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

N1349 2023/08/23

Source Requirements (AIH)
Title Use Cases and Functional Requirements – Artificial Intelligence for Health Data (MPAI-AIH)
Target MPAI Community

Table of Contents

1 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.

Artificial Intelligence for Health data (MPAI-AIH) is an MPAI project supporting advanced forms of health provided by an end-to-end system to collect, AI-based process, access health data, and distribute AI Models updated via federated learning techniques. MPAI is the international, unaffiliated, non-profit Moving Picture, Audio, and Data Coding by Artificial Intelligence organisation developing standards for AI-based data coding. 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.

This document "Use Cases and Functional Requirements: AI Health Data (MPAI-AIH)" introduces the model an AIH Platform where:

1. End Users equipped AIH Frontends, personal devices running an AIH App, to collect, AI process, and upload health data to the AIH Backend.
2. The AIH Backend AI process health data and makes available original and processed health data to Third Party Users for further processing.
3. The AI Models used by the AIH Frontends are periodically collected and the knowledge acquired is used to update the common AI Models using Federated Learning techniques.

The AI Health Platform has characteristics that comply with generally applicable legal frameworks such as the European GDPR. Therefore, the AIH Frontends upload their health data to the AIH Backend with an attached Smart Contract that regulated the access and use of the health data. The AI Backend and Third-Party Users can only have access and use the health data at terms and conditions dictated by the Smart Contract.

This document describes one Use Case (the AIH Platform) and identifies its Functional Requirements. The AIH Platform is assumed to rely on the MPAI AI Framework (MPAI-AIF) standard (see Annex 4 - MPAI Basic) to process health data.

Use Cases and Functional Requirements: AI Health Data (MPAI-AIH) is an attachment to the Call for Technologies: AI for Health Data (MPAI-AIH) [3].

2 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 Union General Data Protection Regulation (GDPR) [5] or the proposal of regulation (Artificial Intelligence Act) [7]. The motivation for the development of this standard comes from the AIM Health Portuguese project.

3 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.

4 Architecture of the AI Health Secure Platform

This section introduces the overall architecture of the AI Health Secure Platform.

4.1 Introduction

The 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 1.

