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

**Public document**

|  |  |
| --- | --- |
| **N1256** | 2023/06/14 |
| **Source** | Requirements (AIH) |
| **Title** | MPAI-AIH Use Cases and Functional Requirements WD0.4 |
| **Target** | MPAI Members |

Table of Contents

[1 Introduction 2](#_Toc137229880)

[2 Scope 3](#_Toc137229881)

[3 Definition of Terms 3](#_Toc137229882)

[4 Architecture of the AI Health Secure Platform 5](#_Toc137229883)

[4.1 Introduction 5](#_Toc137229884)

[4.2 Actors 5](#_Toc137229885)

[4.3 Services 6](#_Toc137229886)

[5 Healthcare use case 7](#_Toc137229887)

[5.1 User health data collection 7](#_Toc137229888)

[5.2 Background process 7](#_Toc137229889)

[5.3 Access to and process of AIH data 8](#_Toc137229890)

[5.4 User stories 8](#_Toc137229891)

[5.5 Verification of AIH data access 12](#_Toc137229892)

[6 AIH Federated Learning System 12](#_Toc137229893)

[6.1 Introduction 12](#_Toc137229894)

[6.2 Federated Learning Mobile applications 13](#_Toc137229895)

[6.3 Centralized Services 14](#_Toc137229896)

[7 Federated Learning in AI Health 14](#_Toc137229897)

[7.1.1 Federated Learning Processes 14](#_Toc137229898)

[7.2 MPAI-AIH Federated Components 15](#_Toc137229899)

[8 MPAI-AIH Requirements 15](#_Toc137229900)

[8.1 Functional requirements 15](#_Toc137229901)

[8.1.1 List of Smart Contract functional requirements 15](#_Toc137229902)

[8.1.2 List of Data Type and Usage requirements 16](#_Toc137229903)

[8.1.3 Federated Learning requirements 17](#_Toc137229904)

[8.2 API Requirements 17](#_Toc137229905)

[8.2.1 API: AIH Platform Back-end <-> Platform Front-end 17](#_Toc137229906)

[8.2.2 API: AIH Platform Back-end (Federated Learning) <--> AIH Platform Back-End 17](#_Toc137229907)

[8.2.3 API: AIH Platform Back-end <-> Third Party 17](#_Toc137229908)

[8.2.4 API: AIH Platform Back-end <-> Blockchain 17](#_Toc137229909)

[8.3 Data Management 18](#_Toc137229910)

[8.3.1 Data definition 18](#_Toc137229911)

[8.3.2 Data collection: 18](#_Toc137229912)

[8.4 Data Requirements 19](#_Toc137229913)

[8.4.1 Data accessibility: 19](#_Toc137229914)

[8.4.2 Data findability: 19](#_Toc137229915)

[8.4.3 Data reusability: 19](#_Toc137229916)

[8.4.4 Data interoperability: 19](#_Toc137229917)

[8.5 Example Case – UNDER WORK 19](#_Toc137229918)

[8.5.1 COVID-19 prediction using sensor data 19](#_Toc137229919)

[9 References 20](#_Toc137229920)

[Annex 1 - MPAI-wide terms and definitions 21](#_Toc137229921)

[Annex 2 - Notices and Disclaimers Concerning MPAI Standards (Informative) 24](#_Toc137229922)

[Annex 3 - The Governance of the MPAI Ecosystem (Informative) 26](#_Toc137229923)

[Annex 4 - API Requirements 28](#_Toc137229924)

[4.1 API: AIH Platform Back-end <-> Platform Front-end 28](#_Toc137229925)

[4.2 API: AIH Platform Back-end (Federated Learning) <--> AIH Platform Back-end 29](#_Toc137229926)

[4.3 API: AIH Platform Back-end System <-> Third Parties 29](#_Toc137229927)

[4.4 API: AIH Platform Back-end System <-> Blockchain 30](#_Toc137229928)

[Annex 5 - Data Types 31](#_Toc137229929)

[5.1 Aggregated Health Data Format 31](#_Toc137229930)

[5.1.1 Time-Series data 31](#_Toc137229931)

# 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 Table 1 if they are specific to this project and in *Table 2* if they are common to all MPAI Standards.

The MPAI Ecosystem is fully specified in [1]. 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 [2].

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.

# Scope

Artificial Intelligence for Health data (MPAI-AIH) is an MPAI project supporting advanced forms of health support provided by an end-to-end system to collect, AI-based process, access health data, and distribute AI Models updated via federated learning techniques. The functionalities of the project are enabled by the AI Health Secure Platform whose technical characteristics should comply with generally applicable legal frameworks such as the European GDPR.

The motivation for the development of this standard comes from the AIM Health Portuguese project.

# Definition of Terms

Table 1 - MPAI-AIH 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. |
| AIH Model Instance | A local End-User instance of the AIH model. |
| AIH Federated Learning System (FLS) | The system aggregating the data describing each of the AIH Model Instances for the creation the data describing a global Model for AIH System-wide distribution. |
| 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

