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

# MPAI Reference Software

# AI Framework
# MPAI-AIF

**V1.0**

**WARNING**

Use of the technologies described in this Technical Specification may infringe patents, copyrights or intellectual property rights of MPAI Members or non-members.

MPAI and its Members accept no responsibility whatsoever for damages or liability, direct or consequential, which may result from the use of this Technical Specification.

Readers are invited to review Annex 2 – Notices and Disclaimers.

# AI Framework
# V1.0


1 Introduction (Informative) ..... 2
2 Scope of the MPAI-AIF Reference Software ..... 4
3 Terms and definitions (Normative) ..... 4
4 References ..... 6
4.1 Normative references ..... 6
4.2 Informative references ..... 6
5 Reference software architecture ..... 7
5.1 General ..... 7
6 Example application (Informative) ..... 8
6.1 Use case description ..... 8
6.2 JSON Metadata ..... 9
7 How to use the Reference Software ..... 9
7.1 Software installation ..... 9
7.2 MPAI Store Simulation ..... 10
7.3 Boot Process ..... 10
7.4 Software licence and location ..... 10
Annex 1 - MPAI-wide terms and definitions ..... 11
Annex 2 - Notices and Disclaimers Concerning MPAI Standards (Informative) ..... 14
Annex 3 - The Governance of the MPAI Ecosystem (Informative) ..... 16
Annex 4 - Patent declarations ..... 18
Annex 5 - JSON Metadata ..... 19
AIF Metadata ..... 19
AIW Metadata ..... 19


## 1 Introduction (Informative)

In recent years, Artificial Intelligence (AI) and related technologies have been applied to a broad range of applications, have started affecting the lives 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 an informed decision about their applicability. Central to this is the notion of Performance, defined as a set of attributes characterising a reliable and trustworthy implementation.

For the aforementioned reasons, to fully achieve the MPAI mission, Technical Specifications shall 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 Standard 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 a provider of Technical, Conformance and Performance Specifications.
- Implementers of MPAI standards.
- MPAI-appointed Performance Assessors.
- The MPAI Store which takes care of the secure distribution of validated Implementations.

*Figure 1* depicts the MPAI-AIF Reference Model under which Implementations of MPAI Application Standards and user-defined MPAI-AIF conforming applications operate.
An AIF Implementation allows the execution of AI Workflows (AIW), composed of basic processing elements called AI Modules (AIM).

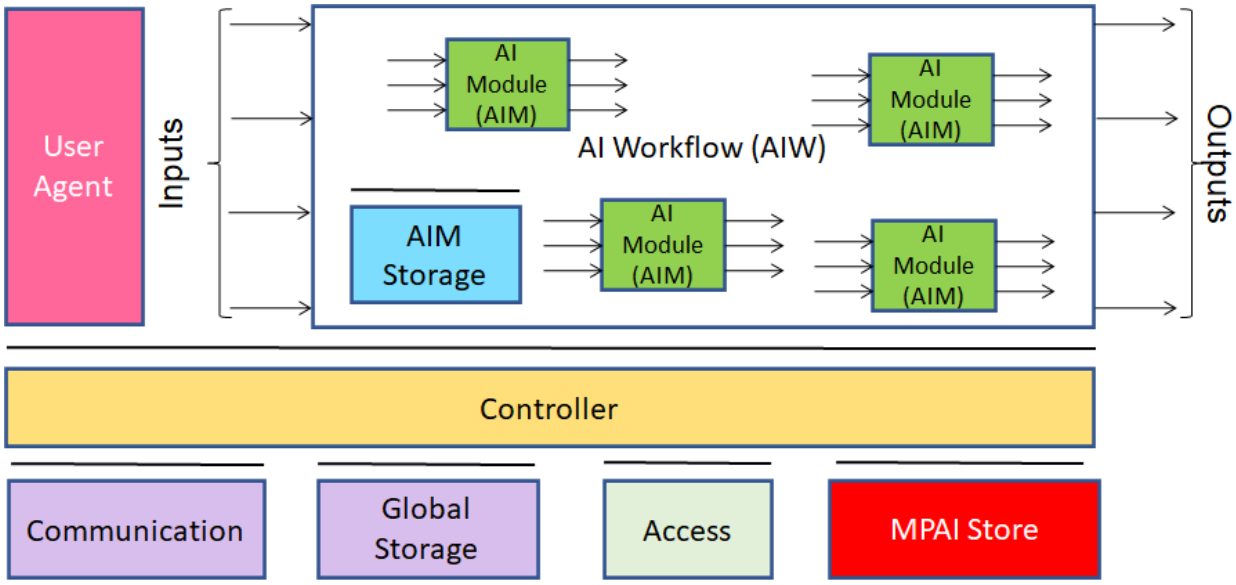


*Figure 1 - The AI Framework (AIF) Reference Model and its Components*

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.
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.
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* – 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 Annex 3.
The chapters and the annexes of this Technical Specification are Normative unless they are labelled as Informative.