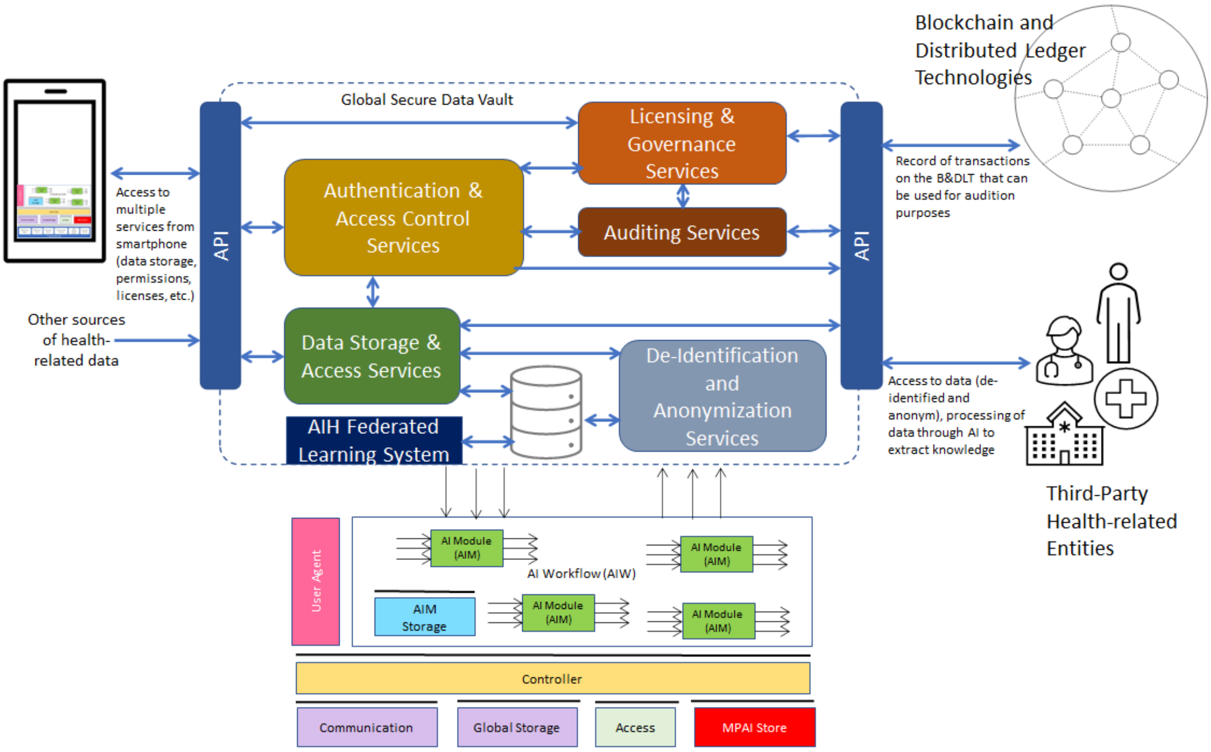

Figure 1 – Reference Model of AIH Secure Platform

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

4.2 Actors

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

1. *AIH Front-end 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 Back-End 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 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 between the Sources and the AIH Platform Back-End.

4.3 Services

Figure 1 depicts the AI Health data system is composed of a set of distributed components and services:

1. The AIH *Front-end*, a smart device (e.g., a smartphone) application (AI-Health app) that is capable of:

1.1. Capturing End User’s Health data, e.g., from Google Fit and Apple Health, and from external biometric sensors that capture Health data.

1.2. Locally storing in a “Secure Data Vault” controlled by the End User (see Figure 2)

1.3. 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.

1.4. Alerting the End User about any deviation of the value of the captured data that may be caused by disease.

1.5. Uploading the processed AI Health data.

1.6. Receiving processed Health data from the AIH Back-End.

The AIH Front-end represents the personal gateway to the user-data and any external biometric sensors that capture health-related data, and the connection with the AIH-Back-End.

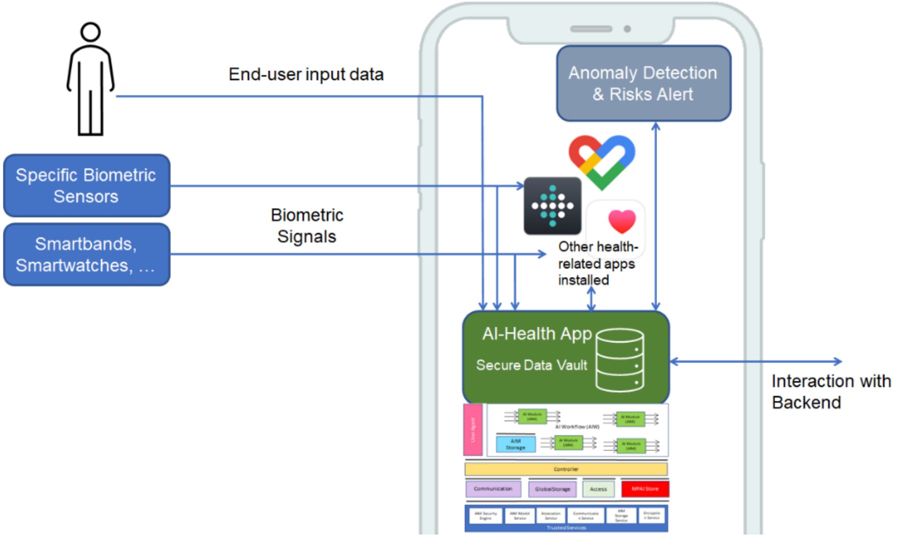

Figure 2 – Front-end architecture of the MPAI-AIH Secure Platform

2. 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 Back-End. 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 Back-End.
3. *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
4. The parties, e.g., the End User storing AIH data and the AIH Back-End.

4.1. The AIH data involved (to be stored or accessed).

4.2. The type of processing present in the AIH Processing Taxonomy performed.

4.3. The access conditions (e.g., timeframe and the End User’s permissions).

5. 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.
6. 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 Front-end and using it under the terms of the Smart contract that has been established between the End User and the AIH Back-End. When an improved model is obtained by the FLS, this is distributed to the client models via a model update, as detailed in Section Annex 3 - .

