



IEEE Standard Adoption of Moving Picture, Audio and Data Coding by Artificial Intelligence (MPAI) Technical Specification Artificial Intelligence Framework (AIF)—Version 2

Developed by the
Collaborative Activities Governance Board

IEEE SA Industry Affiliated Network (IAN)
Base specification contributed by MPAI community



IEEE Std 3301™-2024

IEEE Standard Adoption of Moving Picture, Audio and Data Coding by Artificial Intelligence (MPAI) Technical Specification Artificial Intelligence Framework (AIF)—Version 2

Developed by the

Collaborative Activities Governance Board
of the
IEEE SA Board of Governors

Approved 26 September 2024

IEEE SA Standards Board

Abstract: This standard adopts MPAl AI Framework (MPAl-AIF) Technical Specification Version 2 as an IEEE Standard. The MPAl-AIF Technical Specification specifies architecture, interfaces, protocols, and Application Programming Interfaces (API) of an AI Framework (AIF), especially designed for the execution of AI-based implementation, but also suitable for mixed AI and traditional data processing workflow.

Keywords: adoption, AI framework, AI modules, AIFs, IEEE 3301™, MPAl-AIF

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2024 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 14 November 2024. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 979-8-8557-1336-7 STD27463
Print: ISBN 979-8-8557-1428-9 STDPD27463

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <https://www.ieee.org/about/corporate/governance/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all IEEE standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers involved in technical working groups are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning all standards, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. IEEE Standards documents do not guarantee safety, security, health, or environmental protection, or compliance with law, or guarantee against interference with or from other devices or networks. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document should rely upon their own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE 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.

Translations

The IEEE consensus balloting process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English language version published by IEEE is the approved IEEE standard.

Use by artificial intelligence systems

In no event shall material in any IEEE Standards documents be used for the purpose of creating, training, enhancing, developing, maintaining, or contributing to any artificial intelligence systems without the express, written consent of IEEE SA in advance. “Artificial intelligence” refers to any software, application, or other system that uses artificial intelligence, machine learning, or similar technologies, to analyze, train, process, or generate content. Requests for consent can be submitted using the Contact Us form.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual is not, and shall not be considered or inferred to be, the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE or IEEE SA. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter’s views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s). News releases about IEEE standards issued by entities other than IEEE SA should be considered the view of the entity issuing the release rather than the formal position of IEEE or IEEE SA.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of Governors are not able to provide an instant response to comments or questions, except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or revisions to an IEEE standard is welcome to join the relevant IEEE SA working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the [IEEE SA myProject system](#).¹ An IEEE Account is needed to access the application.

Comments on standards should be submitted using the [Contact Us](#) form.²

¹ Available at: <https://development.standards.ieee.org/myproject-web/public/view.html#landing>.

² Available at: <https://standards.ieee.org/about/contact/>.

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, neither IEEE nor its licensors waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; <https://www.copyright.com/>. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit [IEEE Xplore](#) or [contact IEEE](#).³ For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#).⁴ Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

Patents

IEEE standards are developed in compliance with the [IEEE SA Patent Policy](#).⁵

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at <https://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

Technologies, application of technologies, and recommended procedures in various industries evolve over time. The IEEE standards development process allows participants to review developments in industries, technologies, and practices, and to determine what, if any, updates should be made to the IEEE standard. During this evolution, the technologies and recommendations in IEEE standards may be implemented in ways not foreseen during the standard's development. IEEE standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, data privacy, and interference protection practices and all applicable laws and regulations.

³ Available at: <https://ieeexplore.ieee.org/browse/standards/collection/ieee>.

⁴ Available at: <https://standards.ieee.org/standard/index.html>.

⁵ Available at: <https://standards.ieee.org/about/sasb/patcom/materials.html>.