# 2 Scope of the MPAI-AIF Reference Software

This Reference Software Specification document specifies the use of the Reference Software Implementation (in the following" Software") of the *AI Framework (MPAI-AIF)* Technical Specification [2]. This Reference Software Specification document is composed of four elements:

1. The Reference Software Specification of the MPAI-AIF AIF Technical Specification (this document) downloadable from https://mpai.community/standards/resources/.
2. The associated Reference Software Implementation as an integral part of this document downloadable from [5].
3. A simulated MPAI Store, downloadable from [6].
4. JSON Metadata downloadable from [7].

The Software is composed of:

1. A source code implementation of the MPAI-AIF (normative).
2. A set of libraries required by the Reference Software for execution in a particular environment (informative).
3. A demo implementation of one AIW and its AIMs (informative).
4. Associated metadata of the AIW and the AIMs (informative).
5. Documentation (informative).

The Reference Software is released as source code according to the MPAI modified BSD licence as provided by the General Licence for MPAI Software 1.0 [3] and with the following *disclaimers*:

1. The sole purpose of the Software is to show that a working Implementation of the *MPAI AI Framework (MPAI-AIF)* [4] exists. MPAI disclaims the suitability of the Software for any other purposes and/or its security (See Annex 3 - ).
2. The Software does not aim to provide a ready-to-use product, rather it provides an Implementation of the *MPAI AI Framework (MPAI-AIF)* Technical Specification [4].
3. The only purpose of the demo software of the AIW and the AIMs is to provide an example of a technically correct Implementation.
4. The Software references third-party software such as Zephyr and ST Microelectronics HAL. Users shall verify that they have the right to use such third-party software when exercising the Software.

This version of the MPAI-AIF Reference Software Specification has been developed by the MPAI *AI Framework* Development Committee (AIF-DC).

MPAI may decide to produce new Versions of the MPAI-AIF Reference Software Specification.

# 3 Terms and definitions (Normative)

The Terms used in this Reference Software Specification whose first letter is capital are defined in *Table 1*. The Terms of MPAI-wide applicability are defined in Table 2.

Table 1 - MPAI-AIF 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 Module (AIM) | A processing element receiving AIM-specific Inputs and producing AIM-specific Outputs according to its Function. An AIM may be an aggregation of AIMs. AIMs operate in the Trusted Zone. |

| | |
|---|---|
| AI Workflow (AIW) | A structured aggregation of AIMs implementing a Use Case receiving AIM-specific inputs and producing AIM-specific outputs according to its Function. AIWs operate in the Trusted Zone. |
| AIF Metadata | The data set describing the capabilities of an AIF set by the AIF Implementer. |
| AIM Metadata | The data set describing the capabilities of an AIM set by the AIM Implementer. |
| AIM Storage | A Component to store data of individual AIMs. An AIM may only access its data. The AIM Storage is part of the Trusted Zone. |
| AIW Metadata | The data set describing the capabilities of an AIW set by the AIW Implementer. |
| Channel | A physical or logical connection between an output Port of an AIM and an input Port of an AIM. The term "connection" is also used as synonymous. Channels are part of the Trusted Zone. |
| Communication | The infrastructure that implements message passing between AIMs. Communication operates in the Trusted Zone. |
| Component | One of the 9 AIF elements: Access, AI Module, AI Workflow, Communication, Controller, AIM Storage, Shared Storage, MPAI Store, and User Agent. |
| Controller | A Component that manages and controls the AIMs in the AIWs, so that they execute in the correct order and at the time when they are needed. The Controller operates in the Trusted Zone. |
| Device | A hardware and/or software entity running at least one instance of an AIF. |
| Event | An occurrence acted on by an Implementation. |
| External Port | An input or output Port simulating communication with an external Controller. |
| Knowledge Base | Structured and/or unstructured information made accessible to AIMs via MPAI-specified interfaces. |
| Message | A sequence of Records. |
| MPAI Ontology | A dynamic collection of terms with defined semantics managed by MPAI. |
| MPAI Server | A remote machine executing one or more AIMs. |
| MPAI Store | The repository of Implementations. |
| Port | A physical or logical communication interface of an AIM. |
| Record | Data with a specified Format. |
| Resource policy | The set of conditions under which specific actions may be applied. |
| Shared Storage | A Component to store data shared among AIMs. The Shared Storage is part of the Trusted Zone. |
| Status | The set of parameters characterising a Component. |
| Structure | A composition of Records |
| Swarm Element | An AIF in in a proximity-based scenario. |
| Time Base | The protocol specifying how Components can access timing information. The \Time Base is part of the Trusted Zone. |
| Topology | The set of Channels connecting AIMs in an AIW. |
| Trusted Zone | An environment that contains only trusted objects, i.e., objects that do not require further authentication. |
| User Agent | The Component interfacing the user with an AIF through the Controller |
| Zero Trust | A cybersecurity model primarily focused on data and service protection that assumes no implicit trust [21]. |

# 4 References

## 4.1 Normative references

MPAI-AIF normatively references the following documents:


1. Technical Specification: The Governance of the MPAI Ecosystem (MPAI-GME) V1; https://mpai.community/standards/mpai-gme/.
2. Technical Specification: Artificial Intelligence Framework (MPAI-AIF) V1; https://mpai.community/standards/mpai-aif/
3. The MPAI General Software Licence; https://mpai.community/about/licence/
4. MPAI; Technical Specification: AI Framework (MPAI-AIF) V1.1; https://mpai.community/standards/mpai-aif/.
5. MPAI-AIF Reference software; https://mpai.community/wp-content/uploads/2022/06/refsw.zip
6. MPAI Store Simulation; https://mpai.community/wp-content/uploads/2022/06/mpai_store.zip
7. MPAI-AIF Metadata; https://mpai.community/wp-content/uploads/2022/06/metadata.zip
8. GIT protocol; https://git-scm.com/book/en/v2/Git-on-the-Server-The-Protocols.
9. ZIP format, https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT.
10. Date and Time in the Internet: Timestamps; IETF RFC 3339; July 2002.
11. Uniform Resource Identifiers (URI): Generic Syntax, IETF RFC 2396, August 1998.
12. The JavaScript Object Notation (JSON) Data Interchange Format; https://datatracker.ietf.org/doc/html/rfc8259; IETF rfc8259; December 2017
13. JSON Schema; https://json-schema.org/.
14. BNF Notation for syntax; https://www.w3.org/Notation.html
15. Constrained Application Protocol (CoAP); https://coap.technology/.
16. MPAI; The MPAI Statutes; https://mpai.community/statutes/.
17. MPAI; The MPAI Patent Policy; https://mpai.community/about/the-mpai-patent-policy/.
18. Framework Licence of the Artificial Intelligence Framework Technical Specification (MPAI-AIF); https://mpai.community/standards/mpai-aif/framework-licence/.


## 4.2 Informative references


19. IETF; RFC 7252, Constrained Application Protocol (CoAP); https://datatracker.ietf.org/doc/html/rfc7252
20. Message Passing Interface (MPI), https://www.mcs.anl.gov/research/projects/mpi/
21. Rose, Scott; Borchert, Oliver; Mitchell, Stu; Connelly, Sean; "Zero Trust Architecture"; https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf
22. MPAI Technical Specification; Multimodal Conversation (MPAI-CAE) V1.
23. W. Wang, J. Gao, M. Zhang, S. Wang, G. Chen, T. K. Ng, B. C. Ooi, J. Shao, and M. Reyad, “Rafiki: machine learning as an analytics service system,” Proceedings of the VLDB Endowment, vol. 12, no. 2, pp. 128–140, 2018.
24. Y. Lee, A. Scolari, B.-G. Chun, M. D. Santambrogio, M. Weimer, and M. Interlandi; PRETZEL: Opening the black box of machine learning prediction serving systems; in 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI18), pp. 611–626, 2018.
25. ML.NET [ONLINE]; https://dotnet.microsoft.com/apps/machinelearning-ai/ml-dotnet.
26. D. Crankshaw, X. Wang, G. Zhou, M. J. Franklin, J. E. Gonzalez, and I. Stoica; Clipper: A low-latency online prediction serving system; in NSDI, pp. 613–627, 2017.
27. S. Zhao, M. Talasila, G. Jacobson, C. Borcea, S. A. Aftab, and J. F. Murray; Packaging and sharing machine learning models via the acumos ai open platform; in 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 841–846, IEEE, 2018.
28. Apache Prediction I/O; https://predictionio.apache.org/.

29. D.Sculley, G.Holt, D.Golovin, E. Davydov, T.Phillips, D.Ebner, V. Chaudhary, M. Young, J. Crespo, D.Dennison; Hidden technical debt in Machine learning systems Share; on NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2; December 2015 Pages 2503–2511

# 5 Reference software architecture

## 5.1 General

The Reference Software Implementation of the normative MPAI-AIF Architecture of Figure 1 implements the reference diagram of Figure 2.