## Introduction

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

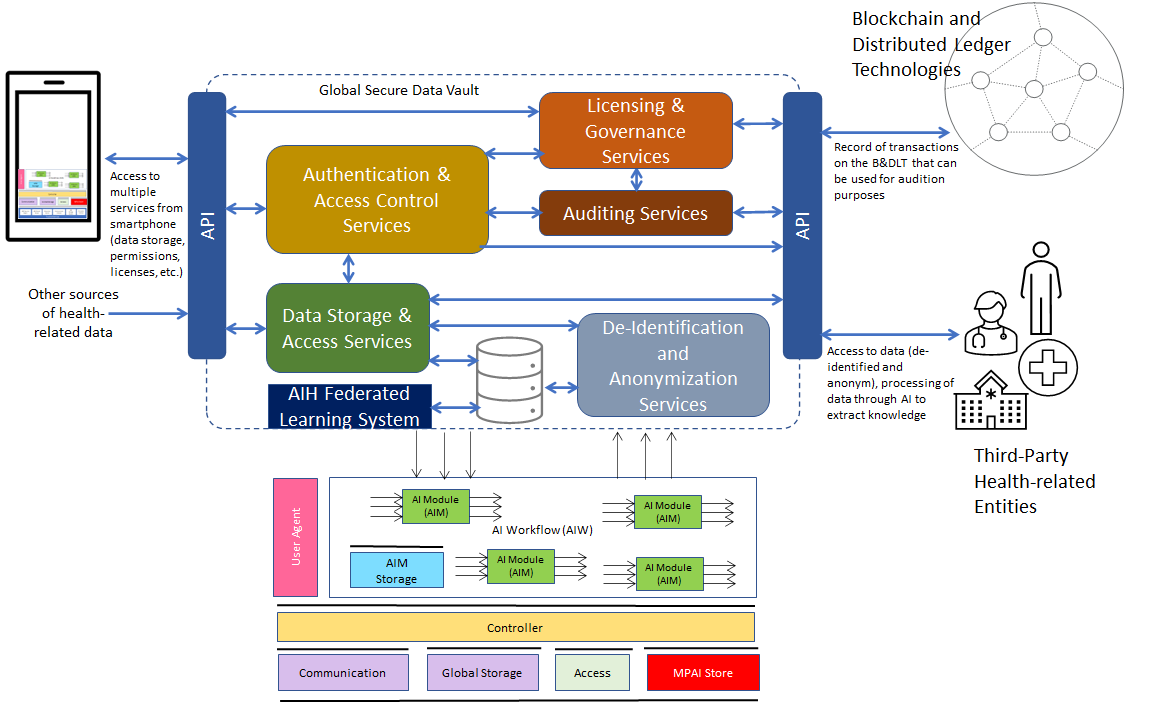


Figure 2 – Reference Model of AIH Secure Platform

A concise description of the AIH Secure Platform is the following:

1. End Users acquire health data using an AIH App running on a handset (AIH Frontend). Health data is AI-processed using the local AI Framework.
2. Data – either processed or not – are uploaded to the AIH Backend according to the terms of the Smart Contract between the End User and the AIH Backend that resides on a Blockchain and regulates the use that the AIH Backend and its Users can make of the uploaded data.
3. The Backend processes Health data using its local AI Framework.
4. Third-Party health-related entities may access the AIH Backend and request AI-processing on AIH Backend available AI data.
5. A Third-Party User may access Health data in the AIH Backend based on the terms of a Smart Contract between the AIH Backend and the End User.

## Actors

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

1. *AIH Frontend User (End User)*: a user whose health-related data are collected by the AIH Platform. The End User controls and audits the access of any Third-Party Entity 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 End User approval to do so.
2. *AIH Backend User (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 AI-based mechanism (or through the orchestration of multiple AI-based 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 are based on approved templates that are verified for legal compliance and technical security before release.
3. *External Data Sources*: represent platforms other than the AIH Platform from which the AIH Platform Backend 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 between the Sources and the AIH Platform Backend.

## Services

Figure 2 depicts the AI Health data system is composed of a set of distributed components and services:

1. The AIH *Frontend*, a smart device (e.g., a smartphone) application (AI-Health app) that is capable of:
   1. Capturing End User’s Health data, e.g., from Google Fit and Apple Health, and from external biometric sensors that capture Health data.
   2. Locally storing in a “Secure Data Vault” controlled by the End User (see Figure 3)
   3. AI processing health-related End User data using standard AIMs and AIWs which perform the computational operations of the End User’s health data, including transformations, training, and inferences.
   4. Alerting the End-User about any deviation of the value of the captured data that may be caused by disease.
   5. Uploading the processed AI Health data.
   6. Receiving processed Health data from the AIH Backend.

The AIH Frontend represents the personal gateway to the user-data and any external biometric sensors that capture health-related data, and the connection with the AIH-Backend.

Diagram

Description automatically generated

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

1. The *AIH Back-end*, composed of a set of tools that implement the necessary services to securely store, de-identify and anonymize data, control entity authentication and access to data, and license and audit the access to AIH data on the AIH Backend. This system gathers anonymized data from all the various sources (the End Users providing it and other External Sources) and acts as a broker gateway between the Third-Part 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 AIH 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 Backend.
   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 Backend 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.
4. The AI Federated Learning system (FLS) orchestrates the learning of a central model for medical diagnosis and prognosis, namely by working as a medical anomaly detection tool, receiving model weights data from the client models at each AIH Frontend and using it under the terms of the Smart contract that has been established between the End User and the AIH Backend. When an improved model is obtained by the FLS, this is distributed to the client models via a model update, as detailed in Section 6.

# Healthcare use case

This section describes the AI Health Secure Platform of Figure 2 and its usage as a use case of MPAI-AIH.

## 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.) and her preferences.
* The AIH app starts collecting relevant data from the End User and securely stores it locally on the Secure Data Vault.
* 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-User analyses the request and decides about permission for such an AIH data export. If the End User grants permission to access her data, the Backend will create a Smart Contract and AIH data will be collected via an API.
* All the AIH data with granted permission are 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 Frontend and in the AIH Backend. 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 and process of AIH data

This section deals with the processing done by the AIH Backend based on the AIH Backend policy and on-demand processing of health data requested by External Parties. A demand may involve transfer and processing of a huge amount of data. Note, 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 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 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 Party 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 a Smart Contract to process the data inside the smart phone.
* A request to process the data in the AIH Backend to serve a request by a Third-Party Entity is executed under the strict terms of the existent Smart Contracts t. The following assertion acts as a paradigm for what is specified in the Smart Contract: the AIH Backend grants the Third-Party Entity 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 properly orchestrated services) that can access the AIH Backend’s AIH data and perform some processing. Access to the service is based on a choice made from a service taxonomy used in Smart Contracts.
* The Third-Part Entity accepts a Smart Contract created by the B&DLT End Users. Access to AIH data is permitted if the Smart Contract is valid.
* Intelligent processing of health data is performed either via an appropriate AIWs or via FLS.

