

The MPAI Metaverse Model (MPAI-MMM)

Presentation of 08 and 15 UTC 1st September 2023



Contents of presentation



- MPAI's approach to the metaverse
- Metaverse functionalities
- M-Instance operation
- Functional Requirements of Processes
- Functional Requirements of Actions
- Functional Requirements of Items
- Functional Requirements of Data Types
- Use Cases
- Functional Profiles
- Community Comments



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
 - \rightarrow require interoperability to bring their components to the market.
 - Integrators: assemble components
 - \rightarrow require ability to assemble third party components.
 - Consumers: use assembled components
 - \rightarrow 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).

8

community

The MPAI ecosystem



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)

13-Sep-

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!





MPAI's approach to metaverse



What is MPAI's relationship with the Metaverse?



- Industry interest in the Metaverse
 New jobs-opportunities-experiences.
- Standards & AI main drivers for the development of the Metaverse
 → MPAI is in both.
- Standards are needed
 → MPAI has already a few.
- Metaverse requires more standards
 MPAI may develop some.
- ► Technology integration for metaverse
 → MPAI develops reference software.



Different types of metaverse interoperability

Prescriptive interoperability (as in media).

The metaverse context is fast-evolving; technology has many configurations.



⁷Lack of interoperability caused by the "Everyone for himself" approach. Conversion services do not guarantee interoperability for all cases.





Functional interoperability

- In general, data format conversion is possible when data formats reference the same functionalities.
- We should specify the functional requirements of:
 - The data provided by an M-Instance to another M-Instance or to a Device.
 - The operation model of an M-Instance.
- Functional interoperability enables the conversion of data with the format of M-Instance_A to data with the format of M-Instance_B.



data_{B.1}

data_{B.1}

data_{B.2}

data_{A.1}

data_{A.1}

The MPAI-MMM – Architecture standard



- Functional Requirements of
 - Processes
 - Actions
 - Items
 - Data Types
- Use Cases
- Functionality Profiles



Introduction



- Technical Specification: MPAI Metaverse Model (MPAI-MMM) – Architecture specifies:
 - Operation Model of an M-Instance.
 - Functional Requirements of
 - Processes running in an M-Instance.
 - Actions performed by Processes.
 - Items on which Actions are performed
 - Data Types supported by the M-Instance.
 - The protocol enabling a Process in M-Instance_A to request a Process in M-Instance_B to execute Actions on Items.
 - The Functional Profiles supported by an M-Instance.

community

The scope of MPAI-MMM – Architecture



MPAI-MMM – Architecture enables interoperability of two or more M-Instances if they:

1. Rely on the same Operation Model.

2. Use:

- 1. Either the same Technologies, or
- 2. Independent Technologies while accessing *appropriate* Conversion Services, and
- 3. The same Profile specified by MPAI-MMM - Architecture.



Terms and Definitions

- Terms are important for any standard.
- More so for a functional standard.
- MPAI-MMM Architecture collects about 100 metaverse-related normative definitions.



Normative References



- 1. MPAI; The MPAI Statutes; https://mpai.community/statutes/.
- 2. MPAI; The MPAI Patent Policy; https://mpai.community/about/thempai-patent-policy/.

3. Framework Licence: MPAI Metaverse Model (MPAI-MMM) – Architecture; <u>https://mpai.community/standards/</u> <u>mpai-mmm/framework-licence/</u>.





Metaverse Functionalities



Functionalities of MPAI-MMM – Architecture



- Characteristics of metaverse.
- Processes and Certification.
- Registration and Accounts
- Users and Personae
- Rules
- Rights
- Locations
- Objects
- M-Instances in general



Metaverse functionalities: what is an M-Instance

- 1. An **M-Instance is composed of Processes** providing the following **functions**:
 - **1. Sensing data** from U-Locations using Devices.
 - 2. Processing the sensed data and producing Items.
 - 3. Producing M-Environments populated by imported or internally generated Items.
 - 4. Processing Objects from this or other M-Instances to affect U- and/or M-Environments in ways that are:
 - 1.Consistent with the **goals** set for the M-Instance.
 - 2.Effected within the capabilities of the M-Instance.
 - 3.In compliance with the **Rules** set for the M-Instance.



Metaverse Functionalities – Processes and Certification

- 2. Process types:
 - **1. Devices**: connecting U-Locations with M-Locations and vice-versa.
 - 2. Users: representing humans as their agents in the M-Instance or on Devices.
 - **3. Apps**: running on Devices. A User may be an App running on a Device.
 - **4. Services**: providing generic Functionalities.

- 3. Certification:
 - **1. Processes** may be connected or imported to an M-Instance if they have passed a Certification Process.
 - **2. Contracts** may be imported to an M-Instance if they have passed a Certification Process.



Metaverse Functionalities – Register

- 4. A human, possibly on behalf of a legal entity, may **Register with an M-Instance**.
- 5. To create an Account, the M-Instance may request:
 - 1. A subset of the human's Personal Profile.
 - 2. WalletIDs
 - 3. Users
 - 4. Personae.

- 6. The Account may include:
 - 1. A subset of the human's Personal Profile.
 - 2. WalletIDs
 - 3. Users
 - 4. Personae
 - 5. Rights and Obligations



Personal Profile, Users, and Personae

- 7. Personal Profile may have a scalable representation to enable an easy extraction of a Personal Profile subset.
- 8. Users
 - 1. Represent humans.
 - 2. Act under the responsibility of a human.
 - 3. May be rendered as Personae.

- 9. Personae may:
 - 1. Faithfully reproduce the visual appearance of the human.
 - 2. Have an altered appearance, compared to that of the human.
 - 3. Have an unrelated visual appearance.
 - 4. Display a presumptive Personal Status in speech, face, and gesture.
 - 5. Be driven by
 - 1. The movements of the human.
 - 2. A Process.



