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

MPAI Technical Specification

MPAI Metaverse Model (MPAI-MMM) Architecture

V1.1

WARNING

Use of the technologies described in this Technical Report 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 Report.

Readers are invited to review Annex 2 - Notices and Disclaimers.

Technical Specification
MPAI Metaverse Model (MPAI-MMM)
– Architecture V1.1

Contents

1 Introduction (Informative)

Metaverse is a word conveying different meanings to different persons and to date some 150 definitions have been formulated. In this document the word metaverse is characterised as "a system that captures data from the real world (in the following, called Universe[1]), processes it, and combines it with internally generated data to create virtual environments that users can interact with". So far, the developers of many systems responding to this characterisation have made technology decisions that best responded to their needs, often without considering the choices that other developers might have made for similar purposes.

Recently, however, there have been mounting concerns that such metaverse "walled gardens" do not fully exploit the opportunities offered by current and expected technologies. Calls have been made to make metaverse instances (in the following, M-Instances) "Interoperable".

MPAI – Moving Picture, Audio, and Data Coding by Artificial Intelligence – the international, unaffiliated, non-profit organisation developing standards for AI-based data coding – has provided initial contributions to M-Instance Interoperability with two Technical Reports:

1. **Technical Report: MPAI Metaverse Model (MPAI-MMM) – Functionalities** [1] has introduced the following elements:

1.1. A set of definitions.
1.2. A set of assumptions, the most important of which are that metaverse:
1.2.1. Is a broad notion that may be used by different industries for different purposes.
1.2.2. Needs technologies, some of which do not even exist with satisfactory performance.
1.2.3. The notion of profile may help improve Interoperability.
1.3. A collection of high-level use cases.
1.4. A collection of exemplary service providers.
1.5. An organised set of ~150 Functionalities.
1.6. A review of the main metaverse-enabling technologies.
1.7. An analysis of metaverse governance needs.
1.8. The MPAI metaverse standardisation roadmap.

2. **Technical Report: MPAI Metaverse Model (MPAI-MMM) – Functionality Profiles** [11] has introduced the following elements:

2.1. A revised and extended list of definitions.
2.2. An operation model of the metaverse based on the notion of Processes performing or requesting other Processes to perform Actions on Items (Items are Data supported by an M-Instance).
2.3. A specification of a first set of Actions, Items, and Data Types enabling:
2.3.1. Detailed descriptions of Use Cases using Actions, Items, and Data Types.
2.3.2. The definition of Functionality Profiles, i.e., subsets of Actions, Items, and Data Types that satisfy a selected subset of identified Functionalities.
2.4. A collection of representative use cases tested against the Operation Model.
2.5. Four initial Functionality Profiles.

[1] Words beginning with a capital letter are defined in Chapter 3.

Technical Specification – MPAI Metaverse Model (MPAI-MMM) – Architecture provides means to achieve M-Instance Interoperability by specifying the Functional Requirements of Processes, Actions, Items, and Data Types that allow Interoperation of two or more M-Instances implementing the Operation Model, executing Processes, and producing Data with Formats that comply with the Functional Requirements of this Technical Specification, possibly via a Conversion Service.

Chapters, Sections, and Annexes are Normative unless they are explicitly identified as Informative.

2 Scope

Technical Specification: MPAI Metaverse Model (MPAI-MMM) – Architecture specifies:

1. Terms and Definitions
2. Operation Model
3. Functional Requirements of Processes, Actions, Items, and Data Types
4. Functional Profiles

enabling Interoperability of two or more metaverse instances (M-Instances) if they:

1. Rely on the Operation Model, and
2. Use the same Profile Architecture, and

2.1. Either the same technologies, or

2.2. Independent technologies while accessing Conversion Services that losslessly transform Data of an M-Instance$_A$ to Data of an M-Instance$_B$.

Note: Full Interoperability may not be achieved if the M-Instances implement different Profiles.

The contents of this Technical Specification: MPAI Metaverse Model (MPAI-MMM) – Architecture is:

1.	Scope	Normative
2.	Terms and Definitions	Normative
3.	Metaverse Functionalities	Informative
4.	Metaverse Operation Model	Normative
5.	Functional Requirements of:	
5.1.	Processes, i.e., Programs executing in an M-Instance	Normative
5.2.	Actions, i.e., Functionalities provided by Processes	Normative
5.3.	Items, i.e., Data and Metadata supported by an M-Instance	Normative
5.4.	Data Types, i.e., Data used in Actions and Items	Normative
6.	Use Cases	Informative
7.	Functional Profiles	Normative

This Technical Specification has been developed by the MMM group of the Requirements Standing Committee. MPAI may decide to publish extensions or new versions of this Technical Specification, or other Technical Specifications of the MPAI-MMM series.

3 Terms and Definitions

Terms beginning with a capital letter have the meaning defined in Table 1. Terms beginning with a small letter have the meaning commonly defined for the context in which they are used. For instance, Table 1 defines *Object*, *Scene,* and *User* but does not define *object*, *scene,* and *human*.

A dash "-" preceding a Term in Table 1 means the following:

1. If the font is normal, the Term in the table without a dash and preceding the one with a dash should be placed before that Term. The notation is used to concentrate in one place all the Terms that are composed of, e.g., the word Decentralised followed by one of the words Application, Autonomous Organisation, Finance, System, and User Identifier, or definitions belonging to the same class, e.g., Action and Items.
2. If the font is *italic,* the Term in the table without a dash and preceding the one with a dash should be placed after that Term. The notation is used to concentrate in one place all the Terms that are composed of, e.g., the word Interface preceded by one of the words Brain-Computer, Haptic, Speech, and Visual.

Table 1 – General Terms and Definitions

Terms	Definitions
Account	An Item that uniquely references a Registered human.
Action	A Functionality provided by a Process.
- Authenticate	The Action of requesting that a Service confirm that an Entity is what it claims to be.
- Author	The Action of Calling a Service to obtain an Entity with associated OutRights to Act on the Entity.
- Change	The Action of requesting that a Service modify the Rights of a User and provide OutRights, e.g., to further Change the Rights.
- Convert	The Action of requesting that a Service change the Format of the Data of an Item into a Format specified by a FormatID.
- Discover	The Action of requesting that a Service provide a DiscoveryOut Item containing: 1. The IDs of the Items relevant to the User's request to Discover expressed in the DiscoverIn Item 2. The OutRights to Act on the DiscoverOut Item.
- Execute	The Action of requesting that a Process execute a Contract.
- Hide	The Action of requesting that a Service make the ID of an Item unavailable and provide OutRights, e.g., to make the ID available again.
- Identify	The Action of requesting that a Service produce an Item from Data & Metadata.
- UM-Import	The Action of a User requesting that a Service read Data & Metadata stored at an Address.
- Inform	The Action of requesting that a Service provide an InformOut Item containing information about an InItem, such as the Metadata of the InItem, with the OutRights to Act on the InformOut Item.
- Interpret	The Action of requesting that a Service provide an InterpretOut Item containing interpretation of an InItem, such as translation or extraction of Personal Status, with the OutRights to Act on the InterpretOut Item.
- MM-Add	The Action of requesting that a Service add an Entity at an M-Location with a Spatial Attitude and provide OutRights to Act on the MM-Added Entity.
- MM-Animate	The Action of requesting that a Service change the features of a Model MM-Embedded at an M-Location with a Stream and provide the OutRights to Act on the MM-Added Entity.
- MM-Disable	The Action of requesting that a Service stop MM-Enabling selected Entities Embedded at an M-Location and provide OutRights to Act on the MM-Disabled Entities.

- MM-Embed	The Composite Action of requesting that a Service MM-Add and MM-Enable an Entity either located at a Service or at an M-Location at a destination M-Location with a Spatial Attitude and provide OutRights to Act on the MM-Embedded Entity.
- MM-Enable	The Action of requesting that a Service accept requests to MM-Send selected Entities MM-Added at an M-Location or to MM-Embed those selected Entities at a destination M-Location and provide OutRights to act on the M-Entities.
- MM-Send	The Action of requesting that a Process forward an Item or Data/Metadata to a Process with appropriate granting Rights.
- Modify	The Action of requesting that a Service produce a new Item from an existing Item by providing new Data and Metadata with the OutRights to further Act on the new Item.
- MU-Actuate	The Action of requesting that a Device present an Entity available at a Device to a U-Location as Media with a Spatial Attitude.
- MU-Render	The Composite Action of requesting that: 1. A Service MM-Send selected Entities Embedded at an M-Location to a Device. 2. The Device MU-Actuate at a U-Location with a Spatial Attitude the Entity received.
- MU-Send	The Action of requesting that a Process transmit an Item to a Device or store an Item at an Address.
- Post	The Action of requesting that a Marketplace include an Asset to its repertory of Assets.
- Register	The Action of a human requesting that an M-Instance grant their Users the Rights to perform Actions in the M-Instance.
- Resolve	The Action of requesting that a Service forward a Request-Action or a Response-Action to a Resolution Service in another M-Instance.
- Track	The Composite Action of requesting that a Service: 1. MM-Embed a Model at an M-Location with a Spatial Attitude. 2. MU-Animate the Model MM-Embedded at an M-Location. 3. MU-Render specified Entities at the M-Location to a U-Location.
- Transact	The Action of a $User_1$ requesting that a Service: 1. Assign Rights on an Asset to $User_2$ (“buyer”). 2. Cause: 2.1. $Wallet_1$ of $User_1$ (“seller”) to be increased by $Value_1$. 2.2. $Wallet_2$ of $User_2$ to be decreased by $Value_2$. $Wallet_3$ of the Service enabling/facilitating the Transaction to be increased by $Value_3$ (optionally).
- UM-Animate	The Composite Action of a User requesting: 1. A Device to 1.1. UM-Capture an animation stream extracted from a scene at a U-Location. 1.2. UM-Send the animation stream and Metadata to a User. 2. A Service to Identify the Animation Stream. 3. A Service to MM-Animate the Model MM-Embedded at the M-Location using the Animation Stream.
- UM-Capture	The Action of requesting that a Device capture Media from a scene at a U-Location.
- UM-Render	The Composite Action of a User requesting:

	1. A Device to: 1.1. UM-Capture a scene at U-Location. 1.2. MM-Send Data and Device-provided Metadata to a User. 2. A Service to: 2.1. Identify an Entity from UM-Sent Data and Metadata. 2.2. MM-Embed the Entity at an M-Location with a Spatial Attitude.
- UM-Send	The Action of a Device acquiring Data & Metadata from an Address.
- Validate	The Action of requesting that a Service verify that a Process has the Rights to perform or request a Process to perform an Action on an Item.
Avatar	A rendered User (synonym of Persona).
Blockchain	A shared immutable ledger stored on a peer-to-peer network of computers.
Certification	The attestation that a Process or Item has specified characteristics.
Connected Autonomous Vehicle	(CAV) A vehicle able to autonomously reach a U-Location by using its own sensing and processing capabilities to generate an M-Instance, sharing its M-Instance with other CAVs and issuing actuation commands.
Conversion	The process of changing the Format of Data.
Data	Information represented in digital form.
- Format	The syntax and semantics of Data.
DataMdata	The combination of Data and Metadata that is not (yet) an Item.
Data Type	Data used in Actions and Items.
- Address	The URL of a storage facility.
- Amount	A number expressing a Value in a Currency.
- Cognitive State	The representation of a User's Personal Status that reflects the way they understand the environment, such as "Confused", "Dubious", "Convinced".
- Coordinates	A set of numbers representing a Position in a Metaverse environment using a coordinate system.
- Currency	A medium of exchange enabling Transactions in a Metaverse environment.
- Emotion	The representation of a User's Personal Status that results from their interaction with an environment, such as "Angry", "Sad", "Determined".
- 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 viewing from above).
- Personal Status	The representation of the information internal to a User characterising their behaviour.
- Point	A point in an M-Environment identified by the set of local Coordinates.
- Point of View	The Spatial Attitude of a Persona watching an Environment.
- Position	The Coordinates of a point in a Metaverse Environment using a Coordinate system.
- Social Attitude	The representation of a User's Personal Status related to the way it intends to position vis-à-vis an environment, e.g., "Respectful", "Confrontational", "Soothing".
- Spatial Attitude	The Position and Orientation of an Entity, and their velocities and accelerations.
- Time	The representation of the measure of time.
Decentralised	
- Application	(dApp) A Process that runs on a decentralised computing system.

- Autonomous Organisation	(DAO) An organisation without centralised leadership, where the main governing rules are typically encoded by means of a Smart Contract.
- Finance	(DeFi) A financial technology based on a secure infrastructure of distributed ledgers like those used by crypto currencies.
- System	A set of dApps enabling a group of Users to make decisions without a centralised entity.
Device	Equipment enabling: - A U-Environment to interact with an M-Instance and/or - An M-Instance to interact with a U-Environment.
Duty	A moral or legal obligation to act or behave.
Entitlement	The state of a User having certain Rights in an M-Instance.
Functional Requirement	A Functionality that is expected to be provided by an entity.
Human	
- *Digital*	Either a Digitised or a Virtual Human.
- *Digitised*	The digital representation of a human.
- *Virtual*	A computer-created Object that has a human appearance when rendered but is not a Digitised Human.
Governance	The action or manner of directing and controlling actors of the Metaverse Ecosystem.
Information and Communication Technologies	(ICT) Technologies that enable the processing and distribution of information via the network.
Interface	A communication pathway enabling systems to interact.
- *Brain-Computer*	(BCI) A communication pathway that allows a human to interact with an M-Instance by sensing and processing the electrical activity of the brain.
- *Haptic*	A communication pathway that allows a human to interact with an M-Instance through bodily movements and sensations.
- *Speech*	A communication pathway that allows a human to interact with an M-Instance using spoken language.
- *Visual*	A communication pathway that allows a human to interact with an M-Instance through bodily movements and visual messages.
Interoperability	The ability of an M-Instance to exchange and make use of the data of another M-Instance.
Item	Data and Metadata supported and identified by an M-Instance.
- Account	An Item that uniquely references a human who has Registered. A User may have more than one Account with one or more Services.
- Activity Data	An Item containing the record of all the Actions made by a User.
- Asset	An Item that may be the object of a Transaction. It may be MM-Embedded at an M-Location or Posted to a Service.
- AuthenticateIn	An Item containing: 1. The Entity or the ID of the Entity to be Authenticated. 2. Information related to the rendering of AuthenticateOut.
- AuthenticateOut	An Item containing the result of processing the Request-Authenticate Action and its rendering.
- Contract	A Program and its Metadata stored on a Device. It is activated by an external entity, e.g., a User, or another activated Contract.
- DiscoverIn	An Item containing: 1. A description of the Items to be Discovered.

	2. Information related to the rendering of DiscoverOut.
- DiscoverOut	An Item containing the description of the Items Discovered and information related to its rendering.
- Entity	Any of the following Items that can be MU-Rendered: Object, Model, Scene, Event, and Experience.
- Event	An Entity that includes selected Entities at an M-Location and their Animations during a period.
- Experience	An Entity comprising User-selected Entities of an Event and the User Interactions with the Entities of the Event.
- Identifier	An Item that uniquely references an Item in an M-Instance.
- InformIn	An Item containing: 1. A description of the Item about which information is requested. 2. Information related to the rendering of InformOut.
- InformOut	An Item containing the description of the Item object of an InformIn.
- Interaction	An Item containing the Request-Action issued by a User on an Entity at an M-Locations and the corresponding Times.
- InterpretIn	An Item containing: 1. The ID or the Item to be Interpreted. 2. Information related to the rendering of InterpretOut.
- InterpretOut	An Item containing the description of the Item object of an InterpretIn Item.
- Ledger	An Item containing a list of Transactions involving Assets.
- Map	An Item containing a structure establishing a correspondence between U-Locations with M-Locations.
- M-Environment	An Identified subset of an M-Instance.
- Message	An Item containing application-specific Data MM-Sent by a Source Process to a Destination Process.
- M-Instance	An implementation of the Common Metaverse Specifications.
- M-Location	An Identified delimited space of an M-Environment.
- Model	An Object representing an object with its features ready to be MM-Animated or UM-Animated.
- Object	An Item with at least one of Audio, Visual, or Haptic perceptibility attributes.
- Obligation	An Item expressing the promise of a Process to perform Actions on Items at M-Locations during a Time.
- Persona	A Model representing a human.
- Personal Profile	An Item containing the Data about the human represented by a User.
- Program	Data that can be executed
- Provenance	The Ledger associated with a specific Asset.
- Request-Action	An Item of the request to a Process to perform an Action as defined in this document.
- Response-Action	An Item containing the response of a Process to a Request-Action as defined in this document.
- Rights	An Item expressing the Process's authorisation to perform Actions on Items at M-Locations during a Time.
- Rules	An Item expressing the terms and conditions under which a User operates in an M-Instance or M-Environment.
- Scene	A possibly hierarchical Composition of Objects having Spatial Attitudes.

- Social Graph	A representation of a User's network of connections with Items, M-Locations, and Processes.
- Stream	An Item made by a continuous flow of Data.
- Transaction	Item representing the changed state of the Accounts and the Rights of a seller User and a buyer User on an Asset and optionally of the Service facilitating/enabling the Transaction
- U-Location	An identifiable delimited portion of the Universe.
- Personal Data	An Item containing Activity Data, Personae, and Social Graph of a User.
- Value	An Amount and the Currency with which the Amount is expressed.
- Wallet	A container of Currency units. In general, a Wallet is implemented outside of the M-Instance.
Level	A subdivision of a Profile indicating the completeness of the user experience provided by the Profile.
Media	1. Data acquired by a Device using a Sensor. 2. Data converted by a Device from an Item to a format that can be presented.
Metadata	An attribute of Data, e.g., of a human, a Process, an M-Location, a U-Location, or an Item.
Metaverse Instance	(M-Instance) A set of Processes providing some or all the following 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 to affect U- and/or M-Environments using Objects in ways that are: 4.1. Consistent with the goals set for the M-Instance. 4.2. Effected within the capabilities of the M-Instance. 4.3. Complying with the Rules set for the M-Instance and applicable laws.
- Actuator	A component of a Device able to MU-Render an Entity and environment Data to a U-Environment.
- Asset	An Item Embedded at an M-Location or Posted to a Service that may be the object of a Transaction.
- Ecosystem	The ensemble of entities and rules ensuring that Metaverse Instances operate in the interest of Metaverse Stakeholders.
- Enabling Service Layer	The set of Services such as payment, security, identity, privacy, etc. that enable operation of an M-Instance.
- Entity	Any of the following Items that can be MU-Rendered: Scene, Object, Model, Event, and Experience.
- Environment	(M-Environment) A portion of an M-Instance run by a Registration system.
- Experience Layer	The set of functions, such as Devices, that generate Experiences.
- Functionality	The attribute of a Process of being endowed with the capability of performing particular Action(s).
- Industry	The collection of players that support the design, development, deployment, operation, and content and service provisioning to Metaverse Instances.

- Instance	(M-Instance) An implementation providing all or a subset of the Metaverse Functionalities.
- Interoperability	The ability of M-Instance #1 to use data from and as intended by M-Instance #2. Interoperability can be Direct or Mediated by a Conversion Service.
- Infrastructure Layer	The set of functions such as network, transport, storage, and (cloud, edge) processing that enable an M-Instance to operate.
- Item	Metaverse-specific Data that may include Metadata that may include Rights.
- Level	A subdivision of a Profile that indicates the degree of completeness of the user experience provided by that Level.
- Location	(M-Location) An identifiable delimited portion of the Metaverse.
- Manager	The entity overseeing the operation of an M-Instance.
- Operation Model	The components and sequence of steps involved in an M-Instance provids Functionalities.
- Operator	The entity overseeing the operation of an M-Environment.
- Partner	A User participating in activities of a Metaverse Operator (i.e., a business customer of an Operator)
- Platform Layer	The set of Services, such as content creation, content discovery, and content access functions that enable an M-Instance to operate.
- Process	The instance of a program being executed.
- Profile	A recognised subset of the Functionalities (Functionality Profile) or Technologies (Technology Profile) specified by the Common Metaverse Specifications.
- Sensor	A Device able to UM-Capture a scene and other environment information as Data.
- Specification	The collection of standards specifying the Technologies and Technology Profiles enabling Metaverse Interoperability.
- Stakeholder	An entity performing a function aimed at achieving a goal in an M-Instance.
- State	The set of values and stored data of an M-Instance at a given time.
- Tool	A Technology or group of Technologies enabling an M-Instance to provide a Functionality.
- Technology	A structured application of scientific and/or technical methods that supports a Functionality.
- User	One or a set of Processes representing a human.
Object	
- *Audio*	The digital representation of an object or a computer-generated Object that can be rendered to and perceived by a human ear.
- *Autonomous*	A Virtual Object animated by a Process giving it the ability to act (e.g., move, speak, respond, execute) with a degree of autonomy.
- *Composite*	An Object that includes more than one Object Type.
- *Digital*	A Digitised or a Virtual Object.
- *Digitised*	The digital representation of an object.
- *Haptic*	An Object with the haptic features of an object able to be rendered to provide haptic sensations in a human.
- *Human*	An Object representing a human.
- *Speech*	The digital representation of a sound emitted by the vocal tract of a human or generated by a computer with similar audio characteristics.
- Type	One of Audio, Visual, Haptic, Olfaction, and Gustation.