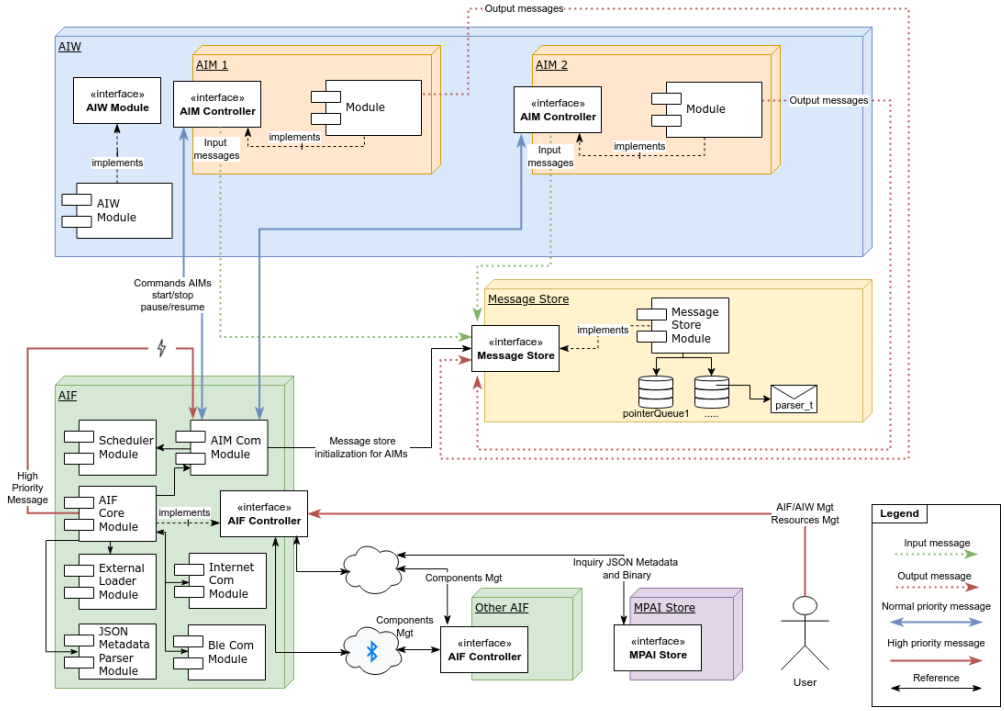


*Figure 2 - Implementation architecture of MPAI-AIF (should originate from a clean figure)*

The Reference Software Implementation has the following characteristics

- Is implemented in C language.
- Implements only MPAI Events (High Priority events)
- Implements messages and a message store.
- Implements an IP communication interface based on CoAP (Constrained Application Protocol [6]) that simulates the MPAI Store via a CoAP server.
- Implements a subset of the MPAI APIs in the form of libraries.
    - Controller API called by User Agent
        - `MPAI_AIFU_Controller_Initialize`
        - `MPAI_AIFU_Controller_Destroy`
        - `MPAI_AIFU_AIW_Start`
        - `MPAI_AIFU_AIW_Pause`
        - `MPAI_AIFU_AIW_Resume`
        - `MPAI_AIFU_AIW_Stop`
        - `MPAI_AIFU_AIM_GetStatus`
    - Controller API called by AIMs
        - `MPAI_AIFM_AIM_Start`
        - `MPAI_AIFM_AIM_Pause`
        - `MPAI_AIFM_AIM_Resume`

    - `MPAI_AIFM_AIM_Stop`
- Parses the JSON Metadata files.

# 6 Example application (Informative)

## 6.1 Use case description

The informative part of the Reference Software depicted in Figure 3 implements a Rehabilitation use case wher the user is requested to perform specific movements in sync with an audio clue.

