requirements 14](#_Toc122358832)

[7.4.1 API: Platform Front-end <-> AIH Platform Back-end 14](#_Toc122358833)

[7.4.2 API: AIH Platform Front-end <--> AIH Platform Back-end (Federated Learning) 15](#_Toc122358834)

[7.4.3 API: Back-end System <-> Third Parties 15](#_Toc122358835)

[7.4.4 API: Back-end System <-> Blockchain 15](#_Toc122358836)

[8 References 15](#_Toc122358837)

[9 Annex A: API Description 16](#_Toc122358838)

[9.1 API: Mobile App <-> Back-end System 16](#_Toc122358839)

[9.2 API: Back-end System <-> Third Parties 16](#_Toc122358840)

[9.3 API: Back-end System <-> B&DLT 17](#_Toc122358841)

# Introduction

In recent years, Artificial Intelligence (AI) and related technologies have been applied to a broad range of applications, have started affecting the life of millions of people and are expected to do so even more in the future. As digital media standards have positively influenced industry and billions of people, so AI-based data coding standards are expected to have a similar positive impact. Indeed, research has shown that data coding with AI-based technologies is generally *more efficient* than with existing technologies for, e.g., compression and feature-based description.

However, some AI technologies may carry inherent risks, e.g., in terms of bias toward some classes of users. Therefore, the need for standardisation is more important and urgent than ever.

The international, unaffiliated, not-for-profit MPAI – Moving Picture, Audio and Data Coding by Artificial Intelligence Standards Developing Organisation has the mission to develop *AI-enabled data coding standards*. MPAI Application Standards enable the development of AI-based products, applications, and services.

As a part of its mission, MPAI has developed standard operating procedures to enable users of MPAI implementations to make informed decision about their applicability. Central to this is the notion of Performance, defined as a set of attributes characterising a reliable, trustworthy, and ethical implementation.

For the aforementioned reasons, to fully achieve the MPAI mission, Technical Specifications must be complemented by an ecosystem designed, created and managed to underpin the life cycle of MPAI standards through the steps of specification, technical testing, assessment of product safety and security, and distribution.

In the following, Terms beginning with a capital letter are defined in **Error! Reference source not found.** if they are specific to this project and in **Error! Reference source not found.** if they are common to all MPAI Standards.

The MPAI Ecosystem is fully specified in [**Error! Reference source not found.**]. It is composed of:

* MPAI as provider of Technical, Conformance and Performance Specifications.
* Implementers of MPAI standards.
* MPAI-appointed Performance Assessors.
* The MPAI Store which takes care of secure distribution of validated Implementations.
* End Users who use Implementations from the MPAI Store.

*Figure 1* depicts the MPAI-AIF Reference Model under which Implementations of MPAI Application Standards and user-defined MPAI-AIF conforming applications operate.

An AIF Implementation allows execution of AI Workflows (AIW), composed of basic processing elements called AI Modules (AIM).

MPAI Application Standards normatively specify Syntax and Semantics of the input and output data and the Function of the AIW and the AIMs, and the Connections between and among the AIMs of an AIW.

In particular, an AIM is defined by its Function and data, but not by its internal architecture, which may be based on AI or data processing, and implemented in software, hardware or hybrid software and hardware technologies.



Figure 1 - The AI Framework (AIF) Reference Model and its Components

MPAI defines Interoperability as the ability to replace an AIW or an AIM Implementation with a functionally equivalent Implementation. MPAI also defines 3 Interoperability Levels of an AIW executed in an AIF:

*Level 1* – Conforming tothe MPAI-AIF Standard.

*Level 2* – Conforming to an MPAI Application Standard.

*Level 3* – Assessed by a Performance Assessor.

MPAI offers Users access to the promised benefits of AI with a guarantee of increased transparency, trust and reliability as the Interoperability Level of an Implementation moves from 1 to 3. Additional information on Interoperability Levels is provided in **Error! Reference source not found.**.

# Scope

Artificial Intelligence for Health data (MPAI-AIH) is an MPAI project supporting collection, AI-based processing, access to health data, and distribution of updated of AI Models. The functionalities of the project are supported by the AI Health Secure Platform whose technical characteristics are defined from the technical point of view considering generally applicable legal concerns.