Metaverse Functionalities – Identifiers and User Data



10. An **Identifier**:

- **1. Uniquely references** an Item, an M-Location, or a Process.
- 2. May be **issued by** a central, distributed, or no **authority**.
- 3. May have a **Format** accepted by more than one M-Instance.
- 4. May enable to trace the Identifier of
 - 1. An Item Back to the **Item** that spawned the **Asset**.
 - 2. A **Modified Item** back to the **Item** that spawned the Modified Item.

community

Metaverse Functionalities – Rules



11. The **Rules** of an M-Instance may include:

- 1. The User Data retention, processing, and access policy.
- 2. The **Rights and Obligations**, e.g., ethical behaviour of Users.
- 3. The **permission**-based or permission-less **access policy** to M-Environments.
- 4. The trusted or **trust-less relationship** between Users and M-Instance.
- 5. The **subset of User Data** required for each Account type.
- 6. The **sanctions** applied to a User whose Actions do not comply with the Rules.



Metaverse Functionalities – Process



12. A **Process may request another Process** in another M-Instance to perform Actions.

13. A User may formulate:

- 1. The **information** it intends to obtain from a Service.
- 2. The **Rights** it intends to acquire on the response of the Service.
- 3. The **rendering** of the response.



Metaverse Functionalities – Functionalities – Rights

14. Rights may include

- . The **maximum number** of:
 - **1. Users** a human can deploy.
 - **2. Personae** a User can be rendered as.
 - **3. Concurrent Users and Objects** an M-Location can support.

- 2. The **ability** to:
 - **1. Create persistent** (as opposed to ephemeral) Scenes at M-Locations.
 - 2. Store, search, and retrieve Experiences.
 - 3. Develop economic activities.
 - 4. Book M-Locations for a duration.
 - 5. Support **persistent storage**.
 - 6. Connect with or import Processes to an M-Instance.
 - 7. Make Transactions to acquire Rights.

Metaverse Functionalities – Perception-related Rights to perform

15. (Perception) A User may, depending on the Rights it holds:

1. Perceive Entities.

- **2.** Perceive scenes at U-Locations as captured by Devices endowed with audio, visual, haptic, and BCI capabilities,
- 3. Import audio, speech, visual, and haptic models.
- 4. Render perceived Entities at a U-Location with spatial attitudes.
- **5.** Receive Messages from a U-Location as a result of an event there.
- 6. Import Data and Metadata from and export Items to an Address.



Metaverse Functionalities – Interaction-related Rights to perform



- 16. (Interaction) A User may, depending on the Rights it holds:
 - **1. Interrogate a Process** about its Functionalities.
 - **2.** Interact with Entities.
 - **3. Send public or private media** (Speech, Visual, and Haptic Messages) to Users.
 - 4. Send selected Entities to a User.
 - **5. Place Entities** with Spatial Attitudes.
 - 6. Animate Models.



Metaverse Functionalities – Rights to Call a Service/1

- 17. (Service call) A User may, depending on the Rights it holds, perform or request a Service to:
 - **1. Author** an Item.
 - 2. Authenticate an Item.
 - **3.** Change the Rights of a Process.
 - **4. Discover** Objects, Processes, Events, or Experiences.

- **5. Find** the Transactions involving an Asset under specified conditions.
- **6. Obtain** an interpretation of an Item, e.g.:

Request	What	From
Extract	Personal Status	Text
Extract	Text	Speech
Translate	Text or Speech	Text or Speech
Extract	Personal Status	Face and/or Gesture



Metaverse Functionalities – Rights to Call a Service/2

7. Obtain Conversion of an Item from a Format to another Format.

- **8. Obtain** information about a Process or an Item.
- 9. Obtain a recommendation of Processes or Items.
- **10.Record** all the Actions made by a User (Activity Data) at an M-Location during a time.
- **11.Create** a representation of a User's network of connections with Items, M-Locations, and Processes (Social Graph).
- **12.Post** an Item that can be Transacted (Asset) to a marketplace.
- 13. Retrieve Objects, Events, or Experiences.
- 14.Store Objects, Events, or Experiences.


Metaverse Functionalities – Rights

18.A User may have as part of its Account or acquire the Rights to:

- **1. Perform** or request another User to perform Actions on Items.
- **2. Develop** an economic activity in an M-Environment.
- 3. Make:
 - 1. An Asset out of an Item.
 - 2. A Transaction on an Asset specifying which Rights to the Asset it grants to the buyer.
 - 3. Make an agreement with another user, e.g., via a Contract.

- **4. Copy or Modify**, i.e., change Data (e.g., by editing it) and/or Metadata of an Item.
- **5. Change** the settings and attributes of Entities of an M-Locations (e.g., SFX during a concert).
- **6. Set** the Point of View from which it perceives a Scene.
- 7. Make inaccessible to any User an Item for which it has Rights.
- 8. Make or request a Process to make Actions in its own or another M-Instance.
- **9. Be rendered** as one of the Account's Personae.



Metaverse Functionalities – M-Location and Object

19. An M-Location may be:

- **1. Private**, i.e., only a User or a group of Users have Rights to it.
- 2. Public, i.e., a Metaverse Operator or a User grant anybody the Rights to enter, move in, and perceive the M-Location.
- 3. Persistent.
- 4. Ephemeral.

20. An **Object** may:

- 1. Be **Virtual**, i.e., Authored in an M-Instance.
- 2. Be **Digitised**, i.e., Digital Twins of objects.
- 3. Be **Autonomous**, i.e., Animated by a Process.
- **4.** Have Audio, Visual, or Haptic attributes, or combinations of these attributes.



