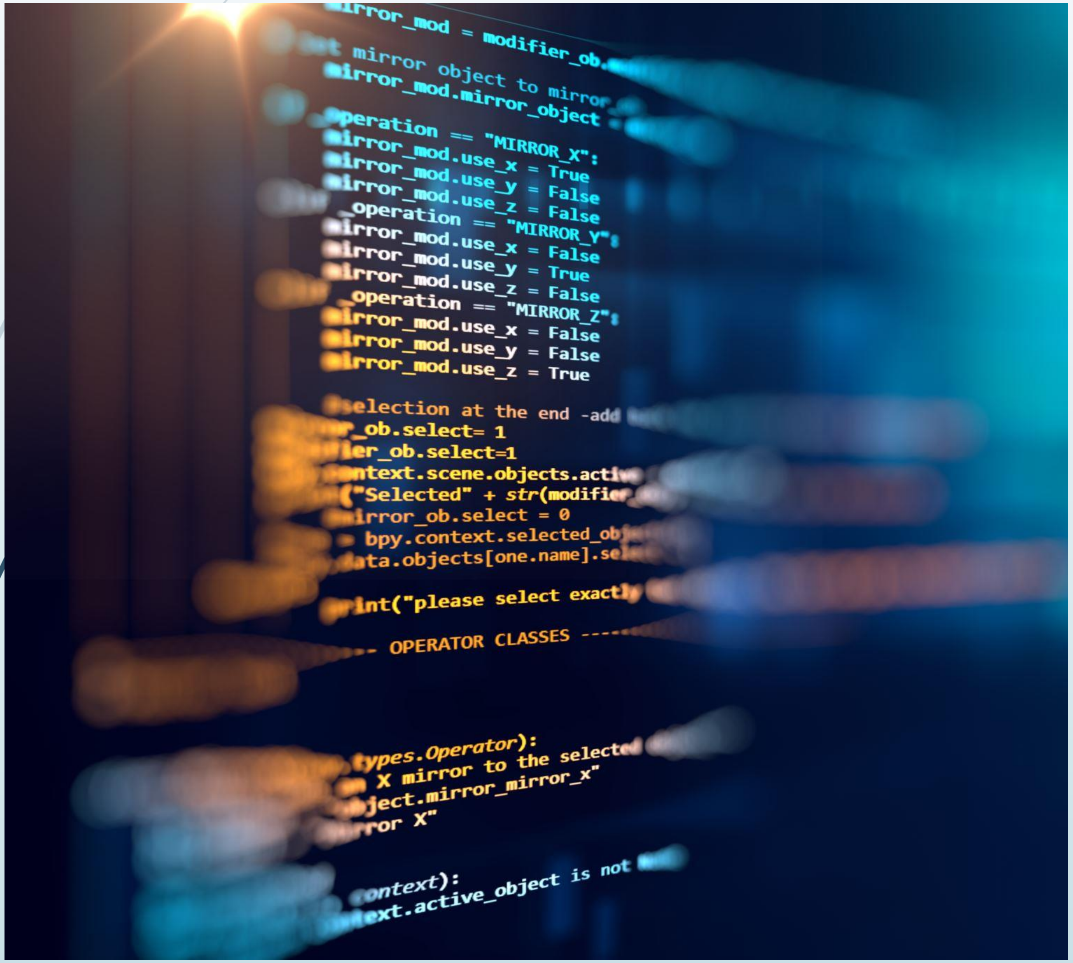
AI Framework (MPAI-AIF)

