



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

MPAI Technical Specification

MPAI Metaverse Model (MPA-MMM) Technologies (MMM-TEC)

V2.0

WARNING

Use of the technologies described in this Technical Specification may infringe patents, copyrights or intellectual property rights of MPAI Members or non-members.

MPAI and its Members accept no responsibility whatsoever for damages or liability, direct or consequential, which may result from the use of this Technical Specification.

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

Technical Specification

MPAI Metaverse Model (MPA-MMM)

Technologies (MMM-TEC) V2.0

1	Foreword	5
2	Introduction	7
3	Scope	8
4	Definitions.....	8
5	References	17
5.1	Normative reference.....	17
5.2	Informative references.....	17
6	Architecture and Operation	17
7	Processes	20
7.1	General	20
7.2	App.....	20
7.3	Device.....	21
7.4	Service.....	21
7.5	User	21
8	Items.....	22
8.1	Cognitive State	22
8.2	Emotion	25
8.3	Face Personal Status.....	29
8.4	Gesture Personal Status.....	30
8.5	Meaning.....	31
8.6	Personal Status	32
8.7	Social Attitude.....	34
8.8	Speech Descriptors.....	41
8.9	Speech Personal Status.....	42
8.10	Summary	43
8.11	Text Descriptors	45
8.12	Text Personal Status.....	46
8.13	Account	47
8.14	Activity Data	48
8.15	Asset	48
8.16	Authentication.....	49
8.17	Basic Certificate	50
8.18	Basic Discovery.....	51
8.19	Basic Information.....	52
8.20	Basic Interpretation	53
8.21	Basic M-Location.....	55
8.22	Basic U-Location.....	56
8.23	Certificate	57
8.24	Contract	57
8.25	Currency	58
8.26	Discovery	58
8.27	Identifier.....	59
8.28	Information.....	60
8.29	Interpretation.....	61

8.30	IPP Message	61
8.31	M-Capabilities.....	63
8.32	M-Environment.....	64
8.33	Message.....	64
8.34	M-Instance.....	65
8.35	M-Location.....	66
8.36	P-Capabilities	67
8.37	Personal Data.....	68
8.38	Personal Profile	69
8.39	Process Action.....	69
8.40	Program	71
8.41	Provenance	72
8.42	Resolution.....	72
8.43	Rights	74
8.44	Rules.....	74
8.45	Summary	75
8.46	Transaction.....	77
8.47	U-Environment.....	77
8.48	U-Location	78
8.49	Universe-Metaverse Map.....	79
8.50	Validation.....	79
8.51	Value	80
8.52	Wallet	81
8.53	3D Model Event Descriptors.....	82
8.54	3D Model Object.....	83
8.55	3D Model Scene Descriptors.....	84
8.56	3D Model Scene Geometry	85
8.57	Annotation.....	87
8.58	Audio Object	88
8.59	Audio Scene Descriptors.....	89
8.60	Audio-Visual Event Descriptors	91
8.61	Audio-Visual Object	92
8.62	Audio-Visual Scene Descriptors	93
8.63	Basic 3D Model Object.....	95
8.64	Basic 3D Model Scene Descriptors.....	96
8.65	Basic Audio Object	97
8.66	Basic Audio Scene Descriptors.....	99
8.67	Basic Audio-Visual Object.....	100
8.68	Basic Audio-Visual Scene Descriptors	101
8.69	Basic Location.....	103
8.70	Basic Object	104
8.71	Basic Scene Descriptors	105
8.72	Basic Speech Object.....	107
8.73	Basic Speech Scene Descriptors	108
8.74	Basic Text Object.....	109
8.75	Basic Visual Object.....	111
8.76	Basic Visual Scene Descriptors.....	112
8.77	Coordinates.....	113
8.78	Instance Identifier.....	114
8.79	Location.....	116

8.80	Object	117
8.81	Orientation.....	119
8.82	Perceptible Entity	120
8.83	Point of View	121
8.84	Position.....	123
8.85	Scene Descriptors.....	124
8.86	Space-Time.....	126
8.87	Spatial Attitude.....	126
8.88	Speech Event Descriptors.....	127
8.89	Speech Object.....	128
8.90	Speech Scene Descriptors	130
8.91	Text Object.....	131
8.92	Time	133
8.93	Visual Event Descriptors.....	134
8.94	Visual Object.....	134
8.95	Visual Scene Descriptors	136
8.96	Avatar	137
8.97	Body Descriptors.....	138
8.98	Face Descriptors.....	139
8.99	Gesture Descriptors.....	141
9	Protocols.....	142
10	Process Actions	146
10.1	Components.....	146
10.2	Authenticate	147
10.3	Author.....	147
10.4	Change.....	148
10.5	Convert.....	148
10.6	Discover	149
10.7	Execute	149
10.8	Hide	149
10.9	Identify	150
10.10	Inform.....	150
10.11	Interpret	151
10.12	MM-Add.....	151
10.13	MM-Anim	152
10.14	MM-Disable	153
10.15	MM-Embed	153
10.16	MM-Enable	154
10.17	MM-Move	154
10.18	MM-PropertyChange	155
10.19	MM-Send	155
10.20	Modify	156
10.21	MU-Actuate.....	156
10.22	MU-Send	157
10.23	Post	157
10.24	Register.....	158
10.25	Resolve.....	158
10.26	Track.....	158
10.27	Transact.....	159
10.28	UM-Capture	159

10.29	UM-Send	160
10.30	Validate	160
11	Profiles	161
11.1	Introduction	161
11.2	Profile structure	162
11.3	Baseline Profile	163
11.4	Finance Profile	163
11.5	Management Profile	164
11.6	High Profile	165
12	Verification Use Cases	165
12.1	Introduction	166
12.2	Use Case Description Language	166
12.3	Friends meet in the metaverse	167
12.4	Virtual Lecture	170
12.5	Virtual Meeting	174
12.6	Hybrid working	176
12.7	eSports Tournament	179
12.8	Virtual Performance	182
12.9	AR Tourist Guide	186
12.10	Virtual Dance School	189
12.11	Virtual Car Showroom	191
12.12	Meeting in Connected Autonomous Vehicles.....	194
12.13	Co-design across metaverses.....	198
12.14	Selling assets on a Marketplace	201
13	MPAI-MMM API	202
13.1	MMM-API	202
13.2	Description	203
13.3	Components.....	203
13.4	Paths	203
13.5	Transaction Service	206
14	Reference Software	206

1 Foreword

The international, unaffiliated, non-profit *Moving Picture, Audio, and Data Coding by Artificial Intelligence (MPAI)* organisation was established in September 2020 in the context of:

1. **Increasing** use of Artificial Intelligence (AI) technologies applied to a broad range of domains affecting millions of people
2. **Marginal** reliance on standards in the development of those AI applications
3. **Unprecedented** impact exerted by standards on the digital media industry affecting billions of people

believing that AI-based data coding standards will have a similar positive impact on the Information and Communication Technology industry.

The design principles of the MPAI organisation as established by the MPAI Statutes are the development of AI-based Data Coding standards in pursuit of the following policies:

1. Publish upfront clear Intellectual Property Rights licensing frameworks.
2. Adhere to a rigorous standard development process.
3. Be friendly to the AI context but, to the extent possible, remain agnostic to the technology thus allowing developers freedom in the selection of the more appropriate – AI or Data Processing – technologies for their needs.

4. Be attractive to different industries, end users, and regulators.
5. Address five standardisation areas:
 1. *Data Type*, a particular type of Data, e.g., Audio, Visual, Object, Scenes, and Descriptors with as clear semantics as possible.
 2. *Qualifier*, specialised Metadata conveying information on Sub-Types, Formats, and Attributes of a Data Type.
 3. *AI Module* (AIM), processing elements with identified functions and input/output Data Types.
 4. *AI Workflow* (AIW), MPAI-specified configurations of AIMs with identified functions and input/output Data Types.
 5. *AI Framework* (AIF), an environment enabling dynamic configuration, initialisation, execution, and control of AIWs.
6. Provide appropriate Governance of the ecosystem created by MPAI Technical Specifications enabling users to:
 1. *Operate* Reference Software Implementations of MPAI Technical Specifications provided together with Reference Software Specifications
 2. *Test* the conformance of an implementation with a Technical Specification using the Conformance Testing Specification.
 3. *Assess* the performance of an implementation of a Technical Specification using the Performance Assessment Specification.
 4. *Obtain* conforming implementations possibly with a performance assessment report from a trusted source through the MPAI Store.

MPAI operates on three solid pillars:

1. The [MPAI Patent Policy](#) specifies the MPAI standard development process and the Framework Licence development guidelines.
2. [Technical Specification: Artificial Intelligence Framework \(MPAI-AIF\) V2.1](#) specifies an environment enabling initialisation, dynamic configuration, and control of AIWs in the standard AI Framework environment depicted in Figure 1. An AI Framework can execute AI applications called AI Workflows (AIW) typically including interconnected AI Modules (AIM). MPAI-AIF supports small- and large-scale high-performance components and promotes solutions with improved explainability.

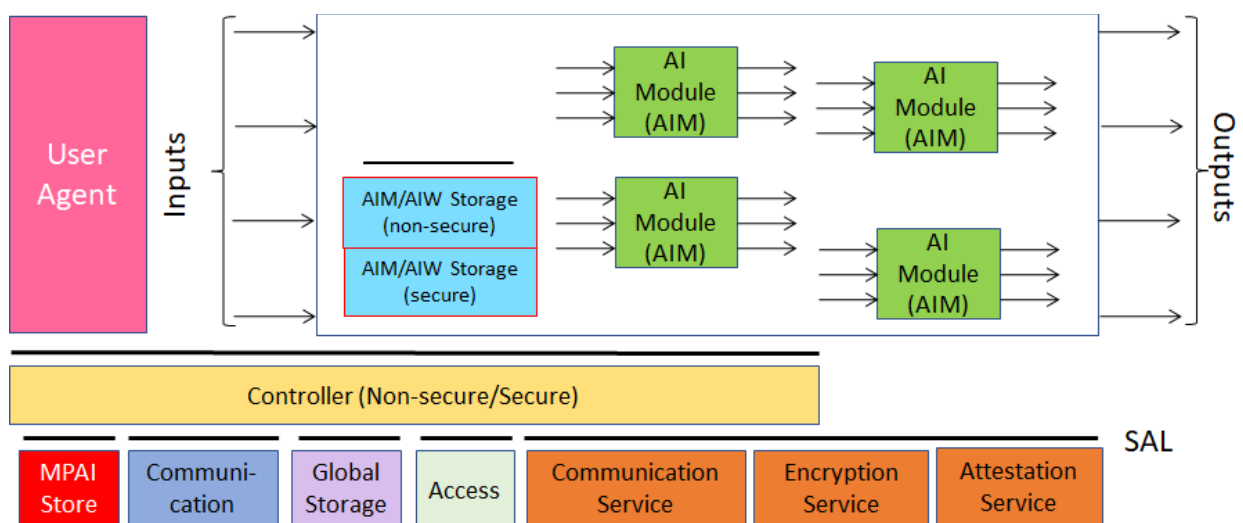


Figure 1 - The AI Framework (MPAI-AIF) V2 Reference Model

3. [**Technical Specification: Data Types, Formats, and Attributes \(MPAI-TFA\) V1.2**](#) specifies Qualifiers, a type of metadata supporting the operation of AIMs receiving data from other AIMs or from input data. Qualifiers convey information on Sub-Types (e.g., the type of colour), Formats (e.g., the type of compression and transport), and Attributes (e.g., semantic information in the Content). Although Qualifiers are human-readable, they are only intended to be used by AIMs. Therefore, Text, Speech, Audio, Visual, and other Data received by or exchanged between AIWs and AIMs should be interpreted as being composed of Content (Text, Speech, Audio, and Visual as appropriate) and associated Qualifiers. For instance, a Text Object is composed of Text Data and Text Qualifier. The specification of most MPAI Data Types reflects this point.

2 Introduction

(Informative)

Many consider the loose "metaverse" concept as one of the most promising evolutionary steps of Information and Communication Technology and some have developed implementations that can be classified as "metaverse". However, metaverse designers and developers have generally made technology decisions that best responded to their needs, with little or no consideration of the need to interact or interoperate with other metaverse instances.

In response to concerns that such metaverse "walled gardens" do not fully exploit the opportunities offered by current and expected technologies, MPAI has developed Technical Reports and Technical Specifications in an attempt to provide specifications whose implementations make metaverse instances "Interoperable":

1. **Technical Report: MPAI Metaverse Model (MPAI-MMM) – Functionalities** introduces definitions, assumptions, a set of high-level use cases, a collection of exemplary service providers, a set of ~150 Functionalities that a metaverse instance might support, a review of the main metaverse-enabling technologies, an analysis of metaverse governance issues, and a standardisation roadmap.
2. **Technical Report: MPAI Metaverse Model (MPAI-MMM) – Functionality Profiles** introduces and revises an extended list of definitions; a metaverse "Operation Model" based on the notion of *Processes* performing or requesting other *Processes* to perform *Actions* on *Items*, i.e., Data, Metadata, and Qualifiers supported by an M-Instance, the type of metaverse to be specified by MPAI; an initial identification of Actions, Items, and Basic Data with Use Cases and Functionality Profiles; a collection of representative use cases tested against the Operation Model; and four initial Functionality Profiles.
3. **Technical Specification – MPAI Metaverse Model (MPAI-MMM) – Architecture (MMM-ARC)** specifies the Functional Requirements of Processes and Actions to provide means to achieve Interoperability between independently designed two or more M-Instances that execute Processes, and produce Data that comply with the MMM-ARC Functional Requirements, if necessary via a Conversion Service.
4. **Technical Specification – MPAI Metaverse Model (MPAI-MMM) – Technologies (MMM-TEC)** specifies or references specifications of Items including Qualifiers that enable interoperability between M-Instances supporting the technologies referenced by the Qualifiers.

This document - **Technical Specification – MPAI Metaverse Model (MPAI-MMM) – Technologies (MMM-TEC) V2.0** - merges and extends the MPAI-ARC V1.1 and the MPAI-TEC V1.0 into an integrated specification by providing the MPAI-MMM APIs and Reference Software. In all Chapters and Sections, Terms beginning with a capital letter are defined in [Table 1](#) online. All MPAI-defined Terms are [accessible online](#). All Chapters are Normative unless they are labelled as Informative.

3 Scope

Technical Specification: MPAI Metaverse Model (MPAI-MMM) – Technologies (MMM-TEC) V2.0 – in this and other MPAI documents also called MMM-TEC V2.0 or MMM-TEC:

1. Specifies the following technologies enabling an M-Instance to interoperate with independently designed and implemented clients and M-Instances:
 1. The Functional Requirements of the *Processes* operating in an M-Instance.
 2. The *Items*, i.e., the Data Types and their Qualifiers recognised in an M-Instance.
 3. The *Protocols* enabling a Process to communicate with another Process.
 4. The *Process Actions* that a Process performs on Items.
 5. The MPAI Metaverse Model *Profiles*.
 6. The MPAI-MMM API.
2. The MMM-TEC Reference Software demonstrating the operation of an M-Instance in one of the ten informative Use Cases included in MMM-TEC V2.0.

Note: Security is assumed to be provided by the Information and Communication Technology Platform supporting the M-Instance.

All MPAI Qualifiers are specified in **Technical Specification: Data Types, Formats, and Attributes (MPAI-TFA) V1.3**. MPAI-TFA specifies Qualifiers for consistent use across all MPAI Technical Specifications.

MMM-TEC has been developed by the MMM Group of the Requirements Standing Committee. MPAI may develop MMM-TEC extensions of new Technical Specifications in the MPAI Metaverse Model work area.

4 Definitions

Capitalised Terms used in MMM-ARC and MMM-TEC have the meaning defined in [Table 1](#). Non-capitalised terms letter have the meaning commonly defined for the context in which they are used or represent an entity in the real world. For instance,

1. Table 1 defines *Object*, *Scene*, and *User* but does not define *object*, *scene*, and *human*.
2. Object indicates an Item but object indicates an entity in the Universe commonly classified as object.

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 first 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 first 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.

All MPAI-defined Terms are available [online](#).

Table 1 - General Terms and Definitions

Terms	Definitions
Account	An Item that uniquely references a Registered human.
Action	A function performed by a Process.
- <i>Authenticate</i>	The Action of a User requesting an Authentication Service to: <ol style="list-style-type: none">1. Confirm that an Item or Process is what it claims to be with an

Authentication Item request.

2. Grant Rights to Authentication Item response based on Model Rights.

The Action of a User requesting an Author Service to:

- *Author*
 1. Produce an Item based on provided Items, Data, Qualifiers.
 2. Grant Rights to the Authored Item based on Model Rights.

The Action of a User requesting a Rights Service to:

- *Change*
 1. Modify the Rights of a Process or an Item based on Model Rights.
 2. Grant the requesting User the Rights to further Change the Rights.

The Action of a Process requesting a Conversion Service to:

- *Convert*
 1. Change the Data of an Item according to a given Qualifier
 2. Grant Rights to the Converted Item based on Model Rights

The Action of a User requesting a Discovery Service to

- *Discover*
 1. Provide Item IDs or Process IDs relevant to the Discovery Item request.
 2. Grant Rights to the Discovery Item response based on Model Rights.

The Action of a Process requesting an Execution Service to:

- *Execute*
 1. Execute a Program.
 2. Grant Rights to the produced Items based on Model Rights.

The Action of a Process requesting an Identification Service to:

- *Hide*
 1. Make the ID of an Item unavailable to all Processes, but the requesting Process.
 2. Grant Rights to the Hidden Item based on Model Rights.

Action of a Process requesting an Identification Service to:

- *Identify*
 1. Produce an Item from Data, Qualifier, and Model Rights provided by the requesting Process.
 2. Grant Rights to the Item based on Model Rights.

The Action of a User requesting an Information Service to:

- *Inform*
 1. Provide information about an Item or Process as contained in the Information Item request.
 2. Grant Rights to Information Item response based on Model Rights.

The Action of a User requesting an Interpretation Service to:

- *Interpret*
 1. Provide information about an Item or Process as contained in the Interpretation Item request.
 2. Grant Rights to Interpretation Item response based on Model Rights.

The Action of a User requesting a Location Service to:

- *MM-Add*
 1. Place an Item at an M-Location with a Spatial Attitude.
 2. Grant Rights based on Model Rights to the MM-Added Item.

Only the User who has MM-Added the Item shall be able to perceive it. Other Users shall be able to perceive it only if the Item is MM-Enabled

The Action of a User requesting that a Location Service:

- *MM-Anim*
 1. Animate an MM-Added or MM-Embedded Item with a stream Item.
 2. Grant Rights based on Model Rights to the MM-Animated Item.

The Action of a User requesting a Location Service to:

- *MM-Disable*
 1. Stop making perceptible an MM-Embedded Item to all Users but the requesting User.
 2. Preserve any change than may have been effected on the MM-Disabled Item.
 3. Grant Rights to the MM-Disabled Item based on Model Rights.

- The Action of a User requesting a Location Service to
- *MM-Embed*
 1. Place an Item at an M-Location with a Spatial Attitude.
 2. Make the Item perceptible.
 3. Grant Rights to the MM-Embedded Item based on Model Rights.
- The Action of a User requesting a Location Service to:
- *MM-Enable*
 1. Add or Change the Spatial Attitude of an MM-Added Item at an M-Location:
 2. Resize the Item by R_x , R_y , R_z along the Item's axes.
 3. Make the Item perceptible.
 4. Grant Rights to the MM-Enabled Item based on Model Rights.
- The Action of a User requesting a Location Service to:
- *MM-Move*
 1. Move an MM-Added/MM-Embedded Item at an M-Location to another MM-Location.
 2. Preserve any change that may have been effected on the Item in previous MM-Add, MM-Embed, or MM-Enable Actions.
 3. Preserve the (un)perceptibility status of the Item.
 4. Grant Rights to the MM-Moved Item based on Model Rights
- The Action of a User requesting a Location Service to modify the characteristics of an Item:
- *MM-PropertyChange*
 1. Resize the Item by R_x , R_y , R_z along the Item's axes.
 2. Display a specific Personal Status (if a Persona).
 3. Grant Rights to the MM-Morphed Item based on Model Rights.
- The Action of a Process requesting a Communication Service:
- *MM-Send*
 1. To send a Message to other Processes.
 2. To grant Rights to the receiving Process on the content of the Message.
- The Action of a User requesting an Identification Service to
- *Modify*
 1. Produce a new Item starting from an existing Item using new Data, Qualifier, and Model Rights to the new Item.
 2. Grant Rights to the Modified Item based on Model Rights.
- The Action of a User requesting an ExIm Service to:
- *MU-Actuate*
 1. Render an Item at a U-Location as Media with a Spatial Attitude
 2. Also render the Scene of the M-Location including the Item, if the M-Location field is present.
 3. Grant Rights to the U-Location based on Model Rights.