Participants

At the time this standard was completed, the Artificial Intelligence Framework Working Group had the following membership:

Stephen Dukes, *Chair*
Andrea Basso, *Vice Chair*
Yu Yuan, *Secretary*

<i>Organization Represented</i>	<i>Name of Representative</i>
OxSenses Corporation	Yu Yuan
Beijing Tsinghua Changung Hospital	Bin Yang
China University of Petroleum Beijing.....	Xiaohe Chen
Electronic and Telecommunications Research Institute (ETRI)	Miran Choi
Imaginary Universes, LLC.....	Stephen Dukes
Moving Picture, Audio and Data Coding by Artificial Intelligence (MPAI)	Andrea Basso
Tongyuan Software Co., Ltd.....	Johnny Lin

The following members of the entity Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

OxSenses Corporation	Electronics and	Intel
China Electronic Standardization	Telecommunications Research	Moving Picture, Audio and Data
Institute	Institute (ETRI)	Coding by Artificial
CRRC Zhuzhou Institute Co.,	Guangdong Yunbai Technology	Intelligence (MPAI)
Ltd.	Co., Ltd.	PetroChina Planning and
Eaton	Imaginary Universes, LLC	Engineering Institute
	Institute of Biomedical	Tongyuan Software Co., Ltd.
	Engineering	

When the IEEE SA Standards Board approved this standard on 26 September 2024, it had the following membership:

David J. Law, *Chair*
Jon Walter Rosdahl, *Vice Chair*
Gary Hoffman, *Past Chair*
Alpesh Shah, *Secretary*

Sara R. Biyabani	Hao Hu	Paul Nikolich
Ted Burse	Yousef Kimiagar	Robby Robson
Stephen Dukes	Joseph L. Koepfinger*	Lei Wang
Doug Edwards	Howard Li	F. Keith Waters
J. Travis Griffith	Xiaohui Liu	Sha Wei
Guido R. Hiertz	John Haiying Lu	Philip B. Winston
Ronald W. Hotchkiss	Kevin W. Lu	Don Wright
	Hiroshi Mano	

*Member Emeritus



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

MPAI Technical Specification

AI Framework MPAI-AIF

V2

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.

Technical Specification

AI Framework (MPAI-AIF) V2

1	Introduction (Informative).....	3
2	Scope of Standard.....	5
3	Terms and Definitions	6
4	References	7
4.1	Normative references.....	7
4.2	Informative references	8
5	Architecture of the AI Framework	9
5.1	AI Framework Components	9
5.1.1	Components for Basic Functionalities	9
5.1.2	Components for Security Functionalities	11
5.2	AI Framework Implementations.....	12
5.3	AIMs.....	12
5.3.1	Implementation types	12
5.3.2	Combination	13
5.3.3	Hardware-software compatibility.....	13
5.3.4	Actual implementations.....	13
6	Metadata	14
6.1	Communication channels and their data types	14
6.1.1	Type system.....	14
6.1.2	Mapping the type to buffer contents	15
6.2	AIF Metadata	15
6.3	AIW/AIM Metadata	17
7	Common features of MPAI-AIF API.....	22
7.1	General.....	22
7.2	Conventions.....	22
7.2.1	API types	22
7.2.2	Return codes	23
7.2.3	High-priority Messages	23
8	Basic API.....	23
8.1	Store API called by Controller	23
8.1.1	Get and parse archive	24
8.2	Controller API called by User Agent	24
8.2.1	General	24
8.2.2	Start/Pause/Resume/Stop Messages to other AIWs	24
8.2.3	Inquire about state of AIWs and AIMs	25
8.2.4	Management of Shared and AIM Storage for AIWs.....	25
8.2.5	Communication management.....	25
8.2.6	Resource allocation management.....	26
8.3	Controller API called by AIMs	26
8.3.1	General	26
8.3.2	Resource allocation management.....	26
8.3.3	Register/deregister AIMs with the Controller.....	27
8.3.4	Send Start/Pause/Resume/Stop Messages to other AIMs	27