08 and 15 UTC 11 September 2023

MPAI.
community

Contents of presentation



- About MPAI
- Basics on MPAI AI data coding
- MPAI-AIF Features
- What’s next

About MPAI

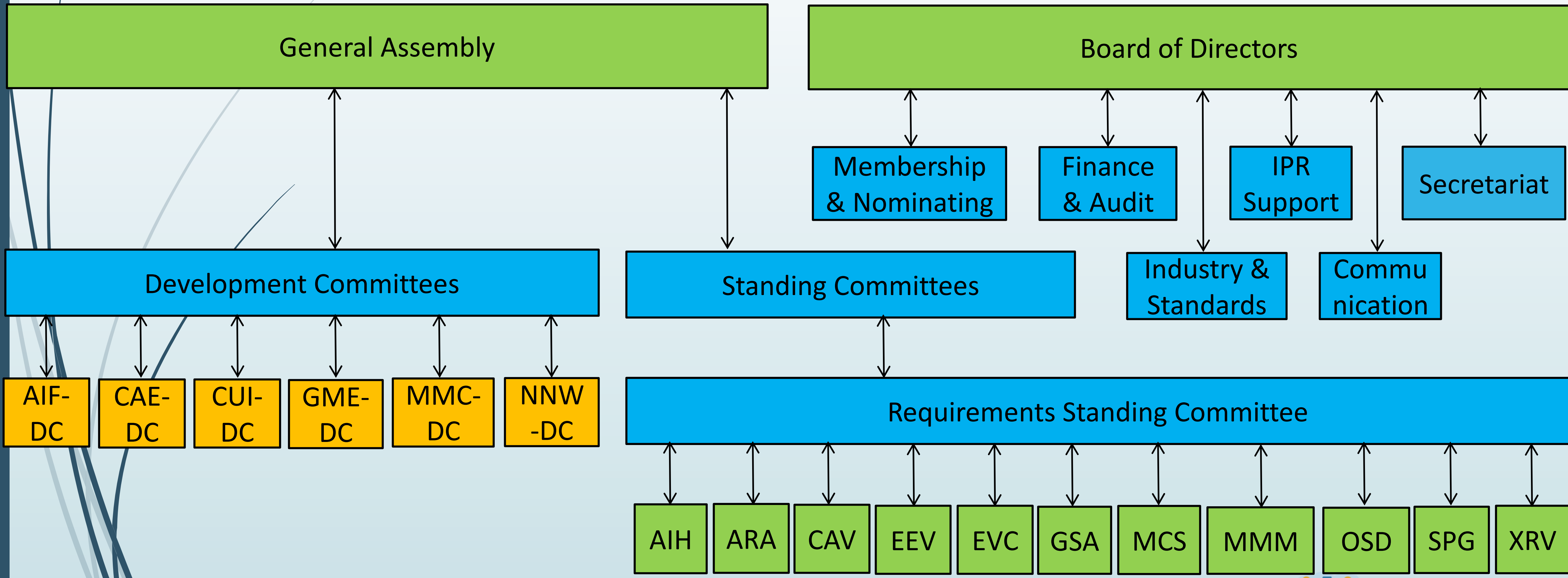