## User stories

This section presents a collection of stories illustrating the interaction between the End Users and Third-Party users with the AIH Platform. These user-stories are expressed in terms of an actor that interacts with the system. The way these user stories are expressed using the following template: “As an <ACTOR> I want to do <ACTION> to <RESULT>”. Each of the expressed user stories also contains an acceptance criterion 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*

|  |  |  |  |
| --- | --- | --- | --- |
| **User-Story** | **As a** | **End-User** | |
| **Number** | **I want to…** | **To…** | **Acceptance Criteria** |
| 1.1 | 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 |
| 1.2 | 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 |
| 1.3 | Preserve and protect his/her health data | Prevent non-authorized access | Assure that only the correct user can:   * Enter the app using the credentials * Access its data |
| 1.4 | 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 |
| 1.5 | 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 |
| 1.6 | 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 |
| 1.7 | 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 |
| **User-Story** | **As a** | **Third-Part User** | |
| **Number** | **I want to…** | **To…** | **Acceptance Criteria** |
| 2.1 | Access the health data catalogue on the system | Select and filter the type of data that interests my needs | Assure that the user can:   * Enter the system * Look at the different types of data available * Filter data according to its preferences |
| 2.2 | Register a new organization on the system | Enable organizations access to the system | Assure that a user can:   * Access the web interface * Proceed with the organization registration * Visualize the organization registration results |
| 2.3 | Register a new user on the system | Enable users to access the system | Assure that a user can:   * Access the web interface * Proceed with its own registration * Visualize the registration results |
| 2.4 | Authenticate on the system | Access the system | Assure that a user can:   * Access the web interface * Present the access credentials * Verify the authentication result |
| 2.5 | Request health data from the system | Obtain data from the system | Assure that a user can:   * Access the web interface * Present the access credentials * Verify the authentication result * Select the data * Request the data download * Create a usage license * Download the data file |
| 2.6 | Create a license to use data | Establish a usage license to use the data | Assure that:   * Request is dully authorized * Availability of the requested data * Create a license that bounds End-users, data and requester * Register the license on the blockchain * Check the result of the license creation |
| 2.7 | Download the user health data | Access the health data | Assure that:   * Data is licenced Data is available * Check the result of the download operation |
| 2.8 | Train or update an AI model collaboratively, using End-users' data | Build/Improve an accurate model | Assure that:   * Data is licenced * Data is available * Define proper AIW and AIMs and publish info on MPAI store * Certificate the validity of the request * Requester can retrieve the trained model |

## Verification of AIH data access

This is a system mechanism that allows End users to verify who has accessed their AIH data. To accomplish this, users may access their AIH data to verify their AIH data processing logs.

# AIH Federated Learning System

## Introduction

This section describes how Health data are processed to extract knowledge embedded in collected Health data.

Because of the system architecture described above, intelligent processing of Health data may happen at two different locations: at the AIH Frontend and at the AIH Backend, at the request of a Third-Party User. Some use cases trigger processing at the AIH Frontend, in response to the End User’s request and interaction, whereas overall system-wide processing services reside in the AIH Backend.

Intelligent processing of Health data follows best practices and state-of-the-art machine learning techniques. It must be technically and socially robust, that is, accurate and reproducible, able to deal with and inform about possible failures, inaccuracies, and errors, aware of the potential repercussions of false positive (and, respectively, negative) responses, 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.

In this view, the main points to be addressed are:

* 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, and the use of algorithmically unbiased models.
* Efficient Implementation Architecture: This addresses the search for the computational organization 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 Health 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 processing of large Health datasets is done in such a way as to preserve the privacy of individuals, namely via the adoption of anonymization and pseudonymization techniques and the use of security-preserving communication and storage methods.
* Knowledge Sharing: This point focuses on how learning from one End user can be transferred and aggregated with learning from another End user, while maintaining End user’s privacy.

## 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.

In practical terms, the processing of AIH data includes:

1. An AI Workflow (AIW) is selected to process the AIH data, 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 the AIMs are downloaded or updated from the “MPAI Store”.
3. The data is processed by the AIM FLS (see Figure 3).

Diagram

Description automatically generated

Figure 3 - Federated Learning training process.

## Centralized Services

This section discusses the centralized learning processes and services offered by the Federated Learning System (FLS) in the AIH Backend. Chief among those are centralized server-client architectures driven by a high-resource server in charge of controlling and distributing intelligent models to client processing.

# Federated Learning in AI Health

Healthcare may leverage FLS 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 improved AI models from information of locally built models (models evolved on End User’s Health data) collected from a pool of devices without leaking personal data. Federated learning models can provide improved data diversity by gathering data from various locations and use cases (e.g., distributed End users, 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 are 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, as the client’s data sets grow, it can be incrementally trained for robustness and adjustment. For example, End-users may have external devices connected to their smartphones which periodically exchange vital information that enables to train/adjust their local machine learning models that issue anomaly alerts. The End user’s model can be incrementally improved and extended by aggregation the knowledge extracted with other models, that is, from different End users. Since, for security, transparency and trustworthiness, no exchange must be permitted between End users’ devices, these improvements are gained by the uploading of model’s weights to the AIH Backend FLS, that aggregates this local knowledge in a new improved global model, that can, in turn, be downloaded by the End users app.

#### 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 Frontend and Third-parties communicate with the AIH Platform Backend using secure APIs, to provide and request data or to request specific AI processing over the AIH health data repositories.

The federated components (locally updated models in the AIMs of the End User’s AIWs) are in the AIH Frontends and are instances of the standard AIMs (and AIWs) that handle healthcare data. For example, AIW service built to detect the possibility of COVID-19 infection by a user may display multiple AIMs: to train, tune and validate the model, make inferences, and make use of functionalities to select, load and pre-process (e.g., apply normalization and cleaning) data. On the other hand, the FLS uses AIWs with different AIMs to further train and update a model to be distributed in the next round.

The MPAI Store is responsible for providing these AIMs. As specified in [2], 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 the desired operations.

It is worth mentioning that multiple instances of AIWs can exist on 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

This chapter provided the functional requirements of the data types exchanged by subsystems of the AIH Platform

The AIH System processes critically important personal data with new AI-enabled technologies. Respondents to the MPAI-AIH call for Technologies are requested to explain how the technologies proposed comply with generally applicable legal concerns such as the European GDPR.