5 Healthcare use case

This section describes the AI Health Secure Platform of Figure 1 and its usage as a use case of MPAI-AIH.

5.1 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 AIH Back-End 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.

5.2 Background process

Background processes may take place in the AIH Front-end and in the AIH 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.

5.3 Access to and process of AIH data

This section deals with the processing done by the AIH Back-End based on the AIH Back-End 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 Back-End 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 Back-End 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 Back-Ends' 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 on behalf of the End Users. Access to AIH data is permitted if the Smart Contract is valid.
- Intelligent processing of health data is performed either via proper AIWs or via FLS.

5.4 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 Entity**	
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 distinct 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 from the third party entity 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

5.5 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.

6 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].

6.1 Federated Learning Processes

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

6.1.1 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.
5. The updated global model is shared with 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 of 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 Back-End FLS, that aggregates this local knowledge in a new improved global model, that can, in turn, be downloaded by the End Users app.

6.1.2 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 their local model to improve the accuracy and customization of the model for herself/ himself.

6.2 MPAI-AIH Federated Components

The AIH Platform Front-end and Third-Parties communicate with the AIH Platform Back-End 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 Front-Ends and are instances of the standard AIMs (and AIWs) that handle healthcare data. For example, an AIW service built to detect the possibility of COVID-19 infection by a user may engage 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 [7], 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.

7 Example Case: COVID-19 prediction using sensor data

Both an AIH Model Instance or the central AIH FLS model can, when dully permitted, make predictions or monitor the health status of the (patient) End users.

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 heartbeat, body temperature, SpO2, external temperature and air quality, are collected from multiple sources, like personal wearable devices and environmental sensors. The data is collected in a decentralized manner, meaning that it is collected directly from the source rather than being centralized into a single location.
- Data pre-processing: The collected data is pre-processed locally, ensuring that it is in a consistent format and that any missing or corrupted data is handled appropriately.
- if the necessary permission has been granted, data may be uploaded to AIH Platform Back-End central system.

- Model training: The AIH Model Instance, i.e., the federated learning model client is trained on the pre-processed data in the AIH Platform Front-end.
- Evaluation: The trained model is always evaluated.
- Deployment: Periodically, the weights of the local trained model are collected via API by the AIH FLS. If an improved central model is achieved, an update will be deployed for the End users AIH Platform Front-end to upgrade the local AIH Model Instance.
- Continual Monitoring: Models' performance is 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.

8 MPAI-AIH Requirements

This chapter provides the functional requirements for 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 [12].

8.1 General Requirements on Data

Machine-readable metadata are essential for automatic discovery of data and services. As such, data should be findable, accessible, interoperable and reusable according to the FAIR principles, even if it is to remain private. Metadata is the descriptor of and data is the thing that is being described.

All data to be collected shall be treated in accordance with existing legal and ethical standards and requirements in the respective countries of data collection and of data usage.

8.1.1 Data accessibility

Data can either be public or confidential, depending on the information and the anonymization or aggregation transformations undertaken.

- 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. In this case, data and meta data are retrievable via their identifier using a standardised communications protocol.
- Metadata are accessible, even when the data are no longer available.

8.1.2 Data findability

- Third Parties can find personal or health data depending on the terms of the Smart Contract detail authorization for access to protected data.
- Open data must be made findable: data and metadata are assigned globally unique and persistent identifiers.

8.1.3 Data reusability

- Metadata and data are released with a clear and accessible data usage license. 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:
    - Metadata and data are associated with detailed provenance.
    - Metadata and data meet domain-relevant community standards.

8.1.4 Data interoperability

- To make data interoperable, standard data and metadata common formats should be used. The documentation used to describe datasets needs to be resolvable using globally unique and persistent identifiers. The documentation in itself needs to be easily findable and accessible.
- Metadata and data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
- Data and metadata include qualified references to other (meta)data.

8.1.5 Security and Privacy

- Personal data will be made available to Third-parties only upon permission of the user assuring that informed consent on data usage and maintenance from the Third-Party that is requesting it has been issued and agreed.
- Every request of authorization permission and concession is tracked in the blockchain using the specific function of the smart contracts.