MPAI stands for Moving Picture, Audio, and Data Coding by Artificial Intelligence.

International, unaffiliated, non-profit SDO.

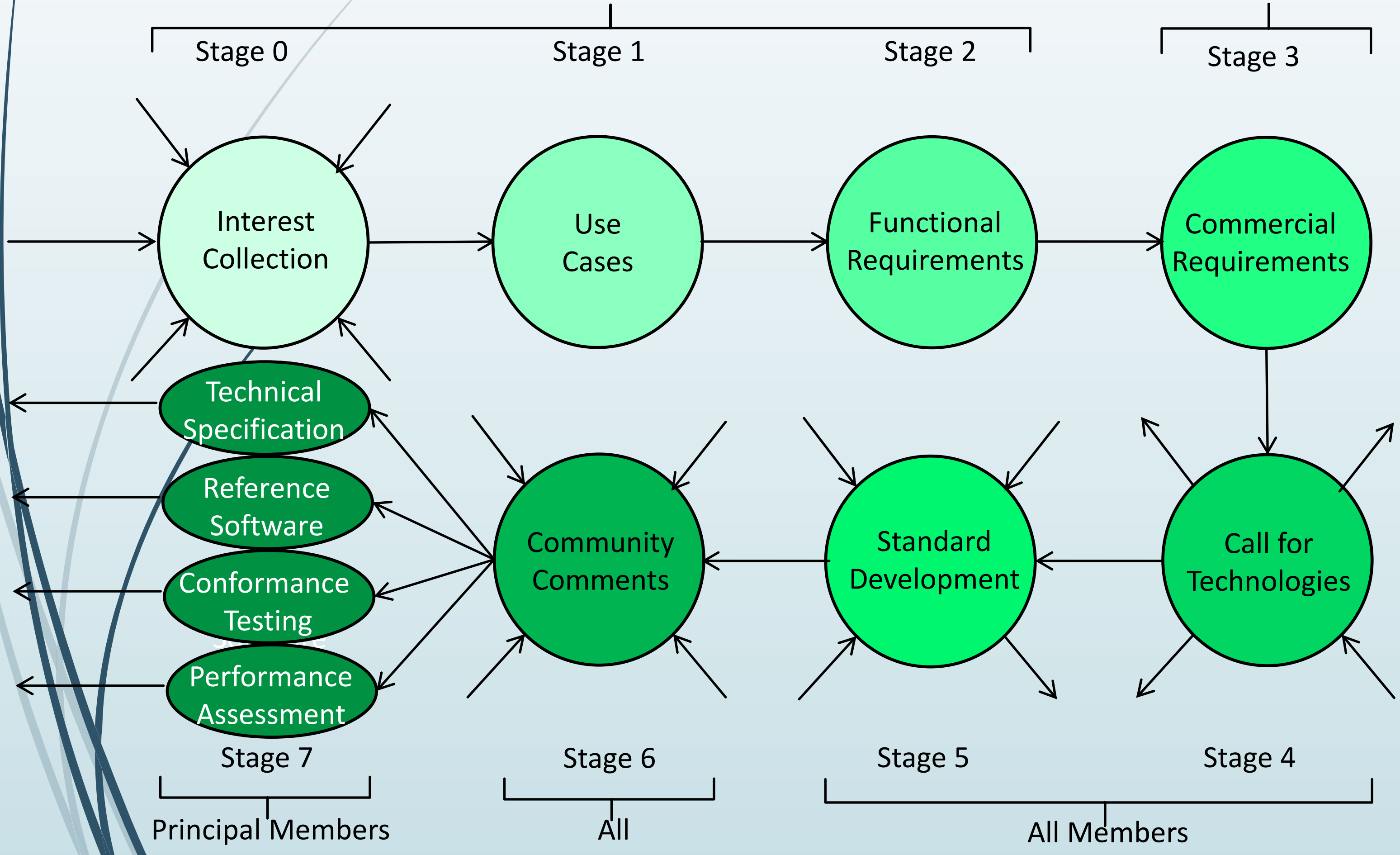
Developing AI-based data coding standards.