## Functional requirements

### List of Smart Contract functional requirements

A Smart Contract shall express the terms and conditions between

1. An End User and the AIH Backend
   1. End User ID
   2. Health Data ID
   3. Health Data Type ID
   4. Date of issuance of Smart Contract
   5. Duration of Smart Contract
   6. List of permitted Processing Types
   7. List of permitted Export Types (Seen from the End User)
2. The AIH Backend and a Third Part Entity #1.
   1. Third-party ID
   2. Health Data ID
   3. Health Data Type ID
   4. Date of issuance of Smart Contract
   5. Duration of Smart Contract
   6. List of permitted Processing Types
   7. List of permitted Export Types (seen from the AIH Backend)
3. The AIH Backend and a Third Part Entity #2.
   1. Third-party ID
   2. Health Data ID
   3. Health Data Type ID
   4. Date of issuance of Smart Contract
   5. Duration of Smart Contract
   6. List of permitted Processing Types.
   7. List of permitted Export Types (seen from the Third-party User)

### List of Data Type and Usage requirements

The sensors generating data are generally calibrated by the AIH Frontend. The calibration information should be specified along with the sensed data.

The list of currently considered requirements is:

1. *Historical User Health Data*, possibly represented in a markup language or JSON for End User’s
   * medical history
   * lab results
   * other information

that can be used to predict disease and inform treatment decision.

1. *Time series data*: Vital sign measurements, such as heart rate and blood pressure, that are collected over time, and can be used to predict disease and monitor treatment progress.
2. *Sensing data*: This includes data from sensors in wearable devices (such as smartwatches, fitness trackers, continuous glucose monitors, wearable shirts, etc.) which can be used for
   * Monitoring vital signs, tracking activity levels, and detecting falls, and can be used to detect abnormal vital sign patterns, such as arrhythmias or hypertension, and to monitor the effectiveness of a treatment.
   * Tracking activity levels, such as steps taken, distance traveled, and calories burned.

This information can be used to monitor physical activity and to promote healthy behaviors. The data format should include the data types and their units of measure.

1. *Geolocation data*: This includes information about the geographic location of End Users, such as their latitude and longitude. This type of data can be used in conjunction with other data types to identify spatial trends and clusters of disease and inform public health interventions and policies. Data considered: latitude (degree, minute, second, hundreds of second), longitude, height above sea level (m).
2. *Social media data*: This includes chats, posts, comments, and other related data. Social media data can be used to understand the public perception of health-related topics, identify trends in health-related behaviors, and monitor the spread of disease or misinformation. This data may be textual (“string”), visual (image, video), and audio.
3. *Textual health data*: This includes unstructured data such as clinical notes, medical literature, and patient-generated data such as text messages, emails, and other types of unstructured data.
4. *Health-related video data*: For example, self-taken videos used for remote demonstration of a particular physical health situation.
5. *Health-related image data*: These include data from the End-User, e.g., self-taken images used for remote demonstration of a particular health situation.
6. *Health-related audio data*: This includes speech and audio recordings, such as an End User speaking, or coughing, which can be used for anomaly prediction and other medical applications (.wav format is preferred).
7. *Genomic data*: This includes DNA sequencing data and other types of genetic information that can be used to predict disease risk and inform personalized medicine.

### Federated Learning requirements

The format of the AIH Frontend’s Neural Network Model used throughout the AIH Platform.

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

Further details of this API are provided in Annex 4 API Requirements.

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

This describes the API that is exposed by the AIH Platform Back-end to other instances of the AIH Model.

### API: AIH Platform Back-end <-> Third Party

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.

Further details of this API are provided in Annex 4 API Requirements.

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

Further details of this API are provided in Annex 4 API Requirements.

## Data Management

### Data definition

Data in AIM Health Platform is stored in the Secure Data Vault, and only the owner of the data is allowed to access and decrypt the data. A secure envelope must be used, containing the user’s health data records. The envelope format is independent of the data it contains. This envelope is represented in a JSON data structure:

* “timestamp”: contains the date and time when the health value was captured;
* “healthdata\_type”: indicates the type of measurement that this data represents. The definition of data types and its usage are illustrated in Subsection [7.6;](bookmark://_Types_of_Health)
* “source”: indicating the source of the health value. It ranges over different values, such as user-introduced health point, measurement obtained by an external sensor, or by any health-related application installed on the smartphone.
* “healthdata”: an array of specific health ‘data’ values.
* “data”: one health data value captured and encrypted with the appropriate data vault encryption key (see Figure below).

Graphical user interface, application

Description automatically generated

An example in JSON for this data structure:

{

"timestamp": "2022-12-15 12:38:07.369915",

"healthdata\_type": "TEMP",

"source": "Apple Watch",

"healthdata": [

{

"data": "OLNdQOZXkOwcZtGpkNzZfw=="

},

{

"data": "GaznHQvd2ePRNG3gPljOHQ=="

}

]

}

### Data collection:

All data to be collected shall be treated in accordance with legal and ethical standards and requirements in the respective countries of data collection. Moreover, personal data will be made available to Third-parties only upon permission of the user after receiving informed consent on data usage and maintenance from the Third-party that is requesting it. Every request of authorization permission and concession is tracked in the blockchain using specific function of the smart contracts, as well as every time a third party withdraws data after it has been authorized. Thus, this information is saved in an immutable and secure way, ensuring transparency between the parts. Any participant providing personal data for Third-parties must be allowed to withdraw the permission to access their data at any time.

## Data Requirements

All data to be collected shall abide to the following categories of requirements: Accessibility, Findability, Reusability and Interoperability.

### Data accessibility:

* Datasets with dissemination level "confidential" will not be shared due to privacy issues. Personal and health data will not be made public but can, depending on the terms of the Smart Contract, be disclosed to Third-parties.
* Anonymized and aggregated data may be disclosed as open data.

### Data findability:

* Third Parties can find personal or health data depending on the terms of the Smart Contract.
* Open data will be made findable.