- *Virtual*	A computer-generated Object that is not a Digitised Object.
- *Visual*	The digital representation of an object captured by an electromagnetic or high-frequency audio signal or computer-generated that can be rendered to and perceived by a human eye.
Oracle	A Service providing information from a U-Environment to a Blockchain.
Privacy	The Rights of a User to keep their Personal Profile secret.
Process	
- App	An application-specific Program executed on a Device.
- Capability	The characteristics of a Process.
- Device	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.
- Service	A Process that can be called to provide specific Functionalities.
- User	A Process representing a human that is UM-Animated by a Stream or MM-Animated by an autonomous agent.
- App	An application-specific Program executed on a Device.
- User	A Process representing a UM-Captured human as a Persona that are either UM-Animated by a Stream or MM-Animated by an autonomous agent.
Profile	A set of base standards and/or their subsets.
- *Functional*	The set of Functionalities offered by a Metaverse Profile.
- *Technology*	The set of Technologies offered by a Metaverse Profile.
Registration	The provisioning by a human of a subset of Personal Data to an M-Instance/Environment to obtain an Account.
Render	The process of making an Item perceptible by human senses.
Representation	Data that represent an entity of a U-Environment in an M-Instance.
Sense of	
- Agency	The subjective awareness of being able to decide, execute, and control one's own actions in an M-Environment.
- Embodiment	The engagement of senses to form a complete M-Instance Experience.
- Presence	The feeling of being in an M-Instance with other Digital Humans for real.
Service	
- *Conversion*	A Service converting the Data produced by an M-Instance_A into Data understood and acted upon by M-Instance_B as intended by M-Instance_A.
Smart Contract	A Program stored on a Blockchain that runs when activated by an external entity, e.g., a User or another Smart Contract.
Token	
- *Fungible*	A representation of an Asset that is interchangeable with other Assets of the same type.
- *Non-Fungible*	(NFT) A unique digital identifier of an Asset that: - Cannot be copied (i.e., a copy is known to be a copy), substituted, or subdivided. - Is recorded in a digital ledger. - Is used to certify Object authenticity and ownership.
Trust-less system	A system allowing a User to make reliable Transactions without trusting or knowing the parties the User makes Transactions with.
Universe	The physical world.

- Location	(U-Location) An identifiable delimited portion of the Universe.
Use Case	An example of how an application domain can be supported by an M-Instance/Environment.
User Keys	The pair of public and private keys where the public key is used to encrypt, and the private key is used to both encrypt and decrypt Data.
User Identifier	
- *Decentralised*	An Identifier that enables the verifiable association with a User without requiring a centralised registry.
- *Self-Sovereign*	A Decentralised Identifier derived from the User's Public Key owned and managed directly by the User based on the knowledge of their own Private Key, e.g., stored in the Crypto Wallet enabled by the Blockchain underpinning the Metaverse Instance.
Wallet	
- *Crypto*	Software or hardware holding the Public and Private Keys of a User to enable them to make Transactions by accessing their Account on a Blockchain.

4 References

4.1 Normative reference

1. MPAI; The MPAI Statutes; https://mpai.community/statutes/.
2. MPAI; The MPAI Patent Policy; https://mpai.community/about/the-mpai-patent-policy/.
3. Framework Licence: MPAI Metaverse Model (MPAI-MMM) – Architecture V1; https://mpai.community/standards/mpai-mmm/framework-licence/.

4.2 Informative references

4. MPAI; Technical Specification – Governance of the MPAI Ecosystem (MPAI-GME) V1.1; https://mpai.community/standards/mpai-gme/
5. MPAI; Technical Specification – AI Framework (MPAI-AIF) V2.0; https://mpai.community/standards/mpai-aif/
6. MPAI; Technical Specification – Portable Avatar Format (MPAI-PAF) V1.1; January 2023; https://mpai.community/standards/mpai-paf/
7. MPAI; Technical Specification – Context-based Audio Enhancement (MPAI-CAE) V2.1; January 2023; https://mpai.community/standards/mpai-cae/
8. MPAI; Technical Specification – Connected Autonomous Vehicle (MPAI-CAV) – Architecture V1.0; September 2023; https://mpai.community/standards/mpai-cav/
9. MPAI; MPAI; Framework Licence: MPAI Metaverse Model (MPAI-MMM) – Architecture; https://mpai.community/standards/mpai-mmm/framework-licence-mpai-metaverse-model-mpai-mmm-architecture/
10. MPAI; Technical Report – MPAI Metaverse Model (MPAI-MMM) – Functionalities; January 2023; https://mpai.community/standards/mpai-mmm/mpai-metaverse-model-mmm-functionality-profiles-mpai-metaverse-model/mmm-functionalities/
11. MPAI; Technical Report – MPAI Metaverse Model (MPAI-MMM) – Functionality Profiles; May 2023; https://mpai.community/standards/mpai-mmm/mpai-metaverse-model-mmm-functionality-profiles/
12. Khronos; Graphics Language Transmission Format (glTF); October 2021; https://registry.khronos.org/glTF/specs/2.0/glTF-2.0.html

5 Metaverse Functionalities

This Chapter collects the Functionalities of an M-Instance supported by this Technical Specification. Terms beginning with a small letter have the common meaning, Terms beginning with a capital letter are defined in Chapter 3.

5.1 M-Instance

An M-Instance

1. Is characterised as a set of Processes providing some or all the following functions:
2. Sensing data from U-Locations using Devices.
3. Processing the sensed data and producing Items.
4. Producing one or more M-Environments populated by Items either imported or internally generated.
5. Processing Objects from the M-Instance or potentially from other M-Instances to affect U- and/or M-Environments in ways that are:

5.1. Consistent with the goals set for the M-Instance.
5.2. Effected within the capabilities of the M-Instance.
5.3. In compliance with the Rules set for the M-Instance and the applicable laws.

6. May operate in a centralised or decentralised way.
7. May be subdivided in administratively separated M-Environments.
8. Monitors:

8.1. The Rights held by Processes on Processes to Act on Items at M-Locations.
8.2. The Entities placed at M-Locations with a Spatial Attitude.

9. Should be secure, e.g., to prevent malicious Users from:

9.1. Accessing the Account of a human who is not the human represented by the User.
9.2. Taking control of a Process.
9.3. Acting on Items for which it does not have Rights.

10. Should withstand denial of service attacks.

5.2 Processes

1. Process types

1.1. Devices: they connect U-Locations with M-Locations and vice-versa.
1.2. Users: they represent humans and are humans' agents in the M-Instance or on Devices.
1.3. Apps: they run on Devices. A User may be an App running on a Device.
1.4. Services: they are generic Processes providing Functionalities.

2. Certification

2.1. Processes may be connected or imported to an M-Instance if they have passed Certification.
2.2. Contracts may be imported to an M-Instance if they have passed a Certification Process.

5.3 Rules

An M-Instance may be governed by Rules that may include:

1. The Personal 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 required for each Account type.
6. The sanctions applied to a User whose Actions do not comply with the Rules.

5.4 Registration

1. A human, possibly on behalf of a legal entity, may Register with an M-Instance.

2. The M-Instance may
2.1. Request:
2.1.1. A subset of the human's Personal Profile.
2.1.2. WalletIDs
2.1.3. Users
2.1.4. Personae
2.2. Create an Account with AccountID including:
2.2.1. The subset of the human's Personal Profile.
2.2.2. WalletIDs
2.2.3. Users
2.2.4. Personae
2.2.5. Rights and Obligations
3. Personal Profile may have a scalable representation to enable an easy creation of a subset.

5.5 Rights

5.5.1 General Rights

Rights may include:
1. The maximum number of:
1.1. Users a human can deploy.
1.2. Personae a User can be rendered as.
1.3. Concurrent Users and Objects an M-Location can support.
2. The ability to:
2.1. Create persistent (as opposed to ephemeral) Scenes at M-Locations.
2.2. Store, search, and retrieve Experiences.
2.3. Develop economic activities.
2.4. Book M-Locations for a duration.
2.5. Support persistent storage.
2.6. Connect with or import Processes to an M-Instance.
2.7. Make Transactions to acquire Rights.

5.5.2 Perception Rights

A User may, depending on the Rights it holds:
1. Perceive scenes at U-Locations as captured by Devices endowed with audio, visual, haptic, and BCI capabilities, and the spatial attitudes of their objects.
2. Receive Messages from a U-Location as a result of an event there.
3. Import audio, speech, visual, and haptic models.
4. Perceive Entities.
5. Render perceived Entities at a U-Location with spatial attitudes.
6. Import Data and Metadata from and export Items to an Address.

5.5.3 Interaction Rights

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 Speech, Visual, and Haptic Messages to Users.
4. Send selected Entities to a User.
5. Place Entities with Spatial Attitudes.
6. Animates Models.

5.5.4 Service access Rights

A User may, depending on the Rights it holds, perform or request a Process to perform:

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

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. Posts an Item that can be Transacted (Asset) to a marketplace.
13. Retrieve Objects, Events, or Experiences.
14. Store Objects, Events, or Experiences.
15. Formulate:

15.1. The information it intends to obtain from a Service.
15.2. The Rights it intends to acquire on the response of the Service.
15.3. How the response should be rendered.

5.5.5 Acquisition of Rights

A User may acquire, or have as part of its Account 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 an Asset out of an Item.
4. Make a Transaction on an Asset specifying which Rights to the Asset it grants to the buyer.
5. Make an agreement with another user, e.g., via a Contract.
6. Copy or Modify, i.e., change Data (e.g., by editing it) and/or Metadata of an Item.
7. Change the settings and attributes of Entities of an M-Locations (e.g., SFX during a concert).
8. Set the Point of View from which it perceives a Scene.
9. Make inaccessible to any User an Item for which it has Rights.
10. Make or request a Process to make Actions in its own or another M-Instance.
11. Be rendered as one of the Personae uploaded at registration time.

5.6 Personae

A Persona may:

1. Faithfully reproduce the visual appearance of the human.
2. Have their visual appearance altered, 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

5.1. The movements of the human.
5.2. A Process.

5.7 Locations

1. An M-Location may be:
1.1. Private, i.e., only a User or a group of Users have Rights to it.
1.2. Public, i.e., a Metaverse Operator or a User grant anybody the Rights to enter, move in, and perceive the M-Location.
1.3. Persistent
1.4. Ephemeral.
2. A U-Location
2.1. May require that a User hold Rights to Capture a scene, Render a Scene, or store Data at an Address.

5.8 Objects and Scenes

1. An Object may:
1.1. Be Virtual, i.e., Authored in an M-Instance.
1.2. Be Digitised, i.e., Digital Twins of objects.
1.3. Be Autonomous, i.e., Animated by a Process.
1.4. Have Audio, Visual, or Haptic attributes, or combinations of these attributes.
2. A Scene may be composed of:
2.1. Only Virtual Objects.
2.2. Only Digitised Objects.
2.3. Partly Virtual and partly Digitised Object.

5.9 Identifiers

1. All Items have an Identifier.
2. An Identifier:
2.1. Uniquely references an Item or a Process.
2.2. May be issued by a central, distributed, or no authority.
2.3. May have a Format accepted by more than one M-Instance.
2.4. May enable to trace the Identifier of an Item back to the Item that spawned the Asset.
2.5. May enable to trace the Identifier of a Modified Item back to the Item that spawned the Modified Item.

6 Metaverse Operation Model

A M-Instance operates according to the following Metaverse Operation Model.

1. An M-Instance is an Information and Communication Technology platform operating according to this Technical Specification and providing the following general functions:
    1.1. To sense data from U-Locations.
    1.2. To process the sensed data and produce Data.
    1.3. To produce one or more M-Environments populated by Objects that can be either digitised or virtual, the latter with or without autonomy.
    1.4. To process Objects from the M-Instance or potentially from other M-Instances.
    1.5. To affect U- and/or M-Environments using Objects in ways that are:
        1.5.1. Consistent with the goals set for the M-Instance.
        1.5.2. Within the Capabilities of the M-Instance.
        1.5.3. According to the Rules of the M-Instance.

1.5.4. Respecting applicable laws and regulations.

2. An M-Instance
   2.1. May be subdivided in administratively separated M-Environments.
   2.2. May make its Capabilities known.
   2.3. May allow a human to Register with the M-Instance or one of its M-Environments.
   2.4. May request a Registering human to provide a subset of their Personal Profile.
   2.5. May give a Registered human the Rights to deploy one or more:
      2.5.1. Users and Devices each with M-Instance-specific Rights.
      2.5.2. Personae, i.e., human-like, or humanoid Models used by deployed Users to render themselves.
   2.6. May establish Rules that the Users in the M-Instance shall comply with.
   2.7. May penalise Users for lack of compliance with the Rules.
3. An M-Instance
   3.1. Is implemented as a set of Processes providing Functionalities defined by this Technical Specification.
   3.2. Creates Items by processing sensed data and autonomously produced data.
   3.3. Employs Data Types to represent entities such as Amount (of a Currency), length, area, and volume etc.
4. An M-Instance may support four types of Processes:
   4.1. User represents and is under the control of a human who has Registered with the M-Instance. A User may run on the M-Instance or on a Device connected to the M-Instance.
   4.2. Device is one or more Processes running on a physical device connecting M-Locations to U-Locations, specifically the Registered human, by:
      4.2.1. 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.
      4.2.2. Receiving an Entity and rending it as Media with a Spatial Attitude (Position and Orientation) – M-Location to a U-Location.
   4.3. Service provides specific functionalities.
   4.4. App runs on a Device. A Device may run the User Process as an App.

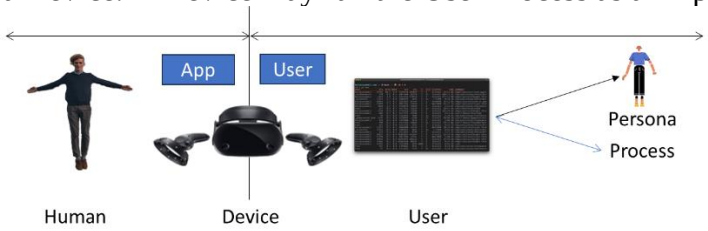


Figure 1 - The human-device-user-persona chain

5. A Process in an M-Instance:
   5.1. Is composed of:
      5.1.1. A Program having a Format supported by the M-Instance.
      5.1.2. Metadata describing the Program and containing the following as a minimum:

Table 2 – Minimum Metadata of a Process

Metadata elements	Details
ProcessID	The ID of 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.

5.2. Provides Functionalities by performing Actions on Items that contain:
  5.2.1. Data whose Format is supported by the M-Instance.
  5.2.2. Metadata describing the Item and containing the following as a minimum:

Table 3 - Minimum Metadata for Item

Metadata elements	Details
ItemID	The ID of the Item.
DescrMdata	Any description of the Item.

5.3. May request other Processes to perform Actions by transmitting a standard Request-Action Item and responding to a Request-Action with a Response-Action:

Table 4 – Elements of Request-Action and Response-Action

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

5.4. Performs the Request if:
  5.4.1. The requesting Process holds the Rights that are required to perform the request.
  5.4.2. The requested Process holds Rights to perform the requested Action on the Item.
5.5. May Perform, or request another Process to perform, Actions on Items even in the absence of Rights, if the Rules so allow.

6. An M-Instance identifies Processes/Items with one more than one Identifier. Each Identifier uniquely refers to only one Process/Item.
7. An M-Instance may require that Processes be Certified if imported and used in the M-Instance.
8. An M-Instance may enable a Process:
  8.1. To communicate to a Process in another M-Instance through an M-Instance's Resolution Service.
  8.2. To obtain conversion of the Format of an Item's Data by calling a Conversion Service.
  8.3. To specify their communication needs by:
    8.3.1. Requesting the needed maximum and average bitrate value.
    8.3.2. Reserving the needed bitrate for a time and a location.
    8.3.3. Requesting that the same simultaneous Experience be provided to a specified number of Devices.

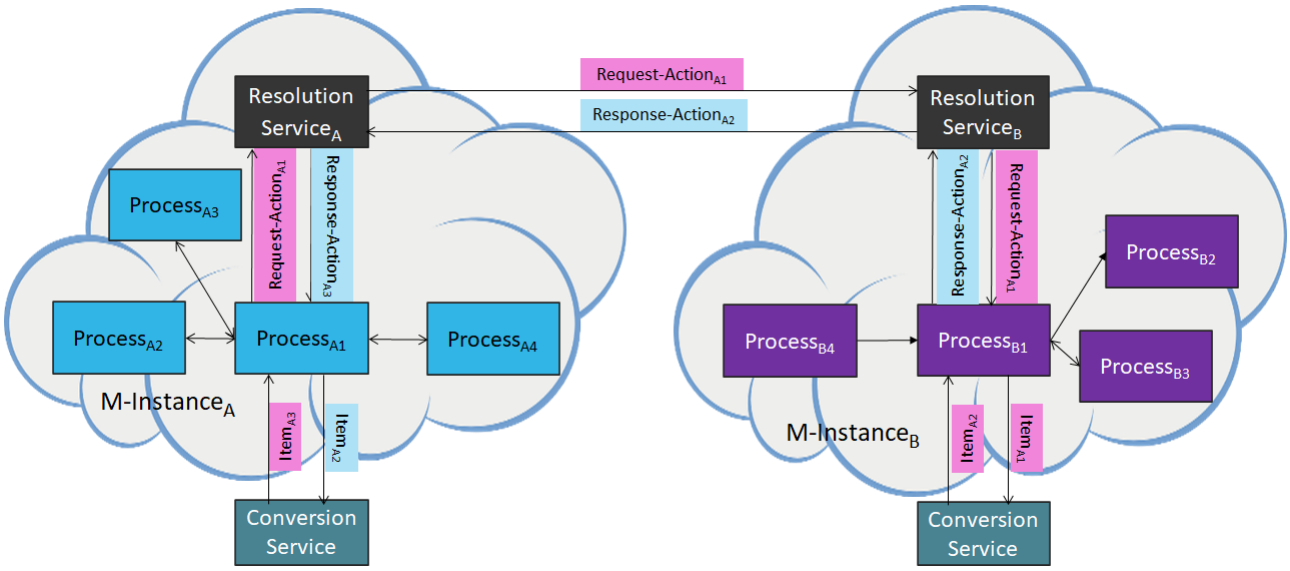


Figure 2 - Resolution and Conversion Services

Future versions of this Technical Specification may specify:

1. Additional Actions and Items.
2. Profiles of Actions and Items.

7 Functional Requirements of Processes

7.1 App

Purpose	An application-specific Program executed on a Device.
Functional Require-ments	The Manager of the M-Instance in which an App will be deployed may request that the Device be subject to certification.
Metadata	AppID: The ID of the App. InRightsID: The ID of the User's Rights to Act on the App. OutRightsID: The ID of the Rights a User may acquire to act on the App. DescrMdata: Any description of the App.

7.2 Device

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 Require-ments	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.
Metadata	DeviceID: The ID of the Device. InRightsID: The ID of the Device's Rights. OutRightsID: The ID of the Rights a User may acquire to act on the Device.

	DescrMdata	Any description of Device.

7.3 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. The Manager of the M-Instance may request certification of a hosted Service..
Metadata	ServiceID: The ID of the Service. InRightsID: The ID of the User's Rights to Act granted to the Service. OutRightsID: The ID of the Rights a User may acquire to Act on the Service. DescrMdata: Any description of the Rights.

7.4 User

Purpose	A Process representing a human.
Functional Requirements	1. A User may perform the following functions: 1.1. The interface of the human with the M-Instance. 1.2. Render the User as a Persona: 1.2.1. UM-Animated by a Stream. 1.2.2. MM-Animated by an autonomous agent. 2. Animation results from an MM-/UM-Animate Action and enabled by a Program run by the User. 3. The Animation Program may be provided: 3.1. By the human. 3.2. By the M-Instance. The Manager of the M-Instance where a User running a proprietary Program is deployed may request that the Program be subject to certification.
Metadata	UserID: ID of User. InRightsID: ID of Rights held by User. AccountIDs: IDs of the Accounts where the User can operate. WalletIDs: IDs of Wallets User has Rights on. PersonalDataID: ID of Personal Data. DescrMdata: Any description of the User.