The motivation for the development of this standardization comes from the AIM Health Portuguese project. This document addresses the technical as well as legal specifications of an AIH System. This document addresses the technical as well as legal specifications of an AIH System

# Definition of Terms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| AI Framework (AIF) | The environment where AIWs are executed. |
| AIH) Data | Health-related data entering the AIH Platform. |
| AI Module (AIM) | A processing element receiving AIM-specific Inputs and producing AIM-specific Outputs according to according to its Function. An AIM may be an aggregation of AIMs. |
| AI Workflow (AIW) | A structured aggregation of AIMs implementing a Use Case receiving AIW-specific inputs and producing AIW-specific inputs according to its Function. |
| AIH Platform | The ICT platform offering AIH services. |
| AIH Platform Back-end | The part of the AIH Platform collecting, storing, and processing health data, and carrying out Federated Learning functions on the AI Models from the Front-end. |
| AIH Platform Front-end | The end-user devices collecting and processing personal health data and updating the AI Models received from the AIH Platform Back-end. |
| AIH Processing Taxonomy | The recognised set of processing that the AIH Platform Back-end can execute. |
| Blockchain | A shared immutable ledger stored on a peer-to-peer network of computers. |
| Data Anonymisation | A mechanism that protects private or sensitive data by erasing or encrypting identifiers that connect an individual to stored data. |
| Data De-identification | A mechanism that breaks the link between data and the individual with whom the data is initially associated. It is a type of data anonymization. |
| End User | The holder of an AIH Platform Front-end instance. |
| External Source | A platform other than the AIH Platform from which the AIH Platform Back-end may collect subsidiary data for the integration of relevant information for health-related predictions. |
| Secure Data Vault | A repository that holds several types of data in a encrypted format. Access to the data is controlled by the user through the presentation of appropriate credentials. |
| Smart Contract | A Program stored on a Blockchain that runs when activated by an external entity, e.g., a User or another Smart Contract. |
| Provenance | A record trail that accounts for the origin of a piece of data (in a database, document or repository) together with an explanation of how and why it got to the present place |
| Third Party Entity | An Entity – excluding the AIH System and the End-User – accessing the AIH Platform Back-end to process some stored AIH data. |
| User | Any entity involved in or accessing the AIH Platform. |

# Architecture of the AI Health Secure Platform

The overall architecture of the AI Health Secure Platform comprises a set of different systems, specific distributed services and APIs as depicted in Figure 2.

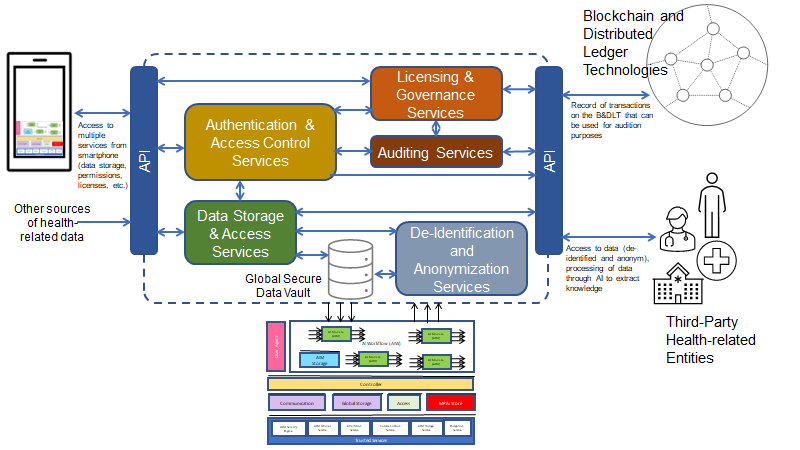


Figure 2 - Reference Model of MPAI-AIH Secure Platform

## Actors

The AI for Health data system identifies and recognizes the following different Users/systems.