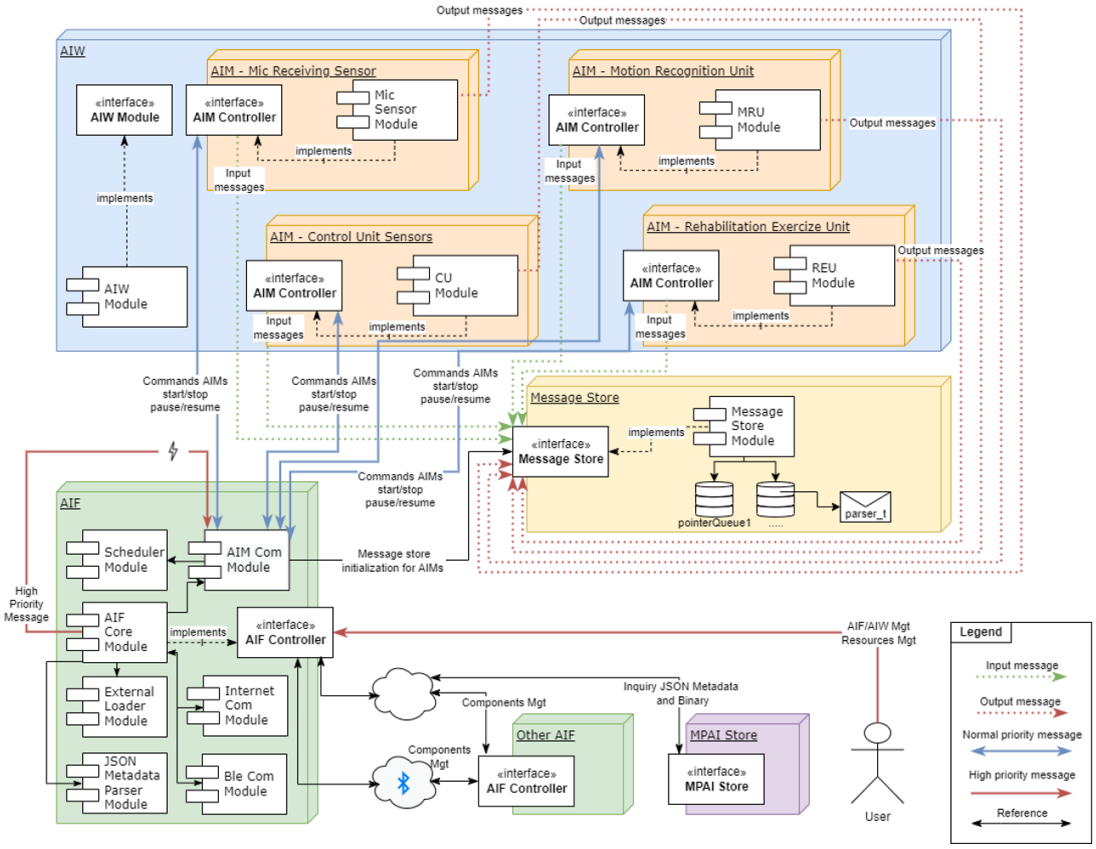


Figure 3 - AIF with a test AIW

The AIW implementing the use case includes the following AIMs:
**VolumePeaksAnalysis:** uses microphones to identify audio pattern (in this case volume peaks) and publish it to the message store's MicPeakDataChannel channel of the.
**ControlUnitSensorsReading:** reads data from all device sensors (like temperature, acceleration, pressure and others) and publish the values to the message store's SensorsDataChannel channel.
**MotionRecognitionAnalysis:** uses data from inertial unit, coming from SensorsDataChannel, to detect motion events such as start, stop etc and publish them to the message store's MotionData-Channel channel.
**MovementsWithAudioValidation:** uses data, cross-referencing it between MotionDataChannel and MicPeakDataChannel, to recognize if the movement is done in a correct way. In particular, it detects a stop event and waits for a volume peak at maximum for 1sec (configurable). It also blinks alternately the LEDs to alert the error.

The system listens via the VolumePeaksAnalysis AIM to the audio signal for specific patterns (i.e., a low frequency impulse coming from a metronome) and simultaneously monitors the movement patterns via the MotionRecognitionAnalysis AIM. The MovementsWithAudioValidation AIM monitors if the movement is detected as correct and executed in synchronization with the specific audio pattern. In all cases, a message is output via the serial port.

### 6.2 JSON Metadata

The Rehabilitation AIW JSON Metadata is given in Annex 5 - .

## 7 How to use the Reference Software

The Software can be executed on the Zephyr real-time operating system (RTOS) using the STM32 Hardware Abstraction Layer (HAL) libraries, on the ST IoT NODE https://www.st.com/en/evaluation-tools/b-l475e-iot01a.html

### 7.1 Software installation

1) Install dependencies:
    a) *cmake* (3.20.0 or above)
    b) *python* (3.6 or above)
2) Install *udev* rules from here
3) Download MPAI Server CoAP jar
4) Run MPAI Server CoAP ($IP_ADDRESS is the CoAP endpoint):

```
java -Dmpai.store.host=$IP_ADDRESS -jar coap-server-0.0.1-SNAPSHOT.jar
```

5) Install PlatformIO Core (v2.7.0) here
6) Configure WLAN (if requested), creating a file *wifi_config.c* like below:

```
#include "wifi_config.h"
char* AUTO_CONNECT_SSID = "<SSID>";
char* AUTO_CONNECT_SSID_PSK = "<PASSWORD>";
```

7) Run these commands:

```
# Change directory to source
> cd iotnode_box_test_sensors

# Build project
> platformio run

# Upload firmware
> platformio run --target upload

# Build specific environment
> platformio run -e disco_l475vg_iot01a

# Upload firmware for the specific environment
> platformio run -e disco_l475vg_iot01a --target upload

# Clean build files
> platformio run --target clean
```

## 7.2 MPAI Store Simulation

The MPAI STORE functionality is simulated via the delivery of the AIW JSON metadata over CoAP/IP, while the AIMs reside on the board as described above. A Java implementation of the CoAP server simulates the MPAI STORE. The source code can be downloaded from https://mpai.community/standards/resources/
In order to run the simulated MPAI STORE ($IP_ADDRESS is the CoAP endpoint):

java -Dmpai.store.host=$IP_ADDRESS -jar coap-server-0.0.1-SNAPSHOT.jar

## 7.3 Boot Process

Implementing the MPAI-AIF specification, the system at the boot time:

- Reads the AIF configuration from MPAI Store
- Reads the AIW configuration from MPAI Store:
  AIW name, Topology, identifying which channel is connected with respective AIM
  List of AIM's used

For each AIM:

- Reads the configuration from MPAI Store
- Initialize it
- Start it

## 7.4 Software licence and location

The Reference Software is released with the MPAI Software Licence and is available from https://www.mpai.community/resources/.

# 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 *1* are 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 Module (AIM) | A processing element receiving AIM-specific Inputs and producing AIM-specific Outputs according to according to its Function. An AIM may be an aggregation of AIMs. |
| AI Workflow (AIW) | A structured aggregation of AIMs implementing a Use Case receiving AIW-specific inputs and producing AIW-specific inputs according to its Function. |
| AIF Metadata | The data set describing the capabilities of an AIF set by the AIF Implementer. |
| AIM Metadata | The data set describing the capabilities of an AIM set by the AIM Implementer. |
| Application Programming Interface (API) | A software interface that allows two applications to talk to each other |
| Application Standard | An MPAI Standard specifying AIWs, AIMs, Topologies and Formats suitable for a particular application domain. |
| Channel | A physical or logical connection between an output Port of an AIM and an input Port of an AIM. The term "connection" is also used as a synonym. |
| Communication | The infrastructure that implements message passing between AIMs. |
| Component | One of the 9 AIF elements: Access, AI Module, AI Workflow, Communication, Controller, Internal Storage, Global Storage, MPAI Store, and User Agent. |
| Conformance | The attribute of an Implementation of being a correct technical Implementation of a Technical Specification. |
| Conformance Tester | An entity authorised by MPAI to Test 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 | Information in digital form. |
| Data Format | The standard digital representation of Data. |
| Data Semantics | The meaning of Data. |