8 Functional Requirements of Actions

8.1 General Actions in an M-Instance

8.1.1 Register

Purpose	The Action of a human requesting that an M-Instance grant selected human's Users the Rights to perform Actions in the M-Instance.	
Request-Action	Time	
	Source	humanID
	Destination	ServiceID
	Requested Action	*Register*
	InItem	PersonalData ∨ PersonalDataID
	InLocation	Address

	OutLocation		ServiceID	
	OutRights		Rights ∨ RightsID	
Response-Action	Success	OutItem	Account ∨ AccountID	
	Error	FaultyReq	The Request is Faulty	
		Wallet	Insufficient Value	

8.1.2 Change

Purpose	The Action of requesting that a Service modify the Rights of a User and provide OutRights, e.g., to further Change the Rights.			
Request-Action	Time			
	Source	UserID		
	Destination	ServiceID		
	Action	*Change*		
	InItems	UserID1 ∧ (Rights ∨ RightsID)		
	OutRights	Rights ∨ RightsID		
Response-Action	Success	OutItem		
	Error	FaultyReq	The Request is Faulty	
		IDs	Incorrect	
		Rights	Missing or incomplete	

8.1.3 Hide

Purpose	The Action of requesting that a Service make the ID of an Item unavailable and provide OutRights, e.g., to make the ID available again.			
Request-Action	Time			
	Source	UserID		
	Destination	ServiceID		
	Requested Action	*Hide*		
	InItem	Item ∨ ItemID		
	OutRights	Rights ∨ RightsID		
Response-Action	Success	OutItem		
	Error	FaultyReq	The Request is Faulty	
		IDs	Incorrect	
		Rights	Missing or incomplete	

8.1.4 Authenticate

Purpose	The Action of requesting that a Service confirm that an Entity is what it claims to be.			
Request-Action	Time			
	Source	UserID		
	Destination	ServiceID		
	Action	*Authenticate*		
	InItems	AuthenticateIn ∨ AuthenticateInID		
	InLocation	M-LocationID ∨ UserID1		
	OutLocation	UserID		
	OutRights	Rights ∨ RightsID		
Response-Action	Success	OutItem	AuthenticateOut ∨ AuthenticateOutID	
	Error	FaultyReq	The Request is Faulty	
		IDs	Incorrect	

	Rights	Missing or incomplete
	M-Location	Out of range
	U-Location	Out of range

8.1.5 Identify

Purpose	The Action of requesting that a Service produce an Item from Data & Metadata.		
Request-Action	Time		
	Source	Process ID	
	Destination	ServiceID	
	Action	*Identify*	
	InItems	DataMdata	
	InLocation	UserID	
	OutLocation	ServiceID	
	OutRight	Rights ∨ RightsID	
Response-Action	Success	OutItem	Item ∨ ItemID
	Error	FaultyReq	The Request is Faulty
		IDs	Incorrect
		Rights	Missing or incomplete

8.1.6 Modify

Purpose	The Action of requesting that a Service produce a new Item from an existing Item by providing new Data and Metadata with the OutRights to further Act on the new Item.		
Request-Action	Time		
	Source	Process ID	
	Destination	ServiceID	
	Action	*Modify*	
	InItems	Item ∧ DataMdata	
	InLocation	ServiceID ∨ UserID	
	OutLocation	ServiceID	
	OutRight	Rights ∨ RightsID	
Response-Action	Success	OutItem	Item ∨ ItemID
	Error	FaultyReq	The Request is Faulty
		IDs	Incorrect
		Rights	Missing or incomplete

8.1.7 Validate

Purpose	The Action of requesting that a Service verify that a Process has the Rights to perform or request a Process to perform an Action on an Item.		
Request-Action	Time		
	Source	ProcessID	
	Destination	ProcessID	
	Action	*Validate*	
	InItem	Request-Action ∨ Request-ActionID	
	InLocation	ProcessID	
	OutLocation	ProcessID	
	OutRights	Rights ∨ RightsID	

Response-Action	Success	OutItem	Item ∨ ItemID	
	Error	Request	Faulty	
		IDs	Incorrect	
		Rights	Missing or incomplete	

8.1.8 Execute

Purpose	The Action of requesting that a Process execute a Contract.			
Request-Action	Time			
	Source	UserID		
	Destination	ProcessID		
	Action	*Execute*		
	InItem	Contract ∨ ContractID		
	OutRights	Rights ∨ RightsID		
Response-Action	Success	OutItem		
	Error	FaultyReq	The Request is Faulty	
		IDs	Incorrect	
		Rights	Missing or incomplete	

8.2 Call a Service

8.2.1 Author

Purpose	The Action of Calling a Service to obtain an Item with associated OutRights to Act on the Item.			
Request-Action	Time			
	Source	UserID		
	Destination	ServiceID		
	Action	*Author*		
	InItems	Item ∨ ItemID ∨ DataMdata		
	InLocation	UserID ∨ Address		
	OutLocation	UserID ∨ ServiceID		
	OutRights	Rights ∨ RightsID		
Response-Action	Success	OutItem	Item ∨ ItemID	
	Error	FaultyReq	The Request is Faulty	
		IDs	Incorrect	
		Rights	Missing or incomplete	
		Wallet error	Insufficient Value	

8.2.2 Discover

Purpose	The Action of requesting that a Service provide a DiscoverOut Item containing: 1. The IDs of the Items relevant to the User's request to Discover or recommended expressed in the DiscoverIn Item 2. The OutRights to Act on the DiscoverOut Item.		
Request-Action	Time		
	Source	UserID	
	Destination	ServiceID	
	Action	*Discover*	
	InItem	DiscoverIn ∨ DiscoverInID	
	InLocation	UserID ∨ ServiceID	

	OutLocation	UserID		
	OutRights	Rights ∨ RightsID		
Response-Action	Success	OutItem	DiscoverOut ∨ DiscoverOutID	
	Error	FaultyReq	The Request is Faulty	
		IDs	Incorrect	
		Rights	Missing or incomplete	

8.2.3 Inform

Purpose	The Action of requesting that a Service provide an InformOut Item containing information about an InItem, such as the Metadata of the InItem, with the OutRights to Act on the InformOut Item.			
Request-Action	Time			
	Source	UserID		
	Destination	ServiceID		
	Action	*Inform*		
	InItem	InformIn ∨ InformInID		
	InLocation	M-LocationID		
	OutLocation	UserID		
	OutRights	Rights ∨ RightsID		
Response-Action	Success	OutItem	InformOut ∨ InformOutID	
	Error	FaultyReq	The Request is Faulty	
		IDs	Incorrect	
		Rights	Missing or incomplete	

8.2.4 Interpret

Purpose	The Action of requesting that a Service provide an InterpretOut Item containing interpretation of an InItem, such as translation or extraction of Personal Status, with the OutRights to Act on the InterpretOut Item.			
Request-Action	Time			
	Source	UserID		
	Destination	ServiceID		
	Action	*Interpret*		
	InItem	InterpretIn ∨ InterpretInID		
	InLocation	M-LocationID ∨ ServiceID		
	OutLocation	UserID		
	OutRights	Rights ∨ RightsID		
Response-Action	Success	OutItem	InterpretOut ∨ InterpretOutID	
	Error	FaultyReq	The Request is Faulty	
		IDs	Incorrect	
		Rights	Missing or incomplete	

8.2.5 Post

Purpose	The Action of requesting that a Marketplace include an Asset to its repertory of Assets.		
Request-Action	Time		
	Source	UserID	
	Destination	ServiceID	
	Action	*Post*	

	InItem	Asset ∨ AssetID	
	InLocation	UserID ∨ ServiceID	
	OutLocation	ServiceID	
	OutRights	Rights ∨ RightsID	
Response-Action	Success	OutItem	
	Error	FaultyReq	The Request is Faulty
		IDs	Incorrect
		Rights	Missing or incomplete
		Wallet	Insufficient Value

8.2.6 Transact

Purpose	The Action of a $User_1$ ("seller") requesting that a Service: 1. Assign Rights on an Asset to $User_2$ ("buyer"). 2. Cause: 2.1. $Wallet_1$ of $User_1$ ("seller") to be increased by $Value_1$. 2.2. $Wallet_2$ of $User_2$ to be decreased by $Value_2$. 2.3. $Wallet_3$ of the Service enabling/facilitating the Transaction to be increased by $Value_3$ (optionally).		
Request-Action	Time		
	Source	UserID	
	Destination	ServiceID	
	Action	*Transact*	
	InItem	Transaction ∨ TransactionID	
	InLocation	UserID ∨ ServiceID	
	OutLocation	UserID ∨ ServiceID	
	OutRights	Rights ∨ RightsID	
Response-Action	Success	OutItems	$AssetID \wedge WalletID_1 \wedge WalletID_2 \wedge WalletID$
	Error	FaultyReq	The Request is Faulty
		IDs	Incorrect
		Rights	Missing or incomplete
		Wallet	$Wallet_2$ has insufficient Value

8.2.7 Convert

Purpose	The Action of requesting that a Service change the Format of the Data of an Item into a Format specified by a FormatID. FormatIDs are defined in the Format Taxonomy (To be Defined).		
Request-Action	Time		
	Source	UserID	
	Destination	ServiceID	
	Action	*Convert*	
	InItem	(Item ∨ ItemID) ∧ FormatID	
	InLocation	ServiceID ∨ M-LocationID	
	OutLocation	ServiceID	
	OutRights	Rights ∨ RightsID	
Response-Action	Success	OutItem	Item ∨ ItemID
	Error	FaultyReq	The Request is Faulty
		IDs	Incorrect
		Rights	Missing or incomplete

8.2.8 Resolve

Purpose	The Action of requesting that a Service in an M-Instance forward a Request-Resolve Item or a Response-Resolve Item to a Resolution Service of another M-Instance.		
Request-Action	Time		
	Source	ProcessID	
	Destination	ServiceID	
	Requested Action	*Resolve*	
	InItem	(Request-Action ∨ Request-ActionID) ∨ (Response-Action ∨ Response-ActionID)	
	InLocation	ProcessID	
	OutLocation	ProcessID	
	OutRights	Rights ∨ RightsID	
Response-Action	Success	OutItem	Item ∨ ItemID
	Error	FaultyReq	The Request is Faulty
		IDs	Incorrect
		Rights	Missing or incomplete

8.3 Manage Entities (Metaverse to Metaverse)

8.3.1 MM-Add

Purpose	The Action of requesting that a Service add an Entity at an M-Location with a Spatial Attitude and provide OutRights to Act on the MM-Added Entity.		
Request-Action	Time		
	Source	UserID	
	Destination	ServiceID	
	Action	*MM-Add*	
	InItem	(Entity ∨ EntityID) ∧ Spatial Attitude	
	InLocation	UserID ∨ ServiceID ∨ M-LocationID	
	OutLocation	M-LocationID	
	OutRights	Rights ∨ RightsID	
Response-Action	Success	OutItem	Entity ∨ EntityID
	Error	FaultyReq	The Request is Faulty
		IDs	Incorrect
		Rights	Missing or incomplete
		Clash	Entity clashes with another Entity
		M-Location	Out of range

8.3.2 MM-Animate

Purpose	The Action of requesting that a Service change the features of a Model MM-Embedded at an M-Location with a Process and provide the OutRights to Act on the MM-Added Entity.	
Request-Action	Time	
	Source	UserID
	Destination	ProcessID
	Action	*MM-Animate*
	InItem	(Model ∨ ModelID) ∧ Spatial Attitude
	InLocation	ServiceID

	OutLocation	M-LocationID	
	OutRights	Rights ∨ RightsID	
Response-Action	Success	OutItem	
	Error	FaultyReq	The Request is Faulty
		IDs	Incorrect
		Rights	Missing or incomplete
		Item mismatch	Entity Data Type and Animation Stream Data Type.

8.3.3 MM-Disable

Purpose	The Action of requesting that a Service stop MM-Enabling selected Entities Embedded at an M-Location and provide OutRights to Act on the MM-Disabled Entities.		
Request-Action	Time		
	Source	UserID	
	Destination	ServiceID	
	Action	*MM-Disable*	
	InItem	List of EntityIDs	
	InLocation	M-LocationID	
	OutLocation	M-LocationID	
	OutRights	Rights ∨ RightsID	
Response-Action	Success	OutItem	
	Error	FaultyReq	The Request is Faulty
		IDs	Incorrect
		Rights	Missing or incomplete
		M-Location	Out of range

8.3.4 MM-Embed

Purpose	The Composite Action of requesting that a Service MM-Add and MM-Enable an Entity either located at a Service or at an M-Location at a destination M-Location with a Spatial Attitude and provide OutRights to Act on the MM-Embedded Entity.		
Request-Action	Time		
	Source	UserID	
	Destination	ServiceID	
	Action	*MM-Embed*	
	InItem	(Entity ∨ EntityID) ∧ Spatial Attitude	
	InLocation	ServiceID ∨ M-LocationID	
	OutLocation	M-LocationID	
	OutRights	Rights ∨ RightsID	
Response-Action	Success	OutItem	
	Error	FaultyReq	The Request is Faulty
		IDs	Incorrect
		Rights	Missing or incomplete
		Clash	Entity clashes with another Entity
		M-Location	Out of range

8.3.5 MM-Enable

Purpose	The Action of requesting that a Service implement requests to MM-Send selected Entities MM-Added at an M-Location per Rights of the requesting User and Provide OutRights to act on the selected M-Entities.		
Request-Action	Time		
	Source	UserID	
	Destination	ServiceID	
	Action	*MM-Enable*	
	InItem	Entity ∨ EntityID	
	InLocation	M-LocationID	
	OutLocation	M-LocationID	
	OutRights	Rights ∨ RightsID	
Response-Action	Success	OutItem	
	Error	FaultyReq	The Request is Faulty
		IDs	Incorrect
		Rights	Missing or incomplete
		M-Location	Out of range

8.3.6 MM-Send

Purpose	The Action of requesting that a Service forward to a Process an Item, or Data/Metadata with OutRights given to the Destination Process to Act on the Item or Data/Metadata.		
Request-Action	Time		
	Source	ProcessID	
	Destination	ProcessID	
	Action	*MM-Send*	
	InItem	Item ∨ ItemID ∨ DataMdata	
	InLocation	ProcessID ∨ M-Location	
	OutLocation	ProcessID ∨ M-Location	
	OutRights	Rights ∨ RightsID	
Response-Action	Success	OutItem	Item ∨ ItemID ∨ DataMdata
	Error	Request	Faulty
		IDs	Incorrect
		Rights	Missing or incomplete

8.4 Manage Entities (Metaverse to Universe)

8.4.1 MU-Actuate

Purpose	The Action of requesting that a Device present an Entity available at a Device to a U-Location as Media with a Spatial Attitude. The Device shall convert the Rights of the Items to rights specific of the U-Location.		
Request-Action	Time		
	Source	UserID	
	Destination	DeviceID	
	Action	*MU-Actuate*	
	InItem	(Entity ∨ EntityID) ∧ Spatial Attitude	
	InLocation	DeviceID	
	OutLocation	U-LocationID	

	OutRights	Metadata		
Response-Action	Success	OutItem	Media	
	Error	FaultyReq	The Request is Faulty	
		IDs	Incorrect	
		Rights	Missing or incomplete	
		U-Location	Out of range	

8.4.2 MU-Render

Purpose	The Composite Action of requesting that: 1. A Service MM-Send selected Entities Embedded at an M-Location to a Device. 2. The Device MU-Actuate the Entity received at a U-Location with a Spatial Attitude.			
Request-Action	Time			
	Source	UserID		
	Destination	ServiceID		
	Action	*MU-Render*		
	InItem	List of Entities ∧ Spatial Attitudes		
	InLocation	M-LocationID		
	OutLocation	U-Location		
	OutRights	Rights ∨ RightsID		
Response-Action	Success	Action result	Media	
	Error	FaultyReq	The Request is Faulty	
		IDs	Incorrect	
		Rights	Missing or incomplete	
		M-Location	Out of range	

8.4.3 MU-Send

Purpose	The Action of requesting that a Process store an Item at an Address.			
Request-Action	Time			
	Source	ProcessID		
	Destination	ServiceID		
	Action	*MU-Send*		
	InItem	Item ∨ ItemID		
	InLocation	M-LocationID ∨ ProcessID		
	OutLocation	Address		
	OutRights	Rights ∨ RightsID		
Response-Action	Success	OutItem		
	Error	FaultyReq	The Request is Faulty	
		IDs	Incorrect	
		Rights	Missing or incomplete	
		Address	Incorrect	

8.4.4 Track

Purpose	The Composite Action of requesting that a Service: 1. MM-Embed a Model at an M-Location with a Spatial Attitude. 2. UM-Animate the Model MM-Embedded at an M-Location. 3. MU-Render specified Entities at the M-Location to a U-Location.

Request-Action	Time		
	Source	UserID	
	Destination	ServiceID	
	Action	*Track*	
	InItem	(Model ∨ ModelID) ∧ Spatial Attitude ∧ M-LocationID	
	InLocation	ServiceID	
	OutLocation	U-LocationID	
	OutRights	Rights ∨ RightsID	
Response-Action	Success	OutItem	Media
	Error	FaultyReq	The Request is Faulty
		IDs	Incorrect
		Rights	Missing or incomplete
		M-LocationID	Out of range
		U-LocationID	Out of range

8.5 Manage Entities (Universe to Metaverse)

8.5.1 UM-Animate

Purpose	The Composite Action of a User requesting: 1. A Device to 1.1. UM-Capture an animation stream extracted from an object at a U-Location. 1.2. UM-Send the animation stream and Metadata to a User. 2. A Service to Identify the Animation Stream. 3. A Service to MM-Animate the Model MM-Embedded at the M-Location using the Animation Stream.		
Request-Action	Time		
	Source	UserID	
	Destination	DeviceID	
	Action	*UM-Animate*	
	InItem	object ∧ (Model ∨ ModelID)	
	InLocation	U-LocationID	
	OutLocation	M-LocationID	
	OutRights	Rights ∨ RightsID	
Response-Action	Success	OutItem	Entity ∨ EntityID
	Error	FaultyReq	The Request is Faulty
		IDs	Incorrect
		Rights	Missing or incomplete
		U-Location	Out of range
		M-Location	Out of range

8.5.2 UM-Capture

Purpose	The Action of requesting that a Device capture Media from a scene at a U-Location.	
Request-Action	Time	
	Source	UserID
	Destination	DeviceID
	Action	*UM-Capture*

	InItem	scene		
	InLocation	U-LocationID		
	OutLocation	DeviceID		
Response-Action	Success	OutItem	Media	
	Error	FaultyReq	The Request is Faulty	
		IDs	Incorrect	
		Rights	Missing or incomplete	
		U-Location	Out of range	

8.5.3 UM-Render

Purpose	The Composite Action of a User requesting: 1. A Device to: 1.1. UM-Capture Media from a scene at U-Location. 1.2. MM-Send Data and Device-provided Metadata to a User. 2. A Service to: 2.1. Identify an Entity from UM-Sent Data and Metadata. 2.2. MM-Embed the Entity at an M-Location with a Spatial Attitude.			
Request-Action	Time			
	Source	UserID		
	Destination	DeviceID		
	Action	*UM-Render*		
	InItem	scene ∧ Spatial Attitude		
	InLocation	U-LocationID		
	OutLocation	M-LocationID		
	OutRights	Rights ∨ RightsID		
Response-Action	Success	OutItem	Entity ∨ EntityID	
	Error	FaultyReq	The Request is Faulty	
		IDs	Incorrect	
		Rights	Missing or incomplete	
		Clash	Entity clashes with another Entity	
		M-Location	Out of range	
		U-Location	Out of range	

8.5.4 UM-Send

Purpose	The Action of a Device transmitting Data & Metadata to a Process.			
Request-Action	Time			
	Source	DeviceID		
	Destination	ProcessID		
	Action	*UM-Send*		
	InItem	DataMdata		
	InLocation	DeviceID ∨ Address		
	OutLocation	ProcessID		
	OutRights	Metadata		
Response-Action	Success	OutItem	DataMdata	
	Error	Request	Faulty	
		IDs	Incorrect	
		Rights	Missing or incomplete	
		U-Location	Out of range	

9 Functional Requirements of Items

9.1 General Items

9.1.1 M-Instance

Purpose	An implementation of MPAI-Metaverse Model – Architecture.
Functional Requirements	1. An M-Instance implements a Functional Profile of MPAI Metaverse Model – Architecture. 2. An M-Instance may 2.1. Add Functionalities not supported by MPAI-Metaverse Model – Architecture. In this case full Interoperability with other M-Instances may not be achieved. 2.2. Enables the creation of M-Environments.
Metadata	M-InstanceID: The ID of the M-Instance. DescrMdata: Any description of the M-Instance.

9.1.2 M-Capabilities

Purpose	An Item providing the list of features of an M-Instance.
Functional Requirements	An M-Instance may show its M-Capabilities, including: 1. Profile of the M-Instance. 2. Data Formats supported by Items (needs a Taxonomy). 3. Services supported (needs a Taxonomy).
Metadata	M-CapabilitiesID: The ID of the M-Capabilities. DescrMdata: Any description of the M-Capabilities.

9.1.3 M-Environment

Purpose	An Identified administrative subset of an M-Instance.
Functional Requirements	An M-Environment may expose Rules concerning the definition, Identification, and access to M-Locations.
Metadata	M-EnvironmentID: The ID of the M-Environment. DescrMdata: Any description of the M-Environment.

9.1.4 Identifier

Purpose	An Item that uniquely references an Item in an M-Instance.
Functional Requirements	An Item can have more than one Identifier. An Identifier of an Item may have a hierarchical structure in case it is useful to Identify an Item based on the M-Instance, the M-Environment: 1. M-InstanceID 2. M-EnvironmentID 3. ItemID.
Metadata	DescrMdata: Any description of the M-Environment.

9.1.5 Rules

Purpose	An Item expressing the terms and conditions under which a human deploys a User in an M-Instance or M-Environment.

Functional Requirements	The Rules 1. Shall express the permissions, obligations, and prohibitions with respect to Actions performed by a User on Items. 2. May grant a User the Rights to: 2.1. Make Transactions. 2.2. Export Items to another Metaverse. 3. May grant a visitor (non-Registered human) limited Rights.
Metadata	RulesID: The ID of the Rules. M-InstanceID: The ID of the M-Instance "where the Rules hold". DescrMdata: Any descriptive Metadata.

9.1.6 Rights

Purpose	An Item expressing the ability of a Process to perform Actions on an Item during Times, at M-Locations, and U-Locations.
Functional Requirements	Rules may waive the need for certain Processes to hold certain Rights to perform certain Actions at certain Times.
Metadata	RightsID: The ID of the Rights. DescrMdata: Any description of the Rights.

9.1.7 Program

Purpose	An Item containing code that can be executed.
Functional Requirements	The code: 1. Shall be executable in the M-Instance. 2. May be subject to certification before being admitted to the M-Instance.
Metadata	ProgramID: The ID of the Program. DescrMdata: Any descriptive Metadata.

9.1.8 Contract

Purpose	A Program stored on a Device that is activated by an external entity, e.g., a User, or another activated Contract.
Functional Requirements	The Data of a Contract shall be in a form that allows execution in the M-Instance. A Contract may be subject to a certification carried out by or on behalf of the M-Instance Manager before it can be imported into an M-Instance.
Metadata	ContractID: The ID of the Contract. DescrMdata: Any description of the Contract.

9.2 Human and User-related Items

9.2.1 Account

Purpose	An Item that uniquely references a human who has Registered.
Functional Requirements	A User may have more than one Account in 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 Personal Data. 3. The Rights held by each User in the M-Instance/M-Environment. 4. The IDs of Devices, Apps, and Users, and Personae.