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 User*: 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 smart contract between that third party and the end-user. The smart contracts used are based on approved templates that are verified for legal compliance before release.
3. *External Data Sources*: represent platforms other than the AIH Platform from which the AIH Platform Back-end 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 components and services that are depicted in Figure 2. These are:

1. The *front-end system*, a smart device (e.g., a smartphone) 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 from the End User. All the collected data will be securely stored in the End User’s smartphone using a “Secure Data Vault” whose access is controlled by the End-User (Figure 3). 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 the local machine learning data analysis.

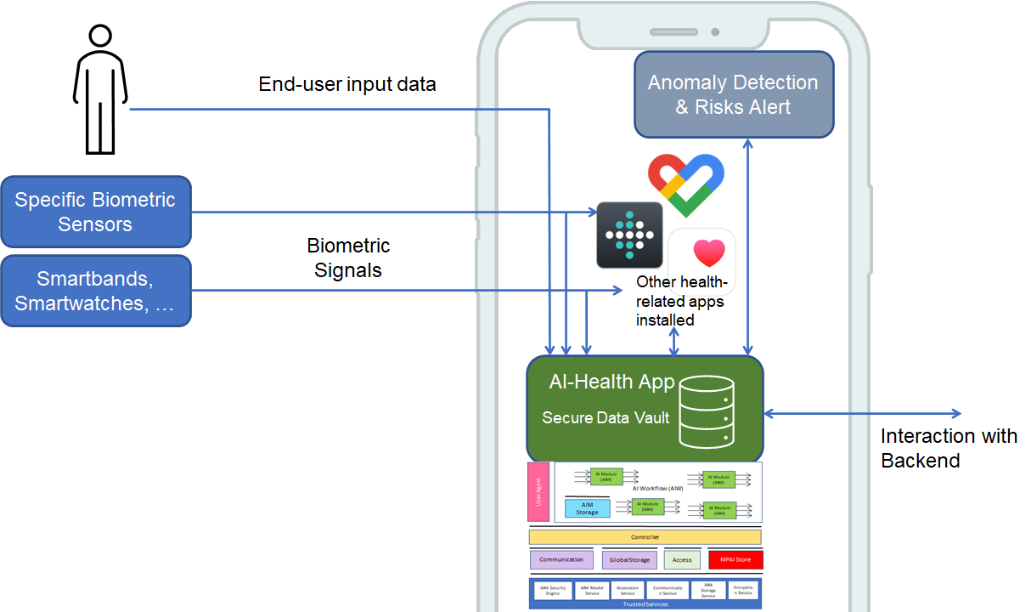


Figure 3 – Front-end architecture of the MPAI-AIH Secure Platform

1. The *back-end system*, composed by a set of tools that implement the necessary services to securely store, de-identify and anonymize data, control entity authentication and access control to data, and license and audit the access to AIH data on the AIH Platform Back-end. This system gathers anonymized data from all the various sources (the End Users providing itit and other other External Sources) and acts as a broker gateway between the entities requesting access to the AIH data and those who will provide it. The backend grants the rights without referring to the identity of the End Users who have provided the data. Of course, the backend may not grant the entity the right to make processing that the End User did not grant to the backend.
2. *Blockchain and Distributed Ledgers* (B&DLT) enable the transparency and auditability of the system. Every provision of and access to health-related data will require the emission of a license in the form of a Smart Contract that will be stored on the B&DLT. This smart contract will contain information about
   1. The parties, e.g., the End-User storing AIH data and the AIH Platform Back-end.
   2. The AIH data involved (to be stored or accessed).
   3. The type of processing present in the AIH Processing Taxonomy performed.
   4. The access conditions (e.g., timeframe and the End-User’s permissions).
3. The *AI services* displayed by the AIH Platform Back-end can be used to directly treat and process the data on the device to extract the specific knowledge sought by the End-User or third parties with contracted rights. These services are 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.

# Healthcare use case

This section presents a simple use case that describes the AI Health Secure Platform of Figure 2 and its usage.

## User health data collection

* The End User signs into the AIH Platform, which is equipped with the necessary security features and initializes the End-User Secure Data Vault.
* The End User configures the AIH 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 AIH app starts collecting relevant data from the End User and securely stores it locally on the Secure Data Vault.
* The End Users are requested for permission to export specific subsets of their AIH data to the AIH Platform and potentially to other Third-Party Users. The End-Users analyse the request and may give permission for such a AIH data export. If the End User gives permission to access their data, the back-end will create a Smart Contract to be accepted by the End User and AIH data will be collected via an API.
* The AIH data from all the End Users participating in the AIH Platform that have given their permission is collected into the AIH Platform Back-end where it will be De-identified and Anonymized.

