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

|  |  |
| --- | --- |
| **N309** | 2021/07/19 |
| **Source** | Leonardo Chiariglione |
| **Title** | Governance of the MPAI ecosystem wd0.3 |
| **Target** | MPAI Members |

Contents

[1 Introduction 1](#_Toc76313976)

[2 Scope of standard 2](#_Toc76313977)

[3 MPAI standards 3](#_Toc76313978)

[4 Definition of Performance 4](#_Toc76313979)

[5 Governance of the ecosystem 4](#_Toc76313980)

[6 The operation of the MPAI ecosystem 5](#_Toc76313981)

[7 The actors of the MPAI ecosystem 6](#_Toc76313982)

[8 Data types 7](#_Toc76313983)

[Annex 1 9](#_Toc76313984)

[Annex 2 11](#_Toc76313985)

# Introduction

Moving Picture, Audio and Data Coding by Artificial Intelligence (MPAI) is an [international association](http://mpai.community/) with the mission to develop *AI-enabled data coding standards*. Research has shown that data coding with AI-based technologies is generally *more efficient* than with existing technol­ogies. Compression and feature-based description are notable examples of coding.

MPAI standards target AI systems implementing MPAI-specified Use Cases. To the extent possible, MPAI subdivides AI systems implementing MPAI-specified Use Cases into functional components called AI Modules (AIM). AI systems implementing a Use Case and AIMs developed in conformance with MPAI standard are both called Implementations.

MPAI defines *standard interfaces* of *AI Modules (AIM)* *combined* and *executed* in an MPAI-specified *AI-Framework* (AIF). AIMs operate on input data with standard formats and produce output data with standard formats.

The 2 basic elements of MPAI standardisation – AIM and AIF – are represented in *Figure 1* and *Figure 2*.

|  |  |
| --- | --- |
| Diagram  Description automatically generated | Diagram  Description automatically generated |
| *Figure 1 – The MPAI AI Module (AIM)* | *Figure 2 – The MPAI AI Framework (AIF)* |

*Figure 1* shows a video from a camera shooting a human face entering the “Video analysis” AIM. This has the function to detect the emotion on the face and the meaning of the sentence the human is uttering. The AIMs can be implemented with neural networks or with Data Processing (DP) technologies. In the latter case, the AIM typically accesses a knowledge base external to the AIM. AIMs can be implemented in software, hardware and mixed hardware-software.

The input data enter the Execution area of the AIF where the workflow of AIMs is executed under the supervision of Management and Control. AIMs communicate via the AIF’s Communication and Storage infrastructure and may access static or slowly changing data sources (e.g., those pos­sibly needed by *Figure 2* to process the data) through Access. The AIF processes the input data and makes available the result as output data.

MPAI believes that *competing* developers striving to provide better *proprietary AIMs* that are interoperable, because they can replace other AIMs conforming to the standard, will naturally create *horizontal markets* of *AI solutions* that build on and further promote AI *innovation*. Unlike that of other DP-based systems, the operation of an AI system may strongly depend on how it has been trained. Therefore, an AI system should be tested for Conformance, i.e., for correct *technical* implementation as it is done for a DP-based system. Unlike a DP-based system, however, an AI system should also be tested for Performance, in order to verify that the implementation satisfies other non-technical requirements.

# Scope of standard

This standard specifies the rules governing the MPAI ecosystem established to achieve the following goals:

1. Development of:
   1. Technical Specification of the AI Framework
   2. Application-oriented Technical Specifications of:
      * 1. The function performed by a Use Case.
        2. The input and output data (e.g., given to and received by the user interface).
        3. The AIMs and their input and output data.
        4. The AIM topology, interconnections and timing.
   3. Reference Software, with equal normative value as the corresponding Technical Specification, as a source code implementation of the AIF and of the workflow exposing all AIM interfaces and technically conforming implementations of the AIMs as source code or compiled.
   4. Conformance Testing standards that allow an implementer to ascertain whether its implementation is technically correct and include
2. The definition of Conformance Testing.
3. The Conformance Testing procedure to be followed for testing.
4. The data sets used for testing.
   1. Standards for Performance Testing of application oriented Technical Specifications that allow an independent, MPAI-appointed Conformance Tester to assess the level of Performance – i.e., Reliability, Robustness, Replicability and Fairness (R3F)– of an implementation under test, that contain:
5. The definition of Performance.
6. The Performance Testing process.
7. The Means – tools, data sets, etc. – used to carry out Performance Testing.
8. The information that a Performance Tester shall provide in support of their results.
   1. Any other non-technical standards, e.g., like the present document.
9. Availability of the above 4 elements – collectively called MPAI Standard – at the following conditions:
   1. By providing the following data to experts.mpai.community:
10. First name and last name
11. Affiliation
12. Country
13. Email address
    1. Without charge for the years 2022 and 2023.
14. Enable End Users and Service Providers to download, build and execute implementations:
    1. authoritatively tested for Conformance, Performance and Security.
    2. not tested for Performance but conforming to the AI Framework specification.