Metadata	AccountID	The ID of the Account.
	humanID	The ID of the Account holder.
	M-InstanceID	The ID of the M-Instance the Account refers to.
	DescrMdata	Any description of the Account.

9.2.2 Activity Data

Purpose	An Item containing the record of all the Actions made by a User.	
Functional Require-ments	Activity Data shall include: 1. The M-Instance ID. 2. The duration the Activity Data refer to. 3. The list of Interactions.	
Metadata	ActivityDataID	The ID of the Activity Data.
	UserID	The ID of the User the Activity Data refers to.
	M-InstanceID	The ID of the M-Instance the Account refers to.
	DescrMdata	Any description of the Activity Data.

9.2.3 Personal Profile

Purpose	An Item containing the Data about the human represented by a User.	
Functional Require-ments	The Personal Profile may include: 1. First Name 2. Last Name 3. Address 4. Nationality. 5. Age.	
Metadata	PersonalPro-fileID	The ID of the Personal Profile.
	humanID	The ID of the human the Personal Profile refers to.
	DescrMdata	Any description of the Personal Profile.

9.2.4 Social Graph

Purpose	A representation of the network of connections with Items and Processes developed by a User.	
Functional Require-ments	The Social Graph represents the Times and M-Locations of Interactions with Items and Processes.	
Metadata	SocialGraphID	The ID of the Social Graph.
	UserID	The ID of the User the Social Graph refers to.
	DescrMdata	Any description of the Social Graph.

9.2.5 Personal Data

Purpose	An Item containing Activity Data, Personae, and Social Graph of a User.	
Functional Require-ments	Personal Data includes the following Data: 1. Rights and Obligations of the Account-holding human. 2. The Personal Profile of the human. 3. The Personae that the Users of the human may assume. 4. The Activity Data of the Users of the human. 5. The Social Graphs of the Users of the human.	
Metadata	PersonalDataID	ID of PersonalData.

AccountID	ID of Account held by User.
PersonalProfileID	ID of Personal Profile.
PersonaIDs	IDs of Personae held User.
ActivityDataIDs	ID of Activity Data
SocialGraphIDs	ID of SocialGraph
DescrMdata	Any description of the Personal Data.

9.3 Items for Process Interaction

9.3.1 Message

Purpose	An Item a Source Process MM-Sends to a Destination Process.	
Functional Requirements	A Message may contain: 1. An Item, or 2. Data and Metadata	
Metadata	MessageID	The ID of the Message.
	DescrMetadata	Any description of the Message.

9.3.2 P-Capabilities

Purpose	An Items containing a description of the characteristics of a Process.	
Functional Requirements	1. Provision of the P-Capabilities Item by a Process is optional. 2. The characteristics of all Processes are: 2.1. List of Actions that can be performed. 2.2. List of Items supported with Data Formats. 2.3. List of Data Types supported. 2.4. The cost of performing an Action. 3. Additionally, a Device has: Apps on board. 4. Additionally, a User has: Human represented by User.	
Metadata	PCapabilitiesID	The ID of P-Capabilities of a Process.
	Descriptive Metadata	Any description of the P-Capabilities.

9.3.3 Request-Action

Purpose	An Item containing the request to a Process to perform an Action as defined in MPAI-MMM - Architecture.	
Functional Requirements	A Request-Action shall include: 1. Time the Request-Action was issued. 2. The Source ProcessID. 3. The Destination ProcessID. 4. The Action requested. 5. The ItemIDs relevant to the Action. 6. The Locations of the Items. 7. The Locations of the output Items produced by the Request-Action. 8. The requested Rights on the output Items.	
Metadata	Request-ActionID	The ID of the Request-Action.
	DescrMdata	Any descriptive Metadata.

9.3.4 Response-Action

Purpose	An Item containing the response of a Process to a Request-Action as defined in MPAI-MMM - Architecture.

Functional Require-ments	The Response-Item shall include: 1. Time the Response-Action was issued. 2. The Source ProcessID (Source refers to the Process that issued the request). 3. The Destination ProcessID. 4. The output Items produced by the Request-Action. 5. Error messages.
Metadata	Request-ActionID: The ID of the Response-Action. DescrMdata: Any descriptive Metadata.

9.4 Items for Service Access

9.4.1 AuthenticateIn

Purpose	An Item containing: 1. The Entity or the ID of the Entity to be Authenticated. 2. Information related to the rendering of AuthenticateOut (optional).
Functional Require-ments	Examples of Entity that may be subject to Authentication can be: 1. Speech produced by a User. 2. The visual appearance of a User. 3. … Information on the rendering of InterpretOut may be provided by: 1. Media type (text, speech, image, etc.) used for rendering. 2. Spatial Attitude of the Object rendering AuthenticateOut.
Metadata	AuthenticateInID: The ID of the AuthenticateIn Item. UserID: The ID of the User generating the AuthenticateIn Item. ServiceID: The ID of the Service providing Authentication Services. RightsID: The ID of the Rights "to Act on the AuthenticateIn Item" granted to the Authentication Service. DescrMdata: Any description of the AuthenticateIn Item.

9.4.2 AuthenticateOut

Purpose	An Item containing the result of the Service Acting on the Request-Authenticate Item and information about its rendering.
Functional Require-ments	AuthenticateOut shall be: 1. Made available to the User as Item, and 1. Rendered to the User as a perceptible Object as specified in AuthenticateIn.
Metadata	AuthenticateOutID: The ID of the Request-Authenticate Item. ServiceID: The ID of the Service providing the AuthenticateOut Item. UserID: The ID of the User having requested the AuthenticateOut Item. RightsID: The ID of the Rights "to Act on the AuthenticateOut Item" granted to UserID. DescrMdata: Any description of the AuthenticateOut Item.

9.4.3 DiscoverIn

Purpose	An Item containing: 1. A description of the Items to be Discovered or recommended. 2. Information related to the rendering of DiscoverOut.

Functional Require-ments	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 DiscoverOut Rendering may be provided by: 1. Media type used for rendering. 2. Spatial Attitude of the Object rendering DiscoverOut.
Metadata	

DiscoverInID	The ID of the DiscoverIn Item.
UserID	The ID of the User generating the DiscoverIn Item.
ServiceID	The ID of the Service providing Discovery Services.
RightsID	The ID of the Rights "to Act on the DiscoverIn Item" granted to the Discovery Service.
DescrMdata	Any description of the DiscoverIn Item.

9.4.4 DiscoverOut

Purpose	An Item containing the description of the Items Discovered and information related to its rendering.
Functional Require-ments	Discovered Items 1. Shall be Identified by their IDs. 2. May be: 2.1. Accompanied by descriptions. 2.2. Rendered to the User as a perceptible Object.
Metadata	

DiscoverOutID	The ID of the Request-Discover Item.
ServiceID	The ID of the Service providing the DiscoverOut Item.
UserID	The ID of the User having requested the DiscoverOut Item.
RightsID	The ID of the Rights "to Act on the DiscoverOut Item" granted to UserID.
DescrMdata	Any description of the DiscoverOut Item.

9.4.5 InformIn

Purpose	An Item containing: 1. A description of the Item about which information is requested. 2. Information related to the rendering of InformOut.
Functional Require-ments	InformIn may refer to: 1. Item Metadata 2. Any other information that a Service may provide on the Item. Information on the rendering of InformOut may be provided by: 1. Media types used for rendering. 2. Spatial Attitude of InformOut rendered Object.
Metadata	

InformInID	The ID of the InformIn Item.
UserID	The ID of the User generating the InformIn Item.
ServiceID	The ID of the Service providing Inform Services.
RightsID	The ID of the Rights "to Act on the InformIn Item" granted to the Inform Service.
DescrMdata	Any description of the InformIn Item.

9.4.6 InformOut

Purpose	An Item containing the description of the Item object of an InformIn.

Functional Requirements	InformOut shall be: 1. Made available to the User as an Item, or 2. Rendered to the User as a perceptible Object.	
Metadata	InformOutID	The ID of the InformOut Item.
	ServiceID	The ID of the Service providing the InformOut Item.
	UserID	The ID of the User having requested the InformIn Item.
	RightsID	The ID of the Rights "to Act on the InformOut Item" granted to UserID.
	DescrMdata	Any description of the InformOut Item.

9.4.7 InterpretIn

Purpose	An Item containing: 1. The ID or the Item to be Interpreted. 2. Information related to the rendering of InterpretOut.	
Functional Requirements	Items candidate for interpretation may be provided as the Item or Identified by ItemID. Information on InterpretOut Rendering may be provided by: 1. Media type used for rendering. 2. Spatial Attitude of InterpretOut rendered Object.	
Metadata	DiscoverInID	The ID of the InterpretIn Item.
	UserID	The ID of the User generating the InterpretIn Item.
	ServiceID	The ID of the Service providing Interpret Services.
	RightsID	The ID of the Rights "to Act on the InterpretIn Item" granted to the Interpret Service.
	DescrMdata	Any description of the InterpretIn Item.

9.4.8 InterpretOut

Purpose	An Item containing the description of the Item provided in response to an InterpretIn Item.	
Functional Requirements	Interpreted Items shall be: 1. Described by the IDs of the Interpreted Item. 2. Rendered to the User as a perceptible Object.	
Metadata	InterpretOutID	The ID of the InterpretOut Item.
	ServiceID	The ID of the Service providing the InterpretOut Item.
	UserID	The ID of the User having requested the InterpretOut Item.
	RightsID	The ID of the Rights "to Act on the InterpretOut Item" granted to UserID.
	DescrMdata	Any description of the InterpretOut Item.

9.5 Finance-related Items

9.5.1 Asset

Purpose	An Item that can be Transacted.
Functional Requirements	An Asset 1. May be: 1.1. MM-Embedded at an M-Location. 1.2. Posted to a Service (e.g., a marketplace). 2. Shall: 2.1. Preserve the Data Formats of the Item that has spawned it.

	2.2. Include the date it was Modified as Asset.
Metadata	AssetID: The ID of the Asset. UserIDs: The ID of the Users "having Rights to the Asset". ProvenanceID: The ID of the Provenance of the Asset. DescrMdata: Any description of the Asset.

9.5.2 Ledger

Purpose	An Item containing a list of Transactions involving Assets.
Functional Requirements	The list of entries of a Ledger may be based on: 1. The types of Items included in the Ledger. 2. The Items at an M-Location. 3. The Users to which a Ledger refers. 4. The duration in time to which a Ledger refers. 5. The Items in a DiscoverOut Item.
Metadata	LedgerID: The ID of the Ledger. UserID: The ID of the User who "has produced the Ledger". DescrMdata: Any descriptive Metadata.

9.5.3 Provenance

Purpose	The Ledger associated and included in the metadata of a specific Asset.
Functional Requirements	The Provenance Item shall include the list of all Transactions executed: 1. On an Asset. 2. Starting from the first Transaction and including the last. 3. The Marketplace on which a Transaction may have been performed.
Metadata	ProvenanceID: The ID of the Provenance. DescrMdata: Any descriptive Metadata.

9.5.4 Transaction

Purpose	Item representing the changed state of the Wallets and the Rights of a seller User and a buyer User on an Asset and optionally of the Service facilitating/enabling the Transaction
Functional Requirements	The Transaction shall represent: 1. The Time the Transaction is performed. 2. The Value moving into the Wallet of User 1 (seller). 3. 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 Value has moved to the seller Wallet. 6. The Time the Value has moved from the buyer Wallet. 7. The Rights to Act on the Asset owned by User1 after Time. 8. The Rights to Act on the Asset owned by User2 after Time.
Metadata	TransactionID: The ID of the Transaction. AssetID: The ID of the Asset the Transaction refers to. UserID: The ID of User1 "who grants the Rights". RightsID: The ID of the Rights held by User1 after Time. WalletID: The ID of the Wallet of User1. TargetUserID: The ID of the User2 "who is granted the Rights". OutRightsID: The ID of the Rights "granted to User2". TargetWalletID: The ID of the Wallet of User2. ServiceID: The ID of the Marketplace.

	ServiceWalletID	The ID of the Wallet of the Marketplace.	
	DescrMdata	Any description of the Transaction.	

9.5.5 Value

Purpose	An Amount and the Currency with which the Amount is expressed.		
Functional Require-ments	Value shall have a representation that enables the expression of the Amount and the Currency used to represent the Amount.		
Metadata	ValueID	The ID of the Value.	
	DescrMdata	Any description of the User.	

9.5.6 Wallet

Purpose	A container of Values.		
Functional Require-ments	A Wallet shall enable the representation of the Amounts in each Currency.		
Metadata	WalletID	The ID of the Wallet.	
	UserID	The ID of the User “having Rights to the Wallet”.	
	RightsID	The ID of the Rights “User has on the Wallet”.	
	DescrMdata	Any description of the User.	

9.6 Perception-related Items

9.6.1 Event

Purpose	An Entity that includes selected Entities at an M-Location during a period.		
Functional Require-ments	An Event shall include: 1. Start Time and End Time. 2. M-LocationID. 3. List of selected Entities of the M-Location.		
Metadata	EventID	The ID of the Event.	
	AuthorID	The ID of the User “who authored the Event”.	
	ParentItemID	The ID of the Entity “from which the Event is derived”.	
	DescrMdata	Any description of the Event.	

9.6.2 Experience

Purpose	An Entity comprising User-selected Entities of an Event and the User Interac-tions with the selected Entities.		
Functional Require-ments	An Experience shall include: 1. ID of User the Experience refers to. 2. EventID 3. Start Time and End Time of the Experience. 4. List of Entities perceived and Interactions made by a User.		
Metadata	ExperienceID	The ID of the Experience.	
	AuthorID	The ID of the User “who authored the Experience”.	
	ParentEntityID	The ID of the Event spawning the Experience.	
	DescrMdata	Any description of the Experience.	

9.6.3 Interaction

Purpose	An Item containing the Actions performed by a User on an Entity.
Functional Requirements	An Interaction includes: 1. The Request-Action issued by a User on an Entity at an M-Location and the corresponding Time. 2. The Response-Action issued by the requested Process.
Metadata	InteractionID: The ID of the Interaction. UserID: The ID of the User "who created the Interaction". EntityID: The ID of Entity "the User Interacted with". DescrMdata: Any description of the Interaction.

9.6.4 Map

Purpose	An Item providing U-Locations and corresponding M-Locations and/or Entities.
Functional Requirements	A Map shall include a list composed of: 1. U-LocationID and Metadata related to the U-LocationID. 2. M-LocationID(s) and/or EntityIDs, and Metadata related to the M-LocationID(s) and/or EntityIDs corresponding to the U-LocationID.
Metadata	MapID: The ID of the Map. AuthorID: The ID of the User "who authored the Map". Auth.ToolID: The ID of the Service "who provided the Authoring tool".

9.6.5 Model

Purpose	An Object having features allowing it to be MM-Animated or UM-Animated.
Functional Requirements	A Model shall include: 1. The type(s) of Media (Audio-Visual-Haptic) composing the Model. 2. The type(s) of Animation the Model can accept.
Metadata	ModelID: The ID of the Object Model. AuthorID: The ID of the User "who authored or imported the DataMdata used to make the Object Model". Auth.ToolID: The ID of the Service "who provided the Authoring tool". DescrMdata: Any description of the Model.

9.6.6 Object

Purpose	An Item with perceptibility attributes that include one of the following Media types: Audio, Visual, and Haptics.
Functional Requirements	An Object shall include: 1. The type(s) of Media (Audio-Visual-Haptic) composing the Model. 2. The Data representation 2.1. Audio Data representation, e.g.: 2.1.1. Mono (e.g., speech) 2.1.2. Stereo 2.1.3. Multichannel 2.1.4. Microphone array 2.1.5. Spatial Audio 2.2. Visual Data representation, e.g.: 2.2.1. Still 2.2.2. Mono 2.2.3. Camera array

	2.2.4. Light field 2.2.5. Holography 2.2.6. RADAR 2.2.7. LiDAR 2.3. Haptic Data representation. 3. The type of Object, e.g.: 3.1. An inanimate Object (e.g., a table). 3.2. A Model (e.g., a Persona). 3.3. An autonomous Object (e.g., a robot). 3.4. An animal, possibly with high accuracy. 3.5. A human, possibly with high accuracy. 4. The Data Format used.
Metadata	Object ID: The ID of the Object Identified by ObjectID. AuthorID: The ID of the User "who authored the Object or imported the DataMdata used to make the Object". Auth.ToolID: The ID of the Service "who provided the Authoring tool" DescrMdata: Any description of the Object.

9.6.7 Scene

Purpose	A possibly hierarchical Composition of Objects having Spatial Attitudes.
Functional Require-ments	1. It should be possible to MM-Embed a Scene at a specified M-Location. 2. The Scene may be MM-/UM-Animated. 3. The Objects in the Scene may be MM-/UM-Animated.
Metadata	SceneID: The ID of the Scene Identified by SceneID AuthorID: The ID of the User "who authored the Scene". Auth.ToolID: The ID of the Service "who provided the Creation tool". DescrMdata: Any description of the Scene.

9.6.8 Stream

Purpose	An Item made of a continuous flow of Data, e.g., an Animation Stream.
Functional Require-ments	Streams may be used to: 1. UM-Animate a Model. 2. UM-Animate an Object. 3. UM-Animate a Scene. 4. Represent a Digitised Object (e.g., a Point Cloud) MM-Embedded at an M-Location.
Metadata	StreamID: The ID of the Stream. AuthorID: The ID of the User "who created or imported the Stream". Auth.ToolID: The ID of the Service "who provided the Creation tool". DescrMdata: Any description of the Stream.

9.6.9 Summary

Purpose	An abridged outline of the content of the utterance(s) of one or more Users possibly including their Personal Statuses.
Functional Require-ments	A Summary contains Text and Personal Status labels.
Metadata	SummaryID: The ID of the Summary Item. UserID: The ID of the User "who authored the Summary".

	DescrMdata	Any description of the Summary.

9.7 Space-related Items

9.7.1 M-Location

Purpose	A delimited space of an M-Instance.	
Functional Requirements	An M-Location: 1. Shall define the space of the M-Instance or M-Environment that is included in the M-Location. 2. May enable the creation of sub-spaces defining sub-M-Locations included in the M-Location (e.g., the rooms of an apartment).	
Metadata	M-LocationID	The ID of the M-Location Item.
	DescrMdata	Any description of the M-Location.

9.7.2 U-Location

Purpose	A delimited portion of the Universe.	
Functional Requirements	A U-Location: 1. Shall define the space in the Universe that is included in the U-Location. 2. May enable the definition of sub-spaces defining sub-U-Locations included in the U-Location. The enforcement of Rights to a U-Location is not part of MPAI-MMM - Architecture.	
Metadata	U-LocationID	The ID of the U-Location.
	DescrMdata	Any description of the U-Location.

10 Functional Requirements of Data Types

10.1 For location and time information

10.1.1 Address

Purpose	The location of a storage facility.
Functional Requirements	The location is identified by a URL.

10.1.2 Coordinates

Purpose	A set of numbers representing Position in an M-Location.
Functional Requirements	Coordinates shall uniquely indicate a point in the space identified by the coordinate system.

10.1.3 Orientation

Purpose	The Euler angles indicating the rotation of an Object.
Functional	MPAI-MMM – Architecture assumes that the principal axis of an Object is the x axis, the y axis has an angle of 90˚ counterclockwise (right-to-left) with the x axis and its z axis points up toward a User viewing from above.

Require-ments	Orientation is expressed as an array of 3 rows and 3 columns: ϕ,θ,ψ 1st order time derivatives of ϕ,θ,ψ 2nd order time derivatives of ϕ,θ,ψ

10.1.4 Point of View

Purpose	The Spatial Attitude of a Persona watching a Scene.
Func-tional Require-ments	Point of View shall express the Position of a Persona with an Orientation watching a Scene. Portable Avatar Format defines the x,y,z of a body [6].

10.1.5 Position

Purpose	The Coordinates of a point in an M-Environment using a Coordinate system.
Func-tional Require-ments	Position is expressed as an array of 3 rows and 3 columns: Cartesian: x,y,z 1st order time derivatives of x,y,z 2nd order time derivatives of x,y,z Spherical: r,ϕ,θ 1st order time derivatives of r,ϕ,θ 2nd order time derivatives of r,ϕ,θ

10.1.6 Spatial Attitude

Purpose	The Position and Orientation of an Entity, and their velocities and accelerations.
Func-tional Require-ments	Spatial Attitude is expressed as an array of 6 rows and 3 columns: Cartesian: x,y,z 1st order time derivatives of x,y,z 2nd order time derivatives of x,y,z or Spherical: r,ϕ,θ 1st order time derivatives of r,ϕ,θ 2nd order time derivatives of r,ϕ,θ and The Euler angles: α,β,γ 1st order time derivatives of α,β,γ 2nd order time derivatives of α,β,γ

10.1.7 Time

Purpose	The representation of the measure of time.
Func-tional Require-ments	Time shall be able to express any value of time on the time axis.

10.2 For Transactions

10.2.1 Amount

Purpose	A number expressing a Value in a Currency.

Functional Requirements	The ability to express an Amount, e.g., a decimal number.

10.2.2 Currency

Purpose	A medium of exchange enabling Transactions in a M-Environment.
Functional Requirements	A Currency shall belong to a list of Currencies.

10.3 For internal state information

10.3.1 Cognitive State

Purpose	The representation of a User's Personal Status that reflects the way they understand the context, such as "Confused", "Dubious", "Convinced".
Functional Requirements	The ability to express a particular Cognitive State with a label from a dictionary of Cognitive States.

10.3.2 Emotion

Purpose	The representation of a User's Personal Status that results from their interaction with a context, such as "Angry", "Sad", "Determined".
Functional Requirements	The ability to express a particular Emotion with a label from a dictionary of Emotions.