- Every time a third party withdraws data after it has been authorized is tracked in the blockchain using the specific function of the smart contracts.
- Any participant providing personal data for Third-parties must be allowed to withdraw the permission to access their data at any time.

8.2 Functional requirements

8.2.1 List of Smart Contract functional requirements

A Smart Contract shall express the terms and conditions between

1. An End User and the AIH Back-End
1.1. End User ID
1.2. Health Data ID
1.3. Health Data Type ID
1.4. Date of issuance of Smart Contract
1.5. Duration of Smart Contract
1.6. List of permitted Processing Types
1.7. List of permitted Export Types (Seen from the End User)
2. The AIH Back-End and a Third Part Entity #1.
2.1. Third-Party ID
2.2. Health Data ID
2.3. Health Data Type ID
2.4. Date of issuance of Smart Contract
2.5. Duration of Smart Contract
2.6. List of permitted Processing Types
2.7. List of permitted Export Types (seen from the AIH Back-End)
3. The Third Part Entity and AIH Back-End #2.
3.1. Third-Party ID
3.2. Health Data ID
3.3. Health Data Type ID
3.4. Date of issuance of Smart Contract
3.5. Duration of Smart Contract
3.6. List of permitted Processing Types.
3.7. List of permitted Export Types (seen from the Third-Party User)

8.2.2 List of Data Types and Usage requirements

MPAI is looking for proposals to represent the following data types and possible associated metadata:

Data Type	Short Description
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/samples
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

Sensors generating data in the edge device are generally calibrated by the AIH Front-end. The calibration information should be specified along with the sensed data.

Next, we present examples of what should be kept for each of the data types and what analytics may be performed in each of the cases. 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.

1. *Historical User Health Data*, possibly represented in a markup language or JSON, with End User's medical history, lab tests/exams results, and other health-related information that can be used to predict abnormal conditions and inform treatment decisions.
2. *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. Examples are given in Table 2

Table 2 – Examples of health data, units of measurement, and representation formats

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

3. *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, 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 a healthy behavior. The data format should include the data types and their units of measure.
4. *Geolocation data*: This includes information about the geographic location of End Users, such as 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).
5. *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.
6. *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.

7. *Health-related video data*: For example, self-taken videos used for remote demonstration of a particular physical health situation.
8. *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.
9. *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).
10. *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.

8.2.3 Aggregated Health Data Format

The Aggregated Health Data Format is simply a container to carry data from a Front-End to the Back-End. Electronic Health Records (EHR) 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 Centre in the Netherlands.

The Aggregated Health Data Format should be wrapped in a secure envelope along with associated encryption methods and containing the user’s health data records. The envelope format should be independent of the data it contains. MPAI AIH healthcare information should be exchanged electronically and wrapped in an adequate envelope.

8.2.4 Federated Learning requirements

This section addresses the format of the AIH Front-end’s Neural Network Models used throughout the AIH Platform. However, MPAI is not seeking proposals for this since Models will depend on the target applications and may change over time.

8.3 API Requirements

8.3.1 List of API functional requirements

An 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.

The system will use REST API interfaces that will provide data access between the components of the AIH Platform.

8.3.2 API: AIH Platform Back-End ↔ Platform Front-end

This API should provide:

1. The necessary services to register, authenticate and control access of the user in the AIH Platform.
2. The mechanisms for storing AI data, granting, and acquiring permissions via smart contracts for data usage, and data usage auditing.

The list of required services and their parameters is:

- **Authentication_register**: Register a new End User on the AIH Platform Back-End.

If a key is specified, the user will be registered in the blockchain using the address (derived from the private key) specified.

- **Authentication_signin**: Sign in the End User on the AIH Platform Back-End and create an authentication and authorization token (JWT).
- **Storage_store**: Store the health data of the End User on the Global Secure DataVault. Data must be encrypted from the End User before the insertion.
- **Storage_retrieve**: Retrieve health data of the owner End User on the Global Secure DataVault.
- **Storage_delete**: Delete the health data of the owner End User on the Global Secure DataVault, only an owner of the data can delete it.
- **Storage_update**: Update the health data of the owner End User on the Global Secure DataVault.
- **Auditing_createSmartContract**: Create a smart contract that contains all the licenses that user and third part entity shares. When is invoked from the user, the contract is endorsed by default.
- **Auditing_createPermission**: Create a new record on the smart contract of the specific Third-Party user that holds a new smart contract for the specific permission (data_query) to be shared.
- **Auditing_validatePermission**: Update the grant of a third party permission, authorized to access to the user health data of the Global Secure DataVault.

