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

MPAI Reference Software

Neural Network Watermarking
MPAI-NNW

WD0.2

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.

Reference Software - Neural Network Watermarking V1

1 Introduction (Informative)

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. Sometimes Technical Reports are produced to provide informative guidance in specific areas for which the development of standards in premature.

Performance Assessment Specifications include 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, Terms beginning with a capital letter are defined in *Table 1* if they are specific to this Standard and in *Table 3* if they are 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 reference [6].

Figure 1 depicts the MPAI-AIF Reference Model under which Implementations of MPAI Application Standards and user-defined MPAI-AIF Conforming applications operate. MPAI is currently developing MPAI-AIF V2 that will compatibly extend MPAI-AIF V1 with security support.

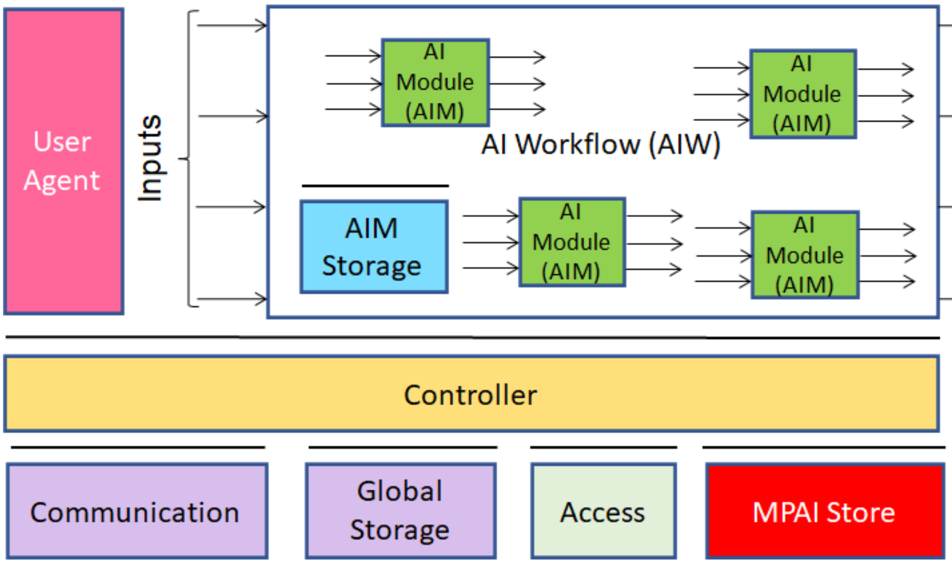

Figure 1 – The AI Framework (AIF) Reference Model and its Components

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 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 1* is an 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, an instance of the IIDRA and of the MPAI Store. Reference [6] provides an informative example of such ecosystem.

The chapters and the annexes of this Technical Specification are Normative, unless they are labelled as Informative.

2 Scope

The MPAI-NNW specifies methodologies to Evaluate the following aspects of a neural network watermarking technology:

- The impact on the performance of a watermarked neural network and its inference.
- The ability of a neural network watermarking detector/decoder to detect/decode a payload when the watermarked neural network has been modified.
- The computational cost of injecting a watermark, and detecting or decoding the payload of the watermarked neural network.

The Reference Software is a Python implementation of the functions specified by the MPAI-NNW standard. The software is designed to evaluate the Imperceptibility, the Robustness and the Computational Cost of a given neural network watermarking technology for the image classification task. Users wishing to implement the standard for other tasks can use this Refence Software as a guide for an implementation for a different task. The Reference Software assumes that the list of neural network watermarking technology IDs of Annex 4 and read every time the software is run.

This Technical Specification has been developed by the MPAI Neural Network Watermarking Development Committee (NNW-DC). As the neural network watermarking area is fast-evolving, MPAI expects it will produce future MPAI-NNW versions providing methods to cope with the evolution of the technology.

3 Terms and Definitions

The terms used in this standard whose first letter is capital have the meaning defined in *Table 1*.