8.3.5	Register Connections between AIMS	28
8.3.6	Using Ports	29
8.3.7	Operations on messages	29
8.3.8	Functions specific to machine learning	31
8.3.9	Controller API called by Controller	31
9	Security API	32
9.1	Data characterization structure	32
9.2	API called by User Agent	33
9.3	API to access Secure Storage	33
9.3.1	User Agent initialises Secure Storage API	33
9.3.2	User Agent writes Secure Storage API	33
9.3.3	User Agent reads Secure Storage API	33
9.3.4	User Agent gets info from Secure Storage API	34
9.3.5	User Agent deletes a p_data in Secure Storage API	34
9.4	API to access Attestation	34
9.5	API to access cryptographic functions	34
9.5.1	Hashing	34
9.5.2	Key management	35
9.5.3	Key exchange	36
9.5.4	Message Authentication Code	36
9.5.5	Cyphers	37
9.5.6	Authenticated encryption with associated data (AEAD)	38
9.5.7	Signature	38
9.5.8	Asymmetric Encryption	39
9.6	API to enable secure communication	39
10	Profiles	39
10.1	Basic Profile	39
10.2	Secure Profile	39
11	Examples (Informative)	40
11.1	AIF Implementations	40
11.1.1	Resource-constrained implementation	40
11.1.2	Non-resource-constrained implementation	40
11.2	Examples of types	40
11.3	Examples of Metadata	40
11.3.1	Metadata of Enhanced Audioconference Experience AIF	41
11.3.2	Metadata of Enhanced Audioconference Experience AIW	41
11.3.3	Metadata of CAE-EAE Analysis Transform AIM	46
11.3.4	Metadata of CAE-EAE Sound Field Description AIM	47
11.3.5	Metadata of CAE-EAE Speech Detection and Separation AIM	48
11.3.6	Metadata of CAE-EAE Noise Cancellation AIM	49
11.3.7	Metadata of CAE-EAE Synthesis Transform AIM	50
11.3.8	Metadata of CAE-EAE Packager AIM	51
Annex 1	MPAI-wide terms and definitions	53
Annex 2	Notices and Disclaimers Concerning MPAI Standards (Informative)	56

1 Introduction (Informative)

MPAI – Moving Picture, Audio, and Data Coding by Artificial Intelligence, the international, un-affiliated, non-profit organisation developing standards for Artificial Intelligence (AI)-based data

coding with clear Intellectual Property Rights licensing frameworks, develops standards in compliance with a rigorous process [2] pursuing the following policies:

1. Be friendly to the AI context but, to the extent possible, agnostic to the technology – AI or Data Processing – used in an implementation.
2. Be attractive to different industries, end users, and regulators.
3. Address three levels of standardisation any of which an implementer can freely decide to adopt:
 - a. Data types, i.e., the data exchanged by systems.
 - b. Components called AI Modules (AIM).
 - c. Connected components called AI Workflows (AIW).
4. Specify the data exchanged by components with a clear semantic to the extent possible.

Technical Specification: AI Framework (MPAI-AIF) V2 enables dynamic configuration, initialisation, and control of AIWs in a standard environment called AI Framework (AIF). Figure 1 depicts the AI Framework.