With clear Intellectual Property Rights licensing frameworks.

The MPAI organisation



The MPAI standard development process



- Develop Use Cases and Functional Requirements.
- Develop Commercial Requirements (Framework Licence).
- Issue Call for Technologies with attached:
  - Functional Requirements.
  - Commercial Requirements.
- Develop standard (MPAI members only).
- SEP holders select patent pool administrator.

MPAI standards for a better AI

- MPAI's data coding standards make explicit the computing workflow of AI applications.
- An MPAI standard **breaks up monolithic AI applications** into a set of interacting components of known data semantics (as far as possible).
- **Developers compete** offering "improved" performance "standard" components.
- Humans can **select applications whose internal operation they can somehow understand**.

MPAI's AI standardisation is "component-based".

An AI application is:

- Subdivided in smaller components: AI modules (AIM).

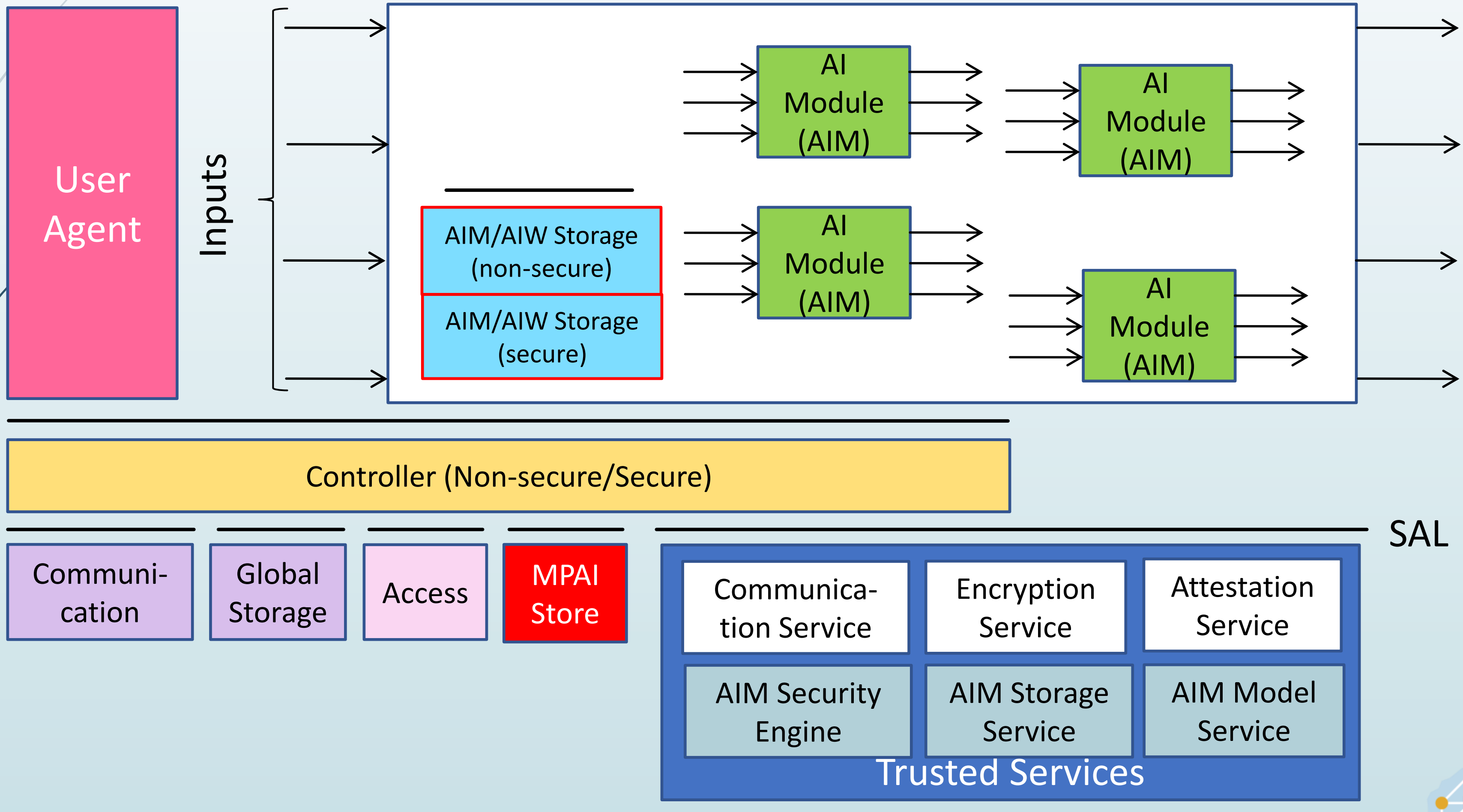
- Aggregated in one or more AI workflows (AIW).