MM-Added Items preserve their (un)perceptibility attributes.
- The Action of a Process requesting an ExIm Service to:
- *MU-Send*
 1. Send a Message to Processes in the Universe.
 2. Grant Rights to Data and Qualifier based on Model Rights.
- The Action of a User requesting that a Marketplace Service:
- *Post*
 1. Include an Asset to its repository.
 2. Grant Rights based on Model Rights if a Transaction based on Model Transaction is performed.
- The Action of a human requesting that a Registration Service:
- *Register*
 1. Open an Account based on the human's Personal Data.
 2. Potentially pays for the Account.
 3. Grant their Users Rights to perform Actions in the M-Instance.
- The Action of:
- *Resolve*
 1. A Process requesting that a Resolution Service set up a session

	between/among two/more than two Processes in two/more than two M-Instances.
	2. A Process or Resolution Service responding to a session request.
	The Action of a User ₁ (“sender”) requesting that a Transaction Service:
	1. Assign Rights on an Asset to User ₂ (“receiver”).
	2. Cause:
- <i>Transact</i>	2.1. Wallet ₁ of User ₁ 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).
	The Action of a Process holding Rights to a U-Location to request an ExIm Service to:
- <i>UM-Capture</i>	1. Capture Data and Qualifier with a Spatial Attitude from Media at U-Location.
	2. Grant Rights to the Process to Identify the Data and Qualifier as Item.
	The Action of a Process requesting an ExIm Service to:
- <i>UM-Send</i>	1. Be allowed to receive a Message from a Service in the Universe.
	2. Grant Rights to Act on the content of the Message.
	The Action of a Process requesting a Validation Service to confirm that a Process hold claimed Rights.
- <i>Validate</i>	
Actuator	A component of a Device able to MU-Render an Item to or MU-Send Items to a U-Environment.
Certification	The attestation that a Process or Item has specified characteristics.
Conversion	The process of Modifying the Data of an Item according to provided Data and Qualifier.
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.
Ecosystem	The ensemble of entities and rules ensuring that Metaverse Instances operate in the interest of Metaverse Stakeholders.
Entitlement	The state of a Process having certain Rights in an M-Instance.
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 a human to interact with M-Instance:

- *Brain-Computer* (BCI) by sensing and processing the electrical activity of the brain.
 - *Haptic* through bodily movements and sensations.
 - *Speech* using spoken language.
 - *Visual* through bodily movements and visual messages.
- Item A Data Type specified by MMM-TEC or other MPAI Technical Specifications for use in an M-Instance.
- An Item that
- *Account*
 - Uniquely references a human registered with an M-Instance
 - Includes the IDs of the human's Personal Profile, Processes, and their Internal Rights.
 - *Activity Data* An Item that records the IPP Protocols executed by a Process.
 - *Asset* An Item that can be Transacted.
 - *Authentication* An Item generated by a Process to request a Service to confirm that an Item is what it claims to be or an Item generated by a Service containing the response to the request.
 - *Basic Certificate* An Item attesting the suitability of a Process for specific MMM-TEC V2.0 usages. Other types of Certificates are possible and are listed in the Certificate Qualifier.
 - *Basic Discovery* An Item generated by a Process to requests that a Service discover Items and Processes. Other types of Discovery Items are possible and are listed in the Certificate Qualifier.
 - *Basic Information* An Item generated by a Process to requests that a Service provide information on Items and Processes. Other types of Information Items are possible and are listed in the Information Qualifier.
 - *Basic Interpretation* An Item generated by a Process to requests that a Service interpret an Items. Other types of Interpretation Items are possible and are listed in the Information Qualifier.
 - *Basic M-Location* An Item representing a region of an M-Instance with Space-Time attributes that is not exposed as further subdivided as a Location.
 - *Basic Object* An Item representing:
 - Data of a specific media type perceptible by a specific device and/or a human.
 - Descriptive Data regarding Sub-Types, Formats and Attributes of the Data (optionally).
 - *Basic Scene Descriptors* An Item representing the Objects of a Basic Scene and their arrangement in the Scene.
 - *Basic Scene Geometry* An Item representing the spatial arrangement of the Objects in a Basic Scene.
 - *Basic U-Location* An Item representing a region of the Universe with Space-Time attributes that is not (exposed as) further subdivided.
 - *Certificate* An Item Process attesting the suitability of a Process for specific usages that is not a Basic Certificate.
 - *Cognitive State* An Item representing the Personal Status Factor representing the internal state of an Entity such as "surprised" or "interested".
 - *Contract* An Item representing terms and conditions or a Program that is executed according to certain terms when conditions are met.

- *Currency* An Item representing the unit of measure of an Amount of a Value.
- *Discovery* An Item representing the description of the Item or Process (or their IDs) to be Discovered.
- *Emotion* An Item representing the Personal Status Factor representing the internal state of an Entity such as that resulting from its interaction with the Context, such as “Angry”, “Sad”, “Determined”.
- *Event Descriptors* An Item representing the series of Scene Descriptors from start and end time.
- *Identifier* An Item that references only one Process or Item in an M-Instance.
- *Information* An Item sent by a Process to a Service requesting information about an Item or Process (or their IDs) to obtain additional information on an Item or Process that is not otherwise available. The Service will respond with an Information response.
- *Interpretation* An Item sent by a Process to a Service requesting interpretation of an Item (or its ID) and the response of the Service.
- *IPP Message* An Item that a Source Process sends to a Destination Process requesting the performance of a Process Action.
- *M-Capabilities* An Item representing the capabilities of an M-Instance or M-Environment.
- *M-Environment* An Identified administrative subset of an M-Instance.
- *Message* An Item that a Source Process MM-Sends to a Communication Service requesting it to deliver it to a Destination Process.
- *M-Instance* An Item representing the Virtual Space created according to the MMM-TEC Technical Specification of the MPAI-Metaverse Model.
- *M-Location* An Item representing a region of an M-Instance with Space-Time attributes that is (exposed as) further subdivided.
- *Object* An Item including a collection of Basic Objects possibly of different Media Types. An Object may have a hierarchical structure where Objects contain Basic Objects and Objects.
- *Orientation* An Item representing an Object’s orientation, orientation velocity, and orientation acceleration.
- *P-Capabilities* An Items containing the list of Processes and Actions that it can perform.
- *Personal Data* An Item containing a human’s Personal Profile and Activity Data of their Users.
- *Personal Profile* An Item containing a human’s Personal Data submitted when Registering with an M-Instance.
- *Personal Status* An Item representing the information internal to an Entity that characterises their behaviour.
- *Point of View* An Item representing the Position and Orientation of an Object in a Virtual Environment excluding velocity and acceleration.
- *Position* An Item representing an Object’s position, velocity, and acceleration.
- *Process Action* An Item that specifies:
 - The Action that a Process has performed, is performing, or is allowed to perform.
 - Time, Source and Destination Complements, and Error Message.

- *Program* An Item containing executable code, e.g., a Process or a Contract that is a Program.
 - *Provenance* An Item containing the list of all Transactions executed on an Asset, first and last included.
 - *Resolution* An Item containing an M-Instance_A's Process_A Request to Resolution Service_A to set up a session involving different M-Instances' Processes and the Responses of the Resolution Services and destination Processes.
 - *Rights* An Item representing the set of Process Actions and the corresponding Levels that a Process may perform.
 - *Rules* An Item containing the set of Rights that Processes May, May not, or Must exercise in the M-Instance:
May: A Process is allowed to exercise the Rights.
May not: A Process is not allowed to exercise the Rights.
Must: A Process must exercise the Rights.
 - *Scene Descriptors* An Item including the Objects of a Scene, the Sub-Scenes, and their arrangement in the Scene.
 - *Scene Geometry* An Item representing the arrangement of the Objects of a Scene that may include Objects and Sub-Scenes in a hierarchical fashion.
 - *Social Attitude* An Item representing Personal Status Factor representing the internal state of an Entity related to the way it intends to position itself vis-à-vis the Context, e.g., "Respectful", "Confrontational", "Soothing".
 - *Space-Time* An Item representing the Spatial Attitude and Time information.
 - *Spatial Attitude* An Item representing the Position and Orientation of an Object, and their velocities and accelerations.
 - *Summary* An Item representing a text-based abridged outline of the utterance(s) of one or more Entities represented by their User ID and including Space-Time, Text, and Personal Statuses.
 - *Time* An Item representing the start time and end time of a duration.
 - *Transaction* An Item representing:
 -The Amount, the WalletID and the Rights on an Asset of a User transferring Rights to another User (Sender).
 - The Amount, the WalletID and the Rights on on the Asset of another User receiving the Rights (Receiver).
 - Optionally the Amount and the WalletID of the Service Provider facilitating/enabling the Transaction.
 - *U-Environment* An Item representing an Identified subset of the Universe.
 - *U-Location* An Item representing a region of the Universe with Space-Time attributes.
 - *Universe-Metaverse Map* An Item representing a list of U-Locations and corresponding M-Locations and/or Items with their Spatial Attitudes.
 - *Validation* An Item sent by a Process to a Service requesting it to validate a claim that another Process makes about its Rights and the response of the Service.
 - *Value* An Item combining an Amount and a Currency.
 - *Wallet* A container of Values.
- Layer
- *Enabling Service* The set of Services such as payment, security, identity, privacy, etc. that enable operation of an M-Instance.

- <i>Experience</i>	The set of functions, such as Devices, that generate Experiences.
- <i>Infrastructure</i>	The set of functions such as network, transport, storage, and (cloud, edge) processing that enable an M-Instance to operate.
- <i>Platform</i>	The set of Services, such as content creation, content discovery, and content access functions that enable an M-Instance to operate.
Level	A type of Right, currently Internal (granted at Registration Time), Acquired (by Process Activity), Granted (by another Process). - . Media - Data acquired by a Device using a Sensor. - Data converted by a Device from an Item to a format that can be presented,
Metaverse	
- Interoperability	The ability of an M-Instance to exchange and make use of the data of a user device or another M-Instance as intended by a user device or M-Instance.
- 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.
- Manager	The entity overseeing the operation of an M-Instance.
- Operation Model	The components and sequence of steps involved in the operation of an M-Instance.
- 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)
- Profile	A recognised subset of the Functionalities (Functionality Profile) or Technologies (Technology Profile) specified by the Common Metaverse Specifications.
- <i>Level</i>	A subdivision of a Profile that indicates the degree of completeness of the user experience provided by that Level.
- Registration	The provisioning by a human of a subset of Personal Data to an M-Instance/Environment to obtain an Account.
- 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.
Oracle	A Service providing U-Environment information to a Blockchain.
Persona	A Model representing a human.
Privacy	The Rights of a User to keep their Personal Profile secret.

Process	
- App	An application-specific Program executed on a Device.
	A Process able to:
	1. UM-Capture Data from a U-Location
	2. UM-Send Data and Metadata to a User.
- Device	and/or
	1. MM-Send an Entity from an M-Location to the Device.
	2. MU-Render an Entity at a U-Location.
- Functionality	The attribute of a Process that is able to perform particular Action(s).
- Service	A Process that can be called to provide specific Functionalities.
- User	A Process representing a human.
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.
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	
- <i>Fungible</i>	A representation of an Asset that is interchangeable with other Assets of the same type.
	(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.
- <i>Non-Fungible</i>	- 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.
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	
- <i>Decentralised</i>	An Identifier that enables the verifiable association with a User without requiring a centralised registry.
	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.
- <i>Self-Sovereign</i>	
Wallet	
- <i>Crypto</i>	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.

5 References

5.1 Normative reference

1. MPAI; [Technical Specification; Multimodal Conversation](#) (MPAI-MMC) V2.2.
2. MPAI; Technical Specification: [Object and Scene Description](#) (MPAI-OSD) V1.3.
3. MPAI; Technical Specification: [Portable Avatar Format](#) (MPAI-PAF) V1.4.
4. MPAI; Technical Specification: [AI Module Profiles](#) (MPAI-PRF) V1.0.
5. MPAI; Technical Specification: [Data Types, Formats, and Attributes](#) (MPAI-TFA) V1.3.
6. ECMA; [ECMA-404 The JSON Data Interchange Standard](#).

5.2 Informative references

7. MPAI; [The MPAI Statutes](#).
8. MPAI; [The MPAI Patent Policy](#).
9. MPAI; Technical Report – [MPAI Metaverse Model](#) (MPAI-MMM) – [Functionalities](#) (MMM-FNC).
10. MPAI; Technical Report – [MPAI Metaverse Model](#) (MPAI-MMM) – [Functionality Profiles](#) (MMM-FPR).
11. MPAI; Technical Specification: [Governance of the MPAI Ecosystem](#) (MPAI-GME) V1.1.
12. MPAI; Technical Specification: [Artificial Intelligence Framework](#) (MPAI-AIF) V2.1.
13. MPAI; Technical Specification – [Connected Autonomous Vehicle](#) (MPAI-CAV) – [Technologies](#) (CAV-TEC) V2.0.
14. MPAI; [Technical Specification; Human and Machine Communication](#) (MPAI-MMC) V2.0.
15. MPAI; MPAI; Framework Licence: [MPAI Metaverse Model](#) (MPAI-MMM) – [Technologies](#).

6 Architecture and Operation

(Informative)

MMM-TEC defines a metaverse instance ([M-Instance](#)) as an Information and Communication Technologies platform populated by [Processes](#) that:

1. *Operate* with various degrees of autonomy and interactivity under the responsibility of the M-Instance whic. Third parties may offer services to Processes, and machines and humans in the *Universe*, i.e., the real world.
2. *Sense* Data from [U-Environments](#), i.e., portions of the Universe.
3. *Produce* three types of [Items](#), i.e., Data that has been [Identified](#) in - and thus recognised by - the M-Instance:
 1. *Digitised* - i.e., sensed from the Universe - possibly animated by activities in the Universe.
 2. *Virtual* - i.e., imported from the Universe as Data or internally generated.
 3. *Mixed* - Digitised and Virtual.
4. *Perform* [Process Actions](#) – either on their initiative, or driven by the actions of humans or machines in the Universe – that combine:
 1. An [Action](#), whose name is possibly prepended by:
 1. *MM*: to indicate Actions performed *inside* the M-Instance, e.g., [MM-Anim](#) using a stream to animate a [3D Model](#) with a [Spatial Attitude](#) (defined as [Position](#), [Orientation](#), and their velocities and accelerations).

2. *MU*: [MU-Actuate](#): to indicate Actions in the M-Instance influencing the Universe, e.g., to render one of its Items to a U-Location as *Media* with a Spatial Attitude.
3. *UM*: to indicate Actions in the Universe influencing the M-Instance, e.g., [UM-Capture](#) to acquire Media at a U-Location with a Spatial Attitude as Data, Qualifier, and Rights.
 2. Items on which the Action is performed or are required for performance, such as Asset, 3D Model, Audio Object, Audio-Visual Scene, etc.
 3. M-Locations and/or U-Locations where the Process Action is performed.
 4. Processes with which the Action is performed.
 5. Time(s) during which the Process Action is requested to be and is performed.
5. *May hold* [Rights](#) on an Item, i.e., may perform the set of Process Actions that are listed as Rights. An Item may include Rights signalling which Processes may perform which Process Actions on it.
6. *Affect* U-Environments and/or M-Instances using Items in ways that are:
 1. Consistent with the goals of the M-Instance as expressed by the [Rules](#).
 2. Within the [M-Capabilities](#) of the M-Instance, e.g., to support [Transactions](#).
 3. Respecting applicable laws and regulations.
7. *Perform* activities strictly inside the M-Instance or have various degrees of interaction with Data sensed from and/or actuated in the Universe.
8. *May be characterised* as:
 1. Services providing specific functionalities, such as content authoring.
 2. Devices connecting the Universe to the M-Instance and the M-Instance to the Universe.
 3. Apps running on Devices.
 4. Users representing and acting on behalf of human entities residing in the Universe and possibly rendered as a Persona, i.e., an avatar.
9. *May request* another Process to perform Process Actions on its behalf by using the [Inter-Process Protocol](#), possibly after [Transacting](#) a [Value](#) (i.e., an *Amount* in a [Currency](#)) to a Wallet.

An M-Instance is managed by an M-Instance Manager. At the initial time, the M-Instance Manager has Rights covering the M-Instance and may decide to define certain subsets inside the M-Instance - called [M-Environments](#) - on which it has Rights and to attach Rights to them.

A [Registering](#) human may:

1. [Request](#) to Register and open an [Account](#) of a certain class.
2. [Be requested](#) to provide their [Personal Profile](#) and possibly to perform a Transaction.
3. [Obtain](#) in exchange a set of Rights that their Processes may perform. Rights have *Levels* indicating that the a specific set of Rights is:
 1. *Internal*, e.g., assigned by the M-Instance at Registration time according to the M-Instance Rules and Account type.
 2. *Acquired*, e.g., obtained by initiative of the Process.
 3. *Granted* to the Process by another Process.

MMM-TEC V2.0 does not specify how an M-Instance verifies that the Process Actions performed by a Process comply with the Process's Rights or the M-Instance Rules. An M-Instance can decide to verify the full set of [Activity Data](#) (the log of performed Process Actions), or to make verifications based on claims by another Process, to make random verifications, or to not make any verification at all. Therefore, MMM-TEC V2.0 does not specify how a M-Instance Manager can sanction non-complying Processes.

In some cases, an M-Instance could be wastefully too costly as an undertaking if all the technologies required by the MMM Technical Specification were mandatorily to be implemented, even if the particular M-Instance had limited scope. MMM-TEC V2.0 specifies [Profiles](#) to facilitate the take-off of M-Instance implementations that conform to the MMM-TEC V2.0 specification without unduly burdening some other implementations.

A Profile includes only a subset of the Actions and Items that are expected to be needed and are shared by a sizeable number of applications. MMM-TEC v2.0 defines four Profiles (see Figure 2):

1. [Baseline Profile](#) enables basic applications such as lecture, meeting, and hang-out.
2. [Finance Profile](#) enables trading activities.
3. [Management Profile](#) enables a controlled ecosystem with more advanced functionalities.
4. [High Profile](#) enables all the functionalities of the Management Profile with a few additional functionalities of its own

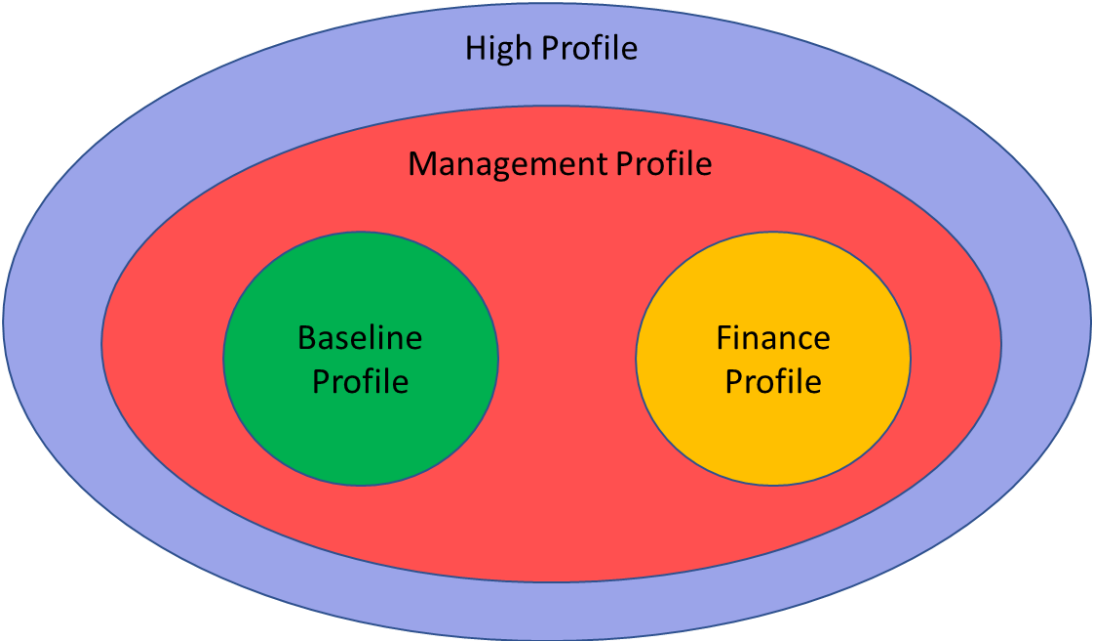


Figure 2 - MMM-TEC V2.0 Profiles

MPAI developed and used some use cases in the two MPAI-MMM Technical Reports published in 2023 and developed to facilitate the MMM-ARC and MMM-TEC Technical Specifications. However, the [Verification Use Cases](#) have been included in MMM-TEC V2.0 to verify that the currently specified Actions and Items support those Use Cases.

Figure 3 gives a summary view of some of the basic MMM elements.

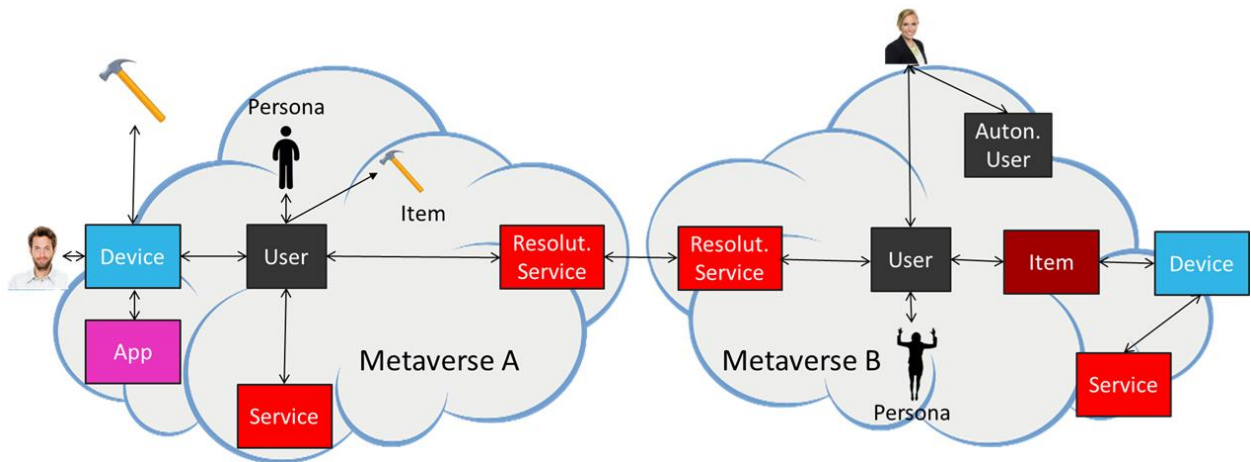


Figure 3 - Main elements of an M-Instance

The fast development of certain technology areas is one of the issues that has so far prevented the development of metaverse interoperability standards. MMM-TEC deals with this issue by providing JSON syntax and semantics for all Items. When needed, the JSON syntax references [Qualifiers](#), MPAI-defined Data Types that supply additional information to the Data in the form of:

1. *Sub-Type* (e.g., the colour space of a Visual Data Type).
2. *Format* (e.g., the compression or the file/streaming format of Speech).
3. *Attributes* (e.g., the Binaural Cues of an Audio Object).

For instance, a Process receiving an Object can understand from the Qualifier referenced in the Object whether it has the required technology to process it, or else it has to rely on a *Conversion Service* to obtain a version of the Object matching its [P-Capabilities](#). This approach should help to prolong the life of the MMM-TEC specification as in many cases only the Qualifier specification will need to be updated, not the MMM-TEC specification.