Table 1 – Table of terms and definitions

Term	**Definition**
Computational cost	The cost of injecting, detecting or decoding a watermark in a neural network or its inference.
Evaluate	Assess a property of a neural network watermarking method using the procedure of [4].
Imperceptibility	A difference in the performance of a neural network before and after the watermark embedding process.
Means	Procedure, tools, dataset or dataset characteristics used to evaluate one or more of Computational cost, Imperceptibility, or Robustness of a neural network watermarking technology.
Modification	The result of a simulated attack performed during Neural Network Watermarking testing.
Neural Network	or NN Model, a set of interconnected information processing nodes

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

	whose connections are affected by Weights.
Neural Network Watermarking	The process of injecting a data payload in the Weights or the activation function of a Neural Network.
Parameter	A set of values characterizing the strength of a Modification.
Payload	The information carried by the watermark.
Robustness	The ability of a watermarked neural network to withstand the impact of modifications in terms of detection and decoding capability.
Tester	The user who evaluates a neural network watermarking technology according to this Technical Specification.
User	The entity Evaluating a Watermarking Technology.
Watermarking Technology Type ID	The identifier of the type of watermarking technology whose Evaluation is enabled by this Reference Software.
Weight	The value by which the connection between two nodes of a Neural Network is multiplied.

4 References

4.1 Normative references

MPAI-AIF normatively references the following documents:

1. MPAI; The MPAI Statutes; https://mpai.community/statutes/
2. MPAI; The MPAI Patent Policy; https://mpai.community/about/the-mpai-patent-policy/.
3. MPAI; Framework Licence of the Artificial Intelligence Framework Technical Specification (MPAI-AIF); https://mpai.community/standards/mpai-aif/framework-licence/
4. MPAI; Technical Specification – Neural Network Watermarking (MPAI-NNW) V1; https://mpai.community/standards/mpai-nnw/
5. PyTorch library; version 1.10 - https://pytorch.org/

4.2 Informative references

6. Technical Specification: The Governance of the MPAI Ecosystem V1, 2021; https://mpai.community/standards/mpai-gme/

5 Software architecture

A User wants to evaluate a neural network watermarking technology applied to a neural network, in the following called watermarked NN, designed for an image classification task.

5.1 Imperceptibility Evaluation

The User shall:

1. have a testing dataset.
2. run *Embedder* or *Embedder_onestep* (depending on the watermarking technology ID specified in Annex 4) to watermark the neural network.
3. run the *Test* function, and enter the watermarked neural network represented in the format of [5] and the testing dataset represented in the format of [5].

The *Test* function will return the following quality measurements:

- Probability of false alarm: $P_{fa} = \frac{fp}{tp+fp+fn+tn}$
- Precision: $\frac{tp}{tp+fp}$ and Recall: $\frac{tp}{tp+fn}$

- Probability of missed detection: $P_{md} = \frac{fn}{tp+fp+fn+tn}$

Hence, the imperceptibility evaluation is Evaluated by comparing the quality measurements of the unwatermarked NN and of the watermarked NN on the same testing dataset.

Test(neural network, testing data)
    return task-dependent quality of the produced inference

5.1.1 Watermark embedding is done after training

The User shall run the *Embedder* function on the trained neural network to be Evaluated and enter the ID of the watermarking technology. The *Embedder* will return the watermarked version of the neural network.

Embedder(*neural network, watermarking technology ID*)
    return watermarked NN

5.1.2 Watermark embedding is done during training

The User shall:

1. have the training dataset represented in the format of [5].
2. run the *Embedder_onestep* function on the trained or untrained neural network (depending on the ID of the watermarking technology type specified in Annex 4).
3. enter the training dataset and the ID of the watermarking technology type.
4. repeat 2. and 3. For the number of steps ***S*** specified by the watermarking technology being Evaluated.

The *Embedder_onestep* will return the neural network after one step of training.

Embedder_onestep(neural network, training_data, watermarking technology ID)
    return neural network

5.2 Robustness Evaluation

The User shall:

1. run *Modification* on the watermarked NN.
2. enter the Modification_ID and its Parameters provided in Table 2. *Modification* returns a modified version of the watermarked NN.
3. run *Detector* on the modified version of the watermarked NN.
4. enter the ID of the watermarking technology type. The *Detector* returns a Boolean: 0 means the watermark is detected and 1 means the watermark is not detected.
5. run *Decoder* on the modified version of the watermarked NN.
6. enter the ID of the watermarking technology type. The *Decoder* returns the retrieved watermark Payload.
7. compute the number of erred bits or symbols between the retrieved watermark Payload and the original watermark Payload.