| | |
|---|---|
| Device | A hardware and/or software entity running at least one instance of an AIF. |
| Ecosystem | The ensemble of the following actors: MPAI, MPAI Store, Implementers, Conformance Testers, Performance Testers and Users of MPAI-AIF Implementations as needed to enable an Interoperability Level. |
| Event | An occurrence acted on by an Implementation. |
| 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. |
| Identifier | A name that uniquely identifies an Implementation. |
| Implementation | 1. An embodiment of the MPAI-AIF Technical Specification, or<br>2. An AIW or AIM of a particular Level (1-2-3). |
| Internal Storage | A Component to store data of the individual AIMs. |
| Interoperability | The ability to functionally replace an AIM/AIW with another AIM/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 be:<br>1. Implementer-specific and satisfying the MPAI-AIF Standard *(Level 1)*.<br>2. Specified by an MPAI Application Standard (*Level 2)*.<br>3. Specified by an MPAI Application Standard and certified by a Performance Assessor (*Level 3)*. |
| Knowledge Base | Structured and/or unstructured information made accessible to AIMs via MPAI-specified interfaces |
| Message | A sequence of Records. |
| 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 procedures, the tools, the data sets and/or the data set characteristics 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 authorised by MPAI to Assess the Performance of an Implementation in a given Application domain |
| Port | A physical or logical communication interface of an AIM. |
| 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 | Data 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. |
| Specification | A collection of normative clauses. |
| 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.<br>(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:<br>1. The formats of the Input/Output data of the AIWs implementing the AIWs.<br>2. The Connections of the AIMs of the AIW.<br>3. The formats of the Input/Output data of the AIMs belonging to the AIW. |
| Testing Laboratory | A laboratory accredited by MPAI to Assess the Grade of Performance of Implementations. |
| Time Base | The protocol specifying how AIF 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. |
| Zero Trust | A cybersecurity model primarily focused on data and service protection that assumes no implicit trust. |

# Annex 2 - Notices and Disclaimers Concerning MPAI Standards (Informative)

The notices and legal disclaimers given below shall be borne in mind when downloading and using approved MPAI Standards.

In the following, "Standard" means the collection of four MPAI-approved and published 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. 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. 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 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. 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 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.

# Annex 3 - 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 implement 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:<br>- AIFs conforming to MPAI-AIF.<br>- 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.<br>- Verifies Implementations' security, e.g., absence of malware.<br>- Indicates unambiguously that Implementations are Level 1. |

**Level 2 Interoperability**

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

| | |
|---|---|
| Implementers' benefits | Upload to the MPAI Store and have globally distributed Implementations of<br>- AIFs conforming to MPAI-AIF.<br>- AIWs and AIMs conforming to MPAI Application Standards. |
| Users' benefits | - Rely on Implementations of AIWs and AIMs whose Functions have been reviewed during standardisation.<br>- 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.<br>- Competition of AIW and AIM Implementations fosters AI innovation. |
| MPAI Store's role | - Tests Conformance of Implementations with the relevant MPAI Standard.<br>- Verifies Implementations' security.<br>- 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<br>- Indicates unambiguously that Implementations are Level 3. |

**The MPAI ecosystem**

Figure 4 is a high-level description of the MPAI ecosystem operation applicable to fully conforming MPAI implementations as specified in the Governance of the MPAI Ecosystem Specification [1]:

1. MPAI establishes and controls the not-for-profit MPAI Store.
2. MPAI appoints Performance Assessors.
3. MPAI publishes Standards.
4. Implementers submit Implementations to Performance Assessors.
5. If the Implementation Performance is acceptable, Performance Assessors inform Implementers and the MPAI Store.
6. Implementers submit Implementations to the MPAI Store tested for Conformance and security.
7. Users download and use Implementations and submit experience scores.