### Data reusability:

Specific Creative Commons Licenses (e.g., CC BY or CC0) can be issued for each deposited dataset, defining either open or restricted access, as long as the terms of the Smart Contract are obeyed. Quality assurance processes will be conducted to ensure high quality of data for maximizing reusability.

### Data interoperability:

To make data interoperable, standard data and metadata common formats will be used.

## Example Case – UNDER WORK

### COVID-19 prediction using sensor data

A workflow using sensor-based data for federated learning to predict COVID-19 would involve several steps, including data collection, pre-processing, model training, and evaluation.

* Data collection: Sensor-based data, such as temperature, SpO2, and air quality, would be collected from multiple sources, like wearable devices and environmental sensors. The data would be collected in a decentralized manner, meaning that it would be collected directly from the source rather than being centralized to a single location.
* Data pre-processing: The collected data would be pre-processed locally ensuring that it is in a consistent format and that any missing or corrupted data is handled appropriately. This might include tasks such as normalization, removal of outliers, and interpolation of missing data.
* Data upload to AIH Platform Back-end.
* Model training: A federated learning model client is trained on the preprocessed data in every AIH Instance.
* Evaluation: The trained model is evaluated
* Deployment: The trained model would be deployed in the devices of the patients or in the medical equipment, where it can be used to make predictions or monitor the health status of the patients.
* Continual Monitoring: The model performance would be monitored over time, and retrained as needed when new data is available. This would ensure that the model remains accurate and up to date.

# References

1. Technical Specification; MPAI Ecosystem Governance (MPAI-GME) V1.1; https://mpai.community/standards/mpai-gme/
2. Technical Specification; AI Framework (MPAI-AIF) V1.1; <https://mpai.community/standards/mpai-aif/>
3. 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.
4. 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:

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.

1. MPAI-wide terms and definitions

The Terms used in this standard whose first letter is capital and are not already included in Table 1are defined in *Table 2.*

*Table 2 – MPAI-wide Terms*

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Access | Static or slowly changing data that are required by an application such as domain knowledge data, data models, etc. |
| AI Framework (AIF) | The environment where AIWs are executed. |
| AI AIMName (AIM) | A data 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 outputs according to the AIW Function. |
| Application Standard | An MPAI Standard designed to enable a particular application domain. |
| Channel | A connection between an output port of an AIM and an input port of an AIM. The term “connection” is also used as synonymous. |
| Communication | The infrastructure that implements message passing between AIMs |
| Composite AIM | An AIM aggregating more than one AIM. |
| Component | One of the 7 AIF elements: Access, Communication, Controller, Internal Storage, Global Storage, Store, and User Agent |
| Conformance | The attribute of an Implementation of being a correct technical Implem­entation of a Technical Specification. |
| Conformance Tester | An entity Testing the Conformance of an Implem­entation. |
| Conformance Testing | The normative document specifying the Means to Test the Conformance of an Implem­entation. |
| Conformance Testing Means | Procedures, tools, data sets and/or data set characteristics to Test the Conformance of an Implem­en­tation. |
| Connection | A channel connecting an output port of an AIM and an input port of an AIM. |
| Controller | A Component that manages and controls the AIMs in the AIF, so that they execute in the correct order and at the time when they are needed |
| Data Format | The standard digital representation of data. |
| Data Semantics | The meaning of data. |
| Ecosystem | The ensemble of actors making it possible for a User to execute an application composed of an AIF, one or more AIWs, each with one or more AIMs potentially sourced from independent implementers. |
| Explainability | The ability to trace the output of an Implementation back to the inputs that have produced it. |
| Fairness | The attribute of an Implementation whose extent of applicability can be assessed by making the training set and/or network open to testing for bias and unanticipated results. |
| Function | The operations effected by an AIW or an AIM on input data. |
| Global Storage | A Component to store data shared by AIMs. |
| Internal Storage | A Component to store data of the individual AIMs. |
| Identifier | A name that uniquely identifies an Implementation. |
| Implementation | 1. An embodiment of the MPAI-AIF Technical Specification, or 2. An AIW or AIM of a particular Level (1-2-3) conforming with a Use Case of an MPAI Applic­ation Standard. |
| Implementer | A legal entity implementing MPAI Technical Specifications. |
| ImplementerID (IID) | A unique name assigned by the ImplementerID Registration Authority to an Implementer. |
| ImplementerID Registration Authority (IIDRA) | The entity appointed by MPAI to assign ImplementerID’s to Implementers. |
| Interoperability | The ability to functionally replace an AIM with another AIW having the same Interoperability Level |
| Interoperability Level | The attribute of an AIW and its AIMs to be executable in an AIF Implem­en­tati­on and to:   1. Be proprietary (Level 1) 2. Pass the Conformance Tes­ting (Level 2) of an Applic­ation Standard 3. Pass the Perform­ance Testing (Level 3) of an Applic­ation Standard. |
| Knowledge Base | Structured and/or unstructured information made accessible to AIMs via MPAI-specified interfaces |
| Message | A sequence of Records transported by Communication through Channels. |
| Normativity | The set of attributes of a technology or a set of technologies specified by the applicable parts of an MPAI standard. |
| Performance | The attribute of an Implementation of being Reliable, Robust, Fair and Replicable. |
| Performance Assessment | The normative document specifying the Means to Assess the Grade of Performance of an Implementation. |
| Performance Assessment Means | Procedures, tools, data sets and/or data set characteristics to Assess the Performance of an Implem­en­tation. |
| Performance Assessor | An entity Assessing the Performance of an Implementation. |
| Profile | A particular subset of the technologies used in MPAI-AIF or an AIW of an Application Standard and, where applicable, the classes, other subsets, options and parameters relevant to that subset. |
| Record | A data structure with a specified structure |
| Reference Model | The AIMs and theirs Connections in an AIW. |
| Reference Software | A technically correct software implementation of a Technical Specific­ation containing source code, or source and compiled code. |
| Reliability | The attribute of an Implementation that performs as specified by the Application Standard, profile and version the Implementation refers to, e.g., within the application scope, stated limitations, and for the period of time specified by the Implementer. |
| Replicability | The attribute of an Implementation whose Performance, as Assessed by a Performance Assessor, can be replicated, within an agreed level, by another Performance Assessor. |
| Robustness | The attribute of an Implementation that copes with data outside of the stated application scope with an estimated degree of confidence. |
| Scope | The domain of applicability of an MPAI Application Standard |
| Service Provider | An entrepreneur who offers an Implementation as a service (e.g., a recommendation service) to Users. |
| Standard | The ensemble of Technical Specification, Reference Software, Confor­man­ce Testing and Performance Assessment of an MPAI application Standard. |
| Technical Specification | (Framework) the normative specification of the AIF.  (Application) the normative specification of the set of AIWs belon­ging to an application domain along with the AIMs required to Im­plem­ent the AIWs that includes:   1. The formats of the Input/Output data of the AIWs implementing the AIWs. 2. The Connections of the AIMs of the AIW. 3. The formats of the Input/Output data of the AIMs belonging to the AIW. |
| Testing Laboratory | A laboratory accredited to Assess the Grade of Performance of Implementations. |
| Time Base | The protocol specifying how Components can access timing information |
| Topology | The set of AIM Connections of an AIW. |
| Use Case | A particular instance of the Application domain target of an Application Standard. |
| User | A user of an Implementation. |
| User Agent | The Component interfacing the user with an AIF through the Controller |
| Version | A revision or extension of a Standard or of one of its elements. |