10.3.3 Social Attitude

Purpose	The representation of a User's Personal Status representing the way User intends to position vis-à-vis other Users, e.g., "Respectful", "Confrontational", "Soothing".
Functional Requirements	The ability to express a particular Social Attitude with a label from a dictionary of Social Emotions.

10.3.4 Personal Status

Purpose	The representation of the information internal to a User characterising their behaviour.
Functional Requirements	Personal Status may include any of Cognitive State, Emotion, Social Attitude or none of them

11 MPAI-MMM Scripting Language

The MPAI-MMM Scripting Language – MMM-Script in the following – serves the twin purposes of providing:

1. A handy tool to describe the Actions performed by Processes in an M-Instance.
2. A compact form that a Process can use to request another Process to perform Actions.

11.1 MMM-Script for Action Description

The performance of any Action in an M-Instance can be expressed as:

Process ActsOn Item | DataMdata | Media

At	Service | User | MLoc | ULoc	Where Item ends up being located with SA
By	Service	Used to perform *Action*
From	Address | ULoc	Address, ULoc where the source is located
Into	Item	Action leads to
Of	User	Item refers to
To	Address | Device | Process | User	Item/Process where Items ends up being placed
With	DataMdata | Stream	Additional Item required to perform Action

Note: SA is used as a compact form for Spatial Attitude.

Table 5 lists the possible combinations of Actions. Composite Actions are divided into elementary Actions.

Table 5 - Action-Item relationships

ProcessA	*General Actions*	*Item*	*Indirect object*	*ProcessB*
human	Registers	Personal Data	By M-Instance	AccountID
User	Changes	Rights	Of User	RightsID
User	Hides	Item		RightsID
User	Identifies	DataMdata	Into Item	ItemID
User	Modifies	Item	With DataMdata Into Item	ItemID
Process	Validates	RequestAction	By Service	
Process	MM-Sends	RequestAction	To Process	
Process	MM-Sends	ResponseAction	To Process	
Process	Executes	Contract		
	Call a Service			
User	Authenticates	AuthenticateIn	At User	AuthenticateOut
User	Authors	DataMdata	By Service At Service	ItemID
User	Discovers	DiscoverIn	By Service At User	DiscoverOut
User	Informs	InformIn	By Service At User	InformOut
User	Interprets	InterpretIn	By Service At User	InterpretOut
User	Posts	Item	At Service By Service	ItemID
User	Transacts	Item	To User To Service By Service	
User	Converts	Item	By Service Into Item At User	Item
	Resolves (composite)	Item	By Service Into Item At User	
Process	- MM-Sends	ReqAct	To Service	ResAct
Service	- MM-Sends	ReqAct	To Service	ResAct
Service	- MM-Sends	ReqAct	To Process	ResAct
	Mng Entities (MM)			
User	MM-Adds	Entity	At M-Location,SA	
	MM-Animates (composite)			
User	- Modify	Model	With Stream Into Item	ItemID
User	- MakeAnim	Item	With Stream	

User	MM-Disables	Entity	At M-Location	
	MM-Embeds (composite)			
User	- MM-Adds	Entity	At M-Location,SA	
User	- MM-Enable	Entity	At M-Location	
User	MM-Enables	Entity	At M-Location	
Process	MM-Sends	Item, DataMdata	To Process	
	Mng Entities (MU)			
Device	MU-Actuates	Entity	At U-Location,SA	
	MU-Renders (composite)			
User	- MM-Sends	Entity	To Device	
Device	- MU-Actuates	Entity	At U-Location,SA	
Process	MU-Sends	Item	To Address	
	Track (composite)			
User	- MM-Embeds	Model	At M-Location,SA	
User	- UM-Animates	Model	With Stream at M-Location	
User	- MU-Renders	Entity	At U-Location,SA	
	Mng Entities (UM)			
	UM-Animates (composite)			
Device	- UM-Captures	Stream	From U-Location	
Device	- UM-Sends	Stream	To User	
User	- Identifies	Stream	Into Item	
User	- MM-Animates	Model	With Stream At M-Location,SA	
Device	UM-Captures	Media	From U-Location To Device	
	UM-Renders (composite)			
Device	- UM-Captures	Media	From U-Location	
Device	- UM-Sends	DataMdata	To User	
User	- Identifies	DataMdata	Into Item	
User	- MM-Embeds	Entity	At M-Location,SA	
Device	UM-Sends	DataMdata	From Address To Process	

11.2 Definition in Backus-Naur form

```
program :=
         | /* empty */
         | one_or_more_statements

one_or_more_statements :=
                         | statement
                         | statement one_or_more_statements

statement :=
           | id action_keyword id modifiers

action_keyword :=
               |“Register”
               | “Change”
               | “Hide”
               | “Authenticate”
               | “Identify”
               | “Modify
               | “Validate”
               | “Execute”
               | “Author”
               | “Discover”
               | “Inform”
               | “Interpret”
               | “Post”
```

```
                | “Transact”
                | “Convert”
                | “Resolve”
                | “MM-Add”
                | “MM-Animate”
                | “MM-Disable”
                | “MM-Embed”
                | “MM-Enable”
                | “MM-Send”
                | “MU-Actuate”
                | “MU-Render”
                | “MU-Send”
                | “Track”
                | “UM-Animate”
                | “UM-Capture”
                | “UM-Render”
                | “UM-Send”

modifiers :=
            | /* empty */
            | one_or_more_modifiers

one_or_more_modifiers :=
                        | modifier
                        | modifier one_or_more_modifiers

modifier :=
           | modifier_keyword id

modifier_keyword :=
          | “At”
          | “By”
          | “From”
          | “Into”
          | “Of”
          | “To”
          | “With”

id :=
    | STRING
    | STRING “@” TIME
    | URL “@” TIME
    | URL “:” STRING “@” TIME
```

12 Use Cases (Informative)

12.1 Introduction

This Informative Chapter collects diverse Metaverse Use Cases where Users request to perform Actions on different types of Items. The goal of this Chapter is to show that the Metaverse elements of this Technical Specification do indeed support a range of representative Use Cases.

Note that, unless disclaimed otherwise, a sentence like “A student attends a lecture held by a teacher in a classroom created by a school manager” means that “a User representing a student

attends a virtual lecture in a virtual classroom Authored by a User representing a school manager and MM-Embedded at an M-Location".

12.2 Use Case Description Language

Metaverse Use Cases involve a plurality of Processes – Users, Devices, Services, Apps – performing or requested by other Processes to perform Actions on a variety of Items.
In a Use Case:

1. Processes (e.g., Users) are sequentially identified by one subscript.
2. Items Acted on by a Process are identified by the subscript of the Process performing an Action on the Item followed by a sequential number.
3. The Locations where the Actions take place are similarly identified by the subscript of the Process performing an Action at the Location followed by a sequential number.
4. If the Actions are performed at different M-Instances, all Processes, Items, and Locations are prefixed by a sequential capital letter.

For instance:

1. $User_{i}$ MM-Embeds $Persona_{i.j}$ at $M\text{-}Location_{i.k}$.
2. $User_{i}$ MU-Renders $Entity_{i.j}$ at $U\text{-}Location_{i.k}$.
3. $User_{A,i}$ MM-Sends $Object_{A.i.j}$ to $User_{B.k}$.

All Use Cases assume that Actions are performed in an M-Instance. When they are performed in the Universe, this is specifically mentioned.

The following abbreviations are used throughout:

MLoc	M-Location
SA	Spatial Attitude
ULoc	U-Location

Note: Persona(AV) is a Persona that can be audio-visually perceived.
Object(AVH) is an Object that can be audio-visual-haptically perceived.

12.3 Virtual Lecture

12.3.1 Description

A student attends a lecture held by a teacher in a classroom created by a school manager:

1. School Manager
   1.1. Authors and embeds a virtual classroom.
   1.2. Pays the teacher.
2. Teacher
   2.1. Is at home.
   2.2. Embeds a persona of theirs from home at the classroom's desk.
   2.3. Embeds and animates a 3D Object.
   2.4. Leaves the classroom.
3. Student
   3.1. Is at home.
   3.2. Pays to attend a lecture and make a copy of their Experience.
   3.3. Embeds a persona of theirs in the classroom.
   3.4. Approaches the teacher's desk to feel the 3D Object with haptic gloves.
   3.5. Stores their lecture Experience.
   3.6. Leaves the classroom and returns home.

12.3.2 MMM-Script representation

Declare: $User_1$ // School manager //
1. $Object(V)_{1.1}$ //Classroom //
2. $MLoc_{1.1}$ // Place#1 (Classroom location) //
3. $Value_{1.1}$ // Lecture consideration //

Declare: $User_2$ // Teacher //
1. $Persona(AV)_{2.1}$ // Teacher's Persona //
1. $MLoc_{2.1}$ // Teacher's home //
2. $MLoc_{2.2}$ // Place#2 (Classroom desk) //
3. $MLoc_{2.3}$ // Place#3 (Experiment Object) //
4. $Object(AVH)_{2.1}$ // Experiment Object //

Declare: $User_3$ // Student //
1. $Persona(AV)_{3.1}$ // Student's Persona //
2. $MLoc_{3.1}$ // Student's home //
3. $MLoc_{3.2}$ // Classroom seat //
4. $Value_{3.1}$ // Lecture fees //
5. $Experience_{3.1}$ // Lecture Experience //
6. $Address_{3.1}$ // Address of Experience storage //

Who	ActsOn	What	Secondary object
Manager	*Authors*	Classroom	*By* AuthorService *With* Data *At* AuthorService
	MM-Embeds	Classroom	*From* AuthorService *At* Place#1
Teacher	*Tracks*	Teacher	*At* Teacher's home *With* SA
	Tracks	Teacher	*At* Place#2 *With* SA
	MM-Disables	Teacher	*At* Teacher's home
	MM-Embeds	Exper. Object	*At* Place#3
Student	*Tracks*	Student	*At* Student's home *With* SA
	Transacts	Lecture fees	To School manager
	Tracks	Student	At Classroom bench *With* SA
	MM-Disables	Student	*At* Student's home
Teacher	*MM-Animates*	Exper. Object	*At* Place#3
Student	*MM-Sends*	Exper. Object	*To* Student
	MU-Sends	Lecture Exper.	*To* Address of Experience storage
Manager	*Transacts*	Lecture cons.	*To* Teacher
Teacher	*MM-Disables*	Teacher	*At* Place#1
	MM-Enables	Teacher	*At* Teacher's home
Student	*MM-Disables*	Student	*At* Place#2
	MM-Enables	Student	*At* Student's home

12.3.3 Actions, Items, and Data Types

Table 6 gives the list of Actions, Items, and Data Types used by the Virtual Lecture Use Case. The Table also gives the Actions implied by the Track Composite Action (MM-Embed, MM-Animate, MM-Send, MU-Render, UM-Capture, MU-Send, and Identify). The list of these Actions will not be repeated in the next tables.

Table 6 – Virtual Lecture Actions, Items, and Data Types

Actions	Items	Data Types
Author	Experience	Amount
Identify	M-Location	Coordinates
MM-Animate	Object(AVH)	Currency
MM-Disable	Object(V)	Spatial Attitude
MM-Embed	Persona(AV)	Value
MM-Send	U-Location	Orientation
MU-Render	Value	Position
MU-Send		
UM-Capture		
UM-Send		
Track		
Transact		

12.4 Virtual Meeting

12.4.1 Description

A meeting manager

1. Authors a meeting room.
2. Deploys a Virtual Meeting Secretary tasked to produce a summary of the conversations, enriched by information about participants' Personal Statuses.

A participant

1. Attends a meeting held in the room.
2. Gets a translation of sentences uttered in languages other than their own.
3. Makes a presentation using a 3D model.

12.4.2 MMM-Script representation

Declare: $User_1$ // Meeting manager //

1. $Object(V)_{1.1}$ // Meeting room //
2. $MLoc_{1.1}$ // Meeting location //
3. $Persona(AV)_{1.1}$ // Virtual Meeting Secretary //
4. $MLoc_{1.2}$ // Place#1 in room (for Virtual Meeting Secretary) //
5. $Summary_{1.1}$ // Meeting Summary //
6. $MLoc_{1.3}$ // Place#2 (for Summary display)

Declare: $User_2$ // Meeting participant #1 //

1. $Persona(AV)_{2.1}$ // Participant #1's Persona //
2. $MLoc_{2.1}$ // Participant#1's home //
3. $MLoc_{2.2}$ // Place#3 in room (for Participant#1) //
4. $Object(AVH)_{2.1}$ // Presentation //
5. $MLoc_{2.3}$ // Place#4 (For presentation display) //
6. $Event_{2.1}$ // Meeting/s recording //
7. $Address_{2.1}$ // Storage (for recording)

Declare: $Process_1$ // Animation Process //

Declare: $User_3$ // Meeting participant #2 //

1. $Persona(AV)_{3.1}$ // Participant #2's Persona //

2. $MLoc_{3.1}$ // Place#5 in room (for Participant#2) //
3. Object(A) $_{3.1}$ // Speech Object#1 (P#2) //
4. Object(A) $_{3.2}$ // Speech Object#2 (P#2) //

Who	ActsOn	What	Secondary object
Manager	MM-Embeds	Meeting room	*At* Meeting room location *With* SA
	MM-Embeds	$Persona_{1.1}$	*At* Place#1 *With* SA
	MM-Animates	$Persona_{1.1}$	
Participant#1	Tracks	$Persona_{2.1}$ (AV)	*At* Participant#1's home *With* SA
	Tracks	$Persona_{2.1}$ (AV)	*At* Place#3 *With* SA
	MM-Disables	$Persona_{2.1}$ (AV)	*From* Participant#1's home
Participant#2	Tracks	$Persona_{3.1}$ (AV)	*At* Participant#1's home *With* SA
	Tracks	$Persona_{3.1}$ (AV)	*At* Place#4 *With* SA
	MM-Disables	Participant#2	*From* Participant#2's home
Participant#1	Authenticates	Participant #2	*At* Participant#1
	Interprets	Speech Object#1	*At* Participant#1
	MM-Embeds	Presentation	*At* Place#5 *With* SA
	MM-Animates	Presentation	
Virtual Secretary	Interprets	Speech Object#2	*At* Meeting manager
	MM-Embeds	Summary	*At* Place#2 *With* SA
Manager	MM-Disables	$Persona_{1.1}$	*From* Place#1
Participant#1	MU-Sends	Recording	*To* Storage
	MM-Embeds	Participant#1	*At* Participant#1's home *With* SA
	MM-Disables	Participant#1	*From* Place#3
Participant#2	MM-Embeds	Participant#1	*At* Participant#2's home *With* SA
	MM-Disables	$Persona_{3.1}$ (AV)	*From* Place#5

12.4.3 Actions, Items, and Data Types

Table 7 gives the list of Actions, Items, and Data Types used by the Virtual Meeting Use Case. For simplicity, the Actions implied by the Track Action have not been added to the Table.

Table 7 – Virtual Meeting Actions, Items, and Data Types.

Actions	Items	Data Types
Authenticate	Event	Coordinates
Interpret	Object(AV)	Orientation
MM-Animate	Object(V)	Position
MM-Disable	Persona(AV)	Spatial Attitude
MM-Embed	Summary	
MM-Send		
Track		

12.5 Hybrid working

12.5.1 Description

A company applies mixed in-presence and remote working policy.
1. Some Workers (R-Workers) attend Company physically.
2. Some Workers (V-Workers) attend Company virtually.
3. All Workers

3.1. Are Authenticated.
3.2. Are present in the Virtual office.
3.3. Communicate by sharing AV messages (Communication of R-Workers' Personae is also mapped to the M-Environment).
3.4. Participate in Virtual meetings.

12.5.2 MMM-Script representation

Declare: $User_1$ // Company manager //
1. $Object(V)_{1.1}$ // Office //
2. $MLoc_{1.1}$ // Office Location //
3. $Persona(AV)_{1.1}$ // Office Gatekeeper //
4. $MLoc_{1.2}$ // Place#1 (for Gatekeeper) //

Declare: $Process_1$ //Animates Office Gatekeeper //

Declare: $User_2$ // R-Worker //
1. $Persona(AV)_{2.1}$ //R-Worker's Persona (R-Persona) //
2. $MLoc_{2.1}$ // Home (R-Worker) //
3. $MLoc_{2.2}$ // Place#2 (R-Worker's Office desk) //
4. $MLoc_{2.3}$ // Place#3 (in meeting room)
5. $Object(AVH)_{2.1}$ // Whiteboard //
6. $MLoc_{2.4}$ // Place#4 (for Whiteboard) //

Declare: $Process_2$ // Animates Whiteboard //

Declare: $User_3$ // V-Worker #1 //
1. $Persona(AV)_{3.1}$ // V-Worker's Persona (V-Persona) //
2. $MLoc_{3.1}$ // V-Worker's home //
3. $MLoc_{3.2}$ // Place#5 (V-Worker's desk) //
4. $Object(A)_{3.1}$ //Speech Object//
5. $MLoc_{3.2}$ // Place#6 (close to R-Worker's desk) //
6. MLoc3.5 // Place#7 (in meeting room) //

Process	Acts	What	Secondary object
Manager	MM-Embeds	Office	*At* Office Location *With* SA
	MM-Embeds	Gatekeeper	*At* Place#1 *With* SA
	MM-Animates	Gatekeeper	
$human_2$	enters company		
R-Worker	Tracks	R-Persona	*At* Place#2 *With* SA
Gatekeeper	Authenticates	R-Persona	*At* Gatekeeper
V-Worker	Tracks	V-Persona	*At* home *With* SA
	MM-Embeds	V-Persona	*At* Place#5 *With* SA
	MM-Sends	Speech Object	*To* R-Worker
	MM-Embeds	V-Persona	*At* Place#6 *With* SA
	MM-Disables	V-Persona	*From* Place#5
	MM-Embeds	V-Persona	*At* Place#7 *With* SA
	MM-Disables	V-Persona	*From* Place#5
R-Worker	MM-Embeds	R-Persona	*At* Place#3 *With* SA
	MM-Disables	R-Persona	*From* Place#2
	MM-Embeds	Whiteboard	*At* Place#4 *With* SA

	MM-Animates	Whiteboard	
	MM-Disables	R-Persona	*From* Place#3
V-Worker	MM-Embeds	V-Persona	*At* home *With* SA
	MM-Disables	V-Persona	*From* Place#7

12.5.3 Actions, Items, and Data Types

Table 8 – Hybrid Working Actions, Items, and Data Types

Actions	**Items**	**Data Types**
Authenticate	Object(V)	Coordinates
MM-Animate	M-Location	Orientation
MM-Disable	Object(A)	Position
MM-Embed	Object(AVH)	Spatial Attitude
MM-Send	Persona(AV)	
Track		

12.6 eSports Tournament

12.6.1 Description

1. Site manager
1.1. Develops a game landscape.
1.2. Makes it available to a game manager.
2. Game manager
2.1. Deploys autonomous characters.
2.2. Places virtual cameras and microphones in the landscape.
3. Captured AV from game landscape is displayed onto a dome screen and streamed online.
MMM-Script representation

12.6.2 MMM-Script representation

Declare: $User_1$ // Site Manager //
1. $Object(AVH)_{1.1}$ // Game landscape
2. $MLoc_{1.1}$ // Game Location //

Declare: $Service_1$ // Author Service //

Declare: $User_2$ // Game manager //
1. $Value_{2.1}$ // Game Location Renting Fees //
2. $Personae_{2.i}$ //Autonomous characters //
3. $M\text{-}Loc_{2.i}$ // Places in Game landscape //
4. $Scene_{2.1}$ //Game's Scene //

Declare: $User_j$ // Players //
1. $Personae_{j.1}$ //Players' characters //
2. $M\text{-}Loc_{j.1}$ // Location in Game landscape //

Declare: $Process_{2.i}$ // Animates Autonomous character //

Declare: $Service_2$ // Microphone/Camera control //

Declare: $Device_1$ //Dome screen //

Declare: $Device_k$ // Online Device of human //

Process	ActsOn	What	Secondary object
Site Mgr	Authors	Game Landscape	*By* AutService *With* Data *At* AuthorService
	MM-Embeds	Game landscape	*From* Service *At* GameLoc *With* SA
Game Mgr	Transacts	Rental Fees	*To* Site Manager
	MM-Embeds	Auton. characters	*At* Places in Game landscape *With* SA
	MM-Animates	Auton. characters	
Player	Tracks	Players' characters	*At* Places in Game landscape *With* SA
Dev. ctrl	Controls	Camera/mike	
Game Mgr	MU-Renders	Game's Scene	*At* Dome screen
			At Online devices

12.6.3 Actions, Items, and Data Types

Table 9 – eSports Tournament Actions, Items, and Data Types.

Actions	Items	Data Types
Author	Object(AVH)	Amount
MM-Animate	Persona (AVH)	Coordinates
MM-Embed	Scene(AVH)	Currency
MU-Render	M-Location	Orientation
Track	U-Location	Position
Transact	Value	Spatial Attitude

12.7 Virtual performance

12.7.1 Description

1. Impresario:
1.1. Acquires Rights to parcel.
1.2. Authors Auditorium
1.3. Embeds Auditorium on Parcel.
2. Participant
2.1. Buys a ticket for an event with the right to stay close to the performance stage for 5 minutes.
2.2. Utters a private speech to another participant.
3. Impresario:
3.1. Collects participants' preferences.
3.2. Interprets participants' mood (Participants Status).
3.3. Generates special effects based on preferences and Participants Status.