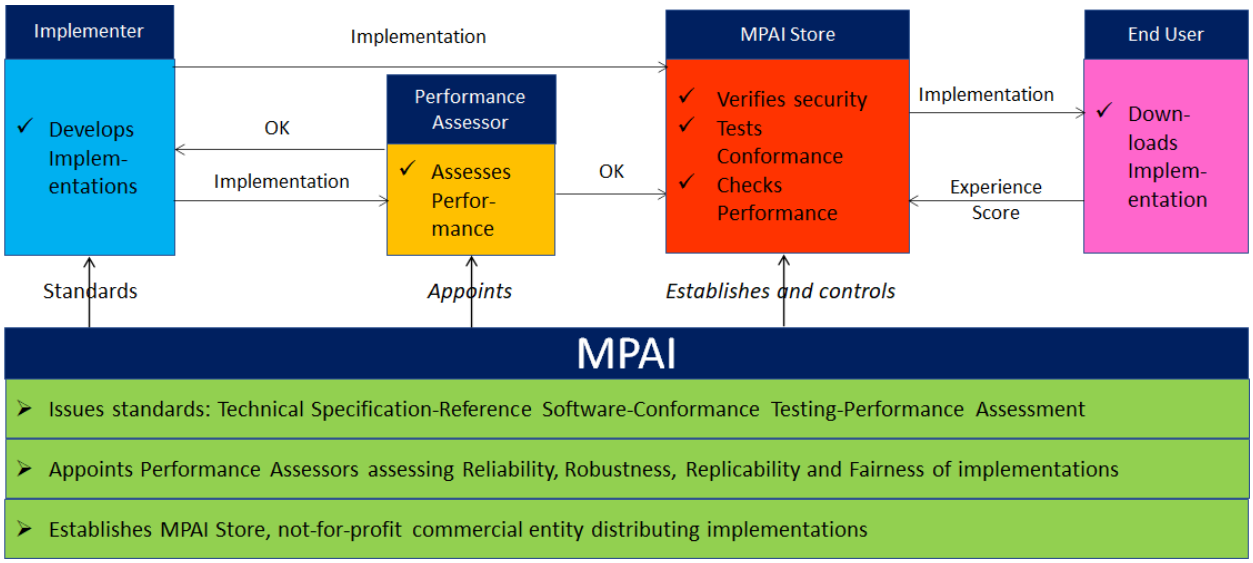


*Figure 4 - The MPAI ecosystem operation*

# Annex 4 - Patent declarations

The MPAI Artificial Intelligence Framework (MPAI-AIF) Technical Specification has been developed according to the process outlined in the MPAI Statutes [16] and the MPAI Patent Policy [17].
The following entities have agreed to licence their standard essential patents reading on the MPAI Artificial Intelligence Framework (MPAI-AIF) Technical Specification according to the MPAI-AIF Framework Licence [18]:

| Entity | Name | Email address |
|---|---|---|
| Speech Morphing, Inc. | Fathy Yassa | fathy@speechmorphing.com |

# Annex 5 - JSON Metadata

## AIF Metadata


```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id":        "https://mpai.community/standards/resources/MPAI-AIF/V1/AIF-metadata.schema.json",
  "title": "MPAI-AIF V1 AIF metadata",
  "ImplementerID": 1,
  "Version": "v1.0",
  "APIProfile": "Main",
  "ResourcePolicies": [
    {
      "Name": "Memory",
      "Minimum": "50000",
      "Maximum": "120000",
      "Request": "80000"
    },
    {
      "Name": "CPUNumber",
      "Minimum": "1",
      "Maximum": "2",
      "Request": "1"
    },
    {
      "Name": "CPU:Class",
      "Minimum": "Low",
      "Maximum": "High",
      "Request": "Low"
    }
  ],
  "Authentication": "admin",
  "TimeBase": "NTP"
}
```


## AIW Metadata


```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id":    "https://mpai.community/standards/resources/MPAI-AIF/V1/AIW-AIM-metadata.schema.json",
  "title": "CAE AIF v1 AIW/AIM metadata",
  "Identifier": {
    "ImplementerID": 1,
    "Specification": {
      "Standard": "MPAI-CAE",
      "AIW": "CAE-REV",
      "AIM": "CAE-REV",
      "Version": "1"
    }
  },
  "APIProfile": "Main",
  "Description": "AIW that implements Use-Case CAE-REV (Rehabilitation Exercises Validation)",
  "Types": [
    {
      "Name": "Sensors_Data_t",
      "Type": "mpai_message_t"
```

```
    },
    {
      "Name": "Mic_Buffer_Data_t",
      "Type": " mpai_message_t "
    },
    {
      "Name": "Mic_Peak_Data_t",
      "Type": " mpai_message_t "
    },
    {
      "Name": "Motion_Data_t",
      "Type": " mpai_message_t "
    }
  ],
  "Ports": [
    {
      "Name": "SensorsDataChannel",
      "Direction": "InputOutput",
      "RecordType": "Sensors_Data_t",
      "Technology": "Software",
      "Protocol": "",
      "IsRemote": false
    },
    {
      "Name": "MicBufferDataChannel",
      "Direction": "InputOutput",
      "RecordType": "Mic_Buffer_Data_t",
      "Technology": "Software",
      "Protocol": "",
      "IsRemote": false
    },
    {
      "Name": "MicPeakDataChannel",
      "Direction": "InputOutput",
      "RecordType": "Mic_Peak_Data_t",
      "Technology": "Software",
      "Protocol": "",
      "IsRemote": false
    },
    {
      "Name": "MotionDataChannel",
      "Direction": "InputOutput",
      "RecordType": "Motion_Data_t",
      "Technology": "Software",
      "Protocol": "",
      "IsRemote": false
    }
  ],
  "Topology": [
    {
      "Output": {
        "AIMName": "MotionRecognitionAnalysis",
        "PortName": "SensorsDataChannel"
      },
      "Input": {
        "AIMName": "ControlUnitSensorsReading",
        "PortName": "SensorsDataChannel"
      }
    },
    {
      "Output": {
```

```
        "AIMName": "MovementsWithAudioValidation",
        "PortName": "MicPeakDataChannel"
      },
      "Input": {
        "AIMName": "VolumePeaksAnalysis",
        "PortName": "MicPeakDataChannel"
      }
    },
    {
      "Output": {
        "AIMName": "",
        "PortName": "MicBufferDataChannel"
      },
      "Input": {
        "AIMName": "VolumePeaksAnalysis",
        "PortName": ""
      }
    },
    {
      "Output": {
        "AIMName": "MovementsWithAudioValidation",
        "PortName": "MotionDataChannel"
      },
      "Input": {
        "AIMName": "MotionRecognitionAnalysis",
        "PortName": "MotionDataChannel"
      }
    }
  ],
  "SubAIMs": [
    {
      "Name": "VolumePeaksAnalysis",
      "Identifier": {
        "ImplementerID": 1,
        "Specification": {
          "Standard": "MPAI-CAE",
          "AIW": "CAE-REV",
          "AIM": "VolumePeaksAnalysis",
          "Version": "1"
        }
      }
    },
    {
      "Name": "ControlUnitSensorsReading",
      "Identifier": {
        "ImplementerID": 1,
        "Specification": {
          "Standard": "MPAI-CAE",
          "AIW": "CAE-REV",
          "AIM": "ControlUnitSensorsReading",
          "Version": "1"
        }
      }
    },
    {
      "Name": "MotionRecognitionAnalysis",
      "Identifier": {
        "ImplementerID": 1,
        "Specification": {
          "Standard": "MPAI-CAE",
          "AIW": "CAE-REV",
```

```
            "AIM": "MotionRecognitionAnalysis",
            "Version": "1"
          }
        }
      },
      {
        "Name": "MovementsWithAudioValidation",
        "Identifier": {
          "ImplementerID": 1,
          "Specification": {
            "Standard": "MPAI-CAE",
            "AIW": "CAE-REV",
            "AIM": "MovementsWithAudioValidation",
            "Version": "1"
          }
        }
      }
    ],
    "Implementations": [
      {
        "BinaryName": "firmware.bin",
        "Architecture": "arm",
        "OperatingSystem": "Zephyr RTOS",
        "Version": "v0.1",
        "Source": "AIMStorage",
        "Destination": ""
      }
    ],
    "ResourcePolicies": [
      {
        "Name": "Memory",
        "Minimum": "50000",
        "Maximum": "120000",
        "Request": "80000"
      },
      {
        "Name": "CPUNumber",
        "Minimum": "1",
        "Maximum": "2",
        "Request": "1"
      },
      {
        "Name": "CPU:Class",
        "Minimum": "Low",
        "Maximum": "High",
        "Request": "Low"
      }
    ],
    "Documentation": [
      {
        "Type": "Tutorial",
        "URI": "https://mpai.community/standards/mpai-cae/"
      }
    ]
  }
```