7 Processes

7.1 General

Process is an instance of a Program running on the computing platform underpinning an M-Instance or on a device connected to it. A Process may perform an MPAI-specified Process Action or request other Processes to perform Process Actions using the **Inter-Process Protocol**. If the Items received have incompatible Qualifiers, a Process may request a Conversion Service to convert the Item so that has a Qualifier compatible with the Process.

Four types of Processes are specified in the following: App, Device, Service, and User. This characterisation has the scope of facilitating the reading of this Technical Specification as all types of Processes are simply Processes. A Process may be defined as performing all the Functions listed in the following.

7.2 App

Functions An instance of an application-specific Program executed on a Device.

Functional Requirements An App may need to be Certified to operate in a given M-Instance.

7.3 Device

A Device may:

1. UM-Capture Data from a U-Location.
2. Additionally, UM-Send Data and Metadata to a User.

Functions

and/or

1. MU-Actuate an Entity at a U-Location.
2. The Device may MU-Actuate after it has "MM-Sent an Item from an M-Location to the Device".

Functional Requirements

To connect and interoperate with an M-Instance, a Device may need to expose its Process Capabilities to the M-Instance.

7.4 Service

Purpose

A Process that provides specific Functionalities to other Processes.

Functional Requirements

A Service may be:

1. One of the natively-supported Services of an M-Instance.
2. Hosted by the M-Instance but provided by a third party.

Services are often specific of an M-Instance. Table 2 names and defines the role of Services that are specified by MPAI (in bold) and some Services of general use referenced by the Verification Use Cases.

Table 2 - A list of Services referenced in the Verification Use Cases

Service	Description
Activity	Keeps track of Processes starting a session.
Authentication	Support User requests for Item Authentication
Author	Enables Users to acquire or develop Items.
Communication	Enables communication between Processes.
Conversion	Converts Data of Items to a different format based on provided Qualifier.
Discovery	Supports Users looking for Items of Processes with given characteristics.
Execution	Executes a Program.
ExIm	Manages input from and output to the Universe.
Identification	Identifies Data and Metadata as Items.
Information	Supports Users looking for information about Items.
Interpretation	Supports users requesting Interpretation of an Item.
Location	Manages positioning and motion of Items.
Parcel	Offers Parcels of M-Instance land to Users.
Post	Posts Items on which a User holds Rights.
Register	Enables humans to Register with an M-Instance.
Rights	Manages the Rights of Processes.
Transact	Enables Users to perform Transactions.

7.5 User

Functions

A Process representing a Registered human.

Functional Requirements

1. A User may perform the following functions:
 - 1.1. Act as the interface of the human with the M-Instance.
 - 1.2. Render the User as a Persona UM-Animated by a Stream or by an autonomous agent.
2. Animation results from an MM-/UM-Anim Action and enabled by a Program run by the User.
3. The Animation Program may be part of the Processes registered by a human or provided by the M-Instance.
4. An M-Instance may not require Registration.

8 Items

An Item is a Data Type specified by MMM-TEC or other MPAI Technical Specifications for use in an M-Instance. All Items used by MMM-TEC V2.0 are listed in Table 1.

An Item may be produced:

1. By the activity of a Process, e.g., by sensing Media and producing Data, Qualifier, and Rights from the Universe.
2. As a result of a request made by a human through a Process of that human or directly to the M-Instance.

MMM-TEC V2.0 specifies or references from other MPAI Technical Specifications a large number of Items. The specification of each Item includes Definition, Functional Requirements, JSON Schema and Semantics.

An Item may optionally include:

1. Data *Qualifier* signalling the relevant characteristics or technologies of a Data Type.
2. *Rights* that are exercised by certain Processes on an Item at certain M- or U-Locations and at certain Times.

The specification of all Data Types – specified by MMM-TEC V2.0 or by other Technical Specifications – are reported below in the following order: MPAI-MMC V2.3, MMM-TEC V2.0, MPAI-OSD V1.3, and MPAI-PAF V1.4.

8.1 Cognitive State

8.1.1 Definition

Cognitive State is a Personal Status Factor representing the internal state of an Entity such as “surprised” or “interested”.

8.1.2 Functional Requirements

Cognitive State can be expressed via several *Modalities*: Text, Speech, Face, and Gestures. (Other Modalities, such as body posture, may be handled in future MPAI Versions.)

Within a given Modality, Cognitive State can be analysed and interpreted via various *Descriptors*. For example, when expressed via Speech, the elements may be expressed through combinations of such features as prosody (pitch, rhythm, and volume variations); separable speech effects (such as degrees of voice tension, breathiness, etc.); and vocal gestures (laughs, sobs, etc.).

Cognitive State is represented by a standard set of labels and associated semantics by two tables:

- A *Label Set Table* containing descriptive labels relevant to the Factor in a three-level format:
 - The CATEGORIES column specifies the relevant categories using nouns (e.g., “ANGER”).
 - The GENERAL ADJECTIVAL column gives adjectival labels for general or basic labels within a category (e.g., “angry”).
 - The SPECIFIC ADJECTIVAL column gives more specific (sub-categorised) labels in the relevant category (e.g., “furious”).
- A *Label Semantics Table* providing the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns of the Label Set Table. For example, for “angry” the semantic gloss is “emotion due to perception of physical or emotional damage or threat.”

Table 1 gives the standardised three-level Basic Cognitive State Label Set.

Table 3 - Basic Cognitive State Label Set

COGNITIVE CATEGORIES	GENERAL ADJECTIVAL	SPECIFIC ADJECTIVAL
AROUSAL	aroused/excited/energetic	cheerful playful lethargic sleepy expectant/anticipating
ATTENTION	attentive	thoughtful distracted/absent-minded vigilant hopeful/optimistic
BELIEF	credulous	sceptical
INTEREST	interested	fascinated curious bored
SURPRISE	surprised	astounded startled
UNDERSTANDING	comprehending	uncomprehending bewildered/puzzled

Table 4 provides the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns above.

Table 4 - Basic Cognitive State Semantics Set

ID	Cognitive State	Meaning
1	aroused/excited/energetic	cognitive state of alertness and energy
2	astounded	high degree of surprised
3	attentive	cognitive state of paying attention
4	bewildered/puzzled	high degree of incomprehension
5	bored	not interested
6	cheerful	energetic combined with and communicating happiness

7	comprehending	cognitive state of successful application of mental models to a situation
8	credulous	cognitive state of conformance to mental models of a situation
9	curious	interest due to drive to know or understand
10	distracted/absent-minded	not attentive to present situation due to competing thoughts
11	expectant/anticipating	attentive to (expecting) future event or events
12	fascinated	high degree of interest
13	interested	cognitive state of attentiveness due to salience or appeal to emotions or drives
14	lethargic	not aroused
15	playful	energetic and communicating willingness to play
16	sceptical	not credulous
17	sleepy	not aroused due to need for sleep
18	surprised	cognitive state due to violation of expectation
19	startled	surprised by a sudden event or perception
20	surprised	cognitive state due to violation of expectation
21	thoughtful	attentive to thoughts
22	uncomprehending	not comprehending

These sets have been compiled in the interests of basic cooperation and coordination among AIM submitters and vendors complemented by a procedure whereby AIM submitters may propose extended or alternate sets for their purposes.

An Implementer wishing to extend or replace a *Label Set Table* for one of the three Factors is requested to do the following:

1. Create a new Label Set Table where:
 1. Proposed additions are clearly marked (in case of extension).
 2. b. All the elements of the target Cognitive State and levels (up to 3) are listed (in case of replacement).
2. Create a new Label Semantics Table where the semantics of elements of the Cognitive State is:
 1. Added to the semantics of the existing Cognitive State (in case of extension).
 2. Provided (in case of replacement). The submitted semantics should have a level of detail comparable to the semantics given in the current *Label Semantics Table*.
3. Submit both tables to the [MPAI Secretariat](#).

The appropriate MPAI Development Committee will examine the proposed extension or replacement. Only the adequacy of the proposed new tables in terms of clarity and completeness will be considered. In case the new tables are not clear or complete, a revision of the tables will be requested.

The accepted Cognitive State Set will be identified as proposed by the submitter and reviewed by the appropriate MPAI Committee and posted to the [MPAI web site](#).

The versioning system is based on a name – MPAI for MPAI-generated versions or “organisation name” for the proposing organisation – with a suffix m.n where m indicates the version and n indicated the subversion.

8.1.3 Syntax

<https://schemas.mpai.community/MMC/V2.3/data/CognitiveState.json>

8.1.4 Semantics

Label	Size	Description
Header	N1 Bytes	Entity Cognitive State Header
- Standard-EntityCognitiveState	9 Bytes	The characters “MMC-ECS-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
EntityCognitiveStateID	N5 Bytes	Identifier of CogState.
EntityCognitiveStateSpaceTime	N7 Bytes	Space-Time info of CogState.
EntityCognitiveStateData	N8 Bytes	Data associated to CogState.
- FusedCogState	N9 Bytes	Integrated CogState Value.
- TextCogState	N10 Bytes	Text CogState Value.
- SpeechCogState	N11 Bytes	Speech CogState Value.
- FaceCogState	N12 Bytes	Face CogState Value.
- GestureCogState	N13 Bytes	Gesture CogState Value.
DescrMetadata	N14 Bytes	Descriptive Metadata

8.1.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Entity Cognitive State (MMC-ECS) if:

1. The Data validates against the Entity Cognitive State ’s JSON Schema.
2. All Data in the Entity Cognitive State ’s JSON Schema:
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.2 Emotion

8.2.1 Definition

Emotion is a Personal Status Factor representing the internal state of an Entity such as that resulting from its interaction with the Context, such as “Angry”, “Sad”, “Determined”.

8.2.2 Functional Requirements

Emotion can be expressed via several *Modalities*: Text, Speech, Face, and Gestures. (Other Modalities, such as body posture, may be handled in future MPAI Versions.)

Within a given Modality, Emotion can be analysed and interpreted via various *Descriptors*. For example, when expressed via Speech, the elements may be expressed through combinations of such features as prosody (pitch, rhythm, and volume variations); separable speech effects (such as degrees of voice tension, breathiness, etc.); and vocal gestures (laughs, sobs, etc.).

Emotion is represented by a standard set of labels and associated semantics by two tables:

- A *Label Set Table* containing descriptive labels relevant to the Factor in a three-level format:
 - The CATEGORIES column specifies the relevant categories using nouns (e.g., “ANGER”).
 - The GENERAL ADJECTIVAL column gives adjectival labels for general or basic labels within a category (e.g., “angry”).
 - The SPECIFIC ADJECTIVAL column gives more specific (sub-categorised) labels in the relevant category (e.g., “furious”).
- A *Label Semantics Table* providing the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns of the Label Set Table. For example, for “angry” the semantic gloss is “emotion due to perception of physical or emotional damage or threat.”

Table 5 gives the standardised three-level Basic Emotion Set partly based on Paul Eckman [19].

Table 5 - Basic Emotion Label Set

EMOTION CATEGORIES	GENERAL ADJECTIVAL	SPECIFIC ADJECTIVAL
ANGER	angry	furious irritated frustrated
CALMNESS	calm	peaceful/serene resigned
DISGUST	disgusted	repulsed
FEAR	fearful/scared	terrified anxious/uneasy
HAPPINESS	happy	joyful content delighted amused
HURT	hurt jealous	insulted/offended resentful/disgruntled bitter
PRIDE/SHAME	proud ashamed	guilty/remorseful/sorry embarrassed
RETROSPECTION	nostalgic	homesick

SADNESS	sad	lonely grief-stricken depressed/gloomy disappointed
---------	-----	--

Table 6 provides the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns above.

Table 6 - Basic Emotion Semantics Set

ID	Emotion	Meaning
1	amused	positive emotion combined with interest (cognitive state)
2	angry	emotion due to perception of physical or emotional damage or threat
3	anxious/uneasy	low or medium degree of fear, often continuing rather than instant
4	ashamed	emotion due to awareness of violating social or moral norms
5	bitter	persistently angry due to disappointment or perception of hurt or injury
6	calm	relatively lacking emotion
7	content	medium or low degree of happiness, continuing rather than instant
8	delighted	high degree of happiness, often combined with surprise
9	depressed/ gloomy	high degree of sadness, continuing rather than instant, combined with lethargy (see AROUSAL)
10	disappointed	sadness due to failure of desired outcome
11	disgusted	emotion due to urge to avoid, often due to unpleasant perception or disapproval
12	embarrassed	shame due to consciousness of violation of social conventions
13	fearful/scared	emotion due to anticipation of physical or emotional pain or other undesired event or events
14	frustrated	angry due to failure of desired outcome
15	furious	high degree of angry
16	grief-stricken	sadness due to loss of an important social contact
17	happy	positive emotion, often continuing rather than instant
18	homesick	sad due to absence from home
19	hurt	emotion due to perception that others have caused social pain or embarrassment
20	insulted/offended	emotion due to perception that one has been improperly treated socially
21	irritated	low or medium degree of angry
22	jealous	emotion due to perception that others are more fortunate or successful
23	joyful	high degree of happiness, often due to a specific event
24	repulsed	high degree of disgusted
25	lonely	sad due to insufficient social contact
26	mortified	high degree of embarrassment
27	nostalgic	emotion associated with pleasant memories, usually of long before
28	peaceful/serene	calm combined with low degree of happiness

29 proud	emotion due to perception of positive social standing
30 resentful/disgruntled	emotion due to perception that one has been improperly treated
31 resigned	calm due to acceptance of failure of desired outcome, often combined with low degree of sadness
32 sad	negative emotion, often continuing rather than instant, often associated with a specific event
33 terrified	high degree of fear

These sets have been compiled in the interests of basic cooperation and coordination among AIM submitters and vendors complemented by a procedure whereby AIM submitters may propose extended or alternate sets for their purposes.

An Implementer wishing to extend or replace a *Label Set Table* for Emotion is requested to do the following:

1. Create a new Label Set Table where:
 1. Proposed additions are clearly marked (in case of extension).
 2. b. All the elements of the Emotion and levels (up to 3) are listed (in case of replacement).
2. Create a new Label Semantics Table where the semantics of elements of the Emotion is:
 1. Added to the semantics of the existing Emotion (in case of extension).
 2. Provided (in case of replacement).
The submitted semantics should have a level of detail comparable to the semantics given in the current *Label Semantics Table*.
3. Submit both tables to the [MPAI Secretariat](#).

The appropriate MPAI Development Committee will examine the proposed extension or replacement. Only the adequacy of the proposed new tables in terms of clarity and completeness will be considered. In case the new tables are not clear or complete, a revision of the tables will be requested.

The accepted Emotion Set will be identified as proposed by the submitter and reviewed by the appropriate MPAI Committee and posted to the [MPAI web site](#).

The versioning system is based on a name – MPAI for MPAI-generated versions or “organisation name” for the proposing organisation – with a suffix m.n where m indicates the version and n indicated the subversion.

8.2.3 Syntax

<https://schemas.mpai.community/MMC/V2.3/data/Emotion.json>

8.2.4 Semantics

Label	Size	Description
Header	N1 Bytes	Entity Emotion Header
- Standard-EntityEmotion	9 Bytes	The characters “MMC-EEM-V”
- Version	N2 Bytes	Major version – 1 or 2 characters

- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
EntityEmotionID	N5 Bytes	Identifier of the Emotion.
EntityEmotionSpaceTime	N7 Bytes	Space-Time info of Emotion
EntityEmotionData	N8 Bytes	Data associated to Emotion.
- FusedEmotion	N9 Bytes	Integrated Emotion Value.
- TextEmotion	N10 Bytes	Text Emotion Value.
- SpeechEmotion	N11 Bytes	Speech Emotion Value.
- FaceEmotion	N12 Bytes	Face Emotion Value.
- GestureCogState	N13 Bytes	Gesture Emotion Value.
DescrMetadata	N14 Bytes	Descriptive Metadata

8.2.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Entity Emotion (MMC-EEM) if:

1. The Data validates against the Entity Emotion 's JSON Schema.
2. All Data in the Entity Emotion 's JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.3 Face Personal Status

8.3.1 Definition

Face Personal Status is a Data Type including the three *Factors*:

1. *Emotion* (such as “angry” or “sad”).
2. *Cognitive State* (such as “surprised” or “interested”).
3. *Social Attitude* (such as “polite” or “arrogant”).

of an Entity's Face Modality.

8.3.2 Functional Requirements

Face Personal Status is added for convenience. However, it is simply the Personal Status of the Face Modality.

8.3.3 Syntax

<https://schemas.mpai.community/MMC/V2.3/data/FacePersonalStatus.json>

8.3.4 Semantics

Label	Size	Description
Header	N1 Bytes	Header of Face Personal Status
- Standard	9 Bytes	The characters “MMC-FPS-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
FacePersonalStatusID	N5 Bytes	Identifier of Face Personal Status.
FacePersonalStatusSpaceTime	N6 Bytes	Space-Time info of Face Personal Status
FacePersonalStatus	N7 Bytes	Face Personal Status
- FaceCognitiveState	N8 Bytes	Cognitive State component of Face Personal Status
- FaceEmotion	N9 Bytes	Emotion component of Face Personal Status
- FaceSocialAttitude	N10 Bytes	Social Attitude component of Face Personal Status
DescrMetadata	N11 Bytes	Descriptive Metadata

8.3.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Face Personal Status (MMC-FPS) if:

1. The Data validates against the Face Personal Status’s JSON Schema.
2. All Data in the Face Personal Status’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conforms with their Data Qualifiers if present.

8.4 Gesture Personal Status

8.4.1 Definition

Gesture Personal Status is a Data Type including the three *Factors*:

1. *Emotion* (such as “angry” or “sad”).
2. *Cognitive State* (such as “surprised” or “interested”).
3. *Social Attitude* (such as “polite” or “arrogant”).

of an Entity's Gesture Modality.

8.4.2 Functional Requirements

Gesture Personal Status is added for convenience. However, it is simply the Personal Status of the Gesture Modality.

8.4.3 Syntax

<https://schemas.mpai.community/MMC/V2.3/data/GesturePersonalStatus.json>

8.4.4 Semantics

Label	Size	Description
Header	N1 Bytes	Header of Gesture Personal Status
- Standard	9 Bytes	The characters “MMC-GPS-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
GesturePersonalStatusID	N5 Bytes	Identifier of Gesture Personal Status.
GesturePersonalStatusSpaceTime	N6 Bytes	Space-Time info of Gesture Personal Status
GesturePersonalStatus	N7 Bytes	Gesture Personal Status
- GestureCognitiveState	N8 Bytes	Cognitive State component of Gesture Personal Status
- GestureEmotion	N9 Bytes	Emotion component of Gesture Personal Status
- GestureSocialAttitude	N10 Bytes	Social Attitude component of Gesture Personal Status
DescrMetadata	N11 Bytes	Descriptive Metadata

8.4.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V1.3 (MMC-GPS) if:

1. The Data validates against the Gesture Personal Status’s JSON Schema.
2. All Data in the Gesture Personal Status’s JSON Schema
 1. Have the specified type.
 2. Validate against their JSON Schemas.
 3. Conform with their Data Qualifiers if present.

8.5 Meaning

8.5.1 Definition

A Data Type representing the syntactic and semantic information of an input text. Meaning is synonym of Text Descriptors.

8.5.2 Functional Requirements

Meaning is used to extract information from text to help the Entity Dialogue Processing AIM to produce a response.

8.5.3 Syntax

<https://schemas.mpai.community/MMC/V2.3/data/Meaning.json>

8.5.4 Semantics

Label	Size	Description
Header	N1 Bytes	Meaning Header
- Standard-Meaning	9 Bytes	The characters “MMC-TXD-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
MeaningID	N5 Bytes	Identifier of Meaning.
Meaning	N6 Bytes	Data set of Meaning
- POS_tagging	N7 Bytes	Results of POS (Part of Speech, e.g., noun, verb, etc.) tagging including information on the question’s POS tagging set and tagged results.
- NE_tagging	N8 Bytes	Results of NE (Named Entity e.g., Person, Organisation, Fruit, etc.) tagging results including information on the question’s tagging set and tagged results.
- Dependency_tagging	N9 Bytes	Results of dependency (structure of the sentence, e.g., subject, object, head of relation, etc.) tagging including information on the question’s dependency tagging set and tagged results.
- SRL_tagging	N10 Bytes	Results of SRL (Semantic Role Labelling) tagging results including information on the question’s SRL tagging set and tagged results. SRL indicates the semantic structure of the sentence such as agent, location, patient role, etc.
DescrMetadata	N11 Bytes	Descriptive Metadata

8.5.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Meaning (MMC-MEA) if:

1. The Data validates against the Meaning’s JSON Schema.
2. All Data in the Meaning’s JSON Schema have the specified type.

8.6 Personal Status

8.6.1 Definition

A Data Type representing the information internal to an Entity that characterises their behaviour.

8.6.2 Functional Requirements

Personal Status is a Data Type composed of three *Factors*:

1. *Emotion* (such as “angry” or “sad”).
2. *Cognitive State* (such as “surprised” or “interested”).

3. *Social Attitude* (such as “polite” or “arrogant”).

Factors are expressed by *Modalities*: Text, Speech, Face, and Gestures. (Other Modalities, such as body posture, may be handled in future MPAI Versions.)

Within a given Modality, the Factors can be analysed and interpreted via various *Descriptors*. For example, when expressed via Speech, the elements may be expressed through combinations of such features as prosody (pitch, rhythm, and volume variations); separable speech effects (such as degrees of voice tension, breathiness, etc.); and vocal gestures (laughs, sobs, etc.).

Each of Emotion, Cognitive State, and Social Attitude Factors is represented by a standard set of labels and associated semantics. For each of these Factors, two tables are provided:

- A *Label Set Table* containing descriptive labels relevant to the Factor in a three-level format:
 - The CATEGORIES column specifies the relevant categories using nouns (e.g., “ANGER”).
 - The GENERAL ADJECTIVAL column gives adjectival labels for general or basic labels within a category (e.g., “angry”).
 - The SPECIFIC ADJECTIVAL column gives more specific (sub-categorised) labels in the relevant category (e.g., “furious”).
- A *Label Semantics Table* providing the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns of the Label Set Table. For example, for “angry” the semantic gloss is “emotion due to perception of physical or emotional damage or threat.”