## Background process

Background processes may take place in the AIH Platform Front-end and in the AIH Platform Back-end. They include:

1. Housekeeping services: a set of ancillary processes tasked with the proper functioning of the system internals. 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 AIH Platform Back-end based on the AIH Platform Back-end 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 External 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 metadata with the appropriate level of detail.
* 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 AIH Platform Back-end to serve a request by an External Party. The following assertion acts as a paradigm for what is specified in the SC: the back-end grants the External Party the right to process the designated data with proc1, proc2, etc.
* 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.

## User stories

A set of user stories has been defined as listed below. These user-stories are expressed in terms of an actor that interacts with the system. The way these user stories are expressed obeys the following template: “As an <ACTOR> I want to do <ACTION> to <RESULT>”. Each of the expressed user stories also contains an acceptance criteria defined to specify how the user story should be validated.

*Table 1. List of user-stories for the End-User AIM-Health Data Vault App*

|  |  |  |
| --- | --- | --- |
| **As a** | **End-User** | |
| **I want to…** | **To…** | **Acceptance Criteria** |
| Import health data from other health apps on my smartphone | Have the data in a single place | Assure that user can:   * Enter the app * Select the data source application * Select the data import operation * Visualize the data that was imported with success |
| Save my health data in a secure way | Avoid privacy breaches and non-authorized accesses | Assure that the user can:   * Enter the app according to an authentication mechanism * If the authentication credentials are appropriate the access is granted * If not, the access is denied |
| Preserve and protect his/her health data | Prevent non-authorized access | Assure that only the correct user can:   * Enter the app using their credentials * Access its data |
| Import health data from health sensors connected to the smartphone | Save health data in a secure repository | Assure that the user can:   * Select the connected sensors * Establish the data flow with the sensors * Select the data that is going to be imported from the sensors * Visualize the imported data |
| Authorize the access to specific stored health data | Concede the anonymized access to health data | The user:   * Properly authenticated and authorized on the application * Receives access requests to its data repository * Accepts the requests and sends data |
| Establish temporary health data access contracts | Authorize who can access my health data and under which conditions | The user:   * Properly authenticated and authorized on the application * Receives requests to access its health data, to perform a set of actions, for a given period of time, from some entity * The user, authorizes or not the access |
| Visualize who has access to my health data | Audit the accesses conducted using a set of specific filters (entity name, date, type of data, among others) | The user:   * Properly authenticated and authorized on the application * Can consult the list of permissions that where given to third-party entities according to multiple filters |
| Check for abuses on its health data | Audit if the contracts are still valid | The user:   * Properly authenticated and authorized on the application * Consults the list of contracts and checks for its validity |
| **As a** | **Thir-Part User** | |

## Intelligent processing of healthcare data

There are two categories of intelligent processing of healthcare data: processing which happens in the front-end, and in the AIH Platform Back-end at the request of an External Party. Intelligent processing of healthcare data follows best practices and 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 the computational organization best adapted to the task at hand in 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 AIH data includes:

1. an AIW (AI Workflow) is selected to process the AIH data, and the AIMs (AI Modules) load the AIH data as needed and may store the AIH data in the “AIM Storage”.
2. The AIW orchestrates the execution of the AIMs, which operate over the AIH data. All of these AIMs 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 (Figure 2).

## Verification of AIH data access

This is a system mechanism that allows users to verify their AIH data access. To accomplish this, the users access their AIH data to verify the logs of processing of their AIH data.

# 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 the AIH Platform 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.

## 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 in the 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. [2]

### 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 by 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 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 fine-tune in their local model to improve the accuracy and customization of the model for herself/ himself.