Metaverse Functionalities – M-Instance (general)

21. An M-Instance:

1. May operate in a **centralised or decentralised** way.

2. Monitors:

- 1. The Rights held by Processes on Processes to Act on Items at M-Locations.
- 2. The Entities placed at M-Locations with a Spatial Attitude.
- **3. Should be secure**, e.g., to prevent malicious Users from:
 - 1. Accessing the Account of a human who is not the human represented by the User.
 - 2. Taking control of a Process.
 - 3. Acting on Items for which it does not have Rights.
- 4. Should withstand denial of service attacks.







Overview of M-Instance Features



election at the end -add ob.select= 1 er_ob.select=1 ntext.scene.objects.active "Selected" + str(modifie rror_ob.select = 0 bpy.context.selected_ob ra.objects[one.name].selected_ob

mint("please select exactle

OPERATOR CLASSES ----

wpes.Operator): X mirror to the selected ject.mirror_mirror_x" ror X" ontext): wt.active_object is not

- Definition of M-Instance
- Functions of an M-Instance
- Minimum Process Metadata
- Communication between Processes
 - Inside an M-Instance
 - Outside of an M-Instance
- Types of Process
- Minimum Item Metadata



M-Instance Definition



An ICT platform with the following general functions:

- 1. To sense data from U-Locations.
- 2. To process the sensed data and produce Data.
- 3. To produce one or more M-Environments populated by Objects that can be either digitised or virtual, the latter with or without autonomy.
- 4. To process Objects from the M-Instance or potentially from other M-Instances.
- 5. To affect U- and/or M-Environments using Objects in ways that are:
 - 1. Consistent with the goals set for the M-Instance.
 - 2. Within the Capabilities of the M-Instance.
 - 3. According to the Rules of the M-Instance.
 - 4. Respecting applicable laws and regulations.
- 6. To operate as and providing the Functionalities of this Technical Specification.

An M-Instance:

- 1. May make its Capabilities known.
- 2. May be administratively subdivided in M-Environments.
- 3. May allow a human to Register with the M-Instance or an M-Environment.
- 4. May request a Registering human to provide a subset of their Personal Profile.
- **5.** Is implemented as **a set of Processes**:
 - 1. Providing Functionalities defined by this Technical Specification.
 - 2. Composed of:
 - 1. A Program having a Format supported by the M-Instance.
 - 2. Metadata describing the Program and containing the following as a minimum:

Table 2 - Minimum Metadata set of a Process

Metadata elements	Details
ProcessID	The ID of the Process.
UserID	The ID of the User having Rights to Act on the Process.
InRightsID	The ID of the User's Rights to Act granted to the Process.
OutRightsID	The ID of the Rights a User may acquire on the Process.
DescrMdata	Any description of the Process.



(Is implemented as a set of Processes)

- 3. Performing Actions on Items if they hold the Rights to do that.
- **4. Requesting other Processes to perform Actions** by transmitting a standard Request-Action Item and responding to a Request-Action with a Response-Action:

Response-Action	Details
Response-Action ID	Unique ID
Emission Time	Tine of Issuance
Source Process ID	Requesting Process ID
Destination Process ID	Requested Process ID
OutItems	In/Output Items required by the Action
	Locations of InItems
	Locations of OutItems
	Expected Rights on OutItems
	Response-ActionResponse-ActionEmissionEmissionSourceProcessDestinationProcessOutItems

45 • community

(Is implemented as a set of Processes)

5. Performing the Request if:

1. The requesting Process holds the Rights that are required to perform the request.

/2. The requested Process holds Rights to perform the requested Action on the Item.

6. Enables a Process:

- 1. To **communicate to a Process in another M-Instance** through an M-Instance's Resolution Service.
- 2. To obtain conversion of the Format of an Item's Data by calling a Conversion Service.
- 3. To specify their communication needs by:
 - 1. Requesting the needed maximum and average bitrate value.
 - 2. Reserving the needed bitrate for a time and a location.
 - 3. Requesting that the same simultaneous Experience be provided to a specified-number of Devices.



7. May support four types of Process:

- 1. <u>Device</u> is one or more **Processes running on a physical device** connecting M-Locations to U-Locations, specifically the Registered human, by:
 - **1. Capturing scenes as Media** and providing Media as Data/Metadata U-Location to M-Location). Data/Metadata shall be Identified as an Item to enable Processes to perform Actions.
 - **2. Receiving an Entity** and rending it as Media with a Spatial Attitude (Position and Orientation) M-Location to a U-Location.
- 2. The <u>User</u> Process **represents and is under the control of a human** Registered with the M-Instance. A User may run on the M-Instance or on a Device connected to the M-Instance.
- 3. <u>Service</u> provides **specific functionalities**.
- 4. <u>App</u> runs on a Device. A Device may run the User Process as an App.







- 8. Provides Functionalities through Processes performing Actions on Items containing:
 - **1. Data** whose Format is supported by the M-Instance.
 - **2. Metadata** describing the Item and containing the following as a minimum:

Table 4 - Minimum	Metadata	for Item
-------------------	----------	----------

Metadata elements	Details
ItemID	The ID of the Item.
UserID	The ID of the User having Rights to Act on the Item.
InRightsID	The ID of the User's Rights to Act granted to the Item.
OutRightsID	The ID of the Rights a User may acquire on the Item.
DescrMdata	Any description of the Item.



- **9. Identifies Processes/Items** with one Identifier and, optionally, with more than one Identifier. Each Identifier uniquely refers to one Process/Item.
- **10.Creates Items** by processing the sensed data and the autonomously produced data.
- **11. Employs Data Types** to represent entities such as Amount (of a Currency), length, area, volume, etc.
- **12.Gives** a Registered human the **Rights to deploy** one or more:
 - 1. Users and Devices each with M-Instance-specific Rights.
 - **2. Personae**, i.e., human-like, or humanoid Models used by deployed Users to render themselves.
- 13.Establishes Rules that the Users in the M-Instance shall comply with.