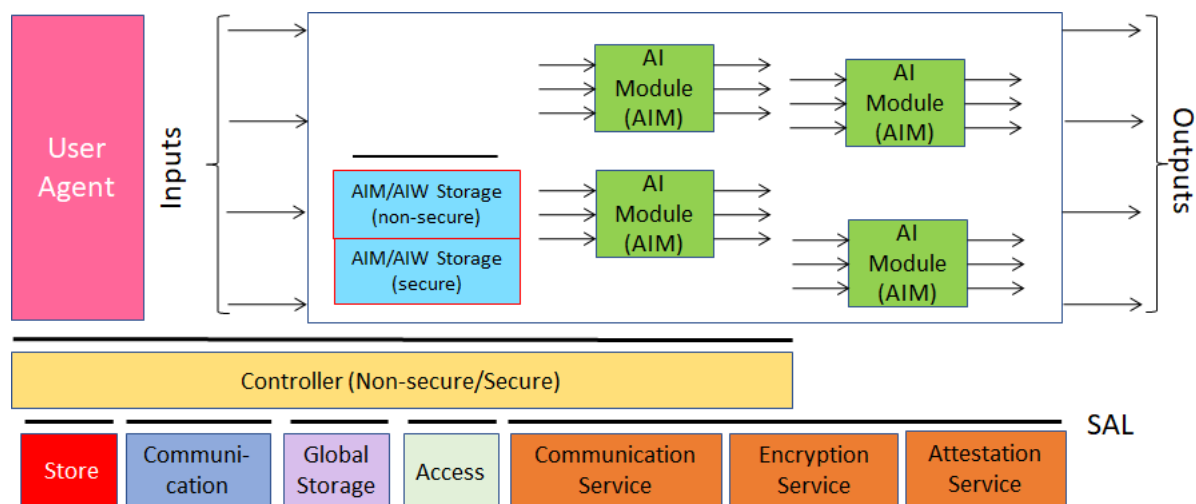


Figure 1 - The AI Framework (MPAI-AIF) V2 Reference Model

Thus, users can exercise AIWs that are both proprietary or standardised by MPAI – i.e., with standard functions and interfaces, with an explicit computing workflow. Developers can compete in providing AIMs with standard functions and interfaces that may have improved performance compared to other implementations. AIMs can execute data processing or Artificial Intelligence algorithms and can be implemented in hardware, software, or hybrid hardware/software.

AIW and its AIMs may have 3 interoperability levels:

Level 1 – Proprietary 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.

MPAI Application 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 Store in Figure 1 is the entity from which Implementations are downloaded. MPAI Standards assume that the AIF, AIW, and AIM Implementations may have been developed by

independent implementers. A necessary condition for this to be possible, is that any AIF, AIW, and AIM implementations be uniquely identified. MPAI has appointed an ImplementerID Registration Authority (IIDRA) to assign unique ImplementerIDs (IID) to Implementers.¹

MPAI Calls “Standard” the combination of Technical Specification, Reference Software Specification, and Conformance Testing Specification. The MPAI-AIF V2 Technical Specification will be accompanied by a Reference Software Specification and a Conformance Testing Specification.

The chapters and the annexes of this Technical Specification are Normative unless they are labelled as Informative. Terms beginning with a capital letter are defined in *Table 1* if specific of this MPAI-AIF Technical Specification, or in *Table 2* if used across MPAI Standards.

2 Scope of Standard

The MPAI *AI Framework* (MPAI-AIF) Technical Specification (in the following also called MPAI-AIF V2) specifies the architecture, interfaces, protocols, and Application Programming Interfaces (API) of an AI Framework specially designed for execution of AI-based implementations, but also suitable for mixed AI and traditional data processing workflows.

MPAI-AIF V2 possesses the following main features in two instances:

Basic functionalities:

1. Independent of the Operating System.
2. Component-based modular architecture with specified interfaces.
3. Interfaces encapsulate Components to abstract them from the development environment.
4. Interface with the Store enabling access to validated Components.
5. Component can be Implemented as:
 - 5.1. Software only, from MCUs to HPC.
 - 5.2. Hardware only.
 - 5.3. Hybrid hardware-software.
6. Component system features are:
 - 6.1. Execution in local and distributed Zero-Trust architectures [28].
 - 6.2. Possibility to interact with other Implementations operating in proximity.
 - 6.3. Direct support to Machine Learning functionalities.
7. The AIF can download an AIW whose identifier has been specified by the User Agent or by a configuration parameter.

Secure functionalities:

1. The AIF provides access to the following Trusted Services:
 - 1.1. A selected range of cyphering algorithms.
 - 1.2. A basic attestation function.
 - 1.3. Secure storage (RAM, internal/external flash, or internal/external/remote disk).
 - 1.4. Certificate-based secure communication.
2. The AIF can execute only one AIW containing only one AIM. The AIM has the following features:
 - 2.1. The AIM may be a Composite AIM.
 - 2.2. The AIMs of the Composite AIM cannot access the Secure API.

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

3. The AIF Trusted Services may rely on hardware and OS security features already existing in the hardware and software of the environment in which the AIF is implemented.

The current version of the Technical Specification: AI Framework (MPAI-AIF) V2 has been developed by the MPAI AI Framework Development Committee (AIF-DC). Future Versions may revise and/or extend the Scope of the Technical Specification.