These sets have been compiled in the interests of basic cooperation and coordination among AIM submitters and vendors complemented by a procedure whereby AIM submitters may propose extended or alternate sets for their purposes.

An Implementer wishing to extend or replace a *Label Set Table* for one of the three Factors is requested to do the following:

1. Create a new Label Set Table where:
 1. Proposed additions are clearly marked (in case of extension).
 2. All the elements of the target Factor and levels (up to 3) are listed (in case of replacement).
2. Create a new Label Semantics Table where the semantics of elements of the target Factor is:
 1. Added to the semantics of the existing target Factor (in case of extension).
 2. Provided (in case of replacement).
The submitted semantics should have a level of detail comparable to the semantics given in the current *Label Semantics Table*.
3. Submit both tables to the MPAI Secretariat (secretariat@mpai.community).

The appropriate MPAI Development Committee will examine the proposed extension or replacement. Only the adequacy of the proposed new tables in terms of clarity and completeness will be considered. In case the new tables are not clear or complete, a revision of the tables will be requested.

The accepted External Factor Set will be identified as proposed by the submitter and reviewed by the appropriate MPAI Committee and posted to the MPAI web site.

The versioning system is based on a name – MPAI for MPAI-generated versions or “organisation name” for the proposing organisation – with a suffix m.n where m indicates the version and n indicated the subversion.

8.6.3 Syntax

<https://schemas.mpai.community/MMC/V2.3/data/PersonalStatus.json>

8.6.4 Semantics

Label	Size	Description
Header	N1 Bytes	Personal Status Header
- Standard-PersonalStatus	9 Bytes	The characters “MMC-EPS-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
PersonalStatusID	N5 Bytes	Identifier of Meaning.
PersonalStatusSpaceTime	N6 Bytes	Space-Time info of PersonalStatus
PersonalStatus	N7 Bytes	Personal Status
- CognitiveState	N8 Bytes	Cognitive State component of Personal Status
- Emotion	N9 Bytes	Emotion component of Personal Status
- SocialAttitude	N10 Bytes	Social Attitude component of Personal Status
DescrMetadata	N11 Bytes	Descriptive Metadata

8.6.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Personal Status (MMC-EPS) if:

1. The Data validates against the Personal Status’s JSON Schema.
2. All Data in the Personal Status’s JSON Schema
 1. Have the specified type.
 2. Validate against their JSON Schemas.
 3. Conform with their Data Qualifiers if present.

8.7 Social Attitude

8.7.1 Definition

Social Attitude is a Personal Status Factor representing the internal state of an Entity related to the way it intends to position itself vis-à-vis the Context, e.g., “Respectful”, “Confrontational”, “Soothing”..

8.7.2 Functional Requirements

Social Attitude can be expressed via several *Modalities*: Text, Speech, Face, and Gestures. (Other Modalities, such as body posture, may be handled in future MPAI Versions.)

Within a given Modality, Social Attitude can be analysed and interpreted via various *Descriptors*. For example, when expressed via Speech, the elements may be expressed through combinations of such features as prosody (pitch, rhythm, and volume variations); separable speech effects (such as degrees of voice tension, breathiness, etc.); and vocal gestures (laughs, sobs, etc.).

Social Attitude is represented by a standard set of labels and associated semantics by two tables:

- A *Label Set Table* containing descriptive labels relevant to the Social Attitude in a three-level format:
 - The CATEGORIES column specifies the relevant categories using nouns (e.g., “ANGER”).
 - The GENERAL ADJECTIVAL column gives adjectival labels for general or basic labels within a category (e.g., “angry”).
 - The SPECIFIC ADJECTIVAL column gives more specific (sub-categorised) labels in the relevant category (e.g., “furious”).
- A *Label Semantics Table* providing the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns of the Label Set Table. For example, for “angry” the semantic gloss is “emotion due to perception of physical or emotional damage or threat.”

These sets have been compiled in the interests of basic cooperation and coordination among AIM submitters and vendors complemented by a procedure whereby AIM submitters may propose extended or alternate sets for their purposes.

An Implementer wishing to extend or replace a *Label Set Table* for Social Attitude is requested to do the following:

1. Create a new Label Set Table where:
 1. Proposed additions are clearly marked (in case of extension).
 2. All the elements of the target Social Attitude and levels (up to 3) are listed (in case of replacement).
2. Create a new Label Semantics Table where the semantics of elements of the Social Attitude is:
 1. Added to the semantics of the existing Social Attitude (in case of extension).
 2. Provided (in case of replacement).
The submitted semantics should have a level of detail comparable to the semantics given in the current *Label Semantics Table*.
3. Submit both tables to the [MPAI Secretariat](#).

Table 7 gives the standardised three-level Basic Social Attitude Set.

Table 7 - Basic Social Attitude Label Set

SOCIAL ATTITUDE	GENERAL	SPECIFIC
CATEGORIES	ADJECTIVAL	ADJECTIVAL
ACCEPTANCE	accepting exclusive/cliquish	welcoming/inviting friendly unfriendly/hostile

AGREEMENT, DISAGREEMENT	like-minded argumentative/disputatious	sarcastic
AGGRESSION	aggressive peaceful submissive	combative/belligerent passive-aggressive mocking
APPROVAL, DISAPPROVAL	admiring/approving disapproving indifferent	awed contemptuous
ACTIVITY, PASSIVITY	assertive passive	controlling permissive/lenient flexible
COOPERATION	cooperative/agreeable uncooperative	subversive/undermining uncommunicative stubborn disagreeable
RESPONSIVENESS	responsive/demonstrative emotional/passionate unresponsive/undemonstrative unemotional/detached	enthusiastic unenthusiastic passionate dispassionate sympathetic merciful
EMPATHY	empathetic/caring kind uncaring/callous	merciless/ruthless self-absorbed selfish/self-serving selfless/altruistic generous
EXPECTATION	optimistic pessimistic	positive sanguine negative/defeatist cynical
EXTROVERSION, INTROVERSION	outgoing/extroverted uninhibited/unreserved	sociable approachable
DEPENDENCE	dependent independent	helpless
MOTIVATION	motivated apathetic/indifferent	inspired excited/stimulated discouraged/dejected dismissive
OPENNESS, TRUST	open honest/sincere reasonable trusting	candid/frank closed/distant dishonest/deceitful responsible/trustworthy/dependable irresponsible distrustful
PRAISING, CRITICISM	laudatory critical	congratulatory flattering belittling

RESENTMENT, FORGIVENESS	forgiving unforgiving/vindictive /spiteful/ vengeful	understanding petty
SELF-PROMOTION	boastful modest/humble/ unassuming	
SELF-ESTEEM	conceited/vain self-deprecating/self-effacing	smug
SOCIAL DOMINANCE, CONFIDENCE	arrogant confident submissive	overconfident forward/presumptuous brazen
SEXUALITY	seductive lewd/bawdy/indecent prudish/priggish	suggestive/risqué/naughty
SOCIAL RANK	polite/courteous/respectful rude/disrespectful commanding/domineering pompous/pretentious obedient rebellious/defiant	condescending/patronizing/snobbish pedantic unaffected servile/obsequious

Table 8 provides the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns above.

Table 8 - Basic Social Attitude Semantics Set

ID	Social Attitude	Meaning
1	accepting	attitude communicating willingness to accept into relationship or group
2	admiring/approving	attitude due to perception that others' actions or results are valuable
3	aggressive	tending to physically or metaphorically attack
4	apathetic/indifferent	showing lack of interest
5	approachable	sociable and not inspiring inhibition
6	argumentative	tending to argue or dispute
7	arrogant	emotion communicating social dominance
8	assertive	taking active role in social situations
9	awed	approval combined with incomprehension or fear
10	belittling	criticising by understating victim's achievements, personal attributes, etc.
11	boastful	tending to praise or promote self
12	brazen	high degree of forwardness/presumption
13	candid/frank	open in linguistic communication
14	closed/distant	not open
15	commanding/domineering	tending to assert right to command
16	combative/belligerent	high degree of aggression, often physical

17	communicative	evincing willingness to communicate as needed
18	conceited/vain	evincing undesirable degree of self-esteem
19	condescending / patronizing / snobbish	disrespectfully asserting superior social status, experience, knowledge, or membership
20	confident	attitude due to belief in own ability
21	congratulatory	wishing well related to another's success or good luck
22	contemptuous	high degree of disapproval and perceived superiority
23	controlling	undesirably assertive
24	cool	repressing outward reaction, often to indicate confidence or dominance, especially when confronting aggression, panic, etc.
25	cooperative/agreeable	communicating willingness to cooperate
26	critical	attitude expressing disapproval
27	cynical	habitually negative, reflecting disappointment or disillusionment
28	dependent	evincing inability to function without aid
29	discouraged/dejected	unmotivated because goals or rewards were not achieved
30	disagreeable	not agreeable
31	disapproving	not approving
32	dishonest/deceitful/insincere	not honest
33	dismissive	actively indicating lack of interest or motivation
34	distrustful	not trusting
35	emotional/passionate	high degree of responsiveness to emotions
36	empathetic/caring	interested in or vicariously feeling others' emotions
37	enthusiastic	high degree of positive response, especially to specific occurrence
38	excited/stimulated	attitude indicating cognitive and emotional arousal
39	exclusive/cliqish	not welcoming into a social group
40	flattering	praising with intent to influence, often insincere
41	flexible	willing to adjust to changing circumstances or needs
42	forward/presumptuous	not observing norms related to intimacy or rank
43	forgiving	tending to forgive improper behaviour
44	friendly	welcoming or inviting social contact
45	generous	tending to give to others, materially or otherwise
46	guilty/remorseful/sorry	regret due to consciousness of hurting or damaging others
47	helpless	high degree of dependence
48	honest/sincere	tending to communicate without deception
49	independent	not dependent
50	indifferent	neither approving nor disapproving
51	inhibited/ reserved/ introverted/ withdrawn	unable or unwilling to participate socially
52	inspired	motivated by some person, event, etc.

53	irresponsible	not responsible
54	kind	tending to act as motivated by empathy or sympathy
55	laudatory	praising
56	lewd/bawdy/indecent	evoking sexual associations in ways beyond social norms
57	like-minded	attitude expressing agreement
58	melodramatic	high or excessive degree of responsiveness or demonstrativeness
59	merciful	tending to avoid punishing others, often motivated by empathy or sympathy
60	merciless/ruthless	not merciful
61	mocking	communicating non-physical aggression, often by imitating a disapproved aspect of the victim
62	modest/humble/unassuming	not boastful
63	motivated	communicating goal-directed emotion and cognitive state
64	negative/defeatist	expressing pessimism, often habitually
65	obedient	evinced tendency to obey commands
66	open	tending to communicate without inhibition
67	optimistic	tending to expect positive events or results
68	outgoing/ extroverted/ uninhibited/ unreserved	not inhibited
69	passive	not assertive
70	passive-aggressive	covertly and non-physically aggressive
71	peaceful	not aggressive
72	pedantic	excessively displaying knowledge or academic status
73	permissive	allowing activity that social norms might restrict
74	pessimistic	tending to expect negative events or results
75	petty	unforgiving concerning small matters
76	polite/courteous/respectful	tending to respect social norms
77	pompous/pretentious	excessively displaying social rank, often above actual status
78	positive	expressing optimism, often habitually
79	prudish/priggish	expressing disapproval of even minor social transgressions, especially related to sex
80	reasonable	evinced willingness to resolve issues through reasoning
81	rebellious/defiant	evinced unwillingness to obey
82	responsible/trustworthy/ dependable	evinced characteristics or behaviour that encourage trust
83	responsive/demonstrative	tending to outwardly react to emotions and cognitive states, often as prompted by others
84	rude/disrespectful	not polite or respectful
85	sanguine	low degree of optimism, often expressed calmly
86	sarcastic	communicating disagreement by pretending agreement in an obviously insincere manner
87	seductive	communicating interest in sexual or related contact

88	self-absorbed	not empathetic due to excessive interest in self
89	self-deprecating/self-effacing	tending to criticize, or fail to praise or promote, self
90	selfish/self-serving	not generous due to excessive interest in own benefit
91	selfless/altruistic	tending to act for others' benefit, sometimes exclusively
92	servile/obsequious	excessively and demonstrably obedient
93	shy	low degree of social inhibition
94	smug	evincing undesirable degree of self-esteem related to perceived triumph
95	stubborn	unwilling to change one's mind or behaviour
96	sociable	comfortable in social situations
97	submissive	tending to submit to social dominance
98	subversive/undermining	communicating intention to work against a victim's goals
99	suggestive/risqué/naughty	evoking sexual associations within social norms
100	supportive	communicating willingness to support as needed
101	sympathetic	empathetic related to others' hurt or suffering
102	trusting	tending to trust others
103	unaffected	not pompous
104	uncaring/callous	not empathetic or caring
105	uncommunicative	not communicative
106	uncooperative	not cooperative
107	understanding	forgiving due to ability to understand motivations
108	unemotional/dispassionate/detached	not emotional, even when emotion is expected
109	unenthusiastic	not enthusiastic
110	unfriendly/hostile	not friendly
111	unresponsive/undemonstrative	not responsive or demonstrative
112	welcoming/inviting	high degree of acceptance with emotional warmth

The appropriate MPAI Development Committee will examine the proposed extension or replacement. Only the adequacy of the proposed new tables in terms of clarity and completeness will be considered. In case the new tables are not clear or complete, a revision of the tables will be requested.

The accepted Social Attitude Set will be identified as proposed by the submitter and reviewed by the appropriate MPAI Committee and posted to the [MPAI web site](#).

The versioning system is based on a name – MPAI for MPAI-generated versions or “organisation name” for the proposing organisation – with a suffix m.n where m indicates the version and n indicated the subversion.

8.7.3 Syntax

<https://schemas.mpai.community/MMC/V2.3/data/SocialAttitude.json>

8.7.4 Semantics

Label	Size	Description
Header	N1 Bytes	Entity Social Attitude Header
- Standard-SocialAttitude	9 Bytes	The characters “MMC-ESA-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SocialAttitudeID	N5 Bytes	Identifier of the Social Attitude.
SocialAttitudeSpaceTime	N7 Bytes	Space-Time info of Social Attitude.
SocialAttitudeData	N8 Bytes	Data associated to Social Attitude.
- FusedSocAtt	N9 Bytes	Integrated Social Attitude Value.
- TextSocAtt	N10 Bytes	Text Social Attitude Value.
- SpeechSocAtt	N11 Bytes	Speech Social Attitude Value.
- FaceSocAtt	N12 Bytes	Face Social Attitude Value.
- GestureSocAtt	N13 Bytes	Gesture Social Attitude Value.
DescrMetadata	N14 Bytes	Descriptive Metadata

8.7.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Entity Social Attitude (MMC-ESA) if:

1. The Data validates against the Entity Social Attitude’s JSON Schema.
2. All Data in the Entity Social Attitude’s JSON Schema
 1. Have the specified type.
 2. Validate against their JSON Schemas.
 3. Conform with their Data Qualifiers if present.

8.8 Speech Descriptors

8.8.1 Definition

A Data Type representing characteristic elements extracted from the input speech, specifically Pitch, Intensity, Tempo, Personal Status, and NNSpeechFeatures in a period of time.

8.8.2 Functional Requirements

Speech Descriptors may include Neural Network Descriptors.

8.8.3 Syntax

<https://schemas.mpai.community/MMC/V2.3/data/SpeechDescriptors.json>

8.8.4 Semantics

Label	Size	Description
Header	N1 Bytes	Speech Descriptors Header
- Standard - SpeechDescriptors	9 Bytes	The characters “MMC-SPD-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	ID of the Metaverse Instance.
SpeechDescriptorsID	N5 Bytes	ID of Speech Descriptors.
SpeechDescriptorsData	N7 Bytes	Data associated with Input Text.
NNSpeechFeatures	N8 Bytes	The output vector of a neural-network using Speech as input.
Duration	N9 Bytes	The Time in which the Speech Descriptors are computed.
Pitch	N10 Bytes	Real number measuring the fundamental frequency of Speech in Hz (Hertz).
Intensity	N11 Bytes	Real number measuring the Energy of Speech in dBs (decibel).
Tempo	N12 Byte	Real number measuring the rate at which specified linguistic units (Phonemes, Syllables, or Words) are produced.
Personal Status	N13 Byte	The Speech Personal Status carried by the input speech.

8.8.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Speech Descriptors (MMC-SPD) if:

1. The Data validates against the Speech Descriptors’ JSON Schema.
2. All Data in the Speech Descriptors’ JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.9 Speech Personal Status

8.9.1 Definition

Speech Personal Status is a Data Type including the three *Factors*:

1. *Emotion* (such as “angry” or “sad”).
2. *Cognitive State* (such as “surprised” or “interested”).
3. *Social Attitude* (such as “polite” or “arrogant”).

of an Entity's Speech Modality.

8.9.2 Functional Requirements

Speech Personal Status is added for convenience. However, it is simply the Personal Status of the Speech Modality.

8.9.3 Syntax

<https://schemas.mpai.community/MMC/V2.3/data/SpeechPersonalStatus.json>

8.9.4 Semantics

Label	Size	Description
Header	N1 Bytes	Header of Speech Personal Status
- Standard	9 Bytes	The characters “MMC-SPS-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SpeechPersonalStatusID	N5 Bytes	Identifier of Speech Personal Status.
SpeechPersonalStatusSpaceTime	N6 Bytes	Space-Time info of Speech Personal Status
SpeechPersonalStatus	N7 Bytes	Speech Personal Status
- SpeechCognitiveState	N8 Bytes	Cognitive State component of Speech Personal Status
- SpeechEmotion	N9 Bytes	Emotion component of Speech Personal Status
- SpeechSocialAttitude	N10 Bytes	Social Attitude component of Speech Personal Status
DescrMetadata	N11 Bytes	Descriptive Metadata

8.9.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Speech Personal Status (MMC-SPS) if:

1. The Data validates against the Speech Personal Status’s JSON Schema.
2. All Data in the Speech Personal Status’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.10 Summary

8.10.1 Definition

A Data Type representing a text-based abridged outline of the utterance(s) of one or more Entities represented by their User ID and including Space-Time, Text, and Personal Statuses.

8.10.2 Functional Requirements

Summary includes:

1. Virtual Space where Summary was generated (M-Instance).
2. Space-Time information in the Virtual Instance.
3. Content for each speaking Entity:
 1. Text
 2. Space-Time information of the Entity the Text refers to.
 3. Personal Status of the Entity the Text refers to.

8.10.3 Syntax

<https://schemas.mpai.community/MMC/V2.3/data/Summary.json>

8.10.4 Semantics

Label	Size	Description
Header	N1 Bytes	Summary Header
- Standard-Item	9 Bytes	The characters “MMC-SUM-V”
- Version	N2 Byte	Major version – 1 or 2 Bytes
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 Bytes
MInstanceID	N4 Bytes	Identifier of M-Instance.
SummaryID	N5 Bytes	Identifier of the Summary.
SummarySpaceTime	N6 Bytes	Space-Time of Summary.
SummaryData[]	N7 Bytes	Data of Summary
- ReportedEntityID	N8 Bytes	ID of the Entity Reported in Summary
- ReportedEntityPersonalStatus	N9 Bytes	Personal Status of Entity Reported in Summary
- ReportedEntitySpaceTime	N10 Bytes	Time-Space info of Entity Reported in Summary
- ReportedEntityTextObject	N11 Bytes	Text Object of Entity Reported in Summary
SummaryData	N12 Bytes	Summary Data.
- SummaryDataLength	N13 Bytes	Number of Bytes in Summary Data
- SummaryDataURI	N14 Bytes	URI of Data of Summary Data
DescrMetadata	N15 Bytes	Descriptive Metadata

8.10.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Summary (MMC-SUM) if:

1. The Data validates against the Summary’s JSON Schema.
2. All Data in the Summary’s JSON Schema

1. Have the specified type
2. Validate against their JSON Schemas
3. Conform with their Data Qualifiers if present.

8.11 Text Descriptors

8.11.1 Definition

A Data Type representing the syntactic and semantic information of a Text.

8.11.2 Functional Requirements

Meaning is an extract of the information from text to help an Entity Dialogue Processing AIM to produce a response.

8.11.3 Syntax

<https://schemas.mpai.community/MMC/V2.3/data/TextDescriptors.json>

8.11.4 Semantics

Label	Size	Description
Header	N1 Bytes	Text Descriptors Header
- Standard - TextDescriptors	9 Bytes	The characters “MMC-TXD-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
TextDescriptorsID	N5 Bytes	ID of Text Descriptors
TextDescriptors	N6 Bytes	Identifier of the AV Object.
- POS_tagging	N7 Bytes	Results of POS (Part of Speech, e.g., noun, verb, etc.) tagging including information on the question’s POS tagging set and tagged results.
- NE_tagging	N8 Bytes	Results of NE (Named Entity e.g., Person, Organisation, Fruit, etc.) tagging results including information on the question’s tagging set and tagged results.
- Dependency_tagging	N9 Bytes	Results of dependency (structure of the sentence, e.g., subject, object, head of relation, etc.) tagging including information on the question’s dependency tagging set and tagged results.
- SRL_tagging	N10 Bytes	Results of SRL (Semantic Role Labelling) tagging results including information on the question’s SRL tagging set and tagged results. SRL indicates the semantic structure of the sentence such as agent, location, patient role, etc.
DesrMetadata	N11 Bytes	Descriptive Metadata

8.11.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Text Descriptors (MMC-TXD) if:

1. The Data validates against the Text Descriptors' JSON Schema.
2. All Data in the Text Descriptors' JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.12 Text Personal Status

8.12.1 Definition

Text Personal Status is a Data Type including the three *Factors*:

1. *Emotion* (such as “angry” or “sad”).
2. *Cognitive State* (such as “surprised” or “interested”).
3. *Social Attitude* (such as “polite” or “arrogant”).

of an Entity's Text Modality.

8.12.2 Functional Requirements

Text Personal Status is added for convenience. However, it is simply the Personal Status of the Text Modality.