- **14. May penalise Users** for lack of compliance with the Rules.
- 15. May **allow a Process** to perform, or request another Process to perform, Actions on Items even **in the absence of Rights**.
- 16. May require that **Processes be certified** for use in the M-Instance.

Future versions of this document may specify:

- **1.** Profiles and Levels of Actions and Items.
- 2. Additional Actions, Items, and Data Types.





Functional Requirements: Processes



Apps and Services

🖿 Арр

Purpose: An application-specific Program executed on a Device.

- Functional Requirements: The Manager of the M-Instance in which an App will be deployed may request that the Device be subject to Certification.
- Service
 - **Purpose**: A Process that can be called to provide specific Functionalities.
 - **Functional Requirements**: A Service may be:
 - 1. One of the Services natively supported by an M-Instance.
 - 2. Hosted by the M-Instance but provided by a third party.
 - A hosted Service may be subject to certification by the Manager of the M-Instance or an entity delegated by them.

Devices

- Purpose: A Process able to:
 - 1. UM-Capture Data from a U-Location
 - 2. UM-Send Data and Metadata to a User.

and/or

- 1. MM-Send an Entity from an M-Location to the Device.
- 2. MU-Render an Entity at a U-Location.

- Functional Requirements: To connect and interoperate with an M-Instance, a Device needs to
 - 1. Exchange Capabilities with the M-Instance.
 - 2. Use the same Item Data Formats or rely on a Conversion Service.

The Manager of the M-Instance to which a Device is connected may request that the Device be subject to certification.



Users

Purpose: A Process representing a human.

- **Functional Requirements**:
 - 1. A User may perform the following functions:
 - 1. The interface of the human with the M-Instance.
 - 2. Render the User as a Persona:
 - 1. UM-Animated by a Stream.
 - 2. MM-Animated by an autonomous agent.
 - 2. An MM-/UM-Animate Action enabled by a Program produces Animation.

- 3. The Animation Program may be provided:
 - 1. By the human.
 - 2. By the M-Instance.
 - 3. The M-Instance may request that a Program executed in the M-Instance be subject to certification.







Functional Requirements: Actions



Overview of Actions



- General Actions in an M-Instance
- Call a Service
- Manage Entities, Media, scenes and Data/Metadata (Metaverse to Metaverse)
- Manage Entities, Media, scenes and Data/Metadata (Metaverse to Universe)
- Manage Entities, Media, scenes and Data/Metadata (Universe to Metaverse)



General Actions in an M-Instance/1

Functional requirements	Actions
To Register with an M-Instance or M-Environment. Only a	Register
human/legal entity – not a User – may do this.	(human)
To Change, i.e., to increase or diminish, the Rights of a Process,	Change
e.g., because new Rights have been acquired or a User has not	(Rights of
complied with the Rules.	Process)
To Hide, i.e., to disable access to an Item to all Processes (unless	<i>Hide</i> (Item)
it was granted to a Process). The Item is accessible again if the	
User has the Rights re-enable the Item.	
To Authenticate, e.g., to confirm that the speech or the face of	Authenticate
a human or an object imported into an M-Instance is from a	(Item)
specific human or U-Location (place in the real world). The	
Request may include Rights to use the information received.	
	commu

General Actions in an M-Instance/2

	Functional requirements	Actions
	To Identify an Item out of Data and Metadata by Identifying	<i>Identify</i> (Item)
	it. For instance, a Device captures Media as Data subject to	
	certain Rights for use in an M-Instance. Unrecognised Data	
	and Metadata become to an Item usable in the M-Instance.	
	To Modify an original Item with (partially) new Data and	<i>Modify</i> (Item)
	Metadata. E.g., a User with Rights on an Item may wish	
	clone it and modify the components of an existing Item.	
	To Convert an Item of a Request-Action or Response-Action	<i>Convert</i> (Item)
	to another Data Format. Convert can be a Service offered by	
	the M-Instance or available outside of the M-Instance.	
	To Verify that a Process has Rights to make an Action on an	Validate (Process)
	Item, to preserve the integrity of M-Instance operation.	
13-Sep-23		oo 🖌 commu

nity

General Actions in an M-Instance/3

Functional requirementsActionsTo Request a Process to perform an Action on Items, by
transmitting a Request-Action to the Process.Request (Request-
Action)To Respond to a Request-Action after performing it by
transmitting a Response-Action to a Process.RespondTo Execute a Contract, i.e., a Program (and Metadata)
activated by an external entity, e.g., a User or another
Blockchain.Execute (Contract)



Call a Service/1

Functional requirements Actions To **Author** an Item by calling a Service and providing it with Data *Author* (Item) and Metadata. Note: An M-Instance can provide a Service, internal or external to the M-Instance, that Users can call to create Items for use in the M-Instance. To **Discover** Items by giving a description of the Items. *Identify* (Item) Comments: An M-Instance may provide a Service that Users can call to find Items or Processes they need or allow a User to Call an external Service to find Items outside of the M-Instance. To Inform about an Item. A User may wish to know more about *Modify* (Item) an Item, starting from its Metadata but potentially including other information the Service has collected on the Item. To Interpret an Item. E.g., a User requests translation of an Author (Item) utterance, recognise a face, convert a sign language message.

13-Sep-23

Call a Service/2

Functional requirements Actions To **Post** an Asset: a User wishing to surrender (part of) its Rights on an *Post* Asset places the Asset at an M-Location or Posts it to a marketplace. (Asset) To Transact an Asset. A User surrenders to another User (part of) its Transact Rights to an Asset, possibly recognising the role of a Service. After (Asset) Transaction, the parties have different Rights on the Asset and may have different statuses of their Wallets. To **Request** a Resolution Service to enable Process_A in M-Instance_A to *Request* communicate with Process_B in a different M-Instance_B. The Resolution (Request-Service can be offered by the M-Instance or available outside of the Action) M-Instance.