12.7.2 MMM-Script representation

Declare: $User_1$ // Impresario //
1. $Value_{1.1}$ /Payment for Land Parcel //
2. $Object(V)_{1.1}$ // Auditorium //
3. $Value_{1.2}$ // Payment for Auditorium authoring //
4. $Object(A)_{1.i}$ // SFX //
5. $M\text{-}Location_{1.i}$ // SFX Places on Auditorium //

6. Value$_{1.3}$ // Consideration for Performance //
7. Participants Status$_{1.1}$ // Status of event participants //

Deckare: Service$_1$ // Content Authoring //

Declare: Service$_2$ // Preference Collection //

Declare: User$_2$ // Performer //
1. Persona$_{2.1}$ // Performer's Persona
2. M-Loc$_{2.1}$ // Performer's home //
3. M-Loc$_{2.2}$ // Stage in Auditorium //

Declare: User$_3$ // Participant #1//
1. Persona$_{3.1}$ /Participant#1's Persona //
2. M-Loc$_{3.1}$ // Home //
3. M-Loc$_{3.2}$ // Seat#1 in Auditorium //
4. Scene$_{3.1}$ // Scene of Stage //
5. Object(A) $_{3.1}$ // Audio Object ///
6. Value$_{3.1}$ // Ticket#1 to event //

User$_4$ // Participant#2//
1. Persona$_{4.1}$ /Participant#2's Persona //
2. M-Loc$_{4.1}$ // Participant#2's Home //
3. M-Loc$_{4.2}$ // Seat#2 in Auditorium //
4. Value$_{4.1}$ // Ticket#2 to event //

Declare: User$_5$ // Land Parcel owner //

Who	**ActsOn**	**What**	**Secondary object**
Impresario	Transacts	Parcel payment	To Parcel Service
	Authors	Auditorium	*By* AutService *With* Data *At* AutService
	Transacts	Authoring Fees	*To* Authoring Service
	MM-Embeds	Auditorium	*From* AutService *At* Parcel *With* SA
	Calls	Preference Service	*At* Preference Service
Performer	Tracks	Performer's Persona	*At* Performer's home *With* SA
	Tracks	Performer's Persona	*At* Stage *With* SA
	MM-Disables	Performer's Persona	*From* home
Participant#1	Tracks	P#1's Persona	*At* home *With* SA
	Transacts	Event's Ticket#1	*At* Participant#1
	Tracks	P#1's Persona	*At* Seat#1 *With* SA
	MM-Disables	P#1's Persona	*From* home
Participant#2	Tracks	P#2's Persona	*At* home *With* SA
	Transacts	Event's Ticket#2	*At* Participant#2
	Embeds	P#2's Persona	*At* Seat#1 *With* SA
	MM-Disables	P#2's Persona $_1$	*From* home
Participant#1	MM-Sends	Audio Object	*To* Participant#2
	Calls	Preference Service	Preference Service
	MM-Sends	Scene$_{3.1}$	*To* Participant#1
Impresario	Calls	Preference Service	*At* Impresario

	Interprets	Participants Status$_{1.1}$	*At* Impresario
	MM-Embeds	SFXs	*At* Auditorium Places *With* SA
	Transacts	Performance Consid.	*To* Performer
Performer	MM-Embeds	Performer's Persona	*At* Home *With* SA
	MM-Disables	Performer's Persona	*From* Stage
Participant#1	MM-Embeds	P#1's Persona	*At* Home *With* SA
	MM-Disables	Persona(AV)$_{3.1}$	*From* Seat#1
Participant#2	MM-Embeds	Persona(AV)$_{4.1}$	*At* Home *With* SA
	MM-Disables	Persona(AV)$_{4.1}$	*From* Seat#2

12.7.3 Actions, Items, and Data Types

Table 10 – Virtual Event Actions, Items, and Data Types.

Actions	**Items**	**Data Types**
Author	Object(A)	Amount
Interpret	Object(AV)	Coordinates
MM-Disable	Persona(AV)	Currency
MM-Embed	M-Location	Orientation
MM-Send	Value	Participants Status
Track		Position
Transact		Spatial Attitude

12.8 AR Tourist Guide

12.8.1 Description

In this Use Case human$_3$ (AR Tourist Guide Service Provider) engages the following humans:

1. human$_1$ to cause their User$_1$ to buy a virtual parcel and develop a virtual landscape suitable for a tourist application.
2. human$_2$ to cause their User$_2$ to develop scenes and autonomous agents for the different places of the landscape.
3. human$_4$ to create an app that alerts the holder of a smart phone running the app.
4. human$_5$ holding a smart phone with the app to perceive Entities and interact with Personae MM-Embedded at M-Locations and MM-Animated.

12.8.2 MMM-Script representation

Declare: User$_1$ // Virtual Land developer//

1. MLoc$_{1.1}$ // Land Parcel //
2. Object(V)$_{1.1}$ // Landscape //
3. Value$_{1.1}$ // Payment for Land Parcel //

Declare: Service$_1$ // Authoring Service //

Declare: User$_2$ // Object developer //

1. Object(AV)$_{2.i}$ // Objects for landscape //
2. MLoc$_{2.i}$ // correspondent to U-Locations //
3. Value$_{2.1}$ // Payment for Objects(AV)$_{2.i}$ //

Declare: User$_3$ // Tourist application developer //

1. Persona$_{3.k}$ // Persona to be MM-Animated //

2. $MLoc_{3.k}$ // correspondent to U-Locations //

Declare: $human_4$ // Software developer //
1. Map // ULoc-MLoc map for mobile app //
2. $Value_{4.1}$ // Payment for Map and App//

Declare: $human_5$ / human holding Device running $human_4$'s App //

Declare: $Device_1$ / Held by $human_5$ /
1. $ULoc_{5.1}$ //

Declare: App_1 //Installed on $Device_1$ //
1. $Message_{5.1}$ // From App_1 to $Device_1$ //

Declare: $User_6$ // Land Parcel Rights holder //

Who	ActsOn	What	Secondary object
$User_1$	Transacts	$Value_{1.1}$	*To* Parcel Rights Holder
	Authors	Tourist Landscape	*At* Parcel Rights Holder
	Embeds	$Object(V)_{1.1}$	*At* Parcel *With* SA
	Transacts	Payment for Parcel	*To* Object Developer
$User_2$	Authors	Objects for landscapes	*At* Authoring Service
	Embeds	Objects for landscapes	*From* Service *At* Landscape *With* SA
	$Transacts_{2.1}$	Payment for App	*To* Tourist application developer
$human_4$	develops	MLoc & ULoc Map	
	develops	App	
	sells	Map and App	*To* $human_3$
App devel.	MM-Embeds	MLoc Personae	At U-Loc correspondance *With* SA
	MM-Animates	MLoc Personae	
$human_5$	comes		*To* U-Location
App	MM-Sends	Message	*To* Device
Device	MM-Sends	Message	*To* App developer
App devel.	MM-Animates	MLoc Persona	*At* M-Location
	MM-Animates	MLoc Persona	
	MU-Renders	Animated Persona	*At* U-Location

12.8.3 Actions, Items, and Data Types

Table 11 – AR Tourist Guide Actions, Items, and Data Types.

Actions	Items	Data Types
Author	Object(AV)	Amount
Author	Object(V)	Coordinates
MM-Animate	Map	Currency
MM-Animate	Message	Orientation
MM-Embed	M-Location	Position
MM-Send	Persona	Spatial Attitude
MU-Render	Service	
MM-Send	U-Location	

Transact	Value	

12.9 Virtual Dance

12.9.1 Description

This Use Cases envisages that:

1. Dance teacher places their virtual secretary Persona animated by an autonomous agent in the dance school.
2. Student #1:
   2.1. Shows up at school.
   2.2. Greets the secretary.
3. Virtual secretary reciprocates greetings.
4. Dance teacher:
   4.1. Places a haptic Persona of theirs in the dance school.
   4.2. Dances with student #1.
5. Student #2:
   5.1. Is at home.
   5.2. Shows up at school.
6. Teacher:
   6.1. Places their haptic Persona close to student #2.
   6.2. Places (replaces) another haptic Persona of theirs close to student #1.
   6.3. Animates the new haptic Persona with autonomous agent dancing with student #1.
   6.4. Dances with student #2.

12.9.2 MMM-Script representation

Declare: $User_1$ // Dance teacher //

1. $Persona(AVH)_{1.1}$ // Dancing persona#1 //
2. $MLoc_{1.1}$ // Place#1 (Teacher's Office) //
3. $Persona(AVH)_{1.2}$ // School Secretary //
4. $MLoc_{1.2}$ // Place#2 (Dancing School //
5. $Persona(AVH)_{1.3}$ // Dancing persona#2 //
6. $MLoc_{1.3}$ // Place#3 (dancing area) //
7. $Object(A)_{1.1}$ // Speech Object#2 (Greetings) //
8. /$MLoc_{1.4}$ // Place#4 (dancing area) //

Declare $User_2$ // Dance student #1 //

1. $Persona(AVH)_{2.1}$ //Student's Persona/
2. $MLoc_{2.1}$ // Student#1's home //
3. //$MLoc_{2.1}$ // Place#5 in dancing area //

$User_3$ // Dance Student #2 //

1. $Persona(AVH)_{3.1}$ // Student's Persona //
2. $MLoc_{3.1}$ // Dance Student#2's home //
3. //$MLoc_{3.1}$ // Place#6 in dancing area //

Who	ActsOn	What	Secondary object
Teacher	Tracks	Persona#1	*At* Home *With* SA
	Tracks	Persona#1	*At* Place#1 *With* SA
	MM-Embeds	Persona#2	*At* Place#2 *With* SA
	MM-Animates	Persona#2	

Student#1	Tracks	Sudent#1's Persona	*At* Student#1's Home *With* SA
	MM-Embeds	Student#1's Persona	*At* Place#5 *With* SA
	MM-Disables	Student#1's Persona	*From* Home
Teacher	Tracks	Teacher's Persona#1	*At* Place#3 *With* SA
Student#2	Tracks	Student#2's Persona	*At* Student#2's Home *With* SA
	Tracks	Student#2's Persona	*At* Place#6 *With* SA
	MM-Disables	Student#2's Persona	*From* Student#2's Home
Teacher)	Tracks	Teacher's Persona#1	*At* Place#4 *With* SA
	MM-Disables	Teacher's Persona#1	*From* Place#3
	MM-Embeds	Teacher's Persona#3	*At* Place#3 *With* SA

12.9.3 Actions, Items, and Data Types

Table 12 – Virtual Dance Actions, Items, and Data Types.

Actions	**Items**	**Data Types**
MM-Animate	M-Location	Orientation
MM-Disable	Object (A)	Position
MM-Embed	Persona (AV)	Spatial Attitude
MM-Send	Persona (AVH)	
Track		

12.10 Virtual Car Showroom

12.10.1Description

This Use Cases envisages that:

1. A car dealer MM-Embeds an MM-Animated Persona in the car showroom (as attendant).
2. A customer:
   2.1. MM-Embeds its Persona in the car showroom.
   2.2. Greets the showroom attendant.
3. The Showroom attendant reciprocates the greeting.
4. The dealer:
   4.1. UM-Animates the attendant.
   4.2. Converses with the customer.
   4.3. Embeds a 3D AVH model of a car.
5. The customer
   5.1. Has a virtual test drive.
   5.2. Buys the car.
   5.3. Returns home.

12.10.2MMM-Script representation

Declare: $User_1$ // Car dealer //

1. $Persona(AV)_{1.1}$ //Car dealer //
2. $MLoc_{1.1}$ // Place#1 (Car dealer's Office) //
3. $Persona(AV)_{1.2}$ //Showroom attendant //
4. $MLoc_{1.2}$ // Place#2 (in Showroom) //
5. $Object(A)_{1.1}$ //Greetings //
6. $M\text{-}Loc_{1.3}$ // Place#3 (in Showroom) //
7. $Model(AVH)_{1.1}$ // 3D Model of car //

Declare: $User_2$ // Customer //
1. Persona(AV)$_{2.1}$ // Customer's Persona //
2. M-Loc$_{2.1}$ //Customer's home //
3. M-Loc$_{2.2}$ // Place#4 in showroom //
4. Persona(AVH)$_{2.1}$ / $User_2$'s Persona for test driving //
5. M-Loc$_{2.3}$ // Place#5 (in virtual car)
6. Value$_{2.1}$ // Payment for car //
7. U-Loc$_{2.1}$ // U-Place#1 (U-Location of Customer) //

Who	ActsOn	What	Secondary object
Car dealer	Tracks	Dealer's Persona#1	*At* Place#1 *With* SA
	MM-Embeds	Dealer's Persona#2	*At* Place#2 *With* SA
	MM-Animates	Dealer's Persona#2	
Customer	Tracks	Customer's Persona	*At* Home *With* SA
	Tracks	Customer's Persona	*At* Place#4 *With* SA
	MM-Disables	Customer's Persona	*From* Home
Car dealer	MM-Sends	Speech Object	*To* Customer
	MM-Embeds	Dealer's Persona	*At* Place#3 *With* SA
	MM-Embeds	Car Model	*At* Place#5 *With* SA
	MM-Animates	Car Model	
Customer	Tracks	Customer's Persona	*At* Place#5 *With* SA
	MM-Disables	Customer's Persona	*From* Place#4
	UM-Renders	Car Model	*At* U-Place#1
	Transacts	Value$_{2.1}$	*To* Dealer
	MM-Disables	Customer's Persona	*From* Place#5
	Tracks	Customer's Persona	*At* Home *With* SA

12.10.3Actions, Items, and Data Types

Table 13 – Virtual Car Showroom Actions, Items, and Data Types.

Actions	Items	Data Types
MM-Animate	Object (A)	Amount
MM-Disable	Persona(AV)	Currency
MM-Embed	Persona(AVH)	Orientation
MM-Send	Scene (AVH)	Position
Track	Value	Spatial Attitude
Transacts		
UM-Animate		

12.11 Drive a Connected Autonomous Vehicle

12.11.1Description

This Use Case considers some of the steps made by a human having rights to an implementation of Technical Specification: Connected Autonomous Vehicle (MPAI-CAV) – Architecture [6]. Chapter 7 of Annex 1 - MPAI Basic provides a high-level summary of the specification.

A CAV rights holder *Registers* with the CAV to access the CAV-created M-Instance by providing:
1. The requested subset of their Personal profile.

2. Two User Processes required to operate a CAV:
   2.1. $User_1$ to operate the Human-CAV Interaction Subsystem.
   2.2. $User_2$ to operate the Autonomous Motion Subsystem.
3. $User_1$'s Personae.
4. WalletIDs.

For simplicity, the Use Case assumes that there are two CAVs: CAV_A and CAV_B and that the CAV_A rights holder ($User_{A.1}$) wants to see the CAV_B Environment in the CAV_B M-Instance:

1. $User_{A.1}$
   1.1. Authenticates the human's voice.
   1.2. Interprets driving instructions from human.
   1.3. Communicates driving instructions to $User_{A.2}$.
2. $User_{A.2}$
   2.1. Gets information about CAV_A position.
   2.2. Gets travel options from Route Planner.
   2.3. Communicates travel options to $User_{A.1}$.
3. $User_{A.1}$
   3.1. Produces Speech Object with travel options.
4. human utters selected option to $User_{A.1}$.
5. $User_{A.1}$
   5.1. Interprets driving instructions from human.
   5.2. Communicates driving instructions to $User_{A.2}$.
6. $User_{A.2}$
   6.1. Gets the Basic Environment Representation from its ESS.
   6.2. Authenticates its peer $User_{B.2}$.
   6.3. Gets elements of the Basic Environment Representation from $User_{B.2}$.
   6.4. Produces Full Environment Representation.
   6.5. Sends a command to the Ego CAV's Motion Actuation Subsystem.
7. $User_{A.1}$
   7.1. Authenticates its peer $User_{A.2}$.
   7.2. Watches CAV_B's Environment.

12.11.2MMM-Script representation

Declare: $human_A$ // CAV_A's rights holder //

Declare: $User_{A.1}$ // CAV_A's HCI //

1. $ULoc_{A.1.1}$ // Place where CAV_A is located //
2. $MLoc_{A.1.1}$ // M-Location corresponding to $ULoc_{A.1.1}$ //
3. $scene_{A.1.1}$ // Scene at $ULoc_{A.1.1}$ //
4. $DataMdata_{A.1.1}$ // Data and Metadata of scene captured by $Device_1$ //
5. $Scene_{A.1.1}$ // Scene of $MLoc_{A.1.1}$ //
6. $Object(A)_{A.1.1}$ // Speech Object #1 //
7. $HCI\text{-}AMSCommand_{A.1.1}$ // Travel request to $User_2$ //
8. $HCI\text{-}AMSCommand_{A.1.2}$ // Travel request to $User_2$ //
9. $Object(A)_{A.1.2}$ //Speech Object #2 //
10. $HCI\text{-}AMSCommand_{A.1.2}$/Travel selection to $User_2$ //

Declare: $Device_1$ // Audiovisual sensor and actuator //

Declare: Route Planner$_{A.1}$ // CAV Process //

Declare: Path Planner$_{A.1}$ // CAV Process //

Declare: Motion Planner$_{A.1}$ // CAV Process //

Declare: Obstacle Avoider$_{A.1}$ // CAV Process //

Declare: Command Issuer$_{A.1}$ // CAV Process //

Declare: User$_{A.2}$ // CAV$_A$'s AMS //
1. AMS-HCIResponse$_{A.2.1}$ // Route selection //
2. Scene$_{A.2.1}$ // CAV$_A$'s Environment //

Declare: User$_{B.2}$ // CAV$_B$'s AMS //
1. Scene$_{B.2.1}$ // CAV$_B$'s scene in ULoc$_{A.1.1}$ //

Declare: User$_{B.1}$ // CAV$_B$'s HCI //

Who	ActsOn	What	Secondary object
human$_A$	Registers		*With* CAV$_A$
User$_{A.1}$	UM-Captures	scene	*At* Device
	UM-Sends	DataMdata	*From* Device *To* User$_{A.1}$
	Identifies	Scene$_{A.1}$	*At* User$_{A.1}$
	Authenticates	Object(AV)$_{A.1.1}$	*At* User$_{A.1}$
	Interprets	Object$_{A.1.1}$(A)	*At* User$_{A.1}$
	MM-Sends	HCI-AMSCmd$_{A.1.1}$	*To* User$_{A.2}$
User$_{A.2}$	MM-Sends	ESS's Scene$_{A.2.1}$	*To* Route Planner
	MM-Sends	AMS-HCIResp$_{A.2.1}$	*To* User$_{A.1}$
User$_{A.1}$	Interprets	Object(A)$_{A.1.3}$	*At* User$_{A.1}$
	MM-Sends	HCI-AMSCmd$_{A.1.2}$	*To* User$_{A.2}$
User$_{A.2}$	Authenticates	User$_{B.2}$	*At* User$_{A.2}$
	MM-Sends	ESS's Scene$_{A.2.2}$	*To* User$_{A.2}$
	MM-Sends	Path$_{A2.1}$	*To* Motion Planner
Motion Planner	MM-Sends	Trajectory$_{A.2.1}$	*To* Obstacle Avoider
Obstacle Avoider	MM-Sends	Trajectory$_{A.2.1}$	*To* Command Issuer
Command Issuer	MM-Sends	AMS-MASCmd$_{A.2.1}$	*To* MotionActuationSubsys
MAS	MM-Sends	MAS-AMS Resp$_{A.2.1}$	*To* Command Issuer
User$_{A.1}$	Authenticates	User$_{B.2}$	*At* User$_{A.1}$
	MM-Sends	Scene$_{B.2.1}$	To User$_{A.1}$

12.11.3Actions, Items, and Data Types

Note: The MPAI-CAV specific Items are included.

Table 14 – Drive a Connected Autonomous Vehicle Actions, Items, and Data Types.

Action	Item	Data Types
Authenticate	AMS-HCIResponse	Spatial Attitude
Interpret	AMS-MASCommand	Coordinates

MM-Embed	Environment Representation	Orientation
MM-Send	HCI-AMSCommand	Position
MU-Render	MAS-AMSResponse	
Register	M-Location	
Request	Object (A)	
Track	Path	
UM-Render	Persona	
	Route	
	Scene	
	Trajectory	

13 Functional Profiles

A standard that only specifies Functionalities may create a significant burden to implementations serving very different needs because the technologies supporting some Functionalities may be costly but seldom used. Profiles defines groups of Functionalities that serve specific application areas while providing a level of Interoperability.

In the following Functional Profile is shortened to Profile.

Table 15 lists the currently identified Actions, Items and Data Types supported by a Profile. Cells with text in italic indicate a classification of Items, Action, and Data Types.

Table 15 – Classified Actions, Items, and Data Types

Items	**Actions**	**Data Types**
General Items	*General Actions*	*For location and time*
M-Instance	Register	Address
M-Capabilities	Change	Coordinates
M-Environment	Hide	Map
Identifier	Authenticate	Orientation
Rules	Identify	Point of View
Rights	Modify	Position
Program	Validate	Spatial Attitude
Contract	Request	Time
Human and User-related Items	Respond	*For Transactions*
Account	Execute	Amount
Activity Data	*Call a Service*	Currency
Personal Profile	Author	*For internal state*
Social Graph	Discover	Cognitive State
Personal Data	Inform	Emotion
Items for Process Interaction	Interpret	Social Attitude
Message	Post	Personal Status
P-Capabilities	Transact	
Items for Service access	Convert	
AuthenticateIn	Resolve	
AuthenticateOut	*Manage Entities (MM)*	
DiscoverIn	MM-Add	
DiscoverOut	MM-Animate	
InformIn	MM-Disable	

InformOut	MM-Embed	
InterpretIn	MM-Enable	
InterpretOut	MM-Send	
Finance-related Items	*Manage Entities (MU)*	
Asset	MU-Actuate	
Ledger	MU-Render	
Provenance	MU-Send	
Transaction	Track	
Value	*Manage Entities (UM)*	
Wallet	UM-Animate	
Perception-related Items	UM-Capture	
Event	UM-Render	
Experience	UM-Send	
Interaction		
Map		
Model		
Object		
Scene		
Stream		
Summary		
Space-related Items		
M-Location		
U-Location		

13.1 Profile structure

The current features MPAI Metaverse Model – Architecture Profiles are:

1. Identified Profiles are Baseline, Management, Finance, and High.
2. The High Profile includes the Management Profile that includes the Baseline and Finance Profiles.
3. The Profile structure includes hierarchical Profiles and one independent Profile.
4. The Baseline, Management, and High Profiles have Levels, currently identified as: Audio only, Audio-Visual, and Audio-Visual-Haptic. The Finance Profile does not have Levels.