This document contains the following Chapters:

1. introduces the types of MPAI standards and the meaning of “normativity”.
2. defines Performance.
3. describes the governance of the MPAI ecosystem.
4. analyses the role of the ecosystem actors.
5. specifies the data types circulating in the MPAI ecosystem.

and two Annexes:

1. defines the terminology used in this document.
2. outlines a possible structure of identifiers.

# MPAI standards

MPAI develops the following standards:

1. The *AI Framework* standard (MPAI-AIF) depicted in *Figure 2*.
2. *Application-oriented* standards that specify Use Cases implementable as workflows and the AIMs required by the Use Cases (see *Figure 1* and *Figure 2*).
3. *Reference software* of the AI Framework standard and of the application standards.
4. *Conformance Testing* of an implementation of an AI Framework, and of a Use Case and its AIMs.
5. *Performance Testing* of an implementation of a Use Case and of its AIMs.

Application-oriented standards normatively specify:

1. *Data formats*: any type of static (time independent) or dynamic (time dependent) data that is used in an AI system, e.g., video, emotion and meaning in *Figure 1*.
2. *AI Modules*: subsystems that are characterised by:
   1. the function performed by the AIMs (e.g., extraction of emotion and meaning).
   2. the data entering and leaving the AIM, e.g., video, emotion and meaning in *Figure 1*.
3. *Use Cases*: an AI system that implements an MPAI-specified Use Case characterised by
   1. the function performed by the AI system.
   2. the data entering and leaving the AI system, e.g., those at the left-hand and right-hand side of *Figure 2*.
   3. the topology and connection of the AIMs in the AI system.

It is up to the implementer to claim conformance of an implementation to:

1. a specific Use Case, its AIMs and their data formats.
2. selected AIMs and their data formats.
3. selected data formats.
4. none of the above.

MPAI defines two types of adherence to an MPAI standard:

1. *Conformance* of an implementation, i.e., the implementation has been tested and declared to be technically correct. Conformance may be claimed for:
   1. a data format.
   2. an AIM.
   3. a Use Case.
2. *Performance* of an implementation, i.e., the implementation (“Implementation”) has been tested and declared to satisfy identified techncial requirements. Performance may be claimed for:
   1. an AIM.
   2. a Use Case.

# Definition of Performance

The Performance of an Implementation is measured against the following attributes:

1. *Reliability*: Implementation performs as specified by the 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.
2. *Robustness*: the ability of the Implementation to cope with data outside of the stated application scope with an estimated degree of confidence.
3. *Replicability*: Performance of an Implementation as Tested by an entity can be replicated, within an agreed level, by another entity.
4. *Fairness*: the training set and/or network is open to testing for bias and unanticipated results so that the extent of applicability of the system can be assessed.

The four attributed are meant to apply to data outside of the training set.

The definition of Performance is specific to an application domain and defined in the context of that domain. Performance of an Implementation is not a binary value. An Implementation can have levels of Performance.

MPAI should consider whether all 4 attributes should mandatorily be tested to obtain a Performance validation or if some attributes can be singled out for testing.

# Governance of the ecosystem

The governance of the ecosystem is ensured by

1. *MPAI* who
   1. Acts as the root of trust.
   2. Defines the rules of governance.
   3. Establishes and controls the MPAI Store.
   4. Certifies Performance Testers.
   5. Develops the 4 components of a standard.
2. *MPAI Stor*e who
   1. tests implementations submitted by implementers for Conformance and Security.
   2. posts software implementations for download.
3. *Performance Testers* who test the Performance of implementations.