8.4 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. The list of required services and their parameters is:

- **Model_Get:** Retrieve global model parameters (W) from the Federated Learning System central node in the Back-End to initialize the Federated Learning process.
- **Model_Store:** Send local parameters (Δw) from the local Federated Learning client to the Back-End for the central Federated Learning System for possible aggregation.

8.4.1 API: AIH Platform Back-End ↔Third party User

This API should provide:

1. The necessary services to register, authenticate and control access of Third-Party Users in the AIH Platform.
2. The service of retrieving the user's health data requested by the Third-Party.

The list of required services and their parameters is:

- **Authentication_register**: Register a new End User on the AIH Platform Back-End. If a key is specified, the user will be registered in the blockchain using the address (derived from the private key) specified.
- **Authentication_signin**: Sign in the third part on the AIH Platform Back-End and create an authentication and authorization token (JWT).
- **Storage_retrieve**: Retrieve health data from the Global Secure DataVault.
- **Storage_import**: Import health data from the back-end system for a specific user address
- **Auditing_createPermission**: Create a new record on the smart contract of the specific Third-Party user that contains a new smart contract for the specific permission (*data_query*) to be shared.

8.4.2 API: AIH Platform Back-End System ↔ Blockchain

This API will provide the necessary services to register and check the validity of the smart contract between the End User and the Third Party. The list of required services and their parameters is:

- **Smartcontract_create**: Registration of a smart contract that contains all the permissions of data sharing authorization by Third-Party entities towards user's data.

- **Smartcontract_permission**: Create a specific permission.

9 References

1. MPAI Standards Resources; https://mpai.community/standards/resources/.
2. MPAI Patent Policy; https://mpai.community/about/the-mpai-patent-policy/.
3. MPAI; Call for Technologies: AI for Health Data (MPAI-AIH); N1348; https://mpai.community/standards/mpai-aih/call-for-technologies/
4. MPAI; Framework Licence: AI for Health Data (MPAI-AIH); N1350; https://mpai.community/standards/MPAI-AIH/framework-licence/
5. MPAI; Template for Responses: AI for Health Data (MPAI-AIH); N1351; https://mpai.community/standards/MPAI-AIH/template-for-responses/
6. Technical Specification; MPAI Ecosystem Governance (MPAI-GME) V1.1; https://mpai.community/standards/mpai-gme/
7. Technical Specification; AI Framework (MPAI-AIF) V1.1; https://mpai.community/standards/mpai-aif/
8. 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.
9. 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.
10. *General Data Protection Regulation (GDPR) https://commission.europa.eu/law/law-topic/data-protection/data-protection-eu_en (Accessed 20th July 2023).*
11. Jacobsen, et al., FAIR Principles: Interpretations and Implementation Considerations. *Data Intelligence* 2020; 2 (1-2): 10–29. doi: https://doi.org/10.1162/dint_r_00024.
12. *Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act) and amending certain Union Legislative Acts https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52021PC0206*

Annex 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 3.*

Table 3 – 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.

Annex 2 - The Governance of the MPAI Ecosystem (Informative)

Level 1 Interoperability

MPAI issues and maintains a standard – called MPAI-AIF [7] – 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 3* 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).