This is depicted in Figure 3. The next Sections provide additional details.

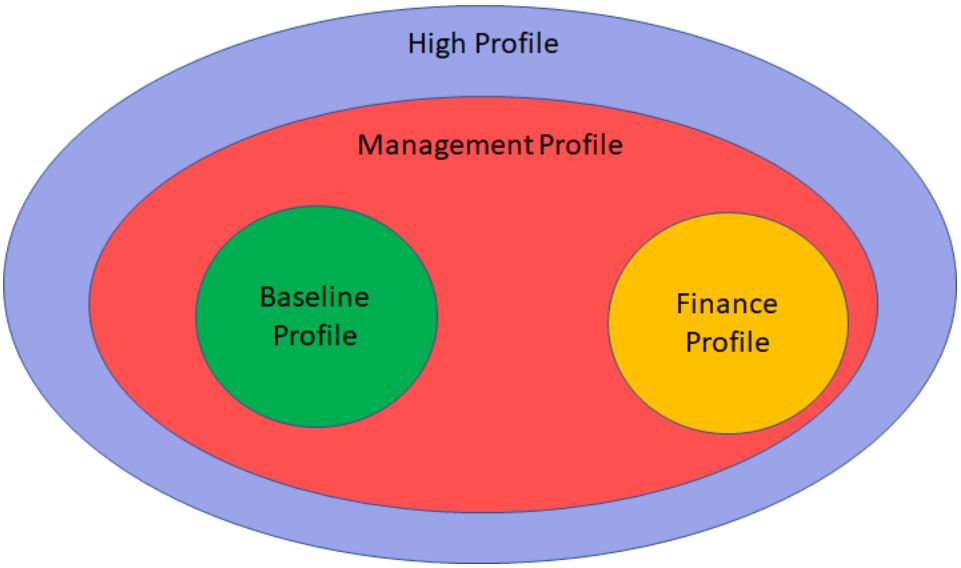


Figure 3 - The currently identified Functional Profiles

Each Profile allocates the supported Actions, Items, and Data Types. While the identified four Profiles serve well the needs conveyed by the identified Functionalities, the consideration of more Functionalities in the future may lead to an increased number of Profiles and potentially Levels.

13.2 Baseline Profile

The Baseline Profile enables a human equipped with a Device supporting the Baseline Profile to allow their Users to perform the functions of Table 16. Currently, this Profile has the following Levels: Audio only; Audio-Visual; and Audio-Visual-Haptic.

Table 16 – Functions, Actions, and Items of the Baseline Profile

Functions	Action	Items
Read Items and Data	UM-Send	Item
Identify Item	Identify	Item, Identifier
Author Entity	Author	Item
Place Entity at an MLoc without perception	MM-Add	Entity, MLoc
Make an MM-Added Entity perceptible	MM-Enable	Entity
Place an Entity 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 Entity at MLoc	MU-Render	Entity, MLoc, ULoc
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

Roughly speaking, this Profile supports applications in a basic form for e.g., lecture, meeting, hang-out.

Table 17 lists the Actions, Entities, and Data Types of the Baseline Functionality Profile.

Table 17 – Actions, Entities, and Data Types of the Baseline Profile

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

13.3 Finance Profile

The Finance Profile enables a human equipped with a Device supporting the Baseline Profile to allow their Users to perform the functions of Table 18. The Finance Profile enables a User to Post Assets and make Transactions. As depicted in *Figure 4*, this Profile is independent of the Basic Profile. The Financial Profile shares some technologies with the Baseline Profile, but the Finance Profile introduces technologies and does not support other Basic Profile technologies. Currently, this Profile does not have Levels.

Table 18 – Functions, Actions, and Items of the Finance Profile

Functions	Action	Items
Register	Register	M-Environment, Account, Activity Data, Personal Profile, Rules, Social Graph
Check that an Item is what it says it is	Authenticate	Item
Make Item inaccessible	Hide	Item
Modify Item into Asset	Modify	Item
Submit Asset to marketplace	Post	Asset
Make a Transaction of an Asset	Transact	Asset, Ledger, Provenance, Rights, Transactions, Value, Wallet
Discover Assets	Discover	DiscoverIn, DiscoverOut
Get information on Asset, User	Inform	InformIn, InformOut
Change User Rights	Change	Rights

Table 19 lists the Actions, Items, and Data Types of the Finance Profile.

Table 19 – Actions, Items, and Data Types of the Finance Profile

Actions	Authenticate	Author	Change	Discover
	Hide	Identify	Inform	MM-Add
	MM-Disable	MM-Embed	MM-Enable	MM-Send
	Modify	MU-Actuate	MU-Render	MU-Send
	Post	Register	Transact	UM-Capture
	UM-Render	UM-Send		
Items	Account	Activity Data	Asset	Identifier
	Ledger	Map	M-Environment	Message

	M-Instance	M-Location	Model	Object
	Personal Profile	Provenance	AuthenticateIn	AuthenticateOut
	DiscoverIn	DiscoverOut	InformIn	InformOut
	Rights	Rules	Scene	Social Graph
	Stream	Transaction	U-Location	Personal Data
	Value	Wallet		
Data Types	Address	Amount	Coordinates	Currency
	Orientation	Position	Spatial Attitude	Time

13.4 Management Profile

The Management Profile enables a controlled ecosystem by supporting all Actions, Items, and Data Types of the Baseline and the Finance Profiles in addition to some others of its own. It enables a human equipped with a Device supporting the Management Profile to allow their Users to perform the functions of Table 20. As depicted in *Figure 4* the Management Profile is a superset of the Baseline and Finance Profiles. Currently, this Profile has the following Levels: Audio; Audio-Visual; and Audio-Visual-Haptic.

Table 20 – Functions, Actions, and Items of the Management Profile

Functions	Action	Items
Register with an M-Environment	(Register)	M-Environment
Make Item inaccessible	Hide	Item
Animate Model with an autonomous Process	MM-Animate	Model
App triggers perception of Entities	UM-Send	Map, Message
Request interpretation of Item	Interpret	InterpretIn, InterpretOut
Save an Experience of an Event	MU-Export	Interaction, Experience, Event
Convert formats	Convert	

Table 21 lists the Actions, Entities, and Data Types required by the Management Profile.

Table 21 –Actions, Item, and Data Types of Management Profile

Actions	Authenticate	Author	Change	Discover
	Hide	Identify	Inform	Interpret
	MM-Add	MM-Animate	MM-Disable	MM-Embed
	MM-Enable	MM-Send	Modify	MU-Actuate
	MU-Render	MU-Send	Post	Register
	Track	Transact	UM-Animate	UM-Capture
	UM-Render	UM-Send		
Items	Account	Activity Data	Asset	AuthenticateIn
	AuthenticateOut	DiscoverIn	DiscoverOut	Event
	Experience	Identifier	InformIn	InformOut
	Interaction	InterpretIn	InterpretOut	Ledger
	Map	M-Environment	Message	M-Instance
	M-Location	Model	Object	Personal Profile
	Provenance	Rights	Rules	Scene
	Social Graph	Stream	Summary	Transaction
	U-Location	Value	Wallet	
Data Types	Address	Amount	Cognitive State	Coordinates

	Currency	Emotion	Map	Orientation
	Personal Status	Point of View	Position	Social Attitude
	Spatial Attitude	Time		

13.5 High Profile

This Profile includes all other Profiles. Table 22 gives the list of Actions, Items and Data Types not included in the Management Profile. Currently, this Profile has the following Levels: Audio; Audio-Visual; and Audio-Visual-Haptic.

Table 22 - Actions, Items, and Data Types of High Profile not in the Management Profile

Actions	Convert	Execute	Resolve	Validate
Item	Contract	M-Capabilities	P-Capabilities	Program
Data Types				

Annex 1 - MPAI Basics (Informative)

1 General

In recent years, Artificial Intelligence (AI) and related technologies have been introduced in a broad range of applications affecting the life of millions of people and are expected to do so much more in the future. As digital media standards have positively influenced industry and billions of people, so AI-based data coding standards are expected to have a similar positive impact. In addition, some AI technologies may carry inherent risks, e.g., in terms of bias toward some classes of users or application domains making the need for standardisation more important and urgent than ever.

The above considerations have prompted the establishment of the international, unaffiliated, not-for-profit Moving Picture, Audio and Data Coding by Artificial Intelligence (MPAI) organisation with the mission to develop *AI-enabled data coding standards* to enable the development of AI-based products, applications, and services.

2 Governance of the MPAI Ecosystem

The technical foundations of the MPAI Ecosystem are currently provided by the Governance of the MPAI Ecosystem [4] developed and maintained by MPAI:

1. Technical Specification.
2. Reference Software Specification.
3. Conformance Testing Specification.
4. Performance Assessment Specification.
5. Technical Report

MPAI published two Technical Reports [10, 11] in the preliminary phases of the MPAI Metaverse Model project. An MPAI Standard is a collection of a variable number of the 5 document types.

Figure 4 depicts the MPAI ecosystem operation for conforming MPAI implementations.

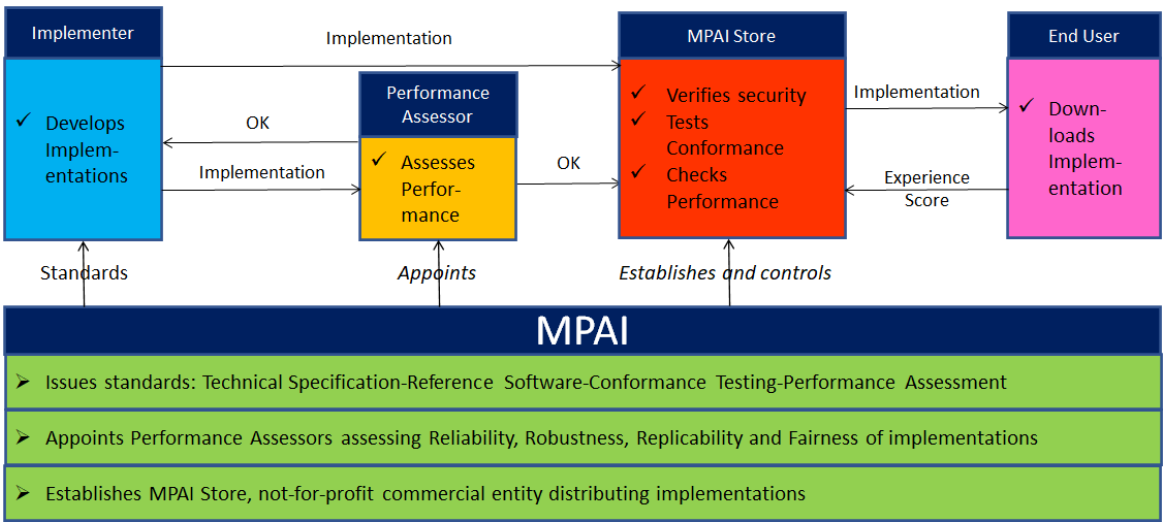


Figure 4 – The MPAI ecosystem operation

Technical Specification: Governance of the MPAI Ecosystem [1] identifies the following roles in the MPAI Ecosystem:

Table 23 - Roles in the MPAI Ecosystem

MPAI	Publishes Standards. Establishes the not-for-profit MPAI Store. Appoints Performance Assessors.
Implementers	Submit Implementations to Performance Assessors. Submit Implementations to the MPAI Store.
Performance Assessors	Inform Implementation submitters and the MPAI Store if Implementation Performance is acceptable.
MPAI Store	Assign unique ImplementerIDs (IID) to Implementers in its capacity as ImplementerID Registration Authority (IIDRA)[2]. Verifies security and Tests Conformance of Implementations.
Users	Download Implementations and report their experience to the MPAI Store.

3 AI Framework

MPAI 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 5* depicts the AI Framework.

MPAI Application Standards normatively specify the Syntax and Semantics of the input and output data and the Function of the AIW and the AIMs, and the Connections between and among the AIMs of an AIW.

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.

[2] At the time of publication of this Technical Report, the MPAI Store was assigned as the IIDRA.

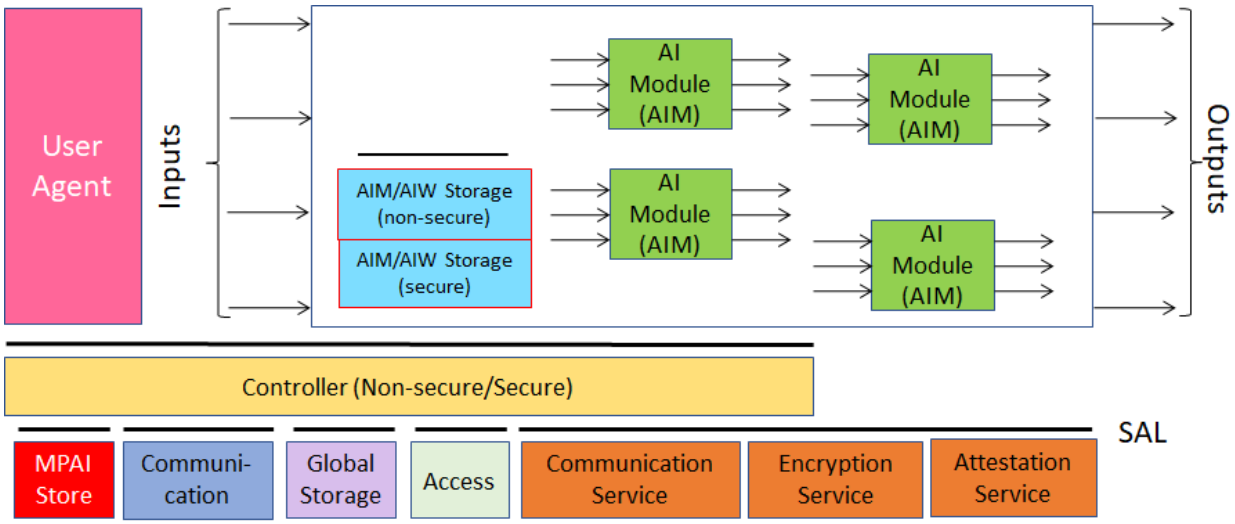


Figure 5 - The AI Framework (MPAI-AIF) V2 Reference Model

An AIW is defined by its Function and input/output Data and by its AIM topology. Likewise, an AIM is defined by its Function and input/output Data. MPAI standards are silent on the technology used to implement the AIM which may be based on AI or data processing, and implemented in software, hardware or hybrid software and hardware technologies.

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.

4 Audio-Visual Scene Description

The ability to describe (i.e., digitally represent) an audio-visual scene is a key requirement of several MPAI Technical Specifications and Use Cases. MPAI has developed Technical Specification: Context-based Audio Enhancement (MPAI-CAE) [6] that includes Audio Scene Descriptors and uses a subset of Graphics Language Transmission Format (glTF) [12] to describe a visual scene.

4.1 Audio Scene Descriptors

Audio Scene Description is a Composite AI Module (AIM) specified by Technical Specification: Context-based Audio Enhancement (MPAI-CAE) [6]. The position of an Audio Object is defined by Azimuth, Elevation, Distance, and Distance Flag.

The Audio Scene Description Composite AIM and its AIMs are depicted in Figure 10.

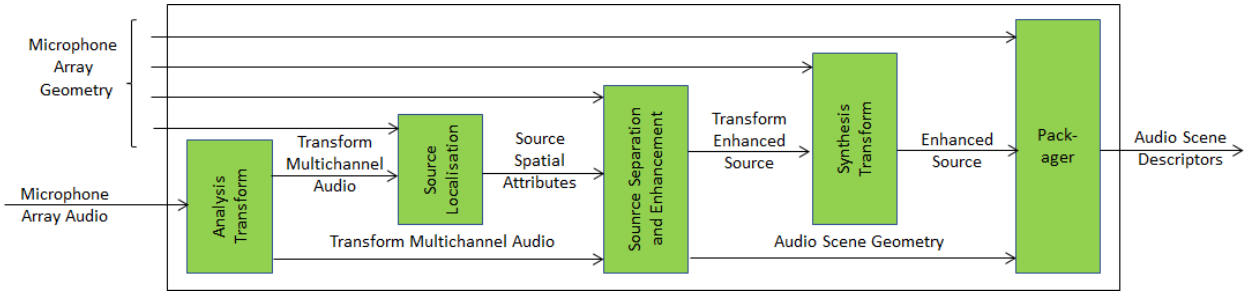


Figure 6 - The Audio Scene Description Composite AIM

5 Personal Status

5.1 General

Personal Status is the set of internal characteristics of a human and a machine making a conversation. Technical Specification: Multimodal Conversation (MPAI-MMC) [8] identifies three Factors of the internal state:

1. *Cognitive State* is a typically rational result from the interaction of a human/avatar with the Environment (e.g., "Confused", "Dubious", "Convinced").
2. *Emotion* is typically a less rational result from the interaction of a human/avatar with the Environment (e.g., "Angry", "Sad", "Determined").
3. *Social Attitude* is the stance taken by a human/avatar who has an Emotional and a Cognitive State (e.g., "Respectful", "Confrontational", "Soothing").

The Personal Status of a human can be displayed in one of the following Modalities: *Text*, *Speech*, *Face*, or *Gesture*. More Modalities are possible, e.g., the body itself as in body language, dance, song, etc. The Personal Status may be shown only by one of the four Modalities or by two, three or all four simultaneously.

5.2 Personal Status Extraction

Personal Status Extraction (PSE) is a composite AIM that analyses the Personal Status conveyed by Text, Speech, Face, and Gesture – of a human or an avatar – and provides an estimate of the Personal Status in three steps:

1. *Data Capture* (e.g., characters and words, a digitised speech segment, the digital video containing the hand of a person, etc.).
2. *Descriptor Extraction* (e.g., pitch and intonation of the speech segment, thumb of the hand raised, the right eye winking, etc.).
3. *Personal Status Interpretation* (i.e., at least one of Emotion, Cognitive State, and Attitude).

Figure 7 depicts the Personal Status estimation process:

1. Descriptors are extracted from Text, Speech, Face Object, and Body Object. Depending on the value of Selection, Descriptors can be provided by an AIM upstream.
2. Descriptors are interpreted and the specific indicators of the Personal Status in the Text, Speech, Face, and Gesture Modalities are derived.
3. Personal Status is obtained by combining the estimates of different Modalities of the Personal Status.

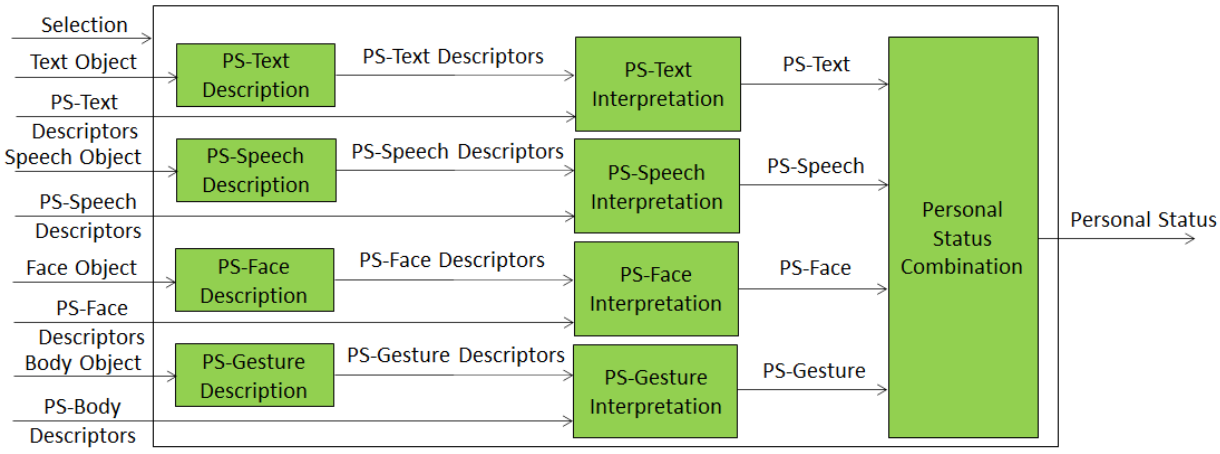


Figure 7 – Reference Model of Personal Status Extraction

An implementation can combine, e.g., the PS-Gesture Description and PS-Gesture Interpretation AIMs into one AIM, and directly provide PS-Gesture from a Body Object without exposing PS-Gesture Descriptors.

5.3 Personal Status Display

A Personal Status Display (PSD) is a Composite AIM receiving Text and Personal Status and generating an avatar producing Text and uttering Speech with the intended Personal Status while the avatar's Face and Gesture show the intended Personal Status. Instead of a ready-to-render avatar, the output can be provided as Compressed Avatar Descriptors. The Personal Status driving the avatar can be extracted from a human or can be synthetically generated by a machine as a result of its conversation with a human or another avatar. Reference Architecture.

Figure 8 represents the AIMs required to implement Personal Status Display.