8.12.3 Syntax

<https://schemas.mpai.community/MMC/V2.2/data/TextPersonalStatus.json>

8.12.4 Semantics

Label	Size	Description
Header	N1 Bytes	Header of Text Personal Status
- Standard	9 Bytes	The characters “MMC-TPS-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
TextPersonalStatusID	N5 Bytes	Identifier of Text Personal Status.
TextPersonalStatusSpaceTime	N6 Bytes	Space-Time info of Text Personal Status
TextPersonalStatus	N7 Bytes	Text Personal Status
- TextCognitiveState	N8 Bytes	Cognitive State component of Text Personal Status
- TextEmotion	N9 Bytes	Emotion component of Text Personal Status
- TextSocialAttitude	N10 Bytes	Social Attitude component of Text Personal Status
DescrMetadata	N11 Bytes	Descriptive Metadata

8.13 Account

8.13.1 Definition

An Item that

1. Uniquely references a human registered with an M-Instance
2. Includes the IDs of the human's Personal Profile, Processes, and their Internal Rights.

8.13.2 Functional Requirements

1. A human may have more than one Account in one or more M-Instances or M-Environments.
2. An Account includes:
 - o The ID of the Registered human.
 - o An M-Instance-specific subset/superset of the Registered human's Personal Data.
 - o The Rights held by the human's Processes.
 - o The IDs
 - Devices
 - Apps
 - Services
 - Users and their Personae.

8.13.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Account.json>

8.13.4 Semantics

Label	Size	Description
Header	N1 Bytes	Account Header
- Standard - Account	9 Bytes	The characters "MMM-ACC-V"
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
M-EnvironmentID	N5 Bytes	Identifier of M-Environment
AccountID	N6 Bytes	Identifier of Account.
Account	N7 Bytes	N is the number of Bytes composing the Account
- humanID	N8 Bytes	Identifier of human.
- PersonalProfileID	N9 Bytes	ID of Personal Profile.
- Processes[]	N10 Bytes	Set of Process
- ProcessID	N11 Bytes	ID of a Process
- RightsID	N12 Bytes	ID of Rights held by ProcessID.

- Personae[]	N13 Bytes Set of Personae of a User having User ID
- PersonaID	N14 Bytes Identifier of Persona
DescrMetadata	N15 Bytes Descriptive Metadata

8.14 Activity Data

8.14.1 Definition

An Item recording the IPP Protocols executed by a Process.

8.14.2 Functional Requirements

The scope of Activity Data can be defined by properly selecting components of the IPP Protocols.

8.14.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/ActivityData.json>

8.14.4 Semantics

Label	Size	Description
Header	N1 Bytes	Activity Data Header
- Standard - ActivityData	9 Bytes	The characters “MMM-ACD-V”
- Version	N2 Bytes	Major version expressed as 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version expressed as 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
ProcessID	N5 Bytes	ID of Process that generated the Activity Data
ActivityDataID	N6 Bytes	Identifier of Activity Data.
ActivityData[]	N7 Bytes	The list of Process Actions.
- IPPMessage	N8 Bytes	IDs of IPPMessage.
DescrMetadata	N9 Bytes	Descriptive Metadata.

8.15 Asset

8.15.1 Definition

An Item that can be Transacted.

8.15.2 Functional Requirements

An Asset

1. Preserves the Format of the Item that spawned it.
2. Includes the Time it was Modified as an Asset.
3. May be:
 - o MM-Embedded at an M-Location.

- Posted to a Service (e.g., a marketplace).
- 4. May include the Provenance, i.e., the history of Transactions.
- 5. May be Modified as an Item and the Modified Item may spawn a new Asset.

8.15.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Asset.json>

8.15.4 Semantics

Label	Size	Description
Header	N1 Bytes	Asset Header
- Standard-Asset	9 Bytes	The characters “MMM-ASS-V”
- Version	N2 Byte	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
AssetID	N5 Bytes	Identifier of Asset.
AssetData[]	N6 Bytes	Metadata of Asset
- SourceItemID	N7 Bytes	ID of Item that spawned this Asset
- AssetDate	N8 Bytes	Time Item was Modified to Asset
- Provenance	N9 Bytes	Provenance or its ID
DescrMetadata	N10 Bytes	Descriptive Metadata

8.16 Authentication

8.16.1 Definition

An Item generated by a Process to request a Service to confirm that an Item is what it claims to be or an Item generated by a Service containing the response to the request.

8.16.2 Functional Requirements

An authentication contains:

1. When a request is made:
 1. The Item (or its ID) to be Authenticated.
 2. The claimed identity of the Item.
2. When a response is made:
 1. A Yes or No.

Examples of Item that may be subject to Authentication are:

1. A Speech produced by a User.
2. The visual appearance (Persona) of a User.

8.16.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Authentication.json>

8.16.4 Semantics

Label	Size	Description
Header	N1 Bytes	Authentication Header
- Standard-Authentication	9 Bytes	The characters “MMM-AUT-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
AuthenticationID	N5 Bytes	Identifier of Authentication Item.
AuthenticationRequest	N6 Bytes	Authentication request
- Item	N7 Bytes	The ItemID.
- PerceptibleEntity	N8 Bytes	Item if Perceptible Entity.
- ClaimedIdentity	N9 Bytes	The claimed identity.
AuthenticationResponse	N10 Bytes	Y for confirmation, N for denial.
DescrMetadata	N11 Bytes	Descriptive Metadata

8.17 Basic Certificate

8.17.1 Definition

An Item attesting the suitability of a Process for specific MMM-TEC V2.0 usages. Other types of Certificates are possible and are listed in the Certificate Qualifier.

8.17.2 Functional Requirements

A Basic Certificate includes:

1. Certificate ID
2. Certificate Authority ID.
3. Actions for which the Process is Certified.

8.17.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/BasicCertificate.json>

8.17.4 Semantics

Label	Size	Description
Header	N1 Bytes	Certificate Header

- Standard - Certificate	9 Bytes	The characters “MMM-BCF-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
CertificateID	N5 Bytes	Identifier of Certificate.
CertificateTime	N6 Bytes	Validity of Certificate.
CertificationAuthorityID	N7 Bytes	Identifier of Certification Authority.
CertifiedProcessActions	N8 Bytes	Process Actions for which Process is Certified.
DescrMetadata	N8 Bytes	Descriptive Metadata.

8.18 Basic Discovery

8.18.1 Definition

An Item generated by a Process to requests that a Service discover Items and Processes. Other types of Discovery Items are possible and are listed in the Certificate Qualifier.

8.18.2 Functional Requirements

Discovery Request includes

1. Textual description of the features of the searched Item or Process.
2. Item ID, the Item, or the Process ID of which similar Items or Processes is being searched.
3. Model Rights for Use of Basic Discovery Response.

Examples of the object of a Discovery Request are:

1. Text specifying the characteristics of the Object or Process.
2. Item ID, [Perceptible Entity](#), or Process ID.
3. Model Rights to use the Basic Discovery Response.

Basic Discovery Response includes:

1. Item IDs, [Perceptible Entity](#), or Process IDs, and Rights to use the Basic Discovery Response.

8.18.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/BasicDiscovery.json>

8.18.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Discovery Header

- Standard-BasicDiscovery	9 Bytes	The characters “MMM-BDV-V”
- Version	N2 Byte	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
BasicDiscoveryID	N5 Bytes	Identifier of Basic Discovery.
BasicDiscovery[]	N6 Bytes	Data of Basic Discovery.
- BasicDiscoveryRequest	N7 Bytes	Data of Request
- Item	N8 Bytes	Item or Item ID provided as an example of what is searched.
- ProcessID	N9 Bytes	Process ID provided as an example of what is searched.
- PerceptibleEntity	N10 Bytes	A perceptible Item.
- MLocation	N11 Bytes	The M-Location where the Item should be searched.
- ModelRights	N12 Bytes	The text containing the expression of Rights for the type of Item being searched.
- BasicDiscoveryResponse	N13 Bytes	Data of Response
- Items[]	N14 Bytes	List of IDs of found Items.
- ItemID	N15 Bytes	the ID of a found Item.
- PerceptibleEntity	N16 Bytes	A perceptible Item.
- MLocationID	N17 Bytes	The M-Location where the Item has been MM-Embedded (can be more precise than the M-Location in the Request).
- Processes[]	N18 Bytes	List of IDs of found Processes.
- ProcessID	N19 Bytes	The ID of a found Process.
- Rights	N20 Bytes	Rights to use Discovery Response.
DescrMetadata	N21 Bytes	Descriptive Metadata

8.19 Basic Information

8.19.1 Definition

An Item generated by a Process to requests that a Service provide information on Items and Processes. Other types of Information Items are possible and are listed in the Information Qualifier.

8.19.2 Functional Requirements

Information Request includes Item ID or Perceptible Item on which Information is requested.

Examples of the object of an Information Request are:

1. Human-readable text.
2. Other human-perceptible Objects.
3. Model Rights for Use of Information Response.

Information Response includes:

1. Human readable Text.
2. Perceptible Item providing information on the Item.
3. Rights for Use of Information Response.

8.19.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/BasicInformation.json>

8.19.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Information Header
- Standard-BasicInformation	9 Bytes	The characters “MMM-BIF-V”
- Version	N2 Byte	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
BasicInformationID	N5 Bytes	Identifier of Basic Information .
BasicInformation[]	N6 Bytes	Data of Basic Information .
- BasicInformationRequest	N7 Bytes	Data of Request
- ItemID	N8 Bytes	Item ID
- PerceptibleEntity	N9 Bytes	The Item about which Interpretation is requested.
- ModelOutRights	N10 Bytes	Model Rights for Use of Information Response.
- BasicInformationResponse	N11 Bytes	Data of Response
- PerceptibleEntity	N12 Bytes	Perceptible Item about the Item.
- Rights	N13 Bytes	Rights to Use of Information Response.
DescrMetadata	N14 Bytes	Descriptive Metadata

8.20 Basic Interpretation

8.20.1 Definition

An Item generated by a Process to requests that a Service interpret an Items. Other types of Interpretation Items are possible and are listed in the Information Qualifier.

8.20.2 Functional Requirements

Interpretation Request includes

1. Item ID or Perceptible Item of which Interpretation is requested.
2. Model Rights to use Interpretation Response.

Examples of Items candidate for an Interpretation Request are:

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

Interpretation Response includes Text and/or Perceptible Item.

8.20.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/BasicInterpretation.json>

8.20.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Interpretation Header
- Standard-BasicInterpretation	9 Bytes	The characters “MMM-BIF-V”
- Version	N2 Byte	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
BasicInterpretationID	N5 Bytes	Identifier of Basic Interpretation.
BasicInterpretationData[]	N6 Bytes	Data of Basic Interpretation.
- BasicDiscoveryRequest	N7 Bytes	Data of Request
- ItemID	N8 Bytes	Item ID
- PerceptibleEntity	N9 Bytes	A Perceptible Item
- Model Rights	N10 Bytes	Model Rights to use Interpretation Response.
- BasicDiscoveryResponse	N11 Bytes	Data of Response
- PerceptibleItem	N12 Bytes	A Perceptible Item
- Rights	N10 Bytes	Rights to use Interpretation Response.
DescrMetadata	N13 Bytes	Descriptive Metadata

8.21 Basic M-Location

8.21.1 Definition

A region of an M-Instance with Space-Time attributes that is not exposed as further subdivided as a Location.

8.21.2 Functional Requirements

Basic M-Location is a portion of an M-Instance space having the following characteristics:

1. Has Space-Time attributes.
2. May cover a limited Space and Time or be unlimited in Space, Time or Space-Time.
3. Is composed of elementary space portions of the M-Instance space represented with a technology recorded in the Location Qualifier.
4. The Space-Time attributes of the Basic M-Location may override the Space-Time information of the technology used to represent the Basic M-Location's elementary space portions.
5. It is continuous, i.e., the Basic M-Location does not include space portions completely surrounded by the Basic M-Location's elementary space portions.
6. May be 0th element of a hierarchy of M-Locations.
7. If further subdivided, the Levels of the hierarchy are renumbered.
8. May have Rights attached governing the Process Actions that a Process may perform in the M-Location.
9. The Basic M-Location Rights may be Original or Granted.
10. A Basic M-Location inherits but may have more Rights than the Rights of the M-Location of Level 1.

8.21.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/BasicMLocation.json>

8.21.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic M-Location Header
- Standard-BasicMLocation	9 Bytes	The characters "MMM-BML-V"
- Version	N2 Bytes	Major version
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version
MInstanceID	N4 Bytes	Identifier of M-Instance.
MEnvironmentID	N5 Bytes	Identifier of M-Environment.
BasicMLocationID	N6 Bytes	Identifier of Basic M-Location.
BasicMLocationSpaceTime	N7 Bytes	Space-Time of the Basic M-Location.
BasicMLocationRights	N8 Bytes	Rights attached to Basic M-Location.
BasicMLocation[]	N9 Bytes	Set of Data defining Basic-M-Location.

- LocationQualifier	N10 Bytes Qualifier of Basic M-Location
DescrMetadata	N11 Bytes Descriptive Metadata.

8.22 Basic U-Location

8.22.1 Definition

A region of the Universe with Space-Time attributes that is not (exposed as) further subdivided.

8.22.2 Functional Requirements

1. Has Space-Time attributes.
2. Is composed of elementary space portions of the U-Environment space represented with a technology recorded in the Location Qualifier.
3. It is continuous, i.e., the Basic U-Location does not include space portions completely surrounded by the Basic U-Location's elementary space portions.
4. May be 0th element of a hierarchy of U-Locations.
5. If further subdivided, the Levels of the hierarchy are renumbered.
6. May have Rights attached governing the Process Actions that a Process may perform in the M-Location.
7. The Basic U-Location Rights may be Original or Granted.
8. A Basic U-Location inherits but may have more Rights than the Rights of the M-Location of Level 1.

8.22.3 Syntax

<https://schemas.mpai.community/MMM/V1.0/data/BasicULocation.json>

8.22.4 Semantics

Label	Size	Description
Header	N1 Bytes	U-Location Header
- Standard	9 Bytes	The characters "MMM-BUL-V"
- Version	N2 Bytes	Major version – 1 or 2 Bytes
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 Bytes
UEnvironmentID	N4 Bytes	The ID of an administratively Identified portion of Universe
ULocationID	N5 Bytes	Identifier of U-Location.
BasicULocationRights	N6 Bytes	Rights attached to Basic U-Location.
BasicULocation[]	N7 Bytes	Set of Data defining Basic-U-Location.
- LocationQualifier	N8 Bytes	Qualifier of Basic U-Location
DescrMetadata	N9 Bytes	Descriptive Metadata.

8.23 Certificate

8.23.1 Definition

An Item Process attesting the suitability of a Process for specific usages that is not a Basic Certificate.

8.23.2 Functional Requirements

The Functional Requirements of a Certificate are specified by the entity issuing it that is recognised in the Certificate Qualifier.

8.23.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Certificate.json>

8.23.4 Semantics

Label	Size	Description
Header	N1 Bytes	Certificate Header
- Standard - Certificate	9 Bytes	The characters “MMM-CRF-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
CertificateID	N5 Bytes	Identifier of Certificate.
CertificateQualifier	N6 Bytes	Qualifies of Certificate.
DescrMetadata	N7 Bytes	Descriptive Metadata

8.24 Contract

8.24.1 Definition

An Item representing terms and conditions or a Program that is executed according to certain terms when conditions are met.

8.24.2 Functional Requirements

A Contract may be subject to Certification before it can be imported into an M-Instance.

8.24.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Contract.json>

8.24.4 Semantics

Label	Size	Description
Header	N1 Bytes	Contract header
- Standard	9 Bytes	The characters “MMM-CTR-V”
- Version	N2 Bytes	Major version – 1 or 2 Bytes
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 Bytes
M-InstanceID	N4 Bytes	Identifier of M-Instance.
ContractID	N5 Bytes	Identifier of Contract.
ContractQualifier	N6 Bytes	Qualifier of Contract.
DescrMetadata	N7 Bytes	Descriptive Metadata

8.25 Currency

8.25.1 Definition

The unit of measure of an Amount of a Value.

8.25.2 Functional Requirements

Amount is expressed as a string. The characters of ISO 4217 are used for the Real Currencies.

8.25.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Currency.json>

8.25.4 Semantics

Label	Size	Description
Header	N1 Bytes	Currency Header
– Standard	9 Bytes	The characters “MMM-CUR-V”
– Version	N2 Byte	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
CurrencyQualifier	N5 Bytes	The Qualifier of a Currency.
DescrMetadata	N6 Bytes	Descriptive Metadata

8.26 Discovery

8.26.1 Definition

An Item representing the description of the Item or Process (or their IDs) to be Discovered.

8.26.2 Functional Requirements

Discovery Request includes the Item or (its ID) of which Authentication is requested and other information proper of the Discovery Qualifier.

Examples of the object of a Discovery Request are:

1. An Item with specific characteristics.
2. A Process performing specific functions.

8.26.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Discovery.json>

8.26.4 Semantics

Label	Size	Description
Header	N1 Bytes	Discovery Header
- Standard-Discovery	9 Bytes	The characters “MMM-DIS-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
Discovery	N5 Bytes	Discovery as identified by its Qualifier.
- DiscoveryQualifier	N6 Bytes	Qualifier of Discovery Item
DescrMetadata	N9 Bytes	Descriptive Metadata

8.27 Identifier

8.27.1 Definition

An Item that references only one Process or Item in an M-Instance.

8.27.2 Functional Requirements

1. A Process or an Item may have more than one Identifier.
2. Each identifier of a Process or an Item identifies only one Process or Item.
3. An Identifier of a Process or an Item may have a hierarchical structure to enable Identification an Item or a Process based on the M-Instance and the M-Environment:
 1. M-InstanceID
 2. M-EnvironmentID
 3. ItemID/ProcessID
4. An Identifier may signal the type of Process or Item it identifies.

8.27.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Identifier.json>

8.27.4 Semantics

Label	Size	Description
Header	N1 Bytes	Identifier Header
- Standard - Identifier	9 Bytes	The characters “MMM-IDF-V”
- Version	N2 Bytes	Major version – 1 or 2 Bytes
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 Bytes
MInstanceID	N4 Bytes	Identifier of M-Instance.
MEnvironmentID	N5 Bytes	Identifier of M-Environment.
Identifier	N6 Bytes	Identifier of Process or Item
DescrMetadata	N6 Bytes	Descriptive Metadata

8.28 Information

8.28.1 Definition

An Item sent by a Process to a Service requesting information about an Item or Process (or their IDs) to obtain additional information on an Item or Process that is not otherwise available. The Service will respond with an Information response.

8.28.2 Functional Requirements

Information Request includes the Item or Process (or their IDs) of which Information is requested and other information proper of the Discovery Qualifier.

8.28.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Information.json>

8.28.4 Semantics

Label	Size	Description
Header	N1 Bytes	Information Header
- Standard-Information	9 Bytes	The characters “MMM-INF-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
InformationID	N5 Bytes	Identifier of Information Item.
InformationData	N6 Bytes	Set of Information Data
- InformationQualifier	N7 Bytes	Qualifier of Information Item
DescrMetadata	N9 Bytes	Descriptive Metadata

8.29 Interpretation

8.29.1 Definition

An Item sent by a Process to a Service requesting interpretation of an Item (or its ID) and the response of the Service. For example, an Interpretation Request of an Item may be:

8.29.2 Functional Requirements

Interpretation Request includes the Item or Process (or their IDs) of which Interpretation is requested and other information proper of the Interpretation Qualifier.

For example, an Interpretation Request of an Item may be:

Request	What	From
Extract Personal Status		Text
Extract Text		Speech
Translate Text or Speech in the target language		Text or Speech
Extract Personal Status		Face and/or Gesture

8.29.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Interpretation.json>

8.29.4 Semantics

Label	Size	Description
Header	N1 Bytes	Interpretation Header
- Standard-Interpretation	9 Bytes	The characters “MMM-INT-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
InterpretationID	N5 Bytes	Identifier of Interpretation Item.
Interpretation	N6 Bytes	Set of Interpretation Data
- InterpretationQualifier	N7 Bytes	ID of Interpretation Format
DescrMetadata	N9 Bytes	Descriptive Metadata

8.30 IPP Message

8.30.1 Definition

An Item that a Source Process sends to a Destination Process requesting the performance of a Process Action.

8.30.2 Functional Requirements

An Inter-Process Protocol (IPP) Message may be a Process Action Request (PA Request) or Process Action Response (PA Response) composed of the following elements. Additional Details at [Protocols](#).

IPP Message Elements	Description
Message ID	ID of PA Request or PA Response.
Response ID	ID of the Message to which this Message is a response. Absent when the Message is a request.
Source Process ID	ID of Process issuing Message.
Process Action	Combination of Action, Items/Processes and Complement (see below).
Resolution Service ID	Resolution Service to which the Message must be sent. Absent when the two Processes are in the same M-Instance.
Destination Process ID	ID of Process to which the Message is sent.
Acknowledgement	Acknowledgement = Y, if successful, or Error message, if failure.

8.30.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/IPPMessage.json>

8.30.4 Semantics

Label	Size	Description
Header	N1 Bytes	IPP Message Header
- Standard-IPPMessage	9 Bytes	The characters “MMM-IPM-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
IPPMessageID	N4 Bytes	Identifier of Message.
IPPMessage	N5 Bytes	Number of Bytes in Message Data
- ResponseID	N6 Bytes	As specified in Inter-Process Protocol .
- SourceProcessID	N7 Bytes	As specified in Inter-Process Protocol .
- ProcessAction	N8 Byte	As specified in Inter-Process Protocol .
- ResolutionServiceID	N9 Bytes	As specified in Inter-Process Protocol .
- DestinationProcessID	N10 Bytes	As specified in Inter-Process Protocol
- Acknowledgement	N11 Bytes	As specified in Inter-Process Protocol .
DescrMetadata	N12 Bytes	Descriptive Metadata.

8.31 M-Capabilities

8.31.1 Definition

An Item representing the capabilities of an M-Instance or M-Environment.

8.31.2 Functional Requirements

Capabilities may only include the Capabilities supported by an M-Instance or M-Environment, respectively.

An E-Environment or M-Environment may make none, some, or all of its Capabilities available, e.g.:

- Profile supported by the M-Environment or M-Environment.
- Actions that can be performed in the the M-Environment or M-Environment.
- Items supported by the M-Environment or M-Environment.
- Data Type Qualifiers supported by the Items.