1. Notices and Disclaimers Concerning MPAI Standards (Informative)

The notices and legal disclaimers given below shall be borne in mind when [downloading](https://www.mpai.community/resources/) and using approved MPAI Standards.

In the following, “Standard” means the collection of four MPAI-approved and [published](https://www.mpai.community/resources/) docum­ents: “Technical Specification”, “Reference Software” and “Conformance Testing” and, where applicable, “Performance Testing”.

Life cycle of MPAI Standards

MPAI Standards are developed in accordance with the [MPAI Statutes](https://mpai.community/statutes/). An MPAI Standard may only be developed when a Framework Licence has been adopted. MPAI Standards are developed by especially established MPAI Development Committees who operate on the basis of consensus, as specified in Annex 1 of the [MPAI Statutes](https://mpai.community/statutes/). While the MPAI General Assembly and the Board of Directors administer the process of the said Annex 1, MPAI does not independently evaluate, test, or verify the accuracy of any of the information or the suitability of any of the technology choices made in its Standards.

MPAI Standards may be modified at any time by corrigenda or new editions. A new edition, however, may not necessarily replace an existing MPAI standard. Visit the [web page](https://mpai.community/resources/) to determine the status of any given published MPAI Standard.

Comments on MPAI Standards are welcome from any interested parties, whether MPAI members or not. Comments shall mandatorily include the name and the version of the MPAI Standard and, if applicable, the specific page or line the comment applies to. Comments should be sent to the [MPAI Secretariat](mailto:secretariat@mpai.community). Comments will be reviewed by the appropriate committee for their technical relevance. However, MPAI does not provide interpretation, consulting information, or advice on MPAI Standards. Interested parties are invited to join MPAI so that they can attend the relevant Development Committees.

Coverage and Applicability of MPAI Standards

MPAI makes no warranties or representations of any kind concerning its Standards, and expressly disclaims all warranties, expressed or implied, concerning any of its Standards, including but not limited to the warranties of merchantability, fitness for a particular purpose, non-infringement etc. MPAI Standards are supplied “AS IS”.

The existence of an MPAI Standard does not imply that there are no other ways to produce and distribute products and services in the scope of the Standard. Technical progress may render the technologies included in the MPAI Standard obsolete by the time the Standard is used, especially in a field as dynamic as AI. Therefore, those looking for standards in the Data Compression by Artificial Intelligence area should carefully assess the suitability of MPAI Standards for their needs.

IN NO EVENT SHALL MPAI BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

MPAI alerts users that practicing its Standards may infringe patents and other rights of third parties. Submitters of technologies to this standard have agreed to licence their Intellectual Property according to their respective Framework Licences.

Users of MPAI Standards should consider all applicable laws and regulations when using an MPAI Standard. The validity of Conformance Testing is strictly technical and refers to the correct implementation of the MPAI Standard. Moreover, positive Performance Assessment of an implementation applies exclusively in the context of the [MPAI Governance](https://mpai.community/standards/mpai-gme/) and does not imply compliance with any regulatory requirements in the context of any jurisdiction. Therefore, it is the responsibility of the MPAI Standard implementer to observe or refer to the applicable regulatory requirements. By publishing an MPAI Standard, MPAI does not intend to promote actions that are not in compliance with applicable laws, and the Standard shall not be construed as doing so. In particular, users should evaluate MPAI Standards from the viewpoint of data privacy and data ownership in the context of their jurisdictions.

Implementers and users of MPAI Standards documents are responsible for determining and complying with all appropriate safety, security, environmental and health and all applicable laws and regulations.

Copyright

MPAI draft and approved standards, whether they are in the form of documents or as web pages or otherwise, are copyrighted by MPAI under Swiss and international copyright laws. MPAI Standards are made available and may be used for a wide variety of public and private uses, e.g., implementation, use and reference, in laws and regulations and standardisation. By making these documents available for these and other uses, however, MPAI does not waive any rights in copyright to its Standards. For inquiries regarding the copyright of MPAI standards, please contact the [MPAI Secretariat](mailto:secretariat@mpai.community).

The Reference Software of an MPAI Standard is released with the [MPAI Modified Berkeley Software Distribution licence](https://mpai.community/about/licence/). However, implementers should be aware that the Reference Software of an MPAI Standard may reference some third party software that may have a different licence.

1. The Governance of the MPAI Ecosystem (Informative)

**Level 1 Interoperability**

With reference to Figure 1 MPAI issues and maintains a standard – called MPAI-AIF – whose components are:

1. An environment called AI Framework (AIF) running AI Workflows (AIW) composed of inter­connected AI Modules (AIM) exposing standard interfaces.
2. A distribution system of AIW and AIM Implementation called MPAI Store from which an AIF Implementation can download AIWs and AIMs.