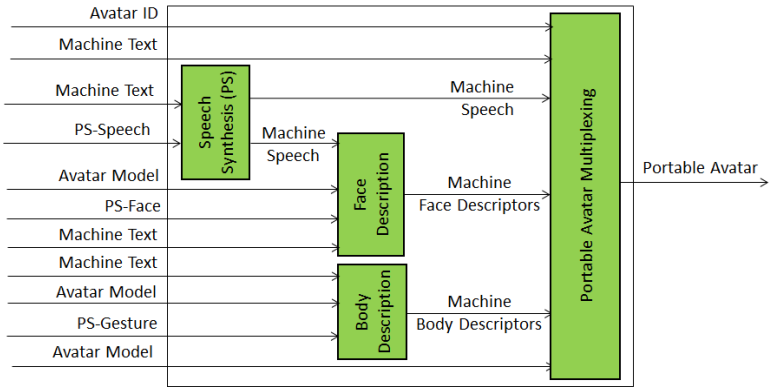


Figure 8 – Reference Model of Personal Status Display

The Personal Status Display operates as follows:

1. Avatar ID is the ID of the Portable Avatar.
2. Machine Text is synthesised as Speech using the Personal Status provided by PS-Speech.
3. Machine Speech and PS-Face are used to produce the Machine Face Descriptors.
4. PS-Gesture and Text are used for Machine Body Descriptors using the Avatar Model.
5. Portable Avatar Multiplexing produces the Portable Avatar.

6 Human-Machine dialogue

Figure 9 depicts the model of the MPAI Personal-Status-based human-machine dialogue.

Audio Scene Description and Visual Scene Description are two front-end AIMs. The former produces 1) Physical Objects, Face and Body Descriptors of the humans, and Visual Scene Geometry; the latter produces Audio Objects and Audio Scene Geometry. A necessary AIM for many applications is Audio-Visual Alignment establishing relationships between Audio and Visual Objects.

Body Descriptors, Physical Objects and Visual Scene Geometry are used by the Spatial Object Identification AIM. This provides the identifier of the Physical Object the human body is indicating by using the Body Descriptors and the Scene Geometry. The Speech extracted from the Audio Scene Descriptor is recognised and passed to the Language Understanding AIM together with the Physical Object ID. The AIM provided a refined text (Text (Language Understanding)) and Meaning (semantic, syntactic, and structural information extracted from input data).

Face and Body Descriptors, Meaning and Speech are used by Personal Status Extraction to extract the Personal Status of the human. Dialogue Processing produces a textual response with an associated machine Personal Status that is congruent with the input Text (Language Understanding)

and human Personal Status. The Personal Status Display AIM produces a synthetic Speech and an avatar representing the machine.

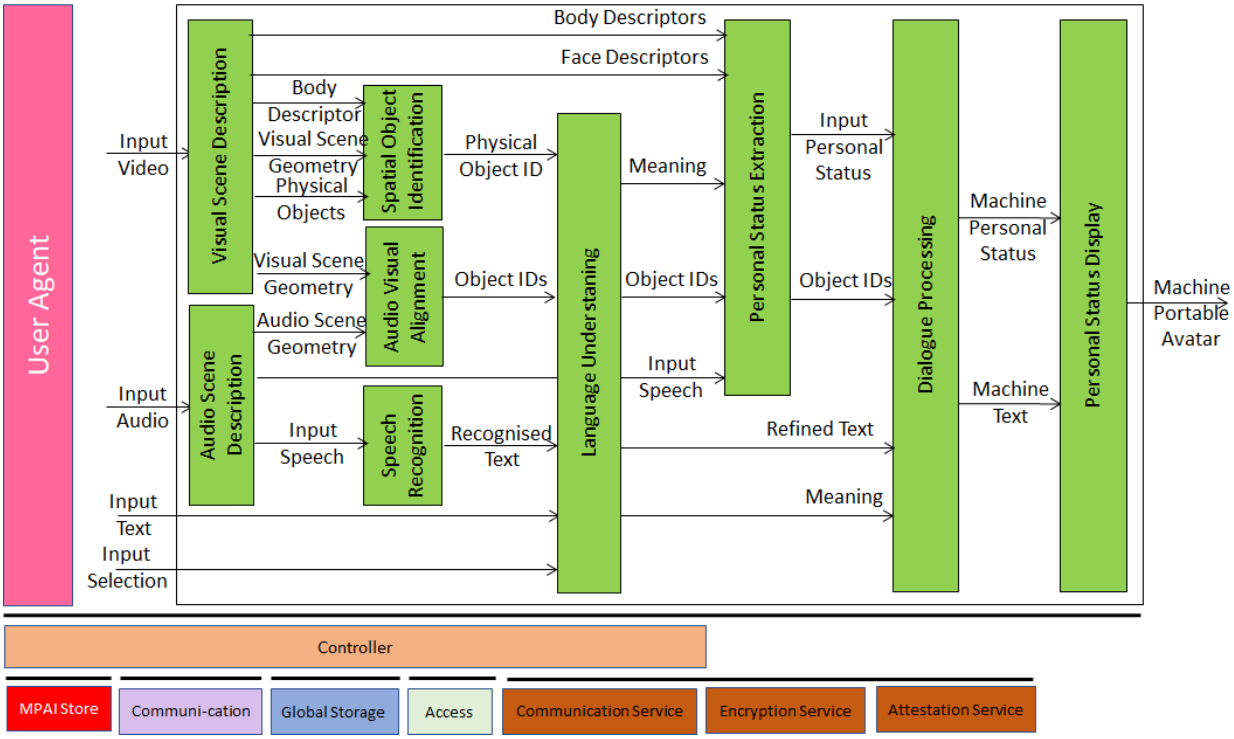


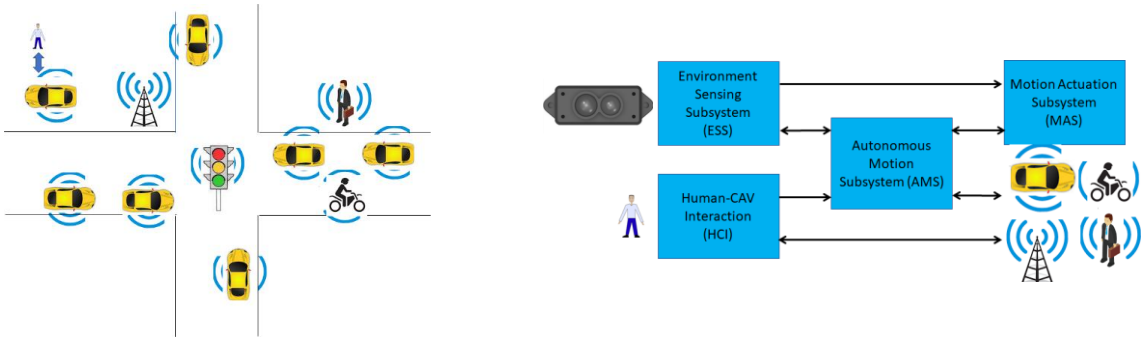
Figure 9 - Personal Status-based Human-Machine dialogue

7 Connected Autonomous Vehicles

MPAI defines a Connected Autonomous Vehicle (CAV) as the information technology-related components of a vehicle enabling it to autonomously reach a destination by:

1. Conversing with humans by understanding their utterances, e.g., a request to be taken to a destination.
2. Acquiring information with a variety of sensors on the physical environment where it is located or traverses like the one depicted in **Error! Reference source not found.**.
3. Planning a Route enabling the CAV to reach the requested destination.
4. Autonomously reaching the destination by:

4.1. Actuating motion in the physical environment.
4.2. Building Digital Representations of the Environment.
4.3. Exchanging elements of such Representations with other CAVs and CAV-aware entities.
4.4. Making decisions about how to execute the Route.
4.5. Acting on the CAV motion actuation to implement the decisions.



MPAI believes in the capability of standards to accelerate the creation of a global competitive CAV market and has published *Technical Specification: Connected Autonomous Vehicle (MPAI-CAV) – Architecture* that includes (see **Error! Reference source not found.**):

1. A CAV Reference Model broken down into four Subsystems.
2. The Functions of each Subsystem.
3. The Data exchanged between Subsystems.
4. A breakdown of each Subsystem in Components of which the following is specified:

4.1. The Functions of the Components.
4.2. The Data exchanged between Components.
4.3. The Topology of Components and their Connections.

5. Subsequently, Functional Requirements of the Data exchanged.
6. Eventually, standard technologies for the Data exchanged.

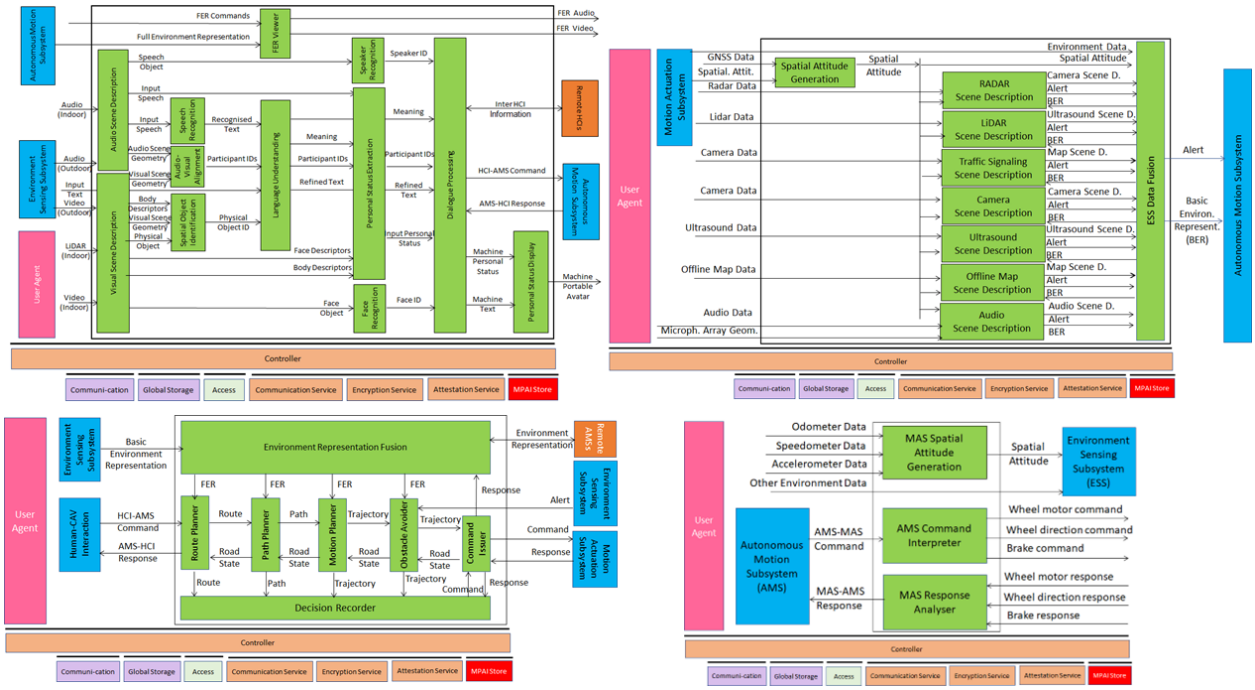


Figure 10 - The MPAI-CAV Subsystems with their Components

Subsystems are implemented as AI Workflows and Components as AI Modules according to Technical Specification: AI Framework (MPAI-AIF) [5].

The Processes of a CAV generate a persistent M-Instance resulting from the integration of:

1. The Environment Representation generated by the Environment Sensing Subsystem by *UM-Capturing* the U-Location being traversed by the CAV.
2. The M-Locations of the M-Instances produced by other CAVs in range CAV that reproduce the U-Locations being traversed by such CAVs to improve the accuracy of the Ego CAV's M-Locations.
3. Relevant Experiences of the Autonomous Motion Subsystem at the M-Location.

Some operations of an implementation of MPAI-CAV can be represented according to the MPAI-MMM – Architecture [9] as shown in Section 12.10.2 Drive a Connected Autonomous Vehicle.

Annex 2 - Notices and Disclaimers Concerning MPAI Standards (Informative)

The notices and legal disclaimers given below shall be borne in mind when downloading and using approved MPAI Standards.

In the following, "Standard" means the collection of four MPAI-approved and published documents: "Technical Specification", "Reference Software" and "Conformance Testing" and, where applicable, "Performance Testing".

Life cycle of MPAI Standards

MPAI Standards are developed in accordance with the MPAI Statutes. An MPAI Standard may only be developed when a Framework Licence has been adopted. MPAI Standards are developed by especially established MPAI Development Committees who operate on the basis of consensus, as specified in Annex 1 of the MPAI Statutes. While the MPAI General Assembly and the Board of Directors administer the process of the said Annex 1, MPAI does not independently evaluate, test, or verify the accuracy of any of the information or the suitability of any of the technology choices made in its Standards.

MPAI Standards may be modified at any time by corrigenda or new editions. A new edition, however, may not necessarily replace an existing MPAI standard. Visit the web page to determine the status of any given published MPAI Standard.

Comments on MPAI Standards are welcome from any interested parties, whether MPAI members or not. Comments shall mandatorily include the name and the version of the MPAI Standard and, if applicable, the specific page or line the comment applies to. Comments should be sent to the MPAI Secretariat. Comments will be reviewed by the appropriate committee for their technical relevance. However, MPAI does not provide interpretation, consulting information, or advice on MPAI Standards. Interested parties are invited to join MPAI so that they can attend the relevant Development Committees.

Coverage and Applicability of MPAI Standards

MPAI makes no warranties or representations of any kind concerning its Standards, and expressly disclaims all warranties, expressed or implied, concerning any of its Standards, including but not limited to the warranties of merchantability, fitness for a particular purpose, non-infringement etc. MPAI Standards are supplied "AS IS".

The existence of an MPAI Standard does not imply that there are no other ways to produce and distribute products and services in the scope of the Standard. Technical progress may render the technologies included in the MPAI Standard obsolete by the time the Standard is used, especially in a field as dynamic as AI. Therefore, those looking for standards in the Data Compression by Artificial Intelligence area should carefully assess the suitability of MPAI Standards for their needs.

IN NO EVENT SHALL MPAI BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF

THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

MPAI alerts users that practicing its Standards may infringe patents and other rights of third parties. Submitters of technologies to this standard have agreed to licence their Intellectual Property according to their respective Framework Licences.

Users of MPAI Standards should consider all applicable laws and regulations when using an MPAI Standard. The validity of Conformance Testing is strictly technical and refers to the correct implementation of the MPAI Standard. Moreover, positive Performance Assessment of an implementation applies exclusively in the context of the MPAI Governance and does not imply compliance with any regulatory requirements in the context of any jurisdiction. Therefore, it is the responsibility of the MPAI Standard implementer to observe or refer to the applicable regulatory requirements. By publishing an MPAI Standard, MPAI does not intend to promote actions that are not in compliance with applicable laws, and the Standard shall not be construed as doing so. In particular, users should evaluate MPAI Standards from the viewpoint of data privacy and data ownership in the context of their jurisdictions.

Implementers and users of MPAI Standards documents are responsible for determining and complying with all appropriate safety, security, environmental and health and all applicable laws and regulations.

Annex 3 - General MPAI Terminology

The Terms used in this standard whose first letter is capital and are not already included in Table 1 are defined in Table 24. To concentrate in one place all the Terms that are composed of a common name followed by other words (e.g., the word Data followed by one of the words Format, Type, or Semantics), the definition given to a Terms preceded by a dash "-" applies to a Term composed by that Term without the dash preceded by the Term that precedes it in the column without a dash.

Table 24 - MPAI-wide Terms

Term	Definition
Access	Static or slowly changing data that are required by an application such as domain knowledge data, data models, etc.
AI Framework (AIF)	The environment where AIWs are executed.
AI Model (AIM)	A data processing element receiving AIM-specific Inputs and producing AIM-specific Outputs according to according to its Function. An AIM may be an aggregation of AIMs.
AI Workflow (AIW)	A structured aggregation of AIMs implementing a Use Case receiving AIW-specific inputs and producing AIW-specific outputs according to the AIW Function.
Application Standard	An MPAI Standard designed to enable a particular application domain.
Channel	A connection between an output port of an AIM and an input port of an AIM. The term "connection" is also used as synonymous.
Communication	The infrastructure that implements message passing between AIMs.
Component	One of the 7 AIF elements: Access, Communication, Controller, Internal Storage, Global Storage, Store, and User Agent
Composite AIM	An AIM aggregating more than one AIM.
Component	One of the 7 AIF elements: Access, Communication, Controller, Internal Storage, Global Storage, Store, and User Agent
Conformance	The attribute of an Implementation of being a correct technical Implementation of a Technical Specification.
- Testing	The normative document specifying the Means to Test the Conformance of an Implementation.
- Testing Means	Procedures, tools, data sets and/or data set characteristics to Test the Conformance of an Implementation.
Connection	A channel connecting an output port of an AIM and an input port of an AIM.
Controller	A Component that manages and controls the AIMs in the AIF, so that they execute in the correct order and at the time when they are needed
Data	Information in digital form.
- Format	The standard digital representation of Data.
- Type	An instance of Data with a specific Data Format.
- Semantics	The meaning of Data.
Descriptor	Coded representation of a text, audio, speech, or visual feature.
Digital Representation	Data corresponding to and representing a physical entity.

Ecosystem	The ensemble of actors making it possible for a User to execute an application composed of an AIF, one or more AIWs, each with one or more AIMs potentially sourced from independent implementers.
Explainability	The ability to trace the output of an Implementation back to the inputs that have produced it.
Fairness	The attribute of an Implementation whose extent of applicability can be assessed by making the training set and/or network open to testing for bias and unanticipated results.
Function	The operations effected by an AIW or an AIM on input data.
Global Storage	A Component to store data shared by AIMs.
AIM/AIW Storage	A Component to store data of the individual AIMs.
Identifier	A name that uniquely identifies an Implementation.
Implementation	1. An embodiment of the MPAI-AIF Technical Specification, or 2. An AIW or AIM of a particular Level (1-2-3) conforming with a Use Case of an MPAI Application Standard.
Implementer	A legal entity implementing MPAI Technical Specifications.
ImplementerID (IID)	A unique name assigned by the ImplementerID Registration Authority to an Implementer.
ImplementerID Registration Authority (IIDRA)	The entity appointed by MPAI to assign ImplementerID's to Implementers.
Instance ID	Instance of a class of Objects and the Group of Objects the Instance belongs to.
Interoperability	The ability to functionally replace an AIM with another AIW having the same Interoperability Level
- Level	The attribute of an AIW and its AIMs to be executable in an AIF Implementation and to: 1. Be proprietary (Level 1) 2. Pass the Conformance Testing (Level 2) of an Application Standard 3. Pass the Performance Testing (Level 3) of an Application Standard.
Knowledge Base	Structured and/or unstructured information made accessible to AIMs via MPAI-specified interfaces
Message	A sequence of Records transported by Communication through Channels.
Normativity	The set of attributes of a technology or a set of technologies specified by the applicable parts of an MPAI standard.
Performance	The attribute of an Implementation of being Reliable, Robust, Fair and Replicable.
- Assessment	The normative document specifying the Means to Assess the Grade of Performance of an Implementation.
- Assessment Means	Procedures, tools, data sets and/or data set characteristics to Assess the Performance of an Implementation.
- Assessor	An entity Assessing the Performance of an Implementation.
Profile	A particular subset of the technologies used in MPAI-AIF or an AIW of an Application Standard and, where applicable, the classes, other subsets, options and parameters relevant to that subset.
Record	A data structure with a specified structure
Reference Model	The AIMs and theirs Connections in an AIW.

Reference Software	A technically correct software implementation of a Technical Specification containing source code, or source and compiled code.
Reliability	The attribute of an Implementation that performs as specified by the Application Standard, profile, and version the Implementation refers to, e.g., within the application scope, stated limitations, and for the period of time specified by the Implementer.
Replicability	The attribute of an Implementation whose Performance, as Assessed by a Performance Assessor, can be replicated, within an agreed level, by another Performance Assessor.
Robustness	The attribute of an Implementation that copes with data outside of the stated application scope with an estimated degree of confidence.
Scope	The domain of applicability of an MPAI Application Standard
Service Provider	An entrepreneur who offers an Implementation as a service (e.g., a recommendation service) to Users.
Standard	A set of Technical Specification, Reference Software, Conformance Testing, Performance Assessment, and Technical Report of an MPAI application Standard.
Technical Specification	(Framework) the normative specification of the AIF. (Application) the normative specification of the set of AIWs belonging to an application domain along with the AIMs required to Implement the AIWs that includes: 1. The formats of the Input/Output data of the AIWs implementing the AIWs. 2. The Connections of the AIMs of the AIW. 3. The formats of the Input/Output data of the AIMs belonging to the AIW.
Testing Laboratory	A laboratory accredited to Assess the Grade of Performance of Implementations.
Time Base	The protocol specifying how Components can access timing information
Topology	The set of AIM Connections of an AIW.
Use Case	A particular instance of the Application domain target of an Application Standard.
User	A user of an Implementation.
User Agent	The Component interfacing the user with an AIF through the Controller
Version	A revision or extension of a Standard or of one of its elements.
Zero Trust	A cybersecurity model primarily focused on data and service protection that assumes no implicit trust.

Annex 4 - Patent declarations (Informative)

Technical Specification: MPAI Metaverse Model (MPAI-MMM) – Architecture has been developed according to the process outlined in the MPAI Statutes [1] and the MPAI Patent Policy [2].

The following table will include references to the entities declaring to agree to licence their standard essential patents reading on *Technical Specification: MPAI Metaverse Model (MPAI-MMM) – Architecture* according to the MPAI-AIF Framework Licence [3]:

Entity	Name	email address