- Executed in a standard environment (AIF).

1 foundational Technical Specification

AI Framework (MPAI-AIF)

The MPAI AI Framework

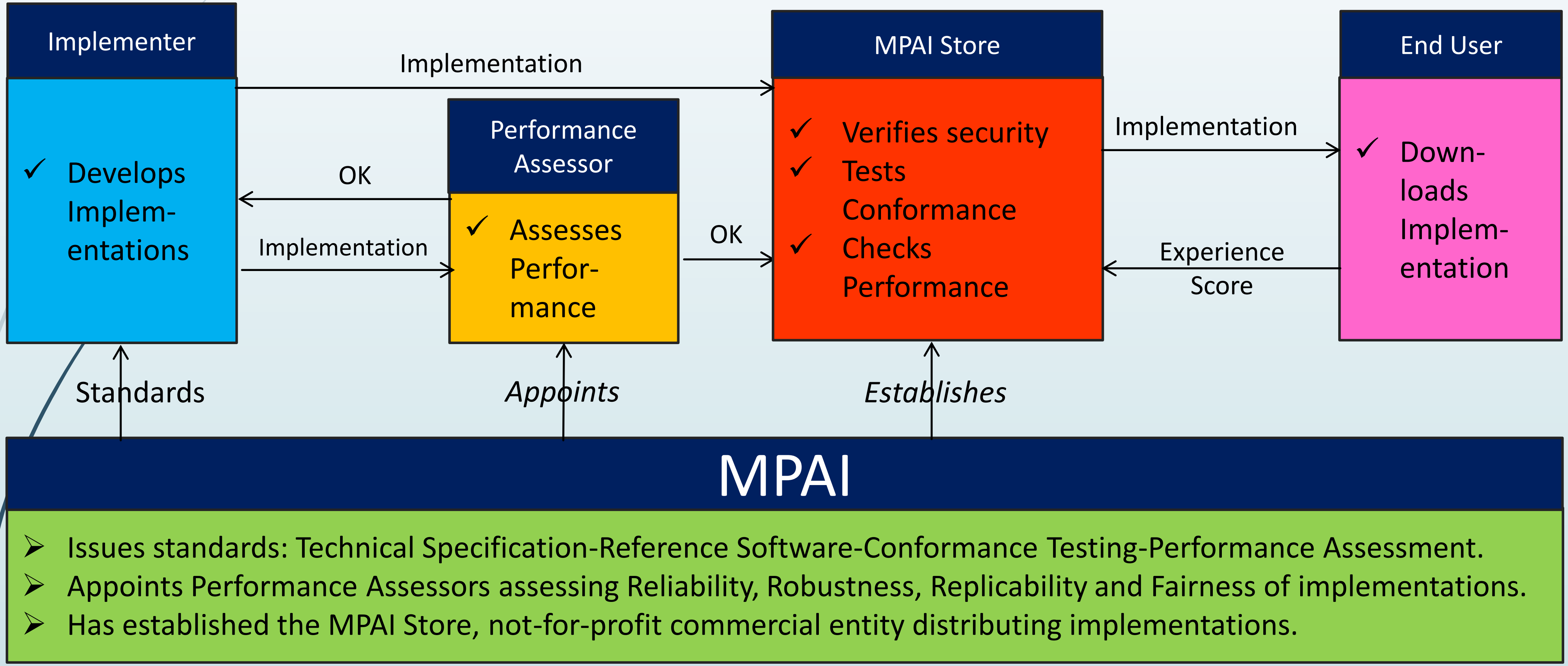


A sustainable MPAI Ecosystem

- **MPAI standards** create an ecosystem composed of:
  - **Developers**: develop components
    → require interoperability to bring their components to the market.
  - **Integrators**: assemble components
    → require ability to assemble third party components.
  - **Consumers**: use assembled components
    → require that the assembled components be trusted.
- The MPAI Store guarantees that AIMs/AIWs are:
  - Interoperable.
  - Trusted.
  - Available.

***1 system Technical Specification*:**
Governance of the MPAI Ecosystem (MPAI-GME).

The MPAI ecosystem



Implementer

✓ Develops Implem-entations

Implementation

Performance Assessor

✓ Assesses Perfor-mance

OK

Implementation

OK

MPAI Store

✓ Verifies security

✓ Tests Conformance

✓ Checks Performance

Implementation

End User

✓ Down-loads Implem-entation

Experience Score

Standards

Appoints

Establishes

MPAI

- ➢ Issues standards: Technical Specification-Reference Software-Conformance Testing-Performance Assessment.
- ➢ Appoints Performance Assessors assessing Reliability, Robustness, Replicability and Fairness of implementations.
- ➢ Has established the MPAI Store, not-for-profit commercial entity distributing implementations.

More published MPAI standards

4 Technical Specifications

1 - Context-based Audio Enhancement (MPAI-CAE)

2 - Compression and Understanding of Financial Data (MPAI-CUI)

3 - Multimodal Conversation (MPAI-MMC)

4 - Neural Network Watermarking (MPAI-NNW)

2 Technical Reports

1 - MPAI Metaverse Model (MPAI-MMM) – Functionalities

2 - MPAI Metaverse Model (MPAI-MMM) – Functionality Profiles

Five standards published for Community Comments to become standards on 29 September

Existing MPAI standards extended

1 - AI Framework V2 (MPAI-AIF)

2 - Multimodal Conversation V2 (MPAI-MMC)

New MPAI standards being approved

3 - Avatar Representation and Animation V1 (MPAI-ARA)

4 - Connected Autonomous Vehicles V1 (MPAI-CAV) – Architecture

5 - MPAI Metaverse Model V1 (MPAI-MMM) – Architecture

Brewing in the pot

Calls for Technologies issued

1 – Artificial Intelligence for Health (MPAI-AIH)

2 – Object and Scene Description (MPAI-OSD)

3 – Extended Reality Venues (MPAI-XRV) - Live Theatrical Stage Performance

New opportunities being explored

1 - AI-based End-to-End Video Coding (MPAI-EEV)

2 - AI-Enhanced Video Coding (MPAI-EVC)

3 - Server-based Predictive Multiplayer Gaming (MPAI-SPG)

MPAI and IEEE

MPAI Technical Specifications adopted as IEEE standards

1. ***MPAI-AIF – 3301-2022***
2. ***MPAI-CAE – 3302-2022***
3. ***MPAI-MMC – 3300-2022***
4. ***MPAI-CUI – 3303-2023***
5. ***MPAI-NNW (on its way)***