These are the elements of the process of Performance Testing:

1. The Performance Testing of an MPAI Use Case shall contain the following Use Case-specific elem­ents:
   1. Definition of Performance.
   2. Performance Testing process.
   3. Means – procedures, tools, data sets, etc. – used to carry out Performance Testing.
   4. Information that a Performance Tester shall provide in support of their results, e.g.,
      1. The 4 attributes of Performance.
      2. The type and amount of information that the Performance Tester should disclose in case the testing has failed.
2. MPAI approves and gives pointers to the Performance Testing Means.
3. The Performance Tester may be an Implementer or a Testing Laboratory appointed by MPAI.
4. The Performance Tester is the sole holders of the MPAI-granted name spaces used to assign Identifiers to successfully Tested Implementations.
5. A Performance Tester may not test the Performance of their implementations.
6. MPAI may not be a Performance Tester.
7. The Implementer shall have the right to obtain the information identified in 1.d from the Performance Tester.
8. Appointment of a Performance Tester applies for a particular domain, has indefinite duration but may be revoked.

Individual Users are free to express a numerical score on their user experience about an Implem­entation and post the score to a reputation system managed by MPAI. MPAI will make the average score public and post the individual textual comments.

The MPAI infrastructure enabling the MPAI ecosystem is based on Identifiers:

1. AIF Identifier contains:
   1. the identity of the Implementer
   2. the version of the MPAI-AIF standard, the Profile
   3. the version of the Implementation
2. The Use Case is identified by:
   1. the MPAI standard
   2. the Profile of the standard
   3. the Version of the standard
3. AIM Identifier contains:
   1. the identity of the Implementer
   2. the version of the MPAI standard and the Profile
   3. the version of the Implementation
4. An AI tool may have an Identifier if it is unique to the AI tool.

The AIF implementing the Use Case needs identifiers, e.g., in the case of a distributed system. Such private identifiers, however, are internal to the AIF and need not be exposed outside of the AIF.

Annex 2 presents a possible structure of Identifiers.

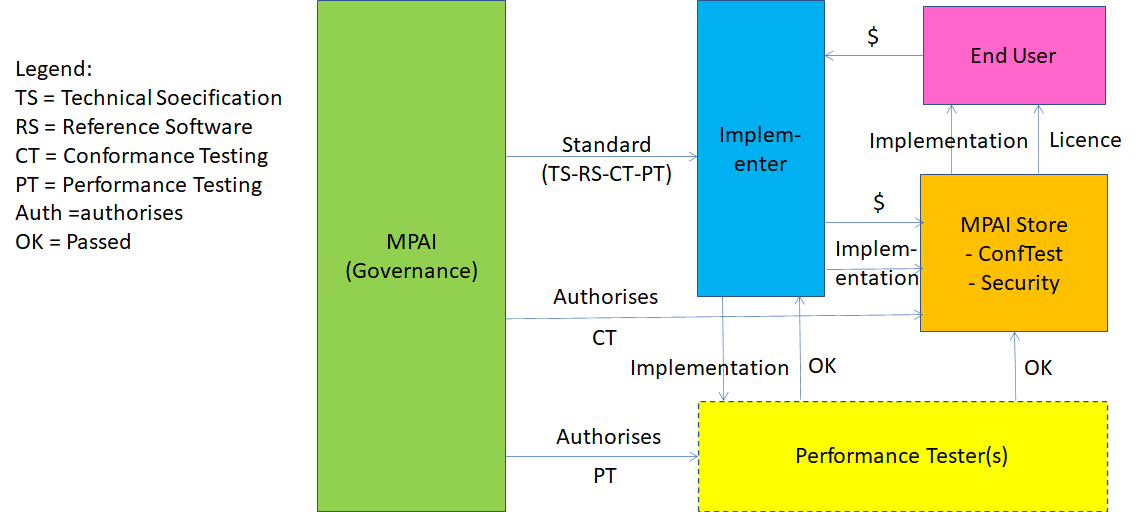
Use of Identifiers is mandatory for Implementations. However, it is optional for implementations.

MPAI will take measures to prevent implementers from attaching Identifiers to entities that do not belong to the MPAI ecosystem.

It should be mentioned that IPR holders may decide to develop Framework Licences and issue licences that are conditional on implementations being Implementations.