# AIM VolumePeaksAnalysis Metadata


```
  {
    "Identifier": {
      "ImplementerID": 1,
```

```
      "Specification": {
        "Name": "CAE",
        "AIW": "REV",
        "AIM": "VolumePeaksAnalysis",
        "Version": "1"
      }
    },
    "Description": "This AIM implements analysis transform function for CAE-
REV that recognizes volume peaks from microphone array audio.",
    "Ports": [],
    "Topology": [],
    "SubAIMs": [],
    "Topology": [],
    "Implementations": [],
    "Documentation": [
      {
        "Type": "Tutorial",
        "URI": "https://mpai.community/standards/mpai-cae/"
      }
    ]
  }
```

## AIM ControlUnitSensorsReading Metadata

```
{
  "Identifier": {
    "ImplementerID": 1,
    "Specification": {
      "Name": "CAE",
      "AIW": "REV",
      "AIM": "ControlUnitSensorsReading",
      "Version": "1"
    }
  },
  "Description": "This AIM implements sensor readings from control unit.",
  "Ports": [],
  "Topology": [],
  "SubAIMs": [],
  "Topology": [],
  "Implementations": [],
  "Documentation": [
    {
      "Type": "Tutorial",
      "URI": "https://mpai.community/standards/mpai-cae/"
    }
  ]
}
```

## AIM MotionRecognitionAnalysis Metadata

```
{
  "Identifier": {
    "ImplementerID": 1,
    "Specification": {
      "Name": "CAE",
      "AIW": "REV",
      "AIM": "MotionRecognitionAnalysis",
      "Version": "1"
    }
  },
```

```
  "Description": "This AIM implements motion recognition analysing data from
inertial unit.",
  "Ports": [],
  "Topology": [],
  "SubAIMs": [],
  "Topology": [],
  "Implementations": [],
  "Documentation": [
    {
      "Type": "Tutorial",
      "URI": "https://mpai.community/standards/mpai-cae/"
    }
  ]
}
```


# AIM MovementsWithAudioValidation Metadata


```
{
  "Identifier": {
    "ImplementerID": 1,
    "Specification": {
      "Name": "CAE",
      "AIW": "REV",
      "AIM": "MovementsWithAudioValidation",
      "Version": "1"
    }
  },
  "Description": "This AIM implements a validation of limbs movements during
rehabilitation exercises, according to music rhythm",
  "Ports": [],
  "Topology": [],
  "SubAIMs": [],
  "Topology": [],
  "Implementations": [],
  "Documentation": [
    {
      "Type": "Tutorial",
      "URI": "https://mpai.community/standards/mpai-cae/"
    }
  ]
}
```