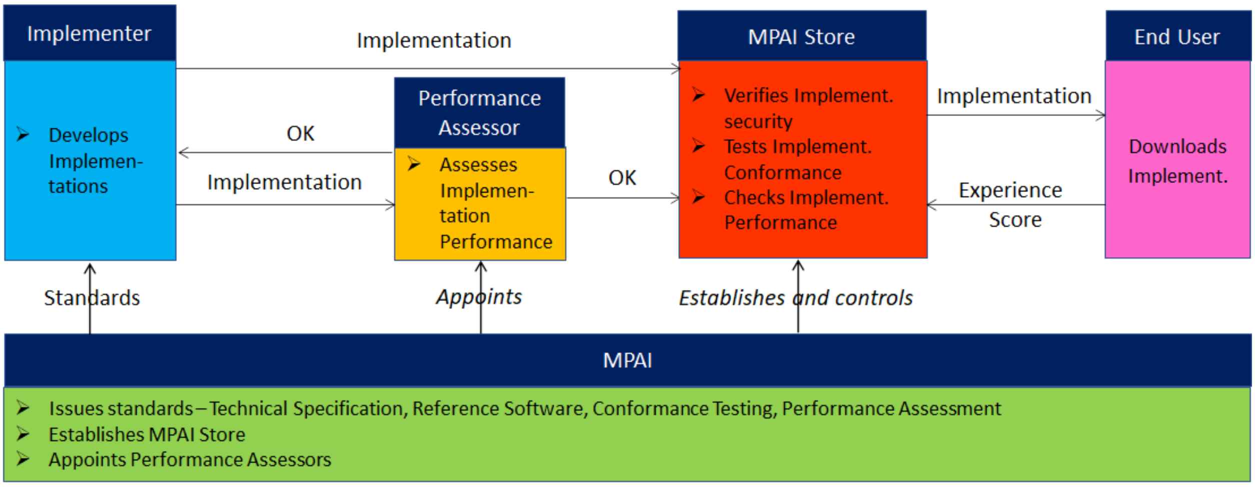
Implementer
Develops Implemen-tations
Implementation
OK
Implementation
Performance Assessor
Assesses Implemen-tation Performance
OK
MPAI Store
Verifies Implement. security
Tests Implement. Conformance
Checks Implement. Performance
Implementation
Experience Score
End User
Downloads Implement.
Standards
Appoints
Establishes and controls
MPAI
Issues standards – Technical Specification, Reference Software, Conformance Testing, Performance Assessment
Establishes MPAI Store
Appoints Performance Assessors

Figure 3 – The MPAI ecosystem operation

Annex 3 - AIH Federated Learning System

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

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 confidentiality-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 (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 2).

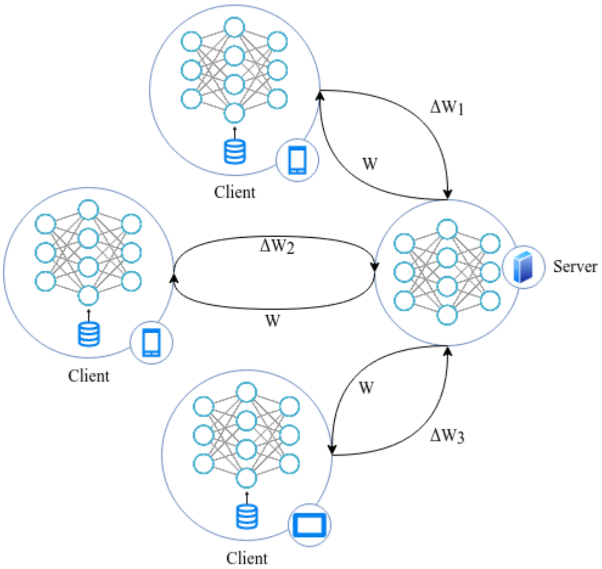

Figure 3 - Federated Learning training process.

The Federated Learning System (FLS) in the AIH Back-End offers centralized learning processes and services. 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.

Annex 4 - MPAI Basic

In recent years, Artificial Intelligence (AI) and related technologies have been introduced in 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 rule, MPAI standards include four documents: Technical Specification, Reference Software Specifications, Conformance Testing Specifications, and Performance Assessment Specifications. The last type of Specification includes standard operating procedures to enable users of MPAI Implementations to make informed decision about their applicability based on the notion of Performance, defined as a set of attributes characterising a reliable and trustworthy implementation.

In the following, if a Term begins with a small letter, it has the commonly used meaning and if with a capital letter, it has either the meaning defined in Table 1 - MPAI-AIH Terms if it is specific to this Technical Report and in *Table 3* if it is common to all MPAI Standards.