A Level 1 Implementation shall be an Implementation of the MPAI-AIF Technical Specification executing AIWs composed of AIMs able to call the MPAI-AIF APIs.

|  |  |
| --- | --- |
| Implementers’ benefits | Upload to the MPAI Store and have globally distributed Implementations of   * AIFs conforming to MPAI-AIF. * AIWs and AIMs performing prop­rietary functions executable in AIF. |
| Users’ benefits | Rely on Implementations that have been tested for security. |
| MPAI Store’s role | * Tests the Conformance of Implementations to MPAI-AIF. * Verifies Implementations’ security, e.g., absence of malware. * Indicates unambiguously that Implementations are Level 1. |

**Level 2 Interoperability**

In a Level 2 Implem­entation, the AIW shall be an Implementation of an MPAI Use Case, and the AIMs shall con­form with an MPAI Applicati­on Standard.

|  |  |
| --- | --- |
| Implementers’ benefits | Upload to the MPAI Store and have globally distributed Implementations of   * AIFs conforming to MPAI-AIF. * AIWs and AIMs conforming to MPAI Application Standards. |
| Users’ benefits | * Rely on Implementations of AIWs and AIMs whose Functions have been reviewed during standardisation. * Have a degree of Explainability of the AIW operation because the AIM Func­tions and the data Formats are known. |
| Market’s benefits | * Open AIW and AIM markets foster competition leading to better products. * Competition of AIW and AIM Implementations fosters AI innovation. |
| MPAI Store’s role | * Tests Conformance of Implementations with the relevant MPAI Standard. * Verifies Implementations’ security. * Indicates unambiguously that Implementations are Level 2. |

**Level 3 Interoperability**

MPAI does not generally set standards on how and with what data an AIM should be trained. This is an important differentiator that promotes competition leading to better solutions. However, the performance of an AIM is typically higher if the data used for training are in greater quantity and more in tune with the scope. Training data that have large variety and cover the spec­trum of all cases of interest in breadth and depth typically lead to Implementations of higher “quality”.

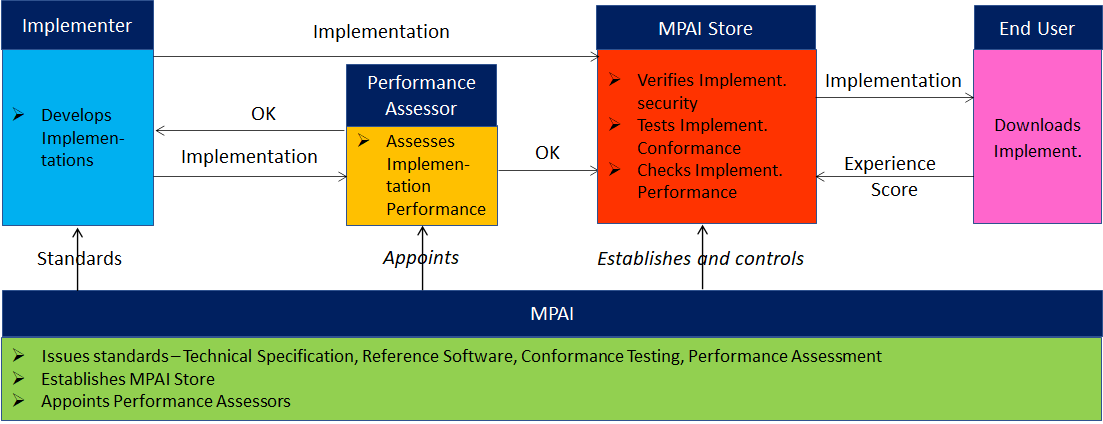
For Level 3, MPAI normatively specifies the process, the tools and the data or the characteristics of the data to be used to Assess the Grade of Performance of an AIM or an AIW.

|  |  |
| --- | --- |
| Implementers’ benefits | May claim their Implementations have passed Performance Assessment. |
| Users’ benefits | Get assurance that the Implementation being used performs correctly, e.g., it has been properly trained. |
| Market’s benefits | Implementations’ Performance Grades stimulate the development of more Performing AIM and AIW Implementations. |
| MPAI Store’s role | * Verifies the Implementations’ security. * Indicates unambiguously that Implementations are Level 3. |

**The MPAI ecosystem**

The following *Figure 4* is a high-level description of the MPAI ecosystem operation applicable to fully conforming MPAI implementations:

1. MPAI establishes and controls the not-for-profit MPAI Store (step 1).
2. MPAI appoints Performance Assessors (step 2).
3. MPAI publishes Standards (step 3).
4. Implementers submit Implementations to Performance Assessors (step 4).
5. If the Implementation Performance is acceptable, Performance Assessors inform Implementers (step 5a) and MPAI Store (step 5b).
6. Implementers submit Implementations to the MPAI Store (step 6); The Store Tests Confor­mance and security of the Implementation.
7. Users download Implementations (step 7).