Manage Entities, Media, scenes and Data/Metadata (MM)

Functional requirements Actions To **MM-Add** (place) an Entity at an M-Location while inhibiting *MM-Add* perception by other Users. E.g., without showing the preparations at (Entity) an M-Location. To **MM-Enable**, i.e., to make an Entity perceptible that was not until *MM-Enable* that moment. E.g., the User only wants the Entity to be seen at a (Entity) given time. To **MM-Embed** an Entity, i.e., to MM-Add and MM-Enable the Entity *MM-Embed* at the same time. (Entity) To **MM-Disable**, i.e., to stop making an Entity perceptible. E.g., the *MM-Disable* User wants the setting of an event not to be seen when the event is (Entity) over.

Manage Entities, Media, scenes and Data/Metadata (MM)



Manage Entities, Media, scenes and Data/Metadata (UM)

		Functional requirements	Actions
To	UM	-Capture a scene at a U-Location as Media. E.g., A User	UM-Capture
ask	s a l	Device to capture a scene at a U-Location as Media.	(scene)
To	UM	-Send Data and Metadata to a Process. E.g., a User asks	UM-Send (Data
a D	evio	ce to transmit the Data corresponding to the Media and	& Metadata)
Dev	/ice	Metadata from the Service to another Process.	
To	UM	-Render, i.e., to present a scene at a U-Location to an	UM-Render
M-I	_002	ation as an Entity with a Spatial Attitude:	(scene)
1.	Dev	vice	
	1.	UM-Captures the scene at the U-Location as Media.	
	2.	UM-Sends the Data and Device Metadata to a Service.	
2.	Ser	vice	
	1.	Identifies the Entity composed of Data and Metadata.	
	2.	MM-Embeds it at the M-Location.	

Manage Entities, Media, scenes and Data/Metadata (UM)

Functional requirements

Actions

To **Animate** a Model using a Process that receives a Stream *UM-Animate* from a U-Location and animates the Model. The Process may (Model) be provided by the M-Instance, the human, or a third-party.



Manage Entities, Media, scenes and Data/Metadata (MU)

Functional requirementsActionsTo MU-Actuate, i.e., to present Media (i.e., Data and Metadata of
perceptible information) available at a Device to a U-Location as an
ActuateMU-
ActuateEntity with a Spatial Attitude. E.g., a Device present an Entity it has
received from the M-Instance via an MM-Send Action and is now
Media to a U-Location as a scene.(Media)

To MU-Render, i.e., to present an Entity at an M-Location to a U-MU-Location as a scene with a Spatial Attitude in two steps: MM-SendRenderthe Entity to a Device and MU-Actuate the Media from the Device.(Entity)To MU-Send, an Item to a Device or to store an Item at an Address.MU-Send(Item)



Manage Entities, Media, scenes and Data/Metadata (MUMU)

Functional requirements	Actions
To Track a Model, i.e., to establish a real-virtual-real round trip, by	Track
1. Placing a Model at an M-Location.	(Model)
2. Animating it with a Stream.	
3. MU-Render the animated Model at a U-Location with a Spatial	
Attitude.	





Functional Requirements - Items



Overview of Items



- General Items
- Human and User-related Items
- Items for Process Interaction
- Items for Service access
- Finance-related Items
- Perception-related Items
- Space-related Items



General Items/1

13-Sep-23

Functional requirements	ltem
An M-Instance	M-
1. Is an abstract entity bearing an Identifier.	Instance
2. May expose its Capabilities.	
An M-Environment	M-
1. Is an abstract entity bearing an Identifier.	Environ-
2. Is hosted by an M-Instance.	ment
3. May expose its Capabilities.	
Capabilities may extend/reduce the Capabilities of its hosting M-Instance.	
An Item or a Process shall bear Identifiers so that:	Identifier
1. An Identifier uniquely references an Item or Process.	
2. An Item can have more than one Identifier.	
An Item may have a hierarchical structure, such as:	
Item: M-InstanceID, M-EnvironmentID, M-LocationID, ItemID.	
Process: M-InstanceID, M-EnvironmentID, ProcessID.	
General Items/2

Functional requirements	ltem
Program is Data (and Metadata) that can be executed in the M-Instance. A Program Item may be subject to certification before being admitted to an M-Instance	Program
 Contract is a special Program that can be activated (Executed) by an external entity, e.g., a User or another already activated Contract. The contract shall include 3 entities: 1. Offer: Rights to Items. 2. Acceptance: By both parties. 3. Consideration: There may be a Transaction. The terms of the Contract are enforced in the jurisdiction of the M-Instance. 	Contract



General Items/3

Functional requirements	ltem
The Rights Item expresses the Actions that a Process can perform on	Rights
Items, at M-/U-Locations, during a period.	
Action1 Item1 Location1 Time11-Time12	
Action2 Item2 Location2 Time21-Time22	
An M-Instance/M-Environment may show its Capabilities , i.e., an Item describing the characteristics of an M-Instance/M-Environment, including:	Capabilities
1. Currencies supported.	
2. Items supported with their Data Formats.	

3. Data Types supported.



Interactions between Processes/1

Functional requirements	ltem
Processes may need to exchange application-level Messages.	Message
A Process should be able to expose its Capabilities, i.e., an Item	Capabilities
containing a description of its characteristics including:	
1. List of Actions that can be performed.	
2. List of Items supported with Data Formats.	
3. List of Data Types supported.	
4. The cost of performing an Action.	
5. Human represented (User and Device)	
6. Apps on board (Device).	



Interactions between Processes/2

Functional requirements	ltem
When a Process requests another Process to perform an Action on its	Request-
behalf, it issues a Request-Action, an Item including:	Action
1. Time the Request-Action was issued.	
2. The Source ProcessID.	
3. The Destination ProcessID.	
4. The Action requested.	
5. The Items relevant to the Action (InItems).	
6. The Locations of the Items (InLocations).	
7. The Locations of the output Items (OutItems) produced by the Request-	
Action (OutLocations).	
8. The requested Rights on the OutItems.	