In general, MPAI Application Standards are defined as aggregations – called AI Workflows (AIW) – of processing elements – called AI Modules (AIM) – executed in an AI Framework (AIF). 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 AIF that executes an AIW. The AIW and its AIMs may have 3 Levels:
Level 1 – Implementer-specific and satisfying the MPAI-AIF Standard.
Level 2 – Specified by an MPAI Application Standard.
Level 3 – Specified by an MPAI Application Standard and certified 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 [1].

Figure 4 depicts the MPAI-AIF Reference Model under which Implementations of MPAI Application Standards and user-defined MPAI-AIF Conforming applications operate [7].

MPAI Application Standards normatively specify the 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.

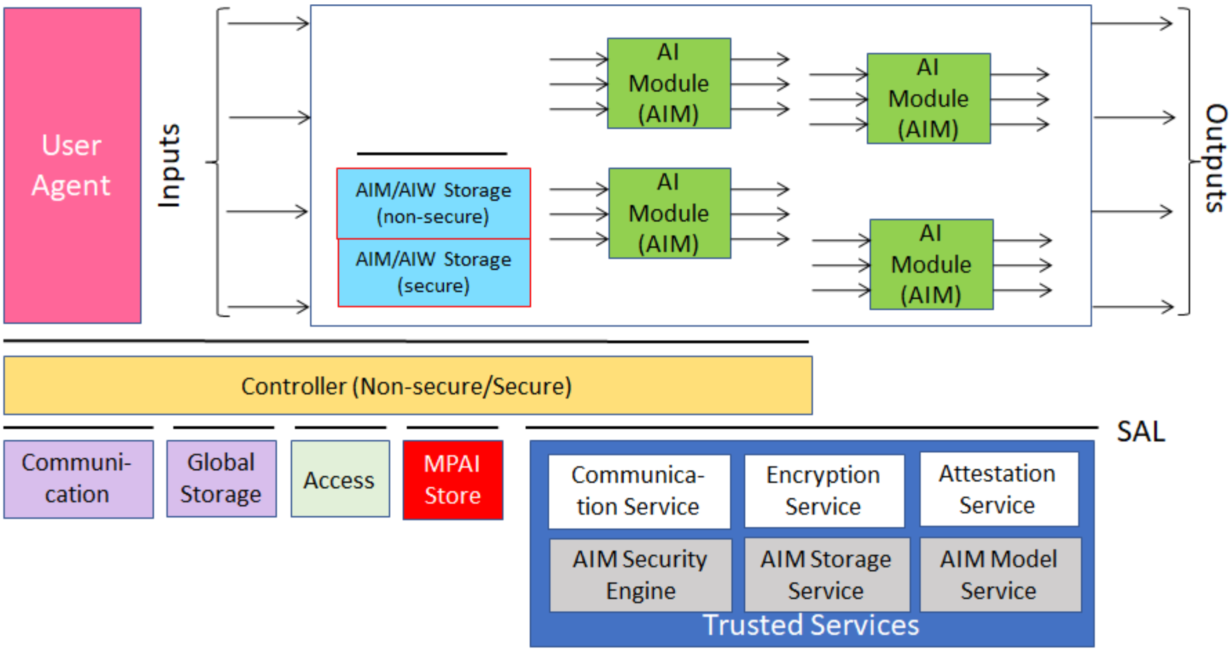

Figure 4 – The AI Framework (MPAI-AIF) V2 Reference Model

It should be noted that 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.

MPAI Standards are designed to enable a User to obtain, via standard protocols, an Implementation of an AIW and of the set of corresponding AIMs and execute it in an AIF Implementation. The MPAI Store in *Figure 4* is the entity from which Implementations are downloaded. MPAI Standards assume that the AIF, AIW, and AIM Implementations may have been developed by independent implementers. A necessary condition for this to be possible, is that any AIF, AIW, and AIM implementations be uniquely identified. MPAI has appointed an ImplementerID Registration Authority (IIDRA) to assign unique ImplementerIDs (IID) to Implementers.[1]

A necessary condition to make possible the operations described in the paragraph above is the existence of an ecosystem composed of Conformance Testers, Performance Assessors, the IIDRA and an instance of the MPAI Store. Reference [1] provides an example of such ecosystem.

[1] At the time of publication of this Technical Report, the MPAI Store was assigned as the IIDRA.