## MPAI-AIH Federated Components

The AIH Platform Font-ends and External Parties communicate with the AIH Platform Back-end 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

## Methodology

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 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.
5. To give informed consent to the applicable law.
6. To be anonymous.
7. To have privacy preserved
8. To have properly represented the User Metadata items that the user wants to present.
9. To register an objection to a User Metadata Item not properly represented in the AIH.
10. Duties of User
11. To respect the laws of their jurisdiction.
12. To respect the laws of the jurisdiction of the AIH.
13. To accept the legal requirements of the AIH.
14. To act in good faith in the AIH.
15. To avoid harm to other Users.
16. To declare whether their displayed identity corresponds to a real human or is fictitious.
17. Rights of Operator
18. Duties of Operator
19. Manages privacy data of User, according to characteristics, e.g. nationality, maintaining compliance with their country’s laws.
20. Governance.

## Functional requirements

### List of specific functional requirement areas

1. Initial list of AIMs with functionality and I/O data
2. AIH Platform Front-end biometric sensor calibration
3. Terms and conditions in a Smart Contract
   1. Front-End to Back-End
   2. Third Part to Back-End
4. Initial list of processing requested by Third Parties
5. Initial list of data type possible target of processing
6. Interface between
   1. Auditing and Licensing & Governance Services
   2. Authentication & Access Control Services and Auditing Services
   3. Authentication & Access Control Services and Licensing & Governance Services
   4. Authentication & Access Control Services and Data Storage & Access Services
   5. Data Storage & Access Services and De-Identification & Anonymization Services
   6. Back-End and the AIF

### 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 AIH into another AIH in the same or different AIH, or the real world

## API Requirements

The system will use REST API interfaces that will provide data access. A special API acts as the interface between the AIH Platform Back-end data and the AI modules for using and processing data. Data may also be collected from External Sources such as public services and third-party entities, using other specialized APIs.

### API: Platform Front-end <-> AIH Platform Back-end

This describes the API that is exposed by the AIH Platform Back-end to the AIH Platform Front-end. This API will provide the necessary services to register, authenticate and control access of the user in the AIH Platform. Moreover, this API will also provide the mechanisms for handling AI data – data storage, permissions for data usage, and data usage auditing.

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

### API: AIH Platform Front-end <--> AIH Platform Back-end (Federated Learning)

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

This describes the API that is exposed by the AIH Platform Back-end to the Third Parties. This API will provide the necessary services to register, authenticate and control access of third-party users in the AIH Platform. In addition, this API will also provide the service of retrieving the user's health data requested by the Third part.

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 describes the API that is exposed by the Blockchain to the AIH Platform Back-end. This API will provide the necessary services to register and check the validity of the smart contract between the End User and the Third Part.

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

# 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.

# Annex A: API Description

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API entry point** | **Method** | **Input Data** | **Output Data** | **Description** |
| /v1/account/create | POST | username  password  type=’End User’ | [Account] | Register a new End User on the AIH Platform Back-end |
| /v1/account/signin | POST | username  password | token | Sign in the End User  on the AIH Platform Back-end and create an authentication and authorization token (JWT) |
| /v1/storage/store | POST | [healthData] |  | Store health data of the End User on the Global Secure DataVault. Data must be encrypted from the End User before the insert. The format of the data is: {} |
| /v1/storage/get | GET |  | [healthData] | Retrieve health data of the owner End User  on the Global Secure DataVault |
| /v1/storage/delete | DELETE | healthDataID |  | Delete health data of the owner End User on the Global Secure DataVault, only the owner of the data can delete it |
| /v1/storage/update | PUT | healthData |  | Update health data of the owner End User on the Global Secure DataVault |
| /v1/audit/updateThirdPartEntityGrants | PUT | thirdparties,  dataAllowed |  | Update the grant of third parties authorized to access to the user health data of the Global Secure DataVault |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API entry point** | **Method** | **Input Data** | **Output Data** | **Description** |
| /v1/account/create | POST | username  password  type=’Third Part’ | [Account] | Register a new third part on the AIH Platform Back-end |
| /v1/account/signin | POST | username  password | jwt | Sign in the third part on the AIH Platform Back-end and create an authentication and authorization token (JWT) |
| /v1/storage/get | GET | [filterCriteria] | [healthData] | Retrieve health data on 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) | [] | Check if a third-party entity is allowed to get End User data, and register the event in the blockchain |