*Figure 4 – The MPAI ecosystem operation*

1. API Requirements

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API entry point** | **Method** | **Input Data** | **Output Data** | **Requirements** |
| /v1/authentication/register | POST | username  password  type=’End User’  [BCpriv\_k] | [Account] | Register a new End User on the AIH Platform Back-end.  If BCpriv is specified, the user will be registered in the blockchain using the address (derived from the private key) specified. |
| /v1/authentication/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 the 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: described in Annex 5 |
| /v1/storage/retrieve | GET |  | [healthData] | Retrieve health data of the owner End User  on the Global Secure DataVault |
| /v1/storage/delete | DELETE | healthDataID |  | Delete the 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 the health data of the owner End User on the Global Secure DataVault |
| /v1/auditing/createSmartContract | PUT | Third-party address |  | Create a smart contract that contains all the licenses that user and third part entity shares. When is called from the user, the contract is endorsed by default. |
| /v1/auditing/createPermission | POST | Third-party address  data\_query  proxy-key |  | Create a new record on the smart contract of the specific third partyuser that contains a new smart contract for the specific permission (data\_query) to be shared. |
| /v1/auditing/validatePermission | PUT | Third-party address  permissionId |  | Update the grant of a third party permission, authorized to access to the user health data of the Global Secure DataVault. |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API entry point** | **Method** | **Input Data** | **Output Data** | **Requirement** |
| /v1/ai/getGlobalModel | GET |  | [W] | Get the global model parameters (W) from the Federated Learning System to initialize the federated learning process. |
| /v1/ai/sendLocalModel | PUT | [Δw] |  | Send the parameters (Δw) to the Federated Learning System, trained with the respective user private dataset. |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API entry point** | **Method** | **Input Data** | **Output Data** | **Requirement** |
| /v1/authentication/register | POST | username  password  type=’End User’  [BCpriv\_k] | [Account] | Register a new End User on the AIH Platform Back-end.  If BCpriv is specified, the user will be registered in the blockchain using the address (derived from the private key) specified. |
| /v1/authentication/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/retrieve | GET | [filterCriteria] | [healthData] | Retrieve health data on the Global Secure DataVault |
| /v1/storage/import | POST | [healthdata]  User\_address |  | Import health data in the back-end system for a specific user address |
| /v1/auditing/createPermission | POST | user\_address  data\_query  proxy-key |  | Create a new record on the smart contract of the specific thirdpart-user that contains a new smart contract for the specific permission (data\_query) to be shared. |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **API entry point** | **Input Data** | **Output Data** | **Requirement** |
| /createSmartContract | userAddress  thirdpartAddress |  | Registration of a smart contract that contains all the permissions of data sharing authorization by Third-party entities towards user's data. |
| /createPermission | userAddress  ThirdpartAddress  data\_query | [] | Create a specific permission |

1. Data Types

Requirements (AIH) has developed the following list of Data Types

|  |  |
| --- | --- |
| **Data Type** | **Comments** |
| **Historical User Health Data** | End User’s medical history, lab results, etc |
| **Time series** | Vital sign measurements, such as heart rate and blood pressure |
| **Sensor** | Data from wearable devices: smartwatches, fitness trackers etc. |
| **Geolocation** | Geographic location of individuals |
| **Social media** | Chats, posts, comments, and other related data |
| **Text** | Unstructured data, e.g., clinical notes and patient-generated data |
| **Audio** | Speech and audio recordings |
| **Video** | Data from endoscopic procedures, laparoscopic surgeries, etc. |
| **Medical images** | X-ray, CT, MRI, and ultrasound images |
| **Genomic** | DNA sequencing data and other types of genetic information |
| **Medical imaging** | 3D images, 4D images (e.g., MRI over time), and multimodal images |

For the purpose of this project, a data format for each data type is selected to ensure compatibility of processed data coming from the different AIH Instances.

## Aggregated Health Data Format

The Aggregated Health Data Format is simply a container to carry data from a Front-End to the Back-End. EHRs improve the efficiency and quality of healthcare by providing healthcare providers with comprehensive, up-to-date, and accurate information about a patient's health history. One example of a data standard used for exchanging healthcare information electronically is the Fast Healthcare Interoperability Resources (FHIR). FHIR was developed by HL7, a global healthcare standards organization and is widely used by healthcare providers and EHR vendors, including companies such as Epic Systems, Cerner, AllScripts, athenahealth, NextGen Healthcare, and Oracle Health Sciences. The format is also being used in hospitals in Europe, including Barts Health NHS Trust in the United Kingdom, University Hospitals Leuven in Belgium, and the Academic Medical Center in the Netherlands. Technically, a tipical EHR contains information about a patient's demographic details, medical history, medication information, laboratory test results, imaging results, and treatment plans.

### Time-Series data

Time-series medical data refers to data collected over time, such as vital sign measurements like heart rate and blood pressure. This data provides valuable information for healthcare providers to monitor a patient's health status, predict potential diseases, and evaluate the progress of treatments.

|  |  |  |
| --- | --- | --- |
| **Name** | **Unit** | **Format** |
| Temperature | Celsius (C) | Floating-point |
| Blood Pressure | Millimeters of mercury (mmHg) | Floating-point |
| Heart rate | Beats per minute (bpm) | Integer |
| Blood glucose | Milligrams per deciliter (mg/dL) | Floating-point |
| Timestamp | YYYY-MM-DD HH:MM:SS | String |
| Respiratory rate | Breaths per minute (bpm) | Integer |
| Hemoglobin A1c | Percentage of Hemoglobin A1c | Floating-point |
| Serum cholesterol | milligrams per deciliter (mg/dL) | Floating-point |
| Serum triglycerides | milligrams per deciliter (mg/dL) | Floating Point |
| Steps | Number of steps in a period | Integer |

The FHIR is equipped to handle time-series medical data. It provides a standard data format for exchanging healthcare information electronically, including time-series data. An JSON example is shown below:

{

"resourceType": "Bundle",

"type": "collection",

"entry": [

{

"resource": {

"resourceType": "Observation",

"subject": {

"reference": "Patient/123"

},

"effectiveDateTime": "2022-03-10T09:30:00Z",

"valueQuantity": {

"value": 37.0,

"unit": "C"

}

}

},

{

"resource": {

"resourceType": "Observation",

"subject": {

"reference": "Patient/123"

},

"effectiveDateTime": "2022-03-10T09:30:00Z",

"valueQuantity": {

"value": 120,

"unit": "mmHg"

}

}

},

{

"resource": {

"resourceType": "Observation",

"subject": {

"reference": "Patient/123"

},

"effectiveDateTime": "2022-03-10T09:30:00Z",

"valueQuantity": {

"value": 80,

"unit": "mmHg"

}

}

},

{

"resource": {

"resourceType": "Observation",

"subject": {

"reference": "Patient/123"

},

"effectiveDateTime": "2022-03-10T09:30:00Z",

"valueQuantity": {

"value": 90,

"unit": "bpm"

}

}

},

{

"resource": {

"resourceType": "Observation",

"subject": {

"reference": "Patient/123"

},

"effectiveDateTime": "2022-03-10T09:30:00Z",

"valueQuantity": {

"value": 120,

"unit": "mg/dL"

}

}

}

]

}