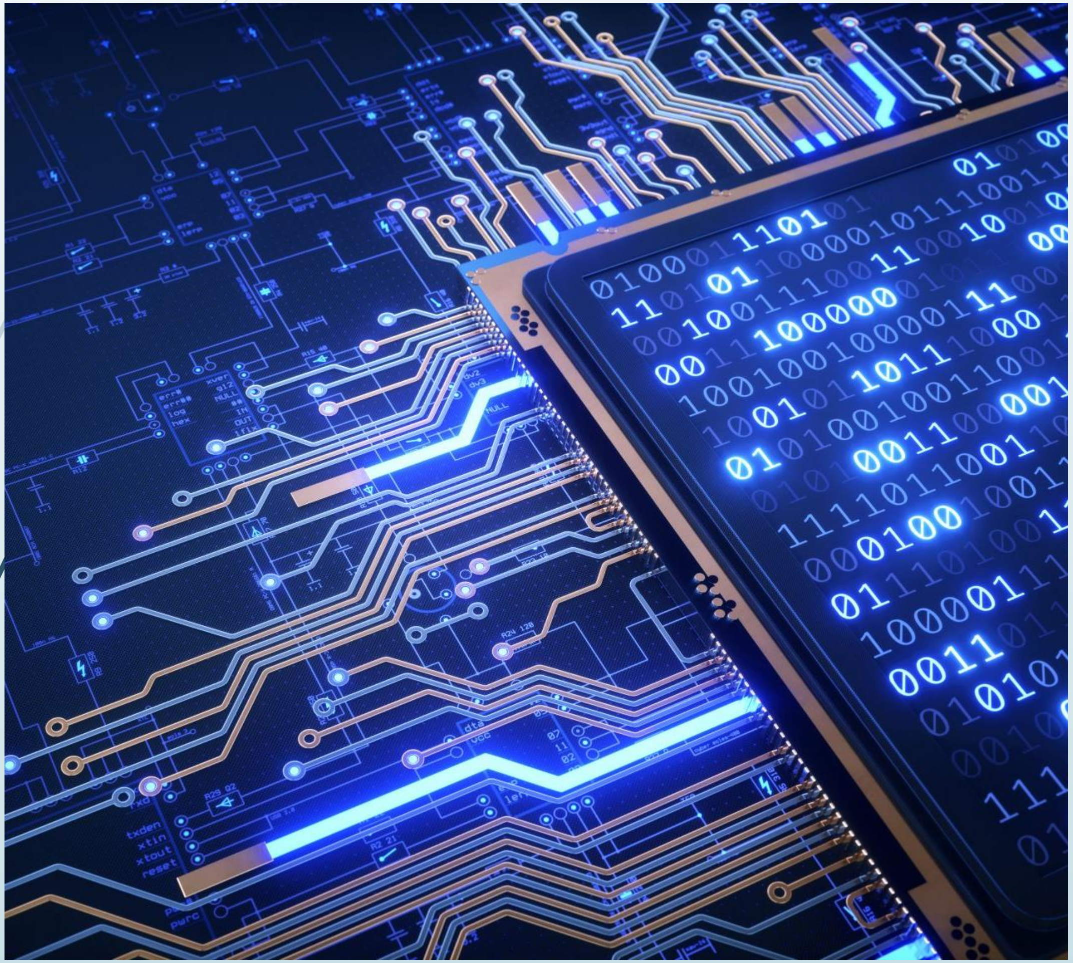
All this achieved in less than 3 years!

Basics on MPAI AI data coding

The MPAI way to AI-based data coding

- Application-oriented MPAI standards **break up monolithic AI applications** into a set of interacting components.
- The **semantics of the data exchanged** between components is known as far as possible.
- **Developers can compete** by providing “standard” components with “improved” performance.
- The MPAI AI Framework standard makes possible this **“Lego-type” approach**:
  - “Applications” (called AI Workflows – **AIWs**)
  - Composed of AI Modules (called **AIMs**)
  - Executed in AI Frameworks (called **AIFs**).

Technical Specification: MPAI AI Framework (MPAI-AIF)



- **Specifies** the following **elements of the AI Framework** specially designed for execution of AI-based implementations.
  - **Architecture**
  - **Interfaces**
  - **Protocols**
  - **Application Programming Interfaces** (API).
- **Mixed** AI and traditional data processing workflows are also supported.