Modification(Modification_ID, neural network, parameters)
    return modified neural network
Detector(neural network, watermarking technology ID)
    return the presence of the watermark or not
Decoder(neural network, watermarking technology ID)
    return the retrieved payload

Table 2. List of modification with their parameters

ID	Modification name	Parameter type	Parameter range
	Modification	Parameter type	Parameter range
0	*Gaussian noise addition*: adding a zero-mean, *S* standard deviation Gaussian noise to a layer in the NN model. This noise addition can be simultaneously applied to a sub-set of layers.	- the layers to be modified by Gaussian noise - the ratio of *S* to standard deviation of the weights in the corresponding layer	- 1 to total number of layers - 0.1 to 0.3
1	*L1 Pruning*: delete the *P*% of the smallest weights, irrespective of their layers.	- the *P* percentage of the deleted weights	- 1% to 90% - 1% to 99.99% when aiming one layer
2	*Random pruning*: delete *R*% of randomly selected weights, irrespective of their layers.	- the *R* percentage of the deleted weights	- 1% to 10%
3	*Quantizing:* reduce to B the number of bits used to represent the weights by **1.** reducing the number of bits based on a sequence of three operations: affine mapping from the weights interval to the $(\mathbf{0}; \mathbf{2}^{\boldsymbol{B}} - \mathbf{1})$ **2.** rounding to the closest integer **3.** backward affine mapping towards the initial weights interval	- the layers to be modified by quantization - the value of *B*	- 1 to total number of layers - 32 to 2
4	*Fine tuning / transfer learning*: resume the training of the *M* watermarked NNs submitted to test, for *E* additional epochs.	- ratio of *E* to the number of epochs in the initial training	- up to 0.5 time the total number of epochs
5	*Knowledge distillation:* train a surrogate network using the inferences of the NN under test as training dataset	- The structure of the architecture - The size of the dataset *D* - The number of epochs *E*	- structures N - 10,000 to 1,000,000 - 1 to 100
6	*Watermark overwriting:* successively insert *R* additional watermarks, with random payloads of the same size as the initial watermark	- *R* number of watermarks successively inserted	- 2 to 4

5.3 Computational cost Evaluation

The User shall either adopt one of the processing environments defined in [4] or define their own processing environment.

Then, the User shall:

1. run *Memory_footprint* on *Embedder* or *Embedder_onestep* or *Decoder* or *Detector* (depending on the watermarking technology ID specified in Annex 4). The *Memory_footprint* function returns the memory footprint of the module under test.
2. run *Time_module* on *Embedder* or *Embedder_onestep* or *Decoder* or *Detector* (depending on the watermarking technology ID specified in Annex 4). The *Time_module* function returns the time required to process the module under test.

6 Reference Software

6.1 Installation requirements

The required modules and their version are:

- python 3.6
- pytorch 1.10 & torchvision 0.11.3
- numpy 1.19.2
- psutils 5.9.3

6.2 Neural Network Watermarking method requirements

The neural network watermarking method shall be a Python class that contains:

- An *Embedder* (or *Embedder_one_step*) function that takes 2 arguments: the model represented in Pytorch format and the Python dictionary which contains all the elements related to the neural network watermarking technology under test.
- An *Embedder_one_step* (or *Embedder*) function that takes 5 arguments: the model, the training dataset, the optimizer, the cost function and the dictionary which contains all the elements related to the neural network watermarking technology under test. The first four elements are Pytorch elements represented in the format of [5] and the fifth is a Python dictionary.
- A *Detector* and/or *Decoder* function that takes 2 arguments: the model represented in Pytorch format and the Python dictionary which contains all the elements related to the neural network watermarking technology under test.

6.3 How to use the Reference Software

The Reference Software contains a folder for the training and testing dataset and four python files:

- *Imperceptibility*.py to Evaluate the Imperceptibility of a neural network watermarking method.
- *Robustness*.py to Evaluate the Robustness of a neural network watermarking method.
- *ComputationalCost*.py to Evaluate the Computational Cost of a neural network watermarking method.
- *Utils*.py which contains functions and import used in different

7 Use cases (Informative)

This chapter provides an overview of possible use cases of MPAI-NNW together with the types of actors playing roles in them. These are provided for information and are not intended to restrict the scope of application of the standard.

The following use cases can relate to both watermarking the NN Model or the NN inference:

- *Identify an NN*
  In this use case, the retrieved Payload conveys information about the NN itself.
- *Identify the actors of an NN*
  Actors are any of NN customer, NN end-user, NN owner, and NN watermarking provider.
  In this use case, the retrieved Payload conveys information about some or all of the following
- *Verify the integrity of an NN*
  In this use case, the Payload conveys information about the NN Model's integrity.
- *Assess the computational cost of injecting, detecting, and decoding a payload*

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 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 Module (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
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.
Zero Trust	A model of cybersecurity 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.