Interactions between Processes/3

Functional Requirements	ltem
When a Process receives and performs a Request-Action, it provides a	Response
Response-Action, i.e., an Item containing:	-Action
1. Time the Response-Action was issued.	
2. The Source ProcessID (Source = Process that issued the request).	
3. The Destination ProcessID.	
4. The output Items produced by performing the Request-Action.	



Human and User-related Items/1

 Account uniquely references a human who has Registered. A human may have more than one Account with one or more M-Instances or M-Environments. An Account shall include: The ID of the Registered human. An M-Instance-specific subset of the Registered human's User Data. The Rights held by each Users in the M-Instance/M-Environment. 	liem
 A human may have more than one Account with one or more M-Instances or M-Environments. An Account shall include: 1. The ID of the Registered human. 2. An M-Instance-specific subset of the Registered human's User Data. 3. The Rights held by each Users in the M-Instance/M-Environment. 	count
 or M-Environments. An Account shall include: 1. The ID of the Registered human. 2. An M-Instance-specific subset of the Registered human's User Data. 3. The Rights held by each Users in the M-Instance/M-Environment. 	
 An Account shall include: 1. The ID of the Registered human. 2. An M-Instance-specific subset of the Registered human's User Data. 3. The Rights held by each Users in the M-Instance/M-Environment. 	
 The ID of the Registered human. An M-Instance-specific subset of the Registered human's User Data. The Rights held by each Users in the M-Instance/M-Environment. 	
 An M-Instance-specific subset of the Registered human's User Data. The Rights held by each Users in the M-Instance/M-Environment. 	
3. The Rights held by each Users in the M-Instance/M-Environment.	
4. The IDs of Devices, Apps, and Users, and Personae.	
5. The validity of:	
1. Rights.	
2. Account.	



Human and User-related Items/2

	Functional Requirements	Item
	Activity Data is an Item containing the record of the Actions made by a User at all	Activity
	M-Locations for a period. Therefore, Activity Data shall include a list of Activities	Data
	and, for each activity:	
	1. The M-LocationID the Activity Data refer to.	
	2. The duration (t_1-t_2) the Activity Data refer to.	
	3. The list of Action.	
	Personal Profile is an Item containing the Data about the human represented by a	Personal
	User. It may include: First Name, Last Name, Address, Country, Age, Biometric data,	Profile
	The Manager of an M-Instance sets <i>Rules</i> , an Item expressing the terms and	Rules
	conditions under which Processes operate in the M-Instance. The Rules may express:	
	1. The ability of a User to perform Actions on Items for which it has Rights.	
	2. The inability of a User to perform Actions on Items for which it has no Rights.	
	3. The duty of a User to perform Actions on Items.	
	4. The ability of a User to make Transactions on the Rights of Items.	

Human and User-related Items/3

Functional Requirements	ltem
Social Graph represents a User's network of connections with Items and Processes	Social
including the following:	Graph
1. The types and the connections with Items and their M-Locations.	
2. The types and the connections with Devices (frequency of use, etc.).	
3. The types and the connections with Services (frequency of use, etc.).	
4. The types and the connections with Users, groups of Users in terms of: Time, M-	
Locations, Declared purpose.	
User Data collects all the Data related to a human and their Users:	User Data
1. Rights held by the human's Users in the M-Instance.	
2. The Personal Profile of the human.	
3. The Personae that the human's Users impersonate.	
4. The Activity Data of the human's Users.	
5. The Social Graphs of the human's Users.	
User Data should allow for easy identification, extraction, and sharing of subsets of	
User Data.	

80 • community

Financial/1

Functional requirements	ltem
Asset is an Item that may be Transacted. An Asset may be:	Asset
1. MM-Embedded at an M-Location.	
2. Posted to a Service.	
An Asset shall:	
1. Preserve the Data Formats of the Item that has spawned it.	
2. Include the date it was created.	
Ledger includes the list of all Transactions:	Ledger
1. Related to Assets.	
2. Starting from the first Transaction and including the last.	
3. The Marketplace on which a Transaction was performed.	
Provenance includes the list of all Transactions:	Provenance
1. Related to an Asset.	
2. Starting from the first Transaction and including the last.	
3. The Marketplace on which a Transaction was performed. Provenance.	
	81 communit

Financial/2

Functional requirements	ltem
Transaction is Item representing the changed state of:	Transaction
1. The Rights on an Asset held by a seller User and a buyer User.	
2. The Accounts of the Users and of the Service facilitating/enabling the	
Transaction (Optional).	
The Transaction shall represent:	
1. The Time the Transaction is performed.	
2. The Value moving into the Wallet of User 1 (seller).	
The Value moved from the Wallet of User2 (buyer).	
4. The Value moved into the Wallet of User 3 (service) - optional.	
5. The Time the Values were moved.	
The Rights to Act owned by User1 after Time.	
7. The Rights to Act owned by User2 after Time.	
	IIII MPAI .
	82 • community

Financial/3

Functional requirements	ltem
Value is expressed by an Amount and the Currency related to the	Value
Amount. It shall have a representation that enables the expression of	
the Amount and the Currency used to represent the Amount.	
A Wallet is a container of Currency units. A Wallet shall enable the	Wallet
representation of:	
1. The Amounts contained in the Wallet for each Currency.	
2. The Transactions performed.	