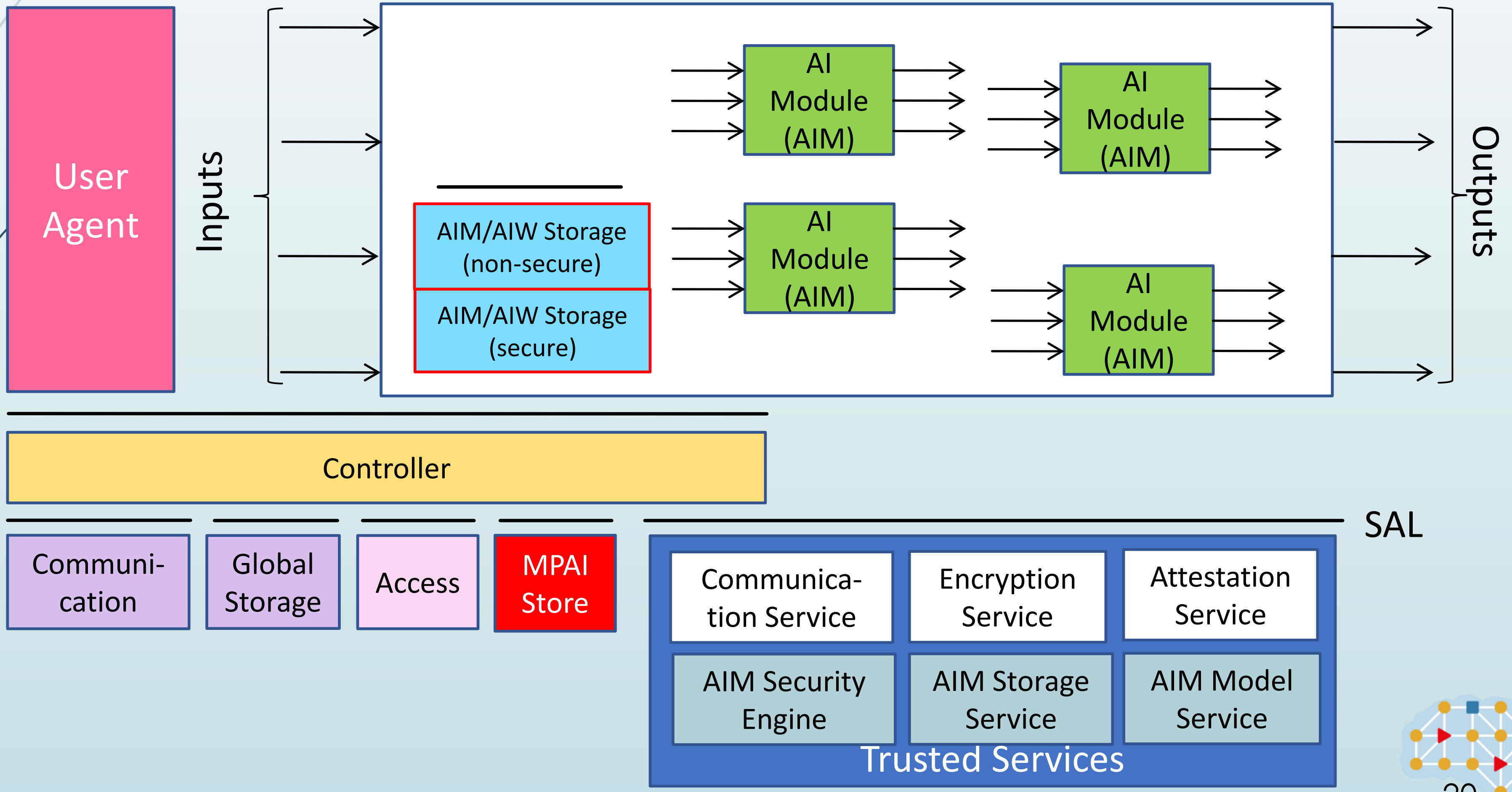
Basic functionalities

- **Independent** of Operating System.
- **Modular** component-based architecture with specified interfaces.
- **Interfaces** encapsulate Components to abstract them from the development environment.
- **MPAI Store interface** enabling access to validated Components.
- Component can be Implemented as:
  - **Software only**, from MCUs to HPC.
  - **Hardware only**.
  - **Hybrid** hardware-software.
- Component system features are:
  - **Execution** in local and distributed Zero-Trust architectures.
  - Possibility to **interact with other Implementations** operating in proximity.
  - Direct support for **Machine Learning** functionalities.
- The AIF can download an AIW whose identifier has been specified by the User Agent or by a configuration parameter.

Secure functionalities

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

MPAI-AIF Reference Model



MPAI-AIF Features

Non-secure AIF Components

Access: to access to static or slowly changing data such as domain knowledge data, data models.

AI Module (AIM): a data processing element receiving Inputs and producing Outputs according to its Function. An AIM may be an aggregation of AIMs.

AI Workflow (AIW): an organised aggregation of AIMs implementing a Use Case.

Communication: connects the Components of an AIF.

Controller: can run May run one or more AIWs and exposes three APIs:
i. ***AIM API*** modules can register, communicate and access the AIF environment; can start, stop, and suspend AIMs.
ii. ***User API U***ser or other Controllers can perform high-level tasks (e.g., switch the Controller on and off, give inputs to the AIW through the Controller).
iii. ***MPAI Store API*** communication between the AIF and the Store.