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
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 own 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, Store, and User Agent.
Composite AIM	An AIM aggregating more than one AIM.
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.
Data Type	An instance of the Data Types defined by 6.1.1.
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.
Group Element	An AIF in a in a proximity-based scenario.
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 a defined semantics managed by MPAI.
MPAI Server	A remote machine executing one or more AIMs.
Remote Port	A Port number associated with a specific remote AIM.
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.
Security Abstraction Layer	(SAL) The set of Trusted Services that provide security functionalities to AIF.
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
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., object 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 [28].
Security Abstraction Layer	A layer acting as a bridge between the AIMs and the Control on one side, and the security functions.

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. GIT protocol, <https://git-scm.com/book/en/v2/Git-on-the-Server-The-Protocols>.
4. ZIP format, <https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT>.
5. Date and Time in the Internet: Timestamps; IETF RFC 3339; July 2002.
6. Uniform Resource Identifiers (URI): Generic Syntax, IETF RFC 2396, August 1998.
7. The JavaScript Object Notation (JSON) Data Interchange Format; <https://datatracker.ietf.org/doc/html/rfc8259>; IETF rfc8259; December 2017
8. JSON Schema; <https://json-schema.org/>.
9. BNF Notation for syntax; <https://www.w3.org/Notation.html>
10. MPAI; The MPAI Ontology; <https://mpai.community/standards/mpai-aif/mpai-ontology/>

11. Framework Licence of the Artificial Intelligence Framework Technical Specification (MPAI-AIF); <https://mpai.community/standards/mpai-aif/framework-licence/>
12. Bormann, C. and P. Hoffman, Concise Binary Object Representation (CBOR), December 2020. <https://rfc-editor.org/info/std94>
13. Schaad, J., CBOR Object Signing and Encryption (COSE): Structures and Process, August 2022. <https://rfc-editor.org/info/std96>
14. IETF Entity Attestation Token (EAT), Draft. <https://datatracker.ietf.org/doc/draft-ietf-rats-eat>
15. IEEE, 1619-2018 — IEEE Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices, January 2019. <https://ieeexplore.ieee.org/servlet/opac?punumber=8637986>
16. IETF, The MD5 Message-Digest Algorithm, April 1992. <https://tools.ietf.org/html/rfc1321.html>
17. [RFC6979] IETF, Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA), August 2013. <https://tools.ietf.org/html/rfc6979.html>
18. [RFC7539] IETF, ChaCha20 and Poly1305 for IETF Protocols, May 2015. <https://tools.ietf.org/html/rfc7539.html>
19. [RFC7919] IETF, Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS), August 2016. <https://tools.ietf.org/html/rfc7919.html>
20. [RFC8017] IETF, PKCS #1: RSA Cryptography Specifications Version 2.2, November 2016. <https://tools.ietf.org/html/rfc8017.html>
21. [RFC8032] IRTF, Edwards-Curve Digital Signature Algorithm (EdDSA), January 2017. <https://tools.ietf.org/html/rfc8032.html>
22. Standards for Efficient Cryptography, SEC 1: Elliptic Curve Cryptography, May 2009. <https://www.secg.org/sec1-v2.pdf>
23. NIST, *FIPS Publication 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, August 2015. <https://doi.org/10.6028/NIST.FIPS.202>
24. NIST, NIST Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation: Methods and Techniques, December 2001. <https://doi.org/10.6028/NIST.SP.800-38A>
25. NIST, NIST Special Publication 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007. <https://doi.org/10.6028/NIST.SP.800-38D>

4.2 Informative references

26. MPAI; Technical Specification: Governance of the MPAI Ecosystem; <https://mpai.community/standards/mpai-gme/>
27. Message Passing Interface (MPI), <https://www.mcs.anl.gov/research/projects/mpi/>
28. Rose, Scott; Borchert, Oliver; Mitchell, Stu; Connelly, Sean; “Zero Trust Architecture”; <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf>
29. MPAI Technical Specification: Context-based Audio Enhancement (MPAI-CAE) V2; <https://mpai.community/standards/mpai-cae/>.
30. MPAI Technical Specification: Connected Autonomous Vehicle - Architecture (MPAI-CAV) V1; <https://mpai.community/standards/mpai-cav/>.
31. MPAI Technical Specification: Compression and Understanding of Industrial Data (MPAI-CUI) V1.1; <https://mpai.community/standards/mpai-cui/>.
32. MPAI Technical Specification: Multimodal Conversation (MPAI-MMC) V2; <https://mpai.community/standards/mpai-mmc/>.
33. MPAI Technical Specification: Neural Network Watermarking (MPAI-MMC) V1; <https://mpai.community/standards/mpai-nnw/>.