8.31.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/MCapabilities.json>

8.31.4 Semantics

Label	Size	Description
Header	N1 Bytes	M-Capabilities Header
- Standard-MCapabilities	9 Bytes	The characters “MMM-MCP-V”
- Version	N2 Bytes	Major version – 1 or 2 Bytes
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 Bytes
MInstanceID	N4 Bytes	Identifier of M-Instance.
MEnvironmentID	N5 Bytes	Identifier of M-Environment.
MCapabilitiesID	N6 Bytes	Identifier of the M-Instance's or M-Environment's Capabilities.
MCapabilities	N7 Bytes	Set of M-Capabilities of a Profile.
Profile	N8 Bytes	“Baseline”, “Finance”, “Management”, or “High”.
Actions[]	N9 Bytes	Actions that can be performed in the the M-Instance or M-Environment.
Items[]	N10 Bytes	List of Item types supported by the M-Instance or M-Environment.
ItemQualifiers[]	N11 Bytes	List of supported Item Qualifiers.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.32 M-Environment

8.32.1 Definition

An Identified administrative subset of an M-Instance.

8.32.2 Functional Requirements

1. An M-Environment implements a Functional Profile of MPAI Metaverse Model – Architecture.
2. The Functional Profile shall be the same or included in the Functional Profile of the M-Instance it is part of.
3. The Item Formats and the Data Type Formats shall be the same or a subset of those supported by the M-Instance it is part of.

8.32.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/MEnvironment.json>

8.32.4 Semantics

Label	Size	Description
Header	N1 Bytes	MEnvironment Header
- Standard - MEnvironment	9 Bytes	The characters “MMM-MEN-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version –1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
MEnvironmentID	N5 Bytes	Identifier of M-EnvironmentID.
MCapabilities	N6 Bytes	Capabilities of M-Environment.
Rules	N7 Bytes	Rules of M-Environment.
DescrMetadata	N8 Bytes	Descriptive Metadata

8.33 Message

8.33.1 Definition

An Item that a Source Process MM-Sends to a Communication Service requesting it to deliver it to a Destination Process.

8.33.2 Functional Requirements

A Message may contain:

1. Item
2. Data
3. Data and Metadata

8.33.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Message.json>

8.33.4 Semantics

Label	Size	Description
Header	N1 Bytes	Message Header
- Standard-Message	9 Bytes	The characters “MMM-MSG-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
MessageID	N4 Bytes	Identifier of Message.
MessagePayload	N5 Bytes	Content of Message
MessagePayloadLength	N6 Bytes	Number of Bytes in Message Payload
MessagePayloadURI	N7 Bytes	URI of Message Data
DescrMetadata	N8 Bytes	Descriptive Metadata

8.34 M-Instance

8.34.1 Definition

A Virtual Space created according to the MMM-TEC Technical Specification of the MPAI-Metaverse Model.

8.34.2 Functional Requirements

1. An M-Instance implements a Functional Profile of MPAI Metaverse Model.
2. An M-Instance may:
 - o Add Functionalities not supported by the MMM-TEC. In this case, full Interoperability with other M-Instances may not be achieved.
 - o Enable the creation of M-Environments.
 - o Expose M-Capabilities.

8.34.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/MInstance.json>

8.34.4 Semantics

Label	Size	Description
Header	N1 Bytes	MInstance Header
- Standard - MInstance	9 Bytes	The characters “MMM-MIN-V”
- Version	N2 Bytes	Major version expressed as 1 or 2 characters
- Dot-separator	1 Byte	The character “.”

- Subversion	N3 Byte	Minor version expressed as 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
Coordinates	N5 Bytes	Coordinate System used by M-Instance.
MCapabilities	N6 Bytes	Capabilities of M-Instance.
Rules	N7 Bytes	Rules of M-Instance.
DescrMetadata	N8 Bytes	Descriptive Metadata.

8.35 M-Location

8.35.1 Definition

A region of an M-Instance with Space-Time attributes that is (exposed as) further subdivided.

8.35.2 Functional Requirements

An M-Location is a portion of an M-Instance space with the following characteristics:

1. Has Space-Time attributes.
2. May cover a limited Space and Time or be unlimited in Space, Time or Space-Time.
3. Is an element of a hierarchy of M-Locations with level $L > 0$.
4. It is composed of Locations of Level L-1.
5. Need not be continuous, i.e., a Level L Location may include space portions completely surrounded by Level L-1 Locations.
6. May have Rights attached governing the Process Actions that a Process may perform in the M-Location.
7. The M-Location Rights may be Original or Granted.
8. An M-Location of Level L-1 inherits but may have more Rights than the Rights of the M-Location of Level L.

A building is an example of M-Location with the following Levels:

1. The building has $L=3$
2. An apartment has $L=2$
3. A room has $L=1$
4. A desk has $L=0$

8.35.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/MLocation.json>

8.35.4 Semantics

Label	Size	Description
Header	N1 Bytes	M-Location Header
- Standard-MLocation	9 Bytes	The characters "MMM-MLC-V"
- Version	N2 Bytes	Major version
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version

MInstanceID	N4 Bytes	Identifier of M-Instance.
MEnvironmentID	N5 Bytes	Identifier of M-Environment.
MLocationID	N6 Bytes	Identifier of M-Location.
MLocation	N7 Bytes	Set of Data defining M-Location.
- BasicMLocations[]	N8 Bytes	Set of Data defining M-Location.
- BasicMLocation	N9 Bytes	A Basic M-Location
-		
BasicMLocationSpaceTime	N10 Bytes	Spatial Attitude and Time of Basic MLocation.
- Mlocations[]	N11 Bytes	Level L-1 M-Location (L may be 0, i.e., Basic M-Location).
- MLocation	N12 Bytes	A Basic M-Location
- MLocationSpaceTime	N13 Bytes	Spatial Attitude and Time of Basic MLocation.
Rights	N14 Bytes	Rights held or granted to M-Location.
DescrMetadata	N15 Bytes	Descriptive Metadata.

8.36 P-Capabilities

8.36.1 Definition

An Items containing the list of Processes and Actions that it can perform.

8.36.2 Functional Requirements

1. The Capabilities of a Processes:
 - o List of Process Actions
 - o Cost for performing each Process Action.
2. In particular
 - o If Device: which Apps it has onboard.
 - o If User: the ID of human represented by User.

8.36.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/PCapabilities.json>

8.36.4 Semantics

Label	Size	Description
Header	N1 Bytes	P-Capabilities Header
- Standard	9 Bytes	The characters “MMM-PCP-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
MEnvironmentID	N5 Bytes	Identifier of M-Environment.
PCapabilitiesID	N6 Bytes	Identifier of P-Capabilities
PCapabilitiesData	N7 Bytes	Identifier of E-Capabilities

- SupportedApps[] N8 Bytes If Process is Device
- humanID N9 Bytes If Process is User
- Actions[] N10 Bytes Actions the Process can perform.
- Items[] N11 Bytes List of Items and Item Data Qualifiers.
- ItemQualifiers[] N12 Bytes The Process Actions that the Process can perform.
- WalletID** N13 Bytes The The ID of the Wallet the Process is connected to.
- DescrMetadata** N14 Bytes Descriptive Metadata

8.37 Personal Data

8.37.1 Definition

An Item containing a human's Personal Profile and Activity Data of their Users.

8.37.2 Functional Requirements

Personal Data includes a human's:

1. Personal Profile.
2. Activity Data of Users.
3. Personae of User.

8.37.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/PersonalData.json>

8.37.4 Semantics

Label	Size	Description
Header	N1 Bytes	Personal Data Header
- Standard PersonalData	9 Bytes	The characters "MMM-PDT-V"
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
PersonalDataID	N5 Bytes	Identifier of Account.
PersonalData	N6 Bytes	Personal Profile and Users' Activity Data.
- PersonalProfileID	N7 Bytes	Identifier of Personal Profile.
- Processes[]	N8 Bytes	Set of human's Processes in the M-Instance
- ProcessID	N9 Bytes	Identifier of ProcessID.
- ActivityData[]	N10 Bytes	Set of ActivityData
- ActivityDataID	N11 Bytes	Identifier of ActivityDataID
- Personae[]	N12 Bytes	Set of Personae of a User
- PersonID	N13 Bytes	Identifier of PersonID
DescrMetadata	N14 Bytes	Descriptive Metadata

8.38 Personal Profile

8.38.1 Definition

An Item containing a human's Personal Data submitted when Registering with an M-Instance.

8.38.2 Functional Requirements

Personal Profile includes humanID and First Name, Last Name, Age, Nationality, and Email of the human. An M-Instance may extend or reduce the elements of Personal Profile.

8.38.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/PersonalProfile.json>

8.38.4 Semantics

Label	Size	Description
Header	N1 Bytes	Personal Profile Header
- Standard - PersonalProfile	9 Bytes	The characters “MMM-PPR-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
humanID	N5 Bytes	ID of the human the Personal Profile refers to.
PersonalProfileID	N6 Bytes	ID of Personal Profile.
PersonalProfile	N7 Bytes	The number of Bytes composing the Personal Profile.
- First Name	N8 Bytes	The human’s given name
- Last Name	N9 Bytes	The human’s family name
- Age	N10 Bytes	The human’s age
- Nationality	N11 Bytes	The human’s country
- Email	N12 Bytes	The human’s address
DescrMetadata	N13 Bytes	Descriptive Metadata

8.39 Process Action

8.39.1 Definition

An Item that specifies:

- The Action that a Process has performed, is performing, or is allowed to perform.
- Time, Source and Destination Complements, and Error Message.

8.39.2 Functional Requirements

Process Action Includes the following elements:

PA Element	Description
------------	-------------

<i>Time</i>	Time of Process Action request emission and Time of Process Action request execution.
<i>Action</i>	One of the standard Actions.
<i>S-Complements</i>	One or more Items or the Process participating in the execution of the Action at the S-Process side. Each Item/Process is preceded by <i>Nil, At, From, Of, or With</i> .
<i>D-Complements</i>	One or more Items or the Process participating in the execution of the Action at the Destination side. Each Item/Process is preceded by <i>Nil, At, From, Of, or With</i> .
<i>Error message</i>	Requested Process does not perform PA request.

8.39.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/ProcessAction.json>

8.39.4 Semantics

Label	Size	Description
Header	N1 Bytes	Process Action Data Header
- Standard - ProcessAction	9 Bytes	The characters “MMM-PAC-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N2 Byte	Minor version – 1 or 2 characters
M-InstanceID	N3 Bytes	Identifier of M-Instance.
ProcessActionID	N4 Bytes	ID of Process Action.
ProcessAction	N5 Bytes	The number of Bytes composing the Process Action.
- Time	N6 Bytes	Specified in Process Action .
- Action	N7 Bytes	Specified in Process Action .
- S-Complements	N8 Bytes	Specified in Process Action .
- NilProc	N9 Bytes	Process ID
- AtProc	N10 Bytes	<i>At</i> followed by Process ID
- FromProc	N11 Bytes	<i>From</i> followed by Process ID
- OfProc	N12 Byte	<i>Of</i> followed by Process ID
- ToProc	N13 Bytes	<i>To</i> followed by Process ID
- WithProc	N14 Bytes	<i>With</i> followed by Process ID
- NilItem	N15 Bytes	Item ID
- AtItem	N16 Bytes	<i>At</i> followed by Item ID
- FromItem	N17 Bytes	<i>From</i> followed by Item ID
- OfItem	N18 Bytes	<i>Of</i> followed by Item ID
- ToItem	N19 Bytes	<i>To</i> followed by Item ID
- WithItem	N20 Bytes	<i>With</i> followed by Item ID
- D-Complements	N21 Bytes	Specified in Process Action .
- NilProc	N22 Byte	Process ID

- AtProc	N23 Bytes	<i>At</i> followed by Process ID
- FromProc	N24 Bytes	<i>From</i> followed by Process ID
- OfProc	N25 Bytes	<i>Of</i> followed by Process ID
- ToProc	N26 Bytes	<i>To</i> followed by Process ID
- WithProc	N27 Bytes	<i>With</i> followed by Process ID
- NilItem	N28 Bytes	Item ID
- AtItem	N29 Bytes	<i>At</i> followed by Item ID
- FromItem	N30 Bytes	<i>From</i> followed by Item ID
- OfItem	N31 Byte	<i>Of</i> followed by Item ID
- ToItem	N32 Byte	<i>To</i> followed by Item ID
- WithItem	N33 Bytes	<i>With</i> followed by Item ID
- Error Message	N34 Bytes	Process Action-specific Message specified in Process Action .
DescrMetadata	N35 Bytes	Descriptive Metadata.

8.40 Program

8.40.1 Definition

An Item containing executable code, e.g., a Process or a Contract that is a Program.

8.40.2 Functional Requirements

A Program may convey Certification information.

8.40.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Program.json>

8.40.4 Semantics

Label	Size	Description
Header	N1 Bytes	Program Header
- Standard - Program	9 Bytes	The characters “MMM-PRG-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
ProgramID	N5 Bytes	Identifier of Program.
CertificateID	N6 Bytes	Certificate ID.
DescrMetadata	N7 Bytes	Descriptive Metadata

8.41 Provenance

8.41.1 Definition

A Data Type containing the list of all Transactions executed on an Asset, first and last included.

8.41.2 Functional Requirements

As a modified Asset is a new Asset, Provenance refers to an Asset with a given ID. An Asset may retain its history.

8.41.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Provenance.json>

8.41.4 Semantics

Label	Size	Description
Header	N1 Bytes	Provenance Header
- Standard - Provenance	9 Bytes	The characters “MMM-PRV-V”
- Version	N2 Byte	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
AssetID	N5 Bytes	Identifier of the Asset.
ProvenanceID	N6 Bytes	Identifier of Provenance.
Provenance[]	N7 Bytes	Set of Provenance Data.
- TransactionID	N8 Bytes	The IDs of the Transactions in the Provenance.
DescrMetadata	N9 Bytes	Descriptive Metadata.

8.42 Resolution

8.42.1 Definition

An Item containing an M-Instance_A's Process_A Request to ResolutionService_A to set up a session involving different M-Instances' Processes and the Responses of the Resolution Services and destination Processes.

8.42.2 Functional Requirements

An Resolution Item is transmitted:

1. In case of Request
 1. From Process_A to ResolutionService_A
 2. From ResolutionService_A to ResolutionService_B

3. From ResolutionService_B to Process_B
2. In case of Response
 1. From Process_B to ResolutionService_B
 2. From ResolutionService_B to ResolutionService_A
 3. From ResolutionService_A to Process_A

8.42.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Resolution.json>

8.42.4 Semantics

Label	Size	Description
Header	N1 Bytes	Resolution Header
- Standard-Resolution	9 Bytes	The characters “MMM-RSL-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
ResolutionID	N5 Bytes	Identifier of Authentication Item.
ProcessAToResolARequest	N6 Bytes	Request of Calling Process to own Resolution Service.
- MInstances[]	N7 Bytes	Set of M-Instances with a participating Process.
- MInstanceID	N8 Bytes	One M-Instance with a participating Process.
- Processes[]	N9 Bytes	Set of participating Processes in M-Instance.
- ProcessID	N10 Bytes	One Participating Process in M-Instance.
ResolAToResolBRequest	N11 Bytes	Request of Calling Process’s Resolution Service
- MInstances[]	N7 Bytes	Set of M-Instances with a participating Process.
- MInstanceID	N8 Bytes	One M-Instance with a participating Process.
- Processes[]	N9 Bytes	Set of participating Processes in M-Instance.
- ProcessID	N10 Bytes	One Participating Process in M-Instance.
ResolBToProcessBRequest	N14 Bytes	Response of Resolution ServiceB
- MInstances[]	N7 Bytes	Set of M-Instances with a participating Process.
- MInstanceID	N8 Bytes	One M-Instance with a participating Process.
- Processes[]	N9 Bytes	Set of participating Processes in M-Instance.
- ProcessID	N10 Bytes	One Participating Process in M-Instance.
ProcessBToResolBResponse	N14 Bytes	Response of ProgressB
- Accept		Y for confirmation, N for denial.
ResolBToResolAResponse	N11 Bytes	Response of Resolution ServiceB
- Error		Access to MetaverseA denied.
- ProcessID		ID of Process sending Accept.
- Accept		Accept of Process ID
ResolAToProcessAResponse	N11 Bytes	Response of Resolution ServiceA
- Error		Access to MetaverseA denied.
- ProcessID		ID of Process sending Accept.

- Accept Accept of Process ID
DescrMetadata N13 Bytes Descriptive Metadata

8.43 Rights

8.43.1 Definition

An Item representing the set of Process Actions and the corresponding Levels that a Process may perform.

8.43.2 Functional Requirements

Rights are determined by the following entities. The exact value may be dependent of the specific Action as specified in [Protocols](#).

Level One of *Internal*, i.e., granted by the M-Instance to the Process, *Acquired*, i.e., obtained by initiative of the Process. or *Granted*, i.e., conceded to the Process by another Process.

Process Action A Process Actions.

8.43.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Rights.json>

8.43.4 Semantics

Label	Size	Description
Header	N1 Bytes	Rights Header
- Standard - Rights	9 Bytes	The characters “MMM-RGT-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
RightsID	N5 Bytes	Identifier of Rights.
Rights[]	N6 Bytes	Set of Rights Data
- Level	N7 Bytes	One of Internal, Acquired, Granted.
- ProcessAction	N8 Bytes	As specified in Functional Requirements.
DescrMetadata	N9 Bytes	Descriptive Metadata

8.44 Rules

8.44.1 Definition

Rules is an Item representing what a Process may, may not or Must Perform.

8.44.2 Functional Requirements

Each Process Action that represents a Rule is preceded by May, May not, or Must exercise in the M-Instance:

May: A Process is allowed to exercise the Rights.

May not: A Process is not allowed to exercise the Rights.

Must: A Process must exercise the Rights.

8.44.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Rules.json>

8.44.4 Semantics

Label	Size	Description
Header	N1 Bytes	Header of Rules Item.
- Standard - Rules	9 Bytes	The characters “MMM-RUL-V”.
- Version	N2 Bytes	Major version expressed as 1 or 2 characters.
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version expressed as 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
MEnvironmentID	N5 Bytes	Identifier of M-Environment.
RulesID	N6 Bytes	The ID of the Rules
RulesData[]	N7 Bytes	The set of Rules.
- May[]	N8 Bytes	The collection of Process Actions preceded by May.
- ProcessAction	N9 Bytes	As defined by Process Actions .
- MayNot	N10 Bytes	The collection of Process Actions preceded by May not.
- ProcessAction	N11 Bytes	As defined by Process Actions .
- Must	N12 Bytes	The collection of Process Actions preceded by Must.
- ProcessAction	N3 Byte	As defined by Process Actions .
DescrMetadata	N14 Bytes	Descriptive Metadata.

8.45 Summary

8.45.1 Definition

A Data Type representing a text-based abridged outline of the utterance(s) of one or more Entities represented by their User ID and including Space-Time, Text, and Personal Statuses.

8.45.2 Functional Requirements

Summary includes:

1. Virtual Space where Summary was generated (M-Instance).
2. Space-Time information in the Virtual Instance.
3. Content for each speaking Entity:
 1. Text
 2. Space-Time information of the Entity the Text refers to.
 3. Personal Status of the Entity the Text refers to.

8.45.3 Syntax

<https://schemas.mpai.community/MMC/V2.3/data/Summary.json>

8.45.4 Semantics

Label	Size	Description
Header	N1 Bytes	Summary Header
- Standard-Item	9 Bytes	The characters “MMC-SUM-V”
- Version	N2 Byte	Major version – 1 or 2 Bytes
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 Bytes
MInstanceID	N4 Bytes	Identifier of M-Instance.
SummaryID	N5 Bytes	Identifier of the Summary.
SummarySpaceTime	N6 Bytes	Space-Time of Summary.
SummaryData[]	N7 Bytes	Data of Summary
- ReportedEntityID	N8 Bytes	ID of the Entity Reported in Summary
- ReportedEntityPersonalStatus	N9 Bytes	Personal Status of Entity Reported in Summary
- ReportedEntitySpaceTime	N10 Bytes	Time-Space info of Entity Reported in Summary
- ReportedEntityTextObject	N11 Bytes	Text Object of Entity Reported in Summary
SummaryData	N12 Bytes	Summary Data.
- SummaryDataLength	N13 Bytes	Number of Bytes in Summary Data
- SummaryDataURI	N14 Bytes	URI of Data of Summary Data
DescrMetadata	N15 Bytes	Descriptive Metadata

8.45.5 Conformance Testing

A Data instance Conforms with MPAI-MMC V2.3 Summary (MMC-SUM) if:

1. The Data validates against the Summary’s JSON Schema.
2. All Data in the Summary’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.46 Transaction

8.46.1 Definition

An Identified subset of the Universe.

8.46.2 Functional Requirements

The Format of a U-Environment is defined by its Location Qualifier.

8.46.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/UEnvironment.json>

8.46.4 Semantics

Label	Size	Description
Header	N1 Bytes	U-Environment Header
- Standard - UEnvironment	9 Bytes	The characters “MMM-UEN-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version –1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
UEnvironmentID	N5 Bytes	Identifier of U-EnvironmentID.
UEnvironmentQualifier	N6 Bytes	Qualifier of the U-Environment
DescrMetadata	N7 Bytes	Descriptive Metadata

8.47 U-Environment

8.47.1 Definition

An Identified subset of the Universe.

8.47.2 Functional Requirements

The Format of a U-Environment is defined by its Location Qualifier.

8.47.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/UEnvironment.json>

8.47.4 Semantics

Label	Size	Description
Header	N1 Bytes	U-Environment Header
- Standard - UEnvironment	9 Bytes	The characters “MMM-UEN-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”

- Subversion	N3 Byte	Minor version –1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
UEnvironmentID	N5 Bytes	Identifier of U-EnvironmentID.
UEnvironmentQualifier	N6 Bytes	Qualifier of the U-Environment
DescrMetadata	N7 Bytes	Descriptive Metadata

8.48 U-Location

8.48.1 Definition

A region of the Universe with Space-Time attributes.

8.48.2 Functional Requirements

A U-Location may:

1. Have an extension limited in Space and Time or be unlimited in Time.
2. Be composed of Basic U-Locations, e.g.:
 1. A room can be a Basic U-Location of the U-Location defined as an apartment.
 2. The apartment can be a U-Location of the "building" U-Location.