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

The Reference Software of an MPAI Standard is released with the MPAI Modified Berkeley Software Distribution licence. However, implementers should be aware that the Reference Software of an MPAI Standard may reference some third-party software that may have a different licence.

Annex 3 The Governance of the MPAI Ecosystem (Informative)

Level 1 Interoperability

With reference to *Figure 1,* MPAI issues and maintains a Technical Specification – called MPAI-AIF – whose components are:

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

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

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

Level 2 Interoperability

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

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

Level 3 Interoperability

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

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

Implementers	May claim their Implementations have passed Performance Assessment.

’ benefits	
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 2* 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 [6]:

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 MPAI Store.
6. Implementers submit Implementations to the MPAI Store
7. MPAI Store verifies security and Tests Conformance of Implementation.
8. Users download Implementations and report their experience to MPAI.

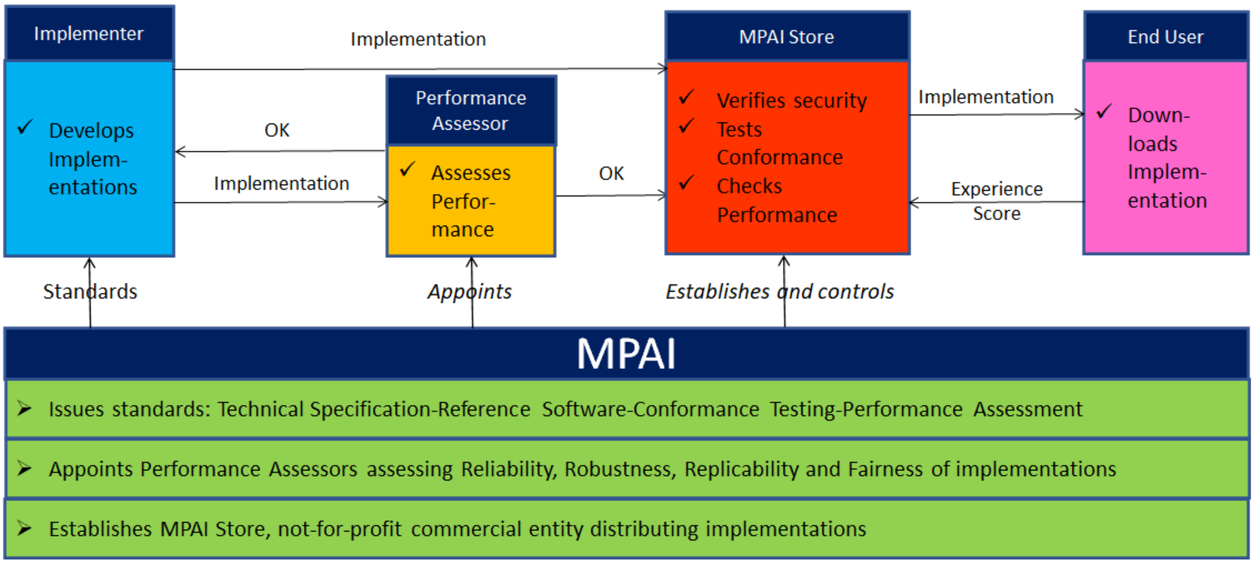
Implementer
✓ Develops Implem-entations
Implementation
OK
Implementation
Performance Assessor
✓ Assesses Perfor-mance
OK
MPAI Store
✓ Verifies security
✓ Tests Conformance
✓ Checks Performance
Implementation
Experience Score
End User
✓ Down-loads Implem-entation
Standards
Appoints
Establishes and controls
MPAI
➢ Issues standards: Technical Specification-Reference Software-Conformance Testing-Performance Assessment
➢ Appoints Performance Assessors assessing Reliability, Robustness, Replicability and Fairness of implementations
➢ Establishes MPAI Store, not-for-profit commercial entity distributing implementations

Figure 2 – The MPAI ecosystem operation

Annex 4 Identifiers of Neural Network Watermarking Technology Types

The table here below identifies the different types of Neural Network Watermarking Technologies.

ID	**Name**	**Features**
0	Direct modification of the weights	Use *Embedder()*
		Does not need access to training dataset
1	Modification of the weights using backpropagation	Use *Embedder_one_step()*
		Need access to training dataset
3	Detection of the watermark in the inference	Does not need access to the weights of the watermarked NN
4	Detection of the watermark in the weights	Need access to the weights of the watermarked NN