To Discover Items, an Item called DiscoverIn is produced that contains:	DiscoverIn
1. A description of the Items to be Discovered.	
2. Information related to the rendering of DiscoverOut.	
Items candidate for Discovery may be described by:	
1. Verbal/text description.	
2. Similar Items.	
3. Belonging to specific M-Instances/M-Environments/M-Locations.	
4. Belonging to specific sections of Activity Data.	
Information on of RenderingDiscoverOut may be provided by:	
1. Media type used for rendering.	
2. Spatial Attitude of the Object rendering DiscoverOut.	
DiscoverOut contains the result of the Service Acting on the Request-	DiscoverOut
Discover Item and information about its rendering as requested in	
DiscoverIn.	

	Functional Requirements	ltem
	To obtain information on an Item, a User produces InformIn, an Item	InformIn
	containing:	
	1. A description of the Item about which information is requested.	
	2. Information related to the rendering of InformOut.	
	InformIn may refer to:	
	1. Item Metadata	
/	2. Any other information that a Service may have on the Item.	
	Information on rendering of InformOut may be provided by:	
	1. Media type used for rendering.	
	Spatial Attitude of the InformOut rendered Object.	
	InformOut is the Item containing the result of the Service Acting on the	InformOut
	Request-Inform Item and information about its rendering as requested in	
	InformIn.	

Functional Requirements	ltem
InterpretIn to obtain interpretation of an Item, e.g., translation of an	InterpretIn
utterance, meaning of a Thai dance movement, etc. It contains:	
1. The ID or the Item to be Interpreted.	
Information related to the rendering of InterpretOut.	
Information on InterpretOut Rendering may be provided by:	
1. Media type used for rendering.	
Spatial Attitude of InterpretOut rendered Object.	
InterpretOut is the Item containing the result of the Service Acting on	InterpretOut
the Request-Interpret Item and information about its rendering. It is rendered as requested in InterpretIn.	



Functional requirements	ltem
An Entity is an Item that can be perceived, i.e., Object, Model, Scene, Event, and	Entity
Experience.	
It is useful to introduce the Event Entity that includes selected Entities at an M-	Event
Location and their Animations during a period. Therefore, an Event shall include:	
1. M-LocationID.	
2. Start Time and End Time.	
3. List of Entities, their Animations, and Interactions.	
It is also useful to introduce the Experience Entity comprising selected Entities of	Experience
an Event and User Interactions with the Entities of the Event. Experience shall	
include:	
1. Start Time and End Time	
2. EventID	
3. List of selected Entities, their Animations, and User Interactions.	

88

community

Functional Requirements	ltem
Object is an Entity representing an object including:	Object
1. The type(s) of Media (Audio-Visual-Haptic) composing the Model.	
2. The Data representation.	
3. The Data Format used.	
Model is an Object representing a real object with its features ready to be MM-	Model
Animated or UM-Animated.	
Persona is a Model representing a human.	Persona



	Functional Requirements	Item
	Scene is a composition of Objects with the following features:	Scene
	1. May be hierarchical.	
	2. May be MM-Embedded at a specified M-Location.	
	3. Represent Objects:	
	1. With a Spatial Attitude.	
	2. Animated by streams or by an autonomous agent.	
	A <i>Stream</i> is made by a continuous flow of Data with the following features:	Stream
	1. May be scalable in space and time.	
/	2. May be used to:	
	1. Animate a Model.	
	Represent a Digitised Object in an M-Instance.	
	Interaction contains the Request-Action issued by a User on an Entity at an	Interaction
	M-Locations and the corresponding Time.	
		90 community

Map contains a structure establishing a correspondence between U- Map Locations with M-Locations e.g., for AR applications). Therefore, a Map shall include:

- 1. The M-Instance the Map refers to.
- 2. For each U-Location having one correspondence with an M-Location:
 - 1. The ID of the M-Location corresponding to the U-LocationID.
 - 2. Metadata related to the U-LocationID.
 - 3. Metadata related to the M-LocationID.



Item with a spatial impact

Functional requirements	Item
M-Location is an Identifiable delimited spatial portion of an M-Instance,	M-Location
e.g., the place occupied by a Persona. An M-Location:	
1. Shall define the space of the M-Instance belonging to the M-Location.	
2. May enable the creation of sub-spaces defining sub-M-Locations.	
 U-Location is an Identifiable delimited spatial portion of the Universe, e.g., the place occupied by a human. A U-Location shall: 1. Define the space in the Universe belonging to the U-Location. 2. Enable the definition of sub-spaces (sub-U-Locations) comprised in the U-Location. The enforcement of Rights to a U-Location is not intended to be part of the MPAL-MMM Architecture 	U-Location

92 **community**



Functional Requirements – Data Types



Overview of Data Types



- For Location and Time information
- For Transactions
- For internal state information
- For spatial information
- For internal status.



Data Types/1

- **1.** For general use
 - 1. Address: A URL.
- 2. For time information
 - 1. Time: A measure of time.
- **3.** For Transactions
 - 1. Currency: A medium of exchange enabling Transactions in an M-Instance.
 - 2. Amount: A number expressing a Value in a Currency.



Data Types/2

4. For spatial information

- *1. Coordinates*: A set of numbers representing a Position in a Metaverse Environment using a coordinate system.
- 2. Orientation: The set of the 3 roll, pitch, yaw angles indicating the rotation around the principal axis (x) of an Object, its y axis having an angle of 90° counterclockwise (right-to-left) with the x axis and its z axis (pointing up toward a viewer from above).
- 3. Point of View: The Spatial Attitude of a User watching the Environment.
- 4. Position: The coordinates of an Object with respect to a coordinate set in a Metaverse Environment.
- 5. Spatial Attitude: The Position and Orientation of an Entity, and their velocities and accelerations.



Data Types/3

5. For internal status

- 1. Cognitive State: A User's Personal Status that reflects the way it understands the Environment, such as "Confused", "Dubious", "Convinced".
- 2. Emotion: A User's Personal Status that results from its interaction with an Environment, such as "Angry", "Sad", "Determined".
- *3. Social Attitude*: The representation of a User's Personal Status related to the way it in-tends to position vis-à-vis an M-Environment, e.g., "Respectful", "Confrontational", "Soothing".
- 4. Personal Status: The information internal to a User characterising its behaviour.