34. MPAI Technical Specification: Portable Avatar Format (MPAI-PAF) V1;
<https://mpai.community/standards/mpai-paf/>.
35. 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.
36. 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.
37. ML.NET [ONLINE]; <https://dotnet.microsoft.com/apps/machinelearning-ai/ml-dotnet>.
38. 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.
39. 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.
40. Apache Prediction I/O; <https://predictionio.apache.org/>.
41. 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
42. Arm; “PSA Certified Crypto API 1.1,” IHI 0086, issue 2, 23/03/2022, <https://arm-software.github.io/psa-api/crypto/1.1/>
43. Arm; “PSA Certified Secure Storage API 1.0,” IHI 0087, issue 2, 23/03/2023, <https://arm-software.github.io/psa-api/storage/1.0/>
44. Arm; “PSA Certified Attestation API 1.0,” IHI 0085, issue 3, 17/10/2022, <https://arm-software.github.io/psa-api/attestation/1.0/>

5 Architecture of the AI Framework

5.1 AI Framework Components

This MPAI-AIF Version adds a Secure Profile with Security functionalities on top of the Basic Profile of Version 1.1 with the following restrictions:

- There is only one AIW containing only one AIM – which may be a Composite AIM.
- The AIM implementer guarantees the security of the AIM by calling the security API.
- The AIF application developer cannot access securely the Composite AIM internals.

5.1.1 Components for Basic Functionalities

Figure 2 specifies the MPAI-AIF Components supported by MPAI-AIF Version 1.1.

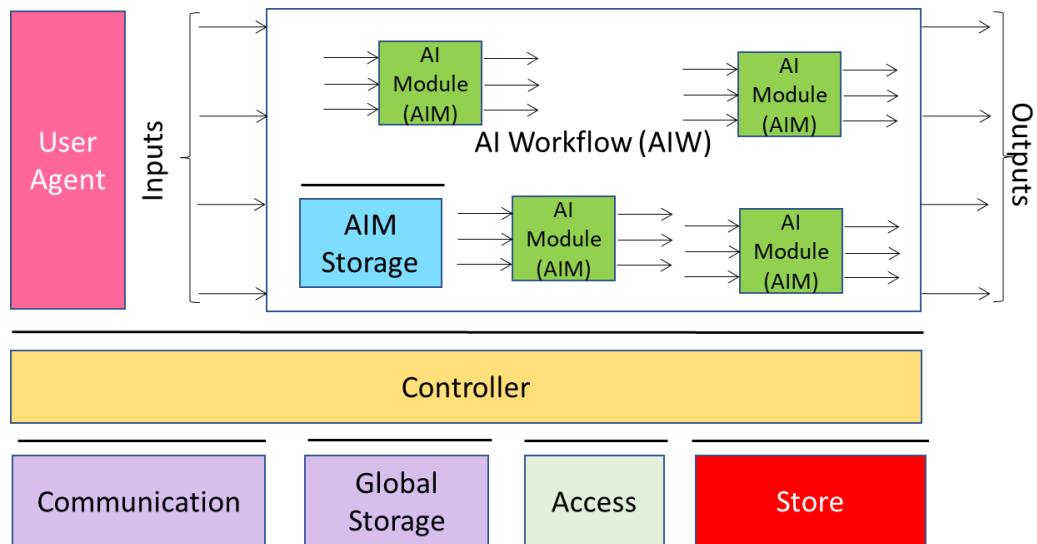


Figure 2 – The MPAI-AIF V1 Reference Model

The specific functions of the Components are:

1. **Controller:**

- 1.1. Provides basic functionalities such as scheduling, communication between AIMs and with AIF Components such as AIM Storage and Global Storage.
- 1.2. Acts as a resource manager, according to instructions given by the User through the User Agent.
- 1.3. Can interact by default to all the AIMs in a given AIF.
- 1.4. Activates/suspends/resumes/deactivates AIWs based on User's or other inputs.
- 1.5. May supports complex application scenarios by balancing load and resources.
- 1.6. Accesses the *Store APIs* to download AIWs and AIMs.
- 1.7. Exposes three APIs:
 - 1.7.1. *AIM APIs* enable AIMs to communicate with it (register themselves, communicate and access the rest of the AIF environment). An AIW is an AIM with additional metadata. Therefore, an AIW uses the same AIM API.
 - 1.7.2. *User APIs* enable User or other Controllers to perform high-level tasks (e.g., switch the Controller on and off, give inputs to the AIW through the Controller).
 - 1.7.3. *Controller-to-Controller API* enables interactions among Controllers.
- 1.8. May run an AIW on different computing platforms and may run more than one AIW.
- 1.9. May communicate with other Controllers.

2. **Communication:** connects the AIF Components via Events or Channels connecting an output Port of an AIM with an input Port of another AIM. Communication has the following characteristics:

- 2.1. The Communication Component is turned on jointly with the Controller.
- 2.2. The Communication Component needs not be persistent.
- 2.3. Channels are unicast and may be physical or logical.
- 2.4. Messages are transmitted via Channels. They are composed of sequences of Records and may be of two types:
 - 2.4.1. High-Priority Messages expressed as up to 16-bit integers.
 - 2.4.2. Normal-Priority Messages expressed as MPAI-AIF defined types (6.1.1).
 - 2.4.2.1. Messages may be communicated through Channels or Events.

3. **AI Module (AIM):** a data processing element with a specified Function receiving AIM-specific inputs and producing AIM-specific outputs having the following characteristics:

- 3.1. Communicates with other Components through Ports or Events.

- 3.2. Includes at least one input Port and one output Port.
- 3.3. May incorporate other AIMs.
- 3.4. May be hot-pluggable, and dynamically register and disconnect itself on the fly.
- 3.5. May be executed:
 - 3.5.1. Locally, e.g., it encapsulates hardware physically accessible to the Controller.
 - 3.5.2. On different computing platforms, e.g., in the cloud or on groups of drones, and encapsulates communication with a remote Controller.
- 4. **AI Workflow (AIW)**: an organised aggregation of AIMs receiving AIM-specific inputs and producing AIM-specific outputs according to its Function implementing a Use Case that is either proprietary or specified by an MPAI Application Standard.
- 5. **Global Storage**: stores data shared by AIMs.
- 6. **AIM Storage**: stores data of individual AIMs.
- 7. **User Agent**: interfaces the User with an AIF through the Controller.
- 8. **Access**: provides access to static or slowly changing data that is required by AIMs such as domain knowledge data, data models, etc.
- 9. **Store**: stores Implementations for users to download by secure protocols.

Note: When different Controllers running on separate computing platforms (Group Elements) interact with one another, they cooperate by requesting one or more Controllers in range to open Remote Ports. The Controllers on which the Remote Ports are opened can then react to information sent by other Controllers in range through the Remote Ports and implement a collective behaviour of choice. For instance: there is a main Controller and the other Controllers in the Group react to the information it sends; or there is no main Controller and all Controllers in the Group behave according to a collective logic specified in the Controllers.

5.1.2 Components for Security Functionalities

The AIF Components have the following features:

- 1. The AIW
 - 1.1. The AIMs in the AIW trust each other and communicate without special security concerns.
 - 1.2. Communication among AIMs in the Composite AIM is non-secure.
- 2. The Controller
 - 2.1. Communicates securely with the MPAI-Store and the User Agent (Authentication, Attestation, and Encryption).
 - 2.2. Accesses Communication, Global Storage, Access and Store via Trusted Services API.
 - 2.3. Is split in two parts:
 - 2.3.1. Secure Controller accesses Secure Communication and Secure Storage.
 - 2.3.2. Non-Secure Controller can access the non-secure parts of the AIF.
 - 2.4. Interfaces with the User Agent in the area where non-secure code is executed.
 - 2.5. Interface with the Composite AIM in the area where secure code is executed,
- 3. AIM/AIW Storage
 - 3.1. Secure Storage functionality is provided through key exchange.
 - 3.2. Non-secure functionality is provided without reference to secure API calls.
- 4. The AIW/AIMs call the Secure Abstraction Layer via API.
- 5. The AIMs of a Composite AIM shall run on the same computing platform.