8.48.3 Syntax

<https://schemas.mpai.community/MMM/V2.0/data/ULocation.json>

8.48.4 Semantics

Label	Size	Description
Header	N1 Bytes	U-Location Header
- Standard-ULocation	9 Bytes	The characters “MMM-ULC-V”
- Version	N2 Bytes	Major version – 1 or 2 Bytes
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 Bytes
UEnvironmentID	N4 Bytes	The ID of an administratively Identified portion of Universe
ULocationID	N5 Bytes	Identifier of U-Location.
ULocation[]	N6 Bytes	Set of Locations and corresponding Time composing the U-Location.
- Location	N7 Bytes	Component U-Locations of the U-Location.
- Time	N8 Bytes	Time of validity of a U-Location in the U-Location
DescrMetadata	N9 Bytes	Descriptive Metadata.

8.49 Universe-Metaverse Map

8.49.1 Definition

An Item providing a list of U-Locations and corresponding M-Locations and/or Items with their Spatial Attitudes.

8.49.2 Functional Requirements

A Map includes a list composed of:

1. U-LocationID and Metadata related to the U-LocationID.
2. M-LocationID(s) and/or ItemIDs.
3. Metadata related to the M-LocationID(s) and/or EntityIDs corresponding to the U-LocationID.

8.49.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/UniverseMetaverseMap.json>

8.49.4 Semantics

Label	Size	Description
Header	N1 Bytes	Universe-Metaverse Map Header
- Standard - UniverseMetaverseMap	9 Bytes	The characters “MMM-UMM-V”
- Version	N2 Bytes	Major version
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version
MInstanceID	N4 Bytes	Identifier of M-Instance.
UEnvironmentID	N5 Bytes	Identifier of U-Environment.
MapData[]	N6 Bytes	Collection of Map elements.
- ULocationID	N7 Bytes	ID of specific U-Environment.
- MLocations[]	N8 Bytes	Corresponding M-Locations.
- MLocationID	N9 Bytes	ID of Corresponding M-Location
- Items[]	N10 Bytes	Items at M-LocationID
- ItemID	N11 Bytes	ID of an Item at M-LocationID
DescrMetadata	N12 Bytes	Descriptive Metadata.

8.50 Validation

8.50.1 Definition

An Item sent by a Process to a Service requesting it to validate a claim that another Process makes about its Rights and the response of the Service.

8.50.2 Functional Requirements

A Validation Item contains one of:

1. Validation request:
 1. The ProcessID
 2. The Rights to be Validated.
2. Validation response:
 1. A Yes or No.

8.50.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Validation.json>

8.50.4 Semantics

Label	Size	Description
Header	N1 Bytes	Validation Header
- Standard-Validation	9 Bytes	The characters “MMM-VLD-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
ValidationID	N5 Bytes	Identifier of Validation Item.
ValidationRequest	N6 Bytes	Validation request
- Process ID	N7 Bytes	The Process ID.
- ClaimedRights	N8 Bytes	The claimed Rights.
ValidationResponse	N8 Bytes	1 for confirmation, 0 for denial.
DescrMetadata	N9 Bytes	Descriptive Metadata

8.51 Value

8.51.1 Definition

An Item combining an Amount and a Currency.

8.51.2 Functional Requirements

The Currency includes currencies in the Universe and in the metaverses.

8.51.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Value.json>

8.51.4 Semantics

Label	Size	Description
Header	N1 Bytes	Value Header
- Standard - Wallet	9 Bytes	The characters “MMM-VAL-V”

- Version	N2 Byte	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
ValueID	N5 Bytes	Identifier of Value.
Value	N6 Bytes	N is the number of Bytes composing the Value
- Amount	N7 Bytes	The number of Bytes of the Amount
- Currency	3 Bytes	The 3-character code of the Currency
DescrMetadata	N8 Bytes	Descriptive Metadata

8.52 Wallet

8.52.1 Definition

A container of Values.

8.52.2 Functional Requirements

A Wallet

1. Includes the set of Values for each Currency.
2. May also include the IDs of the Transactions that cause the Wallet to have the current set of Values.

8.52.3 Syntax

<https://schemas.mpai.community/MMM4/V2.0/data/Wallet.json>

8.52.4 Semantics

Label	Size	Description
Header	N1 Bytes	Wallet Header
- Standard - Wallet	9 Bytes	The characters “MMM-WAL-V”
- Version	N2 Byte	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Byte	Minor version – 1 or 2 characters
M-InstanceID	N4 Bytes	Identifier of M-Instance.
WalletID	N5 Bytes	Identifier of Wallet.
PersonalProfileID	N6 Bytes	Personal Profile of the Wallet Holder.
Wallet	N7 Bytes	Wallet Values organised by Currency.
- ValueID[]	N8 Bytes	ID of Values for Currencies.
- TransactionID[]	N9 Bytes	ID of Transactions affecting a Value.

8.53 3D Model Event Descriptors

8.53.1 Definition

An Item including a series of 3D Model Scene Descriptors for a certain duration.

8.53.2 Functional Requirements

3D Model Event Descriptors contains 3D Model Scene Descriptors for a Time.

8.53.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/3DModelEventDescriptors.json>

8.53.4 Semantics

Label	Size	Description
Header	N1 Bytes	3D Model Event Descriptors Header
· Standard-3DModelEventDescriptors	9 Bytes	The characters “OSD-3DE-V”
· Version	N2 Bytes	Major version – 1 or 2 characters
· Dot-separator	1 Byte	The character “.”
· Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
EventID	N5 Bytes	Identifier of the Event.
EventSpaceTime	17 Bytes	Data about start and end Space-Time.
SceneDescriptors[]	N6 Bytes	Set of Scene Descriptors
- SceneDescriptors	N7 Bytes	Set of AV Scene Descriptors of IDs.
DescrMetadata	N8 Bytes	Descriptive Metadata

8.53.5 Conformance Testing

A Data instance Conforms with MPAI-OSD 3D Model Event Descriptors V1.3 (OSD-3DE) if:

1. The Data validates against the 3D Model Event Descriptors’ JSON Schema.
2. All Data in the 3D Model Event Descriptors’ JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.54 3D Model Object

8.54.1 Definition

A Data Type including a collection of Basic 3D Model Objects.

A 3D Model Object can have a hierarchical structure where 3D Model Objects contain Basic 3D Model Objects and 3D Model Objects.

8.54.2 Functional Requirements

A 3D Model Object may include:

1. ID of a Virtual Space (M-Instance) where it is or intended to be located.
2. ID of the 3D Model Object.
3. Space-Time information of the 3D Model Object.
4. Basic 3D Model Object and 3D Model Objects included in the 3D Model Objects.
5. Annotation data set including:
 1. Annotations
 2. Space-Times of the Annotations.
 3. Rights to perform Actions on the 3D Model Object.
6. The Rights that may be exercised on the 3D Model Object.

Note that.

1. An 3D Model Object that does not include Sub-Scenes and only one Basic 3D Model Object is a Basic 3D Model Object.
2. The Space-Time information of a Basic 3D Model Object and 3D Model Object included in an 3D Model Object may be superseded by the Space-Time information of the 3D Model Object containing them.

8.54.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/3DModelObject.json>

8.54.4 Semantics

Label	Size	Description
Header	N1 Bytes	3D Model Object Header
– Standard-3D ModelObject	9 Bytes	The characters “OSD-3DO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
3DModelObjectID	N5 Bytes	Identifier of the 3D Model Object.
3DModelObjectSpaceTime	N6 Bytes	Space-Time of 3D Model Object.
Basic3DModelObjectCount	N7 Bytes	Set of Parent 3D Model Objects.
Basic3DModelObjects[]	N8 Bytes	Set of Basic 3D Model Objects.

- SpaceTime	N9 Bytes	Space Time of a Basic 3D Model Object in the 3D Model Object.
- Basic3DModelObject	N10 Bytes	A Basic 3D Model Object in the 3D Model Object.
3DModelObjectCount	N11 Bytes	Number of 3D Model Objects.
3DModelObjects[]	N12 Bytes	Set of 3D Model Objects.
- SpaceTime	N13 Bytes	Space Time of an 3D Model Object in the 3D Model Object.
- 3DModelObject	N14 Bytes	A 3D Model Object in the 3D Model Object
Annotations[]	N15 Bytes	Set of 3D Model Object Annotation.
- Annotation	N16 Bytes	An Annotation.
- AnnotationSpaceTime	N17 Bytes	Where Annotation is attached and when it will be active.
- Rights	N18 Bytes	Actions that may be performed on the Annotation
Rights	N19 Bytes	Actions that may be performed on the Object.
DescrMetadata	N20 Bytes	Descriptive Metadata

8.54.5 Conformance Testing

A Data instance Conforms with 3D Model Object (OSD-3DO) V1.3 if:

1. The Data validates against the 3D Model Object's JSON Schema.
2. All Data in the 3D Model Object's JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.55 3D Model Scene Descriptors

8.55.1 Definition

A Data Type including the 3D Model Objects of a scene, their sub-scenes, and their arrangement in the scene.

8.55.2 Functional Requirements

3D Model Scene Descriptors include

1. 3D Model Objects
2. The Descriptors of the Scenes includes in the Scene called Sub-Scenes.
3. Rights that may be exercised on the Scene.

Scenes may be hierarchical, i.e., they may contain Objects and Scenes.

8.55.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/3DModelSceneDescriptors.json>

8.55.4 Semantics

Label	Size	Description
Header	N1 Bytes	3D Model Scene Descriptors Header
- Standard-3DModelSceneDescriptors	9 Bytes	The characters “OSD-3DD-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneDescriptorsID	N5 Bytes	Identifier of Scene Descriptors.
SceneDescriptorsSpaceTime	N6 Bytes	Space and Time of Scene Descriptors.
ObjectCount	N7 Bytes	Number of Objects in Scene.
Objects[]	N8 Bytes	Set of Objects.
- Object or ObjectID	N9 Bytes	Object in the Scene of its ID.
- ObjectSpaceTime	N10 Bytes	Space Time of Object.
SubSceneCount	N11 Bytes	Number of Sub-Scenes in Scene.
SubScenes[]	N12 Bytes	Set of Sub-Scenes in the Scene.
- SubScene or SubSceneID	N13 Bytes	Sub-Scene in the Scene or its ID.
- SubSceneSpaceTime	N14 Bytes	Space Time of Sub-Scene.
DescrMetadata	N15 Bytes	Descriptive Metadata

8.55.5 Conformance Testing

A Data instance Conforms with 3D Model Scene Descriptors (OSD-3SD) V1.3 if:

1. The Data validates against the Scene Descriptors’ JSON Schema.
2. All Data in the Scene Descriptors’ JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.56 3D Model Scene Geometry

8.56.1 Definition

An Data Type including the arrangement of the 3D Model Objects in a scene with their 3D Model Qualifiers.

In the following, Data, Objects, Qualifiers, and (Sub-)Scenes should be read as 3D Model Data, 3D Model Objects, 3D Model Qualifiers, and 3D Model (Sub-)Scenes

8.56.2 Functional Requirements

Scene Geometry includes the arrangements of the Scenes - called Sub-Scenes - in addition to the arrangement of Objects.

8.56.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/3DModelSceneGeometry.json>

8.56.4 Semantics

Label	Size	Description
Header	N1 Bytes	3D Model Scene Geometry Header
- Standard-3DModelSceneGeometry	9 Bytes	The characters “OSD-3SG-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneGeometryID	N5 Bytes	Identifier of Scene Geometry.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SubSceneCount	N7 Bytes	Number of Sub-Scenes in Scene.
SceneGeometrySpaceTime	N8 Bytes	Space and Time of Scene Geometry.
SceneObjects[]	N9 Bytes	Set of Data related to Objects.
- SceneObjectQualifiers	N10 Bytes	Qualifiers of Object.
- SceneObjectSpaceTime	N11 Bytes	Space Time of Object.
SceneSubScenes[]	N9 Bytes	Set of Sub-Scenes.
- SceneSubSceneSpaceTime	N11 Bytes	Space Time of Sub-Scene.
DescrMetadata	N16 Bytes	Descriptive Metadata

8.56.5 Conformance Testing

A Data instance Conforms with 3D Model Scene Geometry (OSD-3SG) V1.3 if:

1. The Data validates against the Scene Geometry’s JSON Schema.
2. All Data in the Scene Geometry’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.57 Annotation

8.57.1 Definition

Annotation is Data attached to an Object or a Scene. As opposed to Qualifier that describes intrinsic properties of an Object, an Annotation is spatially and temporally local and changeable.

8.57.2 Functional Requirements

Elements of an Annotation are:

1. M-Instance ID
2. Annotation ID
3. Annotation Space-Time
4. Annotation Data
 1. JSON Text Objects
 2. Annotation Space-Time in Object or Scene
 3. Permitted Actions on Annotated Data

Annotation Data is text containing the JSON code conforming to the JSON Schema of the Item intended as Annotation. Examples of such Items are Perceptible Entities, Intention, Meaning, and Personal Status and Its components.

8.57.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/Annotation.json>

8.57.4 Semantics

Label	Size	Description
Header	N1 Bytes	Annotation Header
- Standard-Annotation	9 Bytes	The characters “OSD-ANN-V”
- Version	N2 Bytes	Major version – 1 or 2 Bytes
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 Bytes
MInstanceID	N4 Bytes	The Virtual Space whose Object or Scene contains Annotations.
AnnotationID	N5 Bytes	Identifier of Annotation.
Annotation[]	N6 Bytes	The actual Annotation.
- AnnotationJSONText	N7 Bytes	Text of the JSON representing the Data Type used in the Annotation.
- AnnotationSpaceTime	N8 Bytes	Where/when Annotation is attached.
- ProcessActions[]	N9 Bytes	What is possible to do with the Annotation
- ProcessActionID	N10 Bytes	List of possible Process Actions
DescrMetadata	N11 Bytes	Descriptive Metadata

8.57.5 Conformance Testing

A Data instance Conforms with MPAI-OSD V1.3 Annotation (OSD-ANN) if:

1. The Data validates against the Annotation's JSON Schema.
2. All Data in the Annotation's JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.58 Audio Object

8.58.1 Definition

A Data Type including a collection of Basic Audio Objects.

An Audio Object can have a hierarchical structure where Audio Objects contain Basic Audio Objects and Audio Objects.

8.58.2 Functional Requirements

An Audio Object may include:

1. ID of a Virtual Space (M-Instance) where it is or intended to be located.
2. ID of the Audio Object.
3. Space-Time information of the Audio Object.
4. Basic Audio Object and Audio Objects included in the Audio Objects.
5. Annotation data set including:
 1. Annotations
 2. Space-Times of the Annotations.
 3. Rights to perform Actions on the Audio Object.
6. The Rights that may be exercised on the Audio Object.

Note that.

1. An Audio Object that does not include Sub-Scenes and only one Basic Audio Object is a Basic Audio Object.
2. The Space-Time information of a Basic Audio Object, Audio Object included in an Audio Object may be superseded by the Space-Time information of the Audio Object containing it.

8.58.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/AudioObject.json>

8.58.4 Semantics

Label	Size	Description
Header	N1 Bytes	Audio Object Header
– Standard-AudioObject	9 Bytes	The characters “OSD-AUO-V”

– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
AudioObjectID	N5 Bytes	Identifier of the Audio Object.
AudioObjectSpaceTime	N6 Bytes	Space-Time of Audio Object.
BasicAudioObjectCount	N7 Bytes	Set of Parent Audio Objects.
BasicAudioObjects[]	N8 Bytes	Set of Basic Audio Objects.
- SpaceTime	N9 Bytes	Space Time of a Basic Audio Object in the Audio Object.
- BasicAudioObject	N10 Bytes	A Basic Audio Object in the Audio Object.
AudioObjectCount	N11 Bytes	Number of Audio Objects.
AudioObjects[]	N12 Bytes	Set of Audio Objects.
- SpaceTime	N13 Bytes	Space Time of an Audio Object in the Audio Object.
- AudioObject	N14 Bytes	An Audio Object in the Audio Object
Annotations[]	N14 Bytes	Set of Audio Object Annotation.
– Annotation	N15 Bytes	An Annotation.
– AnnotationSpaceTime	N15 Bytes	Where Annotation is attached and when it will be active.
– Rights	N16 Bytes	Actions that may be performed on the Annotation
Rights	N17 Bytes	Actions that may be performed on the Object.
DescrMetadata	N17 Bytes	Descriptive Metadata

8.58.5 Conformance Testing

A Data instance Conforms with Audio Object (OSD-AUO) V1.3 if:

1. The Data validates against the Audio Object’s JSON Schema.
2. All Data in the Audio Object’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.59 Audio Scene Descriptors

8.59.1 Definition

A Data Type including the Audio Objects of a scene, their sub-scenes, and their arrangement in the scene.

8.59.2 Functional Requirements

Audio Scene Descriptors include

1. Audio Objects
2. The Descriptors of the Scenes includes in the Scene called Sub-Scenes.
3. Rights that may be exercised on the Scene.

Scenes may be hierarchical, i.e., they may contain Objects and Scenes.

8.59.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/AudioSceneDescriptors.json>

8.59.4 Semantics

Label	Size	Description
Header	N1 Bytes	Audio Scene Descriptors Header
- Standard-AudioSceneDescriptors	9 Bytes	The characters “OSD-ASD-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneDescriptorsID	N5 Bytes	Identifier of Scene Descriptors.
SceneDescriptorsSpaceTime	N6 Bytes	Space and Time of Scene Descriptors.
ObjectCount	N7 Bytes	Number of Objects in Scene.
Objects[]	N8 Bytes	Set of Objects.
- Object or ObjectID	N9 Bytes	Object in the Scene of its ID.
- ObjectSpaceTime	N10 Bytes	Space Time of Object.
SubSceneCount	N11 Bytes	Number of Sub-Scenes in Scene.
SubScenes[]	N12 Bytes	Set of Sub-Scenes in the Scene.
- SubScene or SubSceneID	N13 Bytes	Sub-Scene in the Scene or its ID.
- SubSceneSpaceTime	N14 Bytes	Space Time of Sub-Scene.
DescrMetadata	N15 Bytes	Descriptive Metadata

8.59.5 Conformance Testing

A Data instance Conforms with Audio Scene Descriptors (OSD-ASD) V1.3 if:

1. The Data validates against the Scene Descriptors’ JSON Schema.
2. All Data in the Scene Descriptors’ JSON Schema
 1. Have the specified type

2. Validate against their JSON Schemas
3. Conform with their Data Qualifiers.

8.60 Audio-Visual Event Descriptors

8.60.1 Definition

An Item including a series of Audio-Visual Scene Descriptors for a certain duration.

8.60.2 Functional Requirements

Audio-Visual Event Descriptors contains Audio-Visual Scene Descriptors for a Time.

8.60.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/AudioVisualEventDescriptors.json>

8.60.4 Semantics

Label	Size	Description
Header	N1 Bytes	Audio-Visual Event Descriptors Header
- Standard-AudioVisualEventDescriptors	9 Bytes	The characters “OSD-AVE-V”
· Version	N2 Bytes	Major version – 1 or 2 characters
· Dot-separator	1 Byte	The character “.”
· Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
EventID	N5 Bytes	Identifier of the Event.
EventSpaceTime	17 Bytes	Data about start and end Space-Time.
SceneDescriptors[]	N6 Bytes	Set of Scene Descriptors
- SceneDescriptors	N7 Bytes	Set of AV Scene Descriptors of IDs.
DescrMetadata	N8 Bytes	Descriptive Metadata

8.60.5 Conformance Testing

A Data instance Conforms with MPAI-OSD Audio-Visual Event Descriptors V1.3 (OSD-AVE) if:

1. The Data validates against the Audio-Visual Event Descriptors’ JSON Schema.
2. All Data in the Audio-Visual Event Descriptors’ JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.61 Audio-Visual Object

8.61.1 Definition

Data whose rendering has both Audio and Visual perceptibility attributes.

8.61.2 Functional Requirements

Audio-Visual Object includes:

1. The ID of a Virtual Space (M-Instance) where it is or will be located.
2. The 3DModel-Speech-Audio-Visual Objects' Space-Time location.
3. The IDs of the 3DModel, Speech, Audio, and Visual Objects' and their Space-Time information.

8.61.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/AudioVisualObject.json>

8.61.4 Semantics

	Label	Size	Description
Header		N1 Bytes	Audio-Visual Object Header
- Standard-AudioVisualObject		9 Bytes	The characters “OSD-AVO-V”
- Version		N2 Byte	Major version – 1 or 2 Bytes
- Dot-separator		1 Byte	The character “.”
- Subversion		N3 Bytes	Minor version – 1 or 2 Bytes
MInstanceID		N4 Bytes	Identifier of M-Instance.
AudioVisualObjectID		N5 Bytes	Identifier of Audio-Visual Object.
AudioVisualObjectSpaceTime		N6 Bytes	Space-Time of Audio-Visual Object
AudioVisualQualifier		N7 Bytes	Qualifier of the Audio-Visual Object
3DModelObjectData		N8 Bytes	3D Model Object Data
- 3DModelObjectID and/or 3DModelObject		N9 Bytes	3D Model Object ID and/or Object
- 3DModelObjectSpaceTime		N10 Bytes	Space-Time of Speech Object
SpeechObjectData		N11 Bytes	Speech Object Data
- SpeechObjectID and/or Speech Object		N12 Bytes	Speech Object ID and/or Object
- SpeechObjectSpaceTime		N13 Bytes	Space-Time of Speech Object
AudioObjectData		N14 Bytes	Audio Object Data
- AudioObjectID and/or Audio Object		N15 Bytes	Audio Object ID and/or Object
- AudioObjectSpaceTime		N16 Bytes	Space-Time of Audio Object

VisualObjectData	N17 Bytes	Visual Object Data
- VisualObjectID and/or Visual Object	N18 Bytes	Visual Object ID and/or Object
- VisualObjectSpaceTime	N19 Bytes	Space-Time of Visual Object
Annotations[]	N20 Bytes	Set of Audio Object Annotation.
- Annotation	N21 Bytes	An Annotation.
- AnnotationSpaceTime	N22 Bytes	Where Annotation is attached and when it will be active.
- Rights	N23 Bytes	Actions that may be performed on the Annotation
Rights	N124Bytes	Actions that may be performed on the Object.
DescrMetadata	N25 Bytes	Descriptive Metadata

8.61.5 Conformance Testing

A Data instance Conforms with Audio-Visual Object V1.3 (OSD-AVO) if:

1. The Data validates against the Audio-Visual Object's JSON Schema.
2. All Data in the Audio-Visual Object's JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.62 Audio-Visual Scene Descriptors

8.62.1 Definition

A Data Type including the Audio-Visual Scene's Objects and Sub-Scenes and their arrangement in the Scene.