# The operation of the MPAI ecosystem

The operation of the MPAI ecosystem is depicted in *Figure 3*



*Figure 3 – The operation of the MPAI ecosystem*

1. MPAI develops the 4 types of standards listed above for MPAI-AIF, and application standards.
2. MPAI appoints Performance Testers.
3. An implementer develops an implementation – AIM or use case:
   1. If the implementation complies with the AIF standard, the implementer may upload it to the MPAI Store.
   2. If the implementation claims to conform to an MPAI application standard, the implem­enter may request a Performance Tester to test the Performance of the implementation.
4. The implementer may upload the implementation to the MPAI store where:
   1. The security of the implementation is tested, if the implementer does not claim adherence to an MPAI standard.
   2. The security and the Conformance of the implementation is tested if the implementer claims adherence to an MPAI standard.
   3. The security, the Conformance and the Performance of the implementation is tested, the last as certified by a Performance Tester.
5. A user wishing to download an implementation from the MPAI Store:
   1. Is sent to the implementer.
   2. Gets a licence from the implementer.
   3. Downloads the implementation from the MPAI Store.

The MPAI Store is a not-for-profit commercial organisation.

Operating the MPAI Store will entail significant costs – e.g., ICT infrastructure, personnel, consultants – that the MPAI Store should recover.

Costs can be recovered from implementers, End Users and Service Providers. Examples are:

1. From implementers:
   1. Conformance and Security Testing cost with a fixed or variable price.
   2. ICT costs: use of space of storage and amount of data streamed at an assigned price for 1 MByte of storage per month and 1 Mbyte of streaming.
2. From end user: a percentage of the amount paid to implementers.
3. From service providers: e.g., pay like an implementer.

This applies also to Conformance and Performance Testing data in case this is stored in the MPAI Store.

# The actors of the MPAI ecosystem

The actors enabling the MPAI ecosystem to operate are given by *Table 1*.

*Table 1 – The actors of the MPAI ecosystem*

|  |  |
| --- | --- |
| **Actor** | **Role** |
| MPAI | Is the non-profit, international, non-affiliated standardisation body who:   1. Standardises    1. AIF    2. Use Cases as Workflows    3. AIMs    4. Conformance Testing of Workflows and AIMs    5. Performance Testing of Workflows and AIMs 2. Defines the rules governing the MPAI ecosystem:    1. Certifies Performance Testers.    2. Assigns Identifier name spaces to Performance Testers.    3. Establishes and controls the operation of the MPAI Store. |
| MPAI Store | 1. Independent commercial not-for-profit entity established and controlled by MPAI 2. Charges MPAI Store users (implementers and end users) on a cost recovery basis 3. Receives implementations from implementers 4. Receives results of Performance Testing of implementations claimed to be in accordance with an MPAI standard 5. Tests submitted implementations for Conformance and security 6. Stores Conformance Testing data and Performance Testing data as guarantor of the ecosystem. Parts of this data will only be offered for download to authorised entities. Availability will be best effort. The MPAI Store may allow publicly available data to be mirrored. 7. Workflows and AIMs may be implementations of MPAI standards or independently developed but suitable to be executed in AIFs. 8. Assigns workflows and AIMs to security experts for testing (either employees or consultants). 9. Submitted non MPAI-specified workflows and AIMs that are approved by security experts will be given a special identifier and made available for distribution 10. Interacts with providers of AIF, Workflows and AIM implementations in order to enforce commercial licenses whenever needed. 11. Distributes workflows and AIMs that are approved by security experts to providers of computational MPAI services and users. 12. The MPAI Store undertakes to make implementations available through high availability ICT infrastructure. |
| Standard developer | An expert contributing to the development of the 4 types of MPAI standard. For the Performance Testing standard, the expert will contribute to defining the nature of the results that a Performance Tester shall deliver. |
| Implementation provider | A possibly commercial entity providing implementations of AIF, Workflows and AIMs. |
| Performance Tester | An entity certified by MPAI to develop tests establishing if a Workflow or AIM:   1. Conforms with one or more MPAI standards 2. Offers sufficient performance and hence can be given an Identifier. 3. Taking into account the requests contained in the Performance Testing part of each standard. |
| Security expert | An entity authorised by MPAI to develop tests establishing if a Workflow or AIM presents outstanding security issues. The Implementation provider shall quickly propose a fix (possibly including the removal of the offending component from the Store). The fix shall be approved/rejected by the MPAI Store. |
| Mirror | A store external to the MPAI Store mirroring non-essential, bulky data generated during the standard approval procedure (for instance, publicly available training data, code, repositories tracking history, or procedures to perform tests). |
| Service provider | An entity running MPAI workflows on their infrastructure (e.g., a private or public cloud). Users can request such entity to run MPAI workflows on their behalf, for free or for a fee. |
| End User | The ultimate beneficiary of the execution of MPAI workflows. May execute non MPAI-specified workflows and AIMs in a local environment |