Figure 3 specifies the MPAI-AIF Components operating in the secure environment created by the Secure Abstraction Layer.

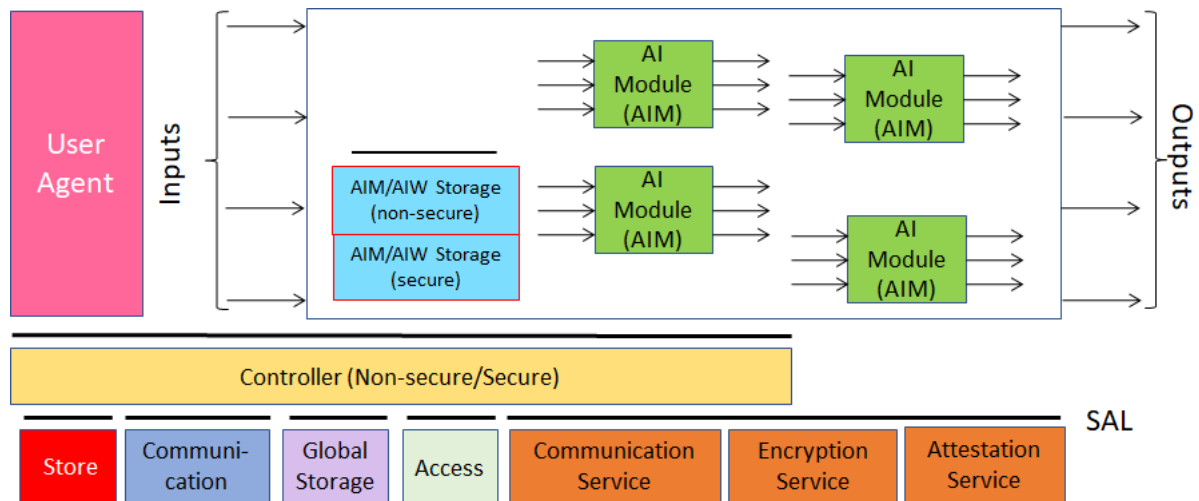


Figure 3 – The MPAI-AIF V2 Reference Model

5.2 AI Framework Implementations

MPAI-AIF enables a wide variety of Implementations:

1. AIF Implementations can be tailored to different execution environments, e.g., High-Performance Computing systems or resource-constrained computing boards. For instance, the Controller might be a process on a HPC system or a library function on a computing board.
2. There is always a Controller even if the AIF is a lightweight Implementation.
3. The API may have different MPAI-defined Profiles to allow for Implementations:
 - 3.1. To run on different computing platforms and different programming languages.
 - 3.2. To be based on different hardware and resources available.
4. AIMs may be Implemented in hardware, software and mixed-hardware and software.
5. Interoperability between AIMs is ensured by the way communication between AIMs is defined, irrespective of whether they are implemented in hardware or software.
6. Use of Ports and Channels ensures that compatible AIM Ports may be connected irrespective of the AIM implementation technology.
7. Message generation and Event management is implementation independent.

5.3 AIMs

5.3.1 Implementation types

AIMs can be implemented in either hardware or software keeping the same interfaces independent of the implementation technology. However, the nature of the AIM might impose constraints on the specific values of certain API parameters and different Profiles may impose different constraints. For instance, Events (easy to accommodate in software but less so in hardware); and persistent Channels (easy to make in hardware, less so in software).

While software-software and hardware-hardware connections are homogeneous, a hybrid hardware-software scenario is inherently heterogeneous and requires the specification of additional communication protocols, which are used to wrap the hardware part and connect it to software. A list of such protocols is provided by the MPAI Ontology [10].

Examples of supported architectures are:

- *CPU-based devices* running an operating system.
- *Memory-mapped devices* (FPGAs, GPUs, TPUs) which are presented as accelerators.