Use Cases



13-Sep-23

Use Cases/1

MPAI has made a general analysis of 18 metaverse application areas

Automotive	Defence	Education
Enterprise	eSports	Events
Finance	Food	Gaming
Healthcare	Hospitality	Professional training
Real estate	Remote work	Retail
Social media	Travel	Virtual spaces



Use Cases/2

MPAI has made an in-depth analysis of 11 use cases

- 1. Attend a Metaverse Event.
- 2. Buy a personal wearable.
- 3. Buy the real twin of an Object.
- 4. Establish a Metaverse Environment.
- 5. Interact with a Metaverse Call Centre.
- 6. Navigate a 3D Object.

- 7. Relax in a Metaverse Environment.
- 8. Social gathering across Metaverse Environments.
- 9. Train Metaverse Hospital staff.10.Visit a Metaverse Environment.

11.Work in a Metaverse Environment.



Methodology to express Use Cases

MPAI has expressed each workflow with Processes, Actions, Items, and Data Types used by the Use Case

Elements of a Use Case

Description

Processes, Items, and Locations Detailed workflow

Workflow and Actions

Actions, Items, and Data Types



Verifying sufficiency if Technical Specification

- MPAI has developed a Use Case Description Language (UCDL).
 Uses UCDL by applying Actions and Items to 9 Use Cases.
- So far, **no need for more** Actions, Items, and Data Types has been identified.
- 1. Xirtual Lecture
- 2. Virtual Meeting
- 3. Hybrid working
- 4. eSports Tournament
- 5. Virtual performance

- 6. AR Tourist Guide
- 7. Virtual Dance
- 8. Virtual Car Showroom.
- 9. Drive a Connected Autonomous Vehicle.





Tracks Persona_{2.1} @ML_{2.1} MM-Embeds Persona_{2.1} @ML_{2.2}

MM-Animates Persona_{2.2} @ML_{2.2}

Virtual dance

Tracks Persona_{1.1} @ML_{1.1}

Transacts Value_{1.1}

MM-Embeds Persona_{1.1} @ML_{1.2}

MM-Disables Persona_{1.1} @ML_{1.1}

Sends Object(A)_{1.1} to Persona_{2.2}

(P_{2.2}) MM-Sends Object(A)_{2.1} to Persona_{1.1}
(P_{2.2}) MM-Sends Object(A)_{2.1} to Persona_{2.1}
(P_{2.1}) MM-Embeds Persona_{2.1} @ML_{2.3}
(P_{2.1}) MM-Disables Persona_{2.1} @ML_{2.2}

(P₂₁₂) MM-Embeds Object(A)_{2.2} @ML_{2.4}

(P_{2.1}) MM-Disables Persona_{2.1} @ML_{2.3}
(P_{2.1}) MM-Embeds Persona_{2.3} @ML_{2.3}
(P₂₁₂) MM-Animates Persona_{2.3} @ML_{2.3}

Tracks Persona@MLTransacts Value1.1MM-Embeds Persona0.1MM-Disables Persona0.1@ML0.1



13-Sep-23



Functional Profiles





Baseline Profile



- Enables a human equipped with a Device supporting the Baseline
 Profile to allow their Users to
 perform the functions of following Table.
- Supports basic forms of lecture, meeting, and hang-out.
- Has the following Levels:
 - Audio only
 - Audio-Visual
 - Audio-Visual-Haptic.



Features of the Baseline Profile/1

Functions	Action	Items
Read Items and Data	UM-Send	ltem
Identify Item	Identify	Item, Identifier
Author Item	Author	Item
Place Item at an MLoc without perception	MM-Add	Entity, MLoc
Make an MM-Added Item perceptible	MM-Enable	Entity
Place an Item at MLoc, with perception	MM-Embed	Entity, MLoc
Stop perception of Scene	MM-Disable	Entity, MLoc
Make available an Object to a User	MM-Send	Entity
Render at ULoc an Item at MLoc	MU-Render	ItemEntity, MLoc, ULoc



Features of the Baseline Profile/2

Functions	Action	Items
Make scene at ULoc available to a Device	UM-Capture	ULoc
Make Data of a Device available to a Process	UM-Send	Device, Process
Transfer Data between Processes	MM-Send	Message
Animate Model @MLoc w/ Data from ULoc	UM-Animate	Model, ULoc
Place Entity @ MLoc	UM-Render	Entity, MLoc, ULoc
Send an Entity at an MLoc to a Device	MM-Send	Entity, Device, MLoc
Place, Animate, and Render Model at MLoc	Track	Model, Stream
Store Item	MU-Send	Item


Actions, Items, and Data Types

	Author	Identify	MM-Add	MM-Embed
Actions	MM-Disable	MM-Enable	MM-Send	MU-Actuate
	MU-Render	MU-Send	Track	UM-Animate
	UM-Capture	UM-Render	UM-Send	
ltems	Identifier	Message	M-Instance	M-Location
	Model	Object	Scene	Stream
	U-Location			
Data Types	Address	Coordinates	Orientation	Position
	Spatial Attitude			





Community Comments



13-Sep-23

Anybody can comment

- Technical Specification: MPAI Metaverse Model – Architecture WD for Community Comments.
- Comments should reach secretariat@mpai.community by 2023/09/21 T 23:58 UTC.
- No specific format required to make comments.
- MPAI intends to publish the standard on 29 September 2023.



What's next for MPAI-MMM



- So far, OpenAPI-based Client and Server developed.
- Soon, publish MPAI-MMM Architecture API.
- Currently developing Table of Content of a Technologies for Metaverse Technical Specification.





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

Join MPAI Share the fun Build the future



13-Sep-23