Global Storage: stores data shared by AIMs.

AIM/AIW Storage: stores data of the individual AIMs (securely/non-securely).

MPAI Store: stores Implementations for users to download.

User Agent: The Component interfacing the user with an AIF through the Controller

AI Framework Implementations

1. AIF **Implementations can be tailored** to different execution environments, e.g., HPC or MCU.
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:
    a. To run on different computing platforms and different programming languages.
    b. To be based on different hardware and resources available.
4. AIMs may be Implemented in **HW, SW and mixed**-HW and SW.
5. AIM Interoperability ensured by **definition of communication** between AIMs, irrespective of HW or SW implementation.
6. Ports and Channels ensure **connection of compatible AIM Ports** irrespective of the AIMs' implementation technology.
7. **Implementation-independent** Message generation and Event management.

Advantages of the MPAI AI Framework (MPAI-AIF)

Enables **creation, execution, composition and update of AIM-based workflows** for high-complexity AI solutions interconnecting multi-vendor AIMs trained to specific tasks, operating in the standard AI framework and exchanging data in standard formats.

Benefit various actors:

- ✓ **Technology providers** can offer their conforming AI technologies to an open market
- ✓ **Application developers** can on the open market for their applications need
- ✓ **Innovation** is fueled by the demand for novel and more performing AI components
- ✓ **Consumers** have a wider choice of better AI applications by a competitive market
- ✓ **Society** lifts the veil of opacity from large, monolithic AI-based applications.

Secure AIF Components

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

JSON metadata

- The **capabilities of the AIF** described by a standard JSON metadata format.
- The **capabilities of the AIW** are described by a standard JSON metadata format.
- The **capabilities of (Composite) AIMs** are described by a standard JSON metadata format (similar to the AIW metadata format).
- An AIF **downloads** suitable AIW and AIMs **from the MPAI Store** using the JSON metadata.

Application Programming Interfaces - Basic

#	API	#	API
8.1	Store API called by Controller	8.3	Controller API called by AIMs
8.1.1	Get and parse archive	8.3.1	General
8.2	Controller API called by User Agent	8.3.2	Resource allocation management
8.2.1	General	8.3.3	Register/deregister AIMs with the Controller
8.2.2	Start/Pause/Resume/Stop Messages to other AIWs	8.3.4	Send Start/Pause/Resume/Stop Messages to other AIMs
8.2.3	Inquire about state of AIWs and AIMs	8.3.5	Register Connections between AIMs
8.2.4	Management of Shared and AIM Storage for AIWs	8.3.6	Using Ports
8.2.5	Communication management	8.3.7	Operations on messages
8.2.6	Resource allocation management	8.3.8	Functions specific to machine learning
		8.3.9	Controller API called by Controller

Application Programming Interfaces - Security

#	API	#	API
9.1	Data characterization structure.	9.5	API to access cryptographic functions
9.2	API called by User Agent	9.5.1	Hashing
9.3	API to access Secure Storage	9.5.2	Key management
9.3.1	User Agent initialises Secure Storage API	9.5.3	Key exchange
9.3.2	User Agent writes Secure Storage API	9.5.4	Message Authentication Code
9.3.3	User Agent reads Secure Storage API	9.5.5	Cyphers
9.3.4	User Agent gets info from Secure Storage API	9.5.6	Authenticated encryption with associated data (AEAD)
9.3.5	User Agent deletes a p_data in Secure Storage API	9.5.7	Signature
9.4	API to access Attestation	9.5.8	Asymmetric Encryption
		9.6	API to enable secure communication

Profiles

Basic Profile

- The Basic Profile utilises:
  1. Non-Secure Controller.
  2. Non-Secure Storage.
  3. Secure Communication enabled by secure communication libraries.
  4. Basic API.

Secure Profile

- The Secure Profile utilises all the technologies in this Technical Specification.

What’s next

What's next?

- Anybody is entitled to comment on AI Framework V2.
- Comments should be sent to secretariat@mpai.community by 2023/09/24 T23:059 UTC
- MPAI plans on **publishing MPAI-AIF as a Technical Specification** at the 36th General Assembly (29 September 2023).
- MPAI plans to continue the **implementation of AIF V1** for more OSs and programming languages than currently available.
- MPAI plan to implement the **Reference Software of MPAI-AIF V2**.

We look forward to working with you on this exciting MPAI project!

Join MPAI
Share the fun
Build the future

MPAI. community