# Data types

The data types proper of the MPAI ecosystem are

*Table 2 – The data types of the MPAI ecosystem*

|  |  |
| --- | --- |
| **Data type** | **Definition** |
| Conformance data | Defined and version controlled by MPAI and freely accessible from the MPAI Store. |
| Performance data | Defined and version controlled by MPAI and accessible either from the MPAI Store or from a third party. |
| Implementations | of workflows and AIMs may be defined by MPAI standards or by implementers.  Implementations intended to be MPAI-conforming are:   1. submitted by providers. 2. validated by MPAI validators. 3. performance-certified by Performance Testers. 4. tested against security issues by MPAI security experts. 5. assigned a unique ID.   Implementations not intended to be MPAI-conforming are:   1. submitted by providers. 2. tested against security issues by MPAI security experts. 3. assigned a unique ID in a different name space.   Calls exist in the MPAI-AIF API to test if a running workflow has been officially certified (i.e., it is compliant with an MPAI standard).  Implementations available on the MPAI Store can be commercial or non-commercial modules. |
| Authorisation-authentication | Workflows and AIMs have an associated free or commercial licence. MPAI-AIF implements a protocol that checks that a user has all the needed privileges to use an implementation, typically implemented through a local licence cache or remote licence store that interacts with the user device and the MPAI Store. |
| Licence | See above |

# Annex 1

**Terminology**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| AI Framework | The environment where AIM-based workflows are executed. |
| AI Module | The basic processing elements receiving processing-specific inputs and producing processing-specific outputs. |
| AI system | A system that uses AI to achieve goals that often could not be achieved or could only be poorly achieved with traditional technologies. |
| Applicability | The function of a Use Case or of an AIM as defined by the relevant standard |
| Component | An AIM or a tool (e.g., a neural network) used by the AIM |
| 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. |
| Identifier | A name that identifies any of the following   1. An Implementer 2. An Implementation 3. A standard, its profiles and its versions. |
| Implementation | An implementation of a Use case or an AIM whose Performance has been Tested to be above a level defined in the relevant Standard. |
| MPAI ecosystem | The ensemble of MPAI, MPAI Store, Performance Testers and Implementers developing and Users utilising Implementations. |
| Normativity | An applicable set of attributes of a technology or a set of technologies specified by an MPAI standard. |
| Performance | The attribute of an Implementation of being Reliable, Robust, Fair and Replicable. |
| Performance Testing | The assessment of the level of Performance of an Implementation. |
| Profile | A particular subset of the technologies that are used in a Use Case standard and, where applicable, the classes, other subsets, options and parameters relevant to that subset. |
| Reliability | The attribute of an Implementation that performs as specified by the 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 Tested by a Testing Entity, can be replicated, within an agreed level, by another Testing Entity. |
| Robustness | The attribute of an Implementation that copes with data outside of the stated application scope with an estimated degree of confidence. |
| Service Provider | An entrepreneur who creates an Implementation that is offered as a service (e.g., a recommendation service) to end users. |
| Standard | A set of Use Cases belonging to an application domain normatively specified by MPAI along with the AIMs required to Implement the Use Cases. MPAI may develop other types of standards. |
| Performance Tester | An Implementer or a Testing Laboratory authorised by MPAI to Test the Performance of an Implementation in a given domain |
| Testing Laboratory | A laboratory accredited by MPAI to Test Implementations for Performance |
| Testing Means | Elements such as tools, procedures, data sets, etc., developed or approved by MPAI to be used when Testing the Performance of an Implementation. |
| Use Case | A particular instance of the application domain covered by a Standard identified as Normative. |
| Version | A revision or extension of a Standard. |

# Annex 2

**Initial ideas for the Identifier structure**

MPAI Identifiers could have the following structure

|  |  |  |
| --- | --- | --- |
| **Field** | **Code** | **c/o** |
| MPAI Identifier | MPAI | MPAI |
| Testing Entity | LengthTE | MPAI |
| Implementer | LengthImp | MPAI |
| Implementation no./version | LengthTI (Tested Implementations) | Implementer |
|  | LengthUT (unTested Implementations) |
| Standard, profile and version | LengthSPV | Performance Tester |
| Checksum | LengthCS | Performance Tester |

Workflow identifiers used created in real time need not have an MPAI-defined structure.