8.62.2 Functional Requirements

Audio-Visual Scene Descriptors includes Scenes in addition to Objects.

8.62.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/AudioVisualSceneDescriptors.json>

8.62.4 Semantics

	Label	Size	Description
Header		N1 Bytes	Audio-Visual Scene Descriptors Header
- Standard-AVSceneDescriptors		9 Bytes	The characters "OSD-AVS-V"
- Version		N2 Bytes	Major version – 1 or 2 characters
- Dot-separator		1 Byte	The character "."
- Subversion		N3 Bytes	Minor version – 1 or 2 characters

MInstanceID	N4 Bytes	Identifier of M-Instance.
AVBasicSceneDescriptorsID	N5 Bytes	Identifier of the AV Object.
ObjectCount	N6 Bytes	Number of Objects in Scene
AVSceneSpaceTime	N7 Bytes	Data about Space and Time
SpeechObjects[]	N8 Bytes	Set of Speech Objects
- SpeechObject	N9 Bytes	Speech Object
- SpeechObjectSpaceTime	N10 Bytes	Space-Time of Speech Object
AudioObjects[]	N11 Bytes	Set of Audio Objects
- AudioObject	N12 Bytes	ID of Audio Object
- AudioObjectSpaceTime	N13 Bytes	Space-Time of Audio Object
VisualObjects[]	N14 Bytes	Set of Visual Objects
- VisualObjectID	N15 Bytes	ID of Visual Object
- VisualObjectSpaceTime	N16 Bytes	Space-Time of Visual Object
AudioVisualObjects[]	N17 Bytes	Set of Audio-Visual Objects
- AudioVisualObjectID	N18 Bytes	ID of Audio-Visual Object
- AudioObjectSpaceTime	N19 Bytes	Space-Time of Audio-Visual Object
SubSceneCount	N20 Bytes	Number of Sub-Scenes in Scene
SpeechSubScenes[]	N21 Bytes	Set of Speech Objects
- SpeechSubScene	N22 Bytes	Speech SubScene
- SpeechSubSceneSpaceTime	N23 Bytes	Space-Time of Speech SubScene
AudioSubScenes[]	N24 Bytes	Set of Audio SubScenes
- AudioSubScene	N25 Bytes	ID of Audio SubScene
- AudioSubSceneSpaceTime	N26 Bytes	Space-Time of Audio SubScene
VisualSubScenes[]	N27 Bytes	Set of Visual SubScenes
- VisualSubSceneID	N28 Bytes	ID of Visual SubScene
- VisualSubSceneSpaceTime	N29 Bytes	Space-Time of Visual SubScene
AudioVisualSubScenes[]	N30 Bytes	Set of Audio-Visual SubScenes
- AudioVisualSubSceneID	N31 Bytes	ID of Audio-Visual SubScene
- AudioSubSceneSpaceTime	N31 Bytes	Space-Time of Audio-Visual SubScene
DescrMetadata	N33 Bytes	Descriptive Metadata

8.62.5 Conformance Testing

A Data instance Conforms with Audio-Visual Scene Descriptors (OSD-AVS) V1.3 if:

1. The Data validates against the Audio-Visual Scene Descriptors' JSON Schema.
2. All Data in the Audio-Visual Scene Descriptors' JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.63 Basic 3D Model Object

8.63.1 Definition

A Data Type including

1. Data representing:
 1. The surface and relevant features of physical objects from the real world, or
 2. Synthetic Data intended for visual rendering, or
 3. A mixture of the two types, and
2. Optionally, descriptive Data regarding Sub-Types, Formats and Attributes of the 3D Model Data.

In the following, Data and Object should be read as 3D Model Data and 3D Model Object.

8.63.2 Functional Requirements

A Basic Object may include:

1. The ID of a Virtual Space (M-Instance) where it is or is intended to be located.
2. The ID of the Basic Object.
3. The ID(s) of Parent Object(s) supporting two cases:
 1. The Parent Object has spawned two (or more) Objects.
 2. Two (or more) Parent Objects have merged into one.
4. The Space-Time information of Parent Objects in an M-Instance.
5. The ID(s) of Child Object(s).
6. The Space-Time information of Child Objects in an M-Instance.
7. The Space-Time information of the Basic Object in an M-Instance.
8. The Qualifier of the specific Data Type.
9. The Rights that can be exercised on the Basic Object.
10. The set of Annotations including, for each Annotation:
 1. Space-Time information of the Annotation.
 2. Rights to perform Actions on the Annotation.

8.63.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/Basic3DModelObject.json>

8.63.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic 3D Model Object Header
– Standard-Basic3DModelObject	9 Bytes	The characters “OSD-B3O-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.

BasicObjectID	N5 Bytes	Identifier of the Basic Object.
BasicObjectSpaceTime	N6 Bytes	Space-Time info of the Basic Object.
Qualifier	N7 Bytes	Qualifier of Basic Data.
BasicObjectAnnotations[]	N8 Bytes	Annotations of Basic Object.
– Annotation	N9 Bytes	ID of Annotation
– AnnotationSpaceTime	N10 Bytes	Where/when Annotation is attached.
Rights	N11 Bytes	Rights to perform Actions of the Basic Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.63.5 Conformance Testing

A Data instance Conforms with Basic Object V1.3 if:

1. The Data validates against the Basic Object's JSON Schema.
2. All Data in the Basic Object's JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers, if present.

8.64 Basic 3D Model Scene Descriptors

8.64.1 Definition

A Data Type including the 3D Model Objects of a scene, their time and arrangement in the scene, and the Rights that may be exercised on the scene.

In the following Object and Scene are to be read as 3D Model Object and 3D Model Scene, respectively.

8.64.2 Functional Requirements

Basic Scene Descriptors include

1. Objects
2. Space-Time information.
3. Rights that may be exercised on the Scene.

The Space-Time of the Objects may be superseded by the Space-Time of the Scene.

8.64.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/BasicAudioSceneDescriptors.json>

8.64.4 Semantics

Label	Size	Description
-------	------	-------------

Header	N1 Bytes	Basic 3D Model Scene Descriptors Header
- Standard-Basic3DModelSceneDescriptors	9 Bytes	The characters “OSD-B3S-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneDescriptorsID	N5 Bytes	Identifier of Scene Descriptors.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SceneDescriptorsSpaceTime	N7 Bytes	Space and Time of Scene Descriptors.
SceneObjects[]	N8 Bytes	Set of Objects.
- SceneObject	N9 Bytes	An Object.
- SceneObjectSpaceTime	N10 Bytes	Space Time of Object.
Rights	N11 Bytes	Rights that may be exercised on the Scene.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.64.5 Conformance Testing

A Data instance Conforms with Basic Scene Descriptors (OSD-B3S) V1.3 if:

1. The Data validates against the Scene Descriptors’ JSON Schema.
2. All Data in the ‘Basic Scene Descriptors’ JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.65 Basic Audio Object

8.65.1 Definition

A Data Type including

1. Data representing:
 1. A sound of the real world, or
 2. Synthetic Data intended for rendering as sound, or
 3. A mixture of the two types, and
2. Optionally, descriptive Data regarding Audio Sub-Types, Formats and Attributes.

In the following, Data, Qualifier, and Object should be read as Audio Data, Audio Qualifiers, and Audio Object, respectively.

8.65.2 Functional Requirements

A Basic Object may include:

1. The ID of a Virtual Space (M-Instance) where it is or is intended to be located.
2. The ID of the Basic Object.
3. The ID(s) of Parent Object(s) supporting two cases:
 1. The Parent Object has spawned two (or more) Objects.
 2. Two (or more) Parent Objects have merged into one.
4. The Space-Time information of Parent Objects in an M-Instance.
5. The ID(s) of Child Object(s).
6. The Space-Time information of Child Objects in an M-Instance.
7. The Space-Time information of the Basic Object in an M-Instance.
8. The Qualifier of the specific Data Type.
9. The Rights that can be exercised on the Basic Object.
10. The set of Annotations including, for each Annotation:
 1. Space-Time information of the Annotation.
 2. Rights to perform Actions on the Annotation.

8.65.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/BasicAudioObject.json>

8.65.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Audio Object Header
– Standard-BasicAudioObject	9 Bytes	The characters “OSD-BAO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
BasicObjectID	N5 Bytes	Identifier of the Basic Object.
BasicObjectSpaceTime	N6 Bytes	Space-Time info of the Basic Object.
Qualifier	N7 Bytes	Qualifier of Basic Data.
BasicObjectAnnotations[]	N8 Bytes	Annotations of Basic Object.
– Annotation	N9 Bytes	ID of Annotation
– AnnotationSpaceTime	N10 Bytes	Where/when Annotation is attached.
Rights	N11 Bytes	Rights to perform Actions of the Basic Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.65.5 Conformance Testing

A Data instance Conforms with Basic Object V1.3 if:

1. The Data validates against the Basic Object's JSON Schema.
2. All Data in the Basic Object's JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers, if present.

8.66 Basic Audio Scene Descriptors

8.66.1 Definition

A Data Type including the Audio Objects of a scene, their time and arrangement in the scene, and the Rights that may be exercised on the scene.

In the following Object and Scene are to be read as Audio Object and Audio Scene, respectively.

8.66.2 Functional Requirements

Basic Scene Descriptors include

1. Objects
2. Space-Time information.
3. Rights that may be exercised on the Scene.

The Space-Time of the Objects may be superseded by the Space-Time of the Scene.

8.66.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/BasicAudioSceneDescriptors.json>

8.66.4 Semantics

	Label	Size	Description
Header		N1 Bytes	Basic Audio Scene Descriptors Header
- Standard-BasicAudioSceneDescriptors		9 Bytes	The characters "OSD-BAS-V"
- Version		N2 Bytes	Major version – 1 or 2 characters
- Dot-separator		1 Byte	The character "."
- Subversion		N3 Bytes	Minor version – 1 or 2 characters
MInstanceID		N4 Bytes	Identifier of M-Instance.
SceneDescriptorsID		N5 Bytes	Identifier of Scene Descriptors.
ObjectCount		N6 Bytes	Number of Objects in Scene.
SceneDescriptorsSpaceTime		N7 Bytes	Space and Time of Scene Descriptors.

SceneObjects[]	N8 Bytes Set of Objects.
- SceneObject	N9 Bytes An Object.
- SceneObjectSpaceTime	N10 Bytes Space Time of Object.
Rights	N11 Bytes Rights that may be exercised on the Scene.
DescrMetadata	N12 Bytes Descriptive Metadata

8.66.5 Conformance Testing

A Data instance Conforms with Basic Scene Descriptors (OSD-BAS) V1.3 if:

1. The Data validates against the Scene Descriptors' JSON Schema.
2. All Data in the Scene Descriptors' JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.67 Basic Audio-Visual Object

8.67.1 Definition

A Data Type including

1. Data representing:
 1. A sound and visual of the real world, or
 2. Synthetic Data intended for rendering as sound and visual, or
 3. A mixture of the two types, and
2. Optionally, descriptive Data regarding Audio-Visual Sub-Types, Formats and Attributes.

In the following, Data, Qualifier, and Object should be read as Audio-Visual Data, Audio-Visual Qualifiers, and Audio-Visual Object, respectively.

8.67.2 Functional Requirements

A Basic Object may include:

1. The ID of a Virtual Space (M-Instance) where it is or is intended to be located.
2. The ID of the Basic Object.
3. The ID(s) of Parent Object(s) supporting two cases:
 1. The Parent Object has spawned two (or more) Objects.
 2. Two (or more) Parent Objects have merged into one.
4. The Space-Time information of Parent Objects in an M-Instance.
5. The ID(s) of Child Object(s).
6. The Space-Time information of Child Objects in an M-Instance.
7. The Space-Time information of the Basic Object in an M-Instance.
8. The Qualifier of the specific Data Type.
9. The Rights that can be exercised on the Basic Object.
10. The set of Annotations including, for each Annotation:
 1. Space-Time information of the Annotation.

2. Rights to perform Actions on the Annotation.

8.67.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/BasicAudioVisualObject.json>

8.67.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Audio-Visual Object Header
– Standard-BasicAudioVisualObject	9 Bytes	The characters “OSD-BMO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
BasicObjectID	N5 Bytes	Identifier of the Basic Object.
BasicObjectSpaceTime	N6 Bytes	Space-Time info of the Basic Object.
Qualifier	N7 Bytes	Qualifier of Basic Data.
BasicObjectAnnotations[]	N8 Bytes	Annotations of Basic Object.
– Annotation	N9 Bytes	ID of Annotation
– AnnotationSpaceTime	N10 Bytes	Where/when Annotation is attached.
Rights	N11 Bytes	Rights to perform Actions of the Basic Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.67.5 Conformance Testing

A Data instance Conforms with Basic Object V1.3 if:

1. The Data validates against the Basic Object’s JSON Schema.
2. All Data in the Basic Object’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers, if present.

8.68 Basic Audio-Visual Scene Descriptors

8.68.1 Definition

A Data Type including the Audio-Visual Objects of a scene, their time and arrangement in the scene, and the Rights that may be exercised on the scene.

In the following Object and Scene are to be read as Audio-Visual Object and Audio-Visual Scene, respectively.

8.68.2 Functional Requirements

Basic Scene Descriptors include

1. Objects
2. Space-Time information.
3. Rights that may be exercised on the Scene.

The Space-Time of the Objects may be superseded by the Space-Time of the Scene.

8.68.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/BasicAudioVisualSceneDescriptors.json>

8.68.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Audio-Visual Scene Descriptors Header
- Standard-BasicAudioVisualSceneDescriptors	9 Bytes	The characters “OSD-BMS-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneDescriptorsID	N5 Bytes	Identifier of Scene Descriptors.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SceneDescriptorsSpaceTime	N7 Bytes	Space and Time of Scene Descriptors.
SceneObjects[]	N8 Bytes	Set of Objects.
- SceneObject	N9 Bytes	An Object.
- SceneObjectSpaceTime	N10 Bytes	Space Time of Object.
Rights	N11 Bytes	Rights that may be exercised on the Scene.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.68.5 Conformance Testing

A Data instance Conforms with Basic Scene Descriptors (OSD-BMS) V1.3 if:

1. The Data validates against the Scene Descriptors’ JSON Schema.
2. All Data in the Scene Descriptors’ JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.69 Basic Location

8.69.1 Definition

A region with Space-Time attributes that is not exposed as further subdivided.

8.69.2 Functional Requirements

A Basic Location

1. Is composed of an arbitrary number of connected spatial units having a Location Format.
2. may have a spatial information that overrides the spatial information of a Right Parallelepiped includes in the Basic Location.

8.69.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/BasicLocation.json>

8.69.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Location Header
– Standard-BasicLocation	9 Bytes	The characters “OSD-BLC-V”
– Version	N2 Bytes	Major version
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version
MInstanceID	N4 Bytes	Identifier of M-Instance.
BasicLocationID	N6 Bytes	Identifier of Basic Location.
BasicLocation[]	N7 Bytes	Set of Right Parallelepipeds defining Basic-M-Location.
– LocationQualifier	N8 Bytes	Spatial unit.
– SpaceTime	N9 Bytes	Space-Time of spatial unit.
DescrMetadata	N10 Bytes	Descriptive Metadata.

8.69.5 Conformance Testing

A Data instance Conforms with Basic Location (OSD-BLC) V1.3 if the Data:

1. The Data validates against the Basic Location’s JSON Schema.
2. All Data in the Basic Location’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas

8.70 Basic Object

8.70.1 Definition

A Data Type including:

1. Data of a specific media type perceptible by a specific device and/or a human.
2. Descriptive Data regarding Sub-Types, Formats and Attributes of the Data (optionally).

8.70.2 Functional Requirements

A Basic Object may include:

1. The ID of a Virtual Space (M-Instance) where it is or is intended to be located.
2. The ID of the Basic Object.
3. The ID(s) of Parent Object(s) supporting two cases:
 1. The Parent Object has spawned two (or more) Objects.
 2. Two (or more) Parent Objects have merged into one.
4. The Space-Time information of Parent Objects in an M-Instance.
5. The ID(s) of Child Object(s).
6. The Space-Time information of Child Objects in an M-Instance.
7. The Space-Time information of the Basic Object in an M-Instance.
8. The Qualifier of the specific Data Type.
9. The Rights that can be exercised on the Basic Object.
10. The set of Annotations including, for each Annotation:
 1. Space-Time information of the Annotation.
 2. Rights to perform Actions on the Annotation.

All Media-specific Basic Objects are given in the following Table

Object name	Acronym	JSON
Basic 3DModel Object	OSD-B3O	Link
Basic Audio Object	OSD-BAO	Link
Basic Audio-Visual Object	OSD-BMO	Link
Basic LiDAR Object	OSD-BLO	Link
Basic Offline Map Object	OSD-BOO	Link
BasicRADAR Object	OSD-BRO	Link
Basic Speech Object	OSD-BSO	Link
Basic Text Object	OSD-BTO	Link
Basic Ultrasound Object	OSD-BUO	Link
Basic Visual Object	OSD-BVO	Link

8.70.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/BasicObject.json>

8.70.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Object Header
– Standard-BasicObject	9 Bytes	The characters “OSD-BOB-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
Basic ObjectID	N5 Bytes	Identifier of the Basic Object.
ParentObjects[]	N6 Bytes	Set of Parent Objects.
- ParentObject	N7 Bytes	A Parent Object.
- ParentObjectSpaceTime	N8 Bytes	SpaceTime of a Parent Object.
ChildObjects[]	N9 Bytes	Set of Child Objects.
- ChildObject	N10 Bytes	Identifier of a Child Object.
- ChildObjectSpaceTime	N11 Bytes	SpaceTime of a Child Object.
Space-Time	N12 Bytes	Space-Time of Data.
Qualifier	N13 Bytes	Qualifier of Data.
Annotations[]	N14 Bytes	Set of Data Annotations.
– Annotation	N15 Bytes	An Annotation
– AnnotationSpaceTime	N16 Bytes	Where Annotation is attached - when it will be active.
– Rights	N17 Bytes	Actions that may be performed on the Annotation.
Rights	N18 Bytes	Rights to perform Process Actions on the Object.
DescrMetadata	N19 Bytes	Descriptive Metadata

8.70.5 Conformance Testing

A Data instance Conforms with Basic Object (OSD-BOB) V1.3 if:

1. The Data validates against the Basic Object’s JSON Schema.
2. All Data in the Basic Object’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.71 Basic Scene Descriptors

8.71.1 Definition

A Data Type representing the Objects of a scene and their arrangement in the scene.

8.71.2 Functional Requirements

Basic Scene Descriptors include:

1. ID of a Virtual Space where it is or will be located.
2. ID of the Scene Descriptors.
3. The number of Objects in the Scene.
4. Basic Scene Space-Time.
5. Objects that include, for each Object:
 1. The Object ID or the Object.
 2. Space-Time values potentially different from their intrinsic Space Times values.
6. Rights that may be exercised on the Basic Scene

The following Table gives the Media Type-specific Basic Scene Descriptors

Basic Scene Descriptors name	Acronym	JSON
3DModel Basic Scene Descriptors	OSD-3BS	Link
Audio Basic Scene Descriptors	OSD-ABS	Link
Audio-Visual Basic Scene Descriptors	OSD-MBS	Link
LiDAR Basic Scene Descriptors	OSD-LBS	Link
Offline Map Basic Scene Descriptors	OSD-OBS	Link
RADAR Basic Scene Descriptors	OSD-RBS	Link
Speech Basic Scene Descriptors	OSD-SBS	Link
Text Basic Scene Descriptors	OSD-TBS	Link
Ultrasound Basic Scene Descriptors	OSD-UBS	Link
Visual Basic Scene Descriptors	OSD-VBS	Link

8.71.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/BasicSceneDescriptors.json>

8.71.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Scene Descriptors Header
- Standard-BasicSceneDescriptors	9 Bytes	The characters “OSD-BSD-”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
BasicSceneDescriptorsID	N5 Bytes	Identifier of the Basic Scene.
SpaceTime	N6 Bytes	Data about Space-Time of Basic Scene.
ObjectCount	N7 Bytes	Number of Objects in Scene.
Objects[]	N16 Bytes	Set of Objects.
- ObjectID and/or Object	N17 Bytes	ObjectID or Object.
- ObjectSpaceTime	N18 Bytes	Space-Time of Object.

8.71.5 Conformance Testing

A Data instance Conforms with Basic Scene Descriptors (OSD-BSD) V1.3 if:

1. The Data validates against the Basic Scene Descriptors' JSON Schema.
2. All Data in the Basic Scene Descriptors' JSON Schema
 1. Have the specified type.
 2. Validate against their JSON Schemas.
 3. Conform with their Data Qualifiers.

8.72 Basic Speech Object

8.72.1 Definition

A Data Type including:

1. Data representing:
 1. Content represented as Data whose rendering has vocal attributes, or
 2. Synthetic Data intended for rendering as speech, or
 3. A mixture of the two types, and
2. Optionally, descriptive Data regarding Speech Sub-Types, Formats and Attributes.

In the following, Data, Qualifier, and Object should be read as Speech Data, Speech Qualifiers, and Speech Object, respectively.

8.72.2 Functional Requirements

A Basic Object may include:

1. The ID of a Virtual Space (M-Instance) where it is or is intended to be located.
2. The ID of the Basic Object.
3. The ID(s) of Parent Object(s) supporting two cases:
 1. The Parent Object has spawned two (or more) Objects.
 2. Two (or more) Parent Objects have merged into one.
4. The Space-Time information of Parent Objects in an M-Instance.
5. The ID(s) of Child Object(s).
6. The Space-Time information of Child Objects in an M-Instance.
7. The Space-Time information of the Basic Object in an M-Instance.
8. The Qualifier of the specific Data Type.
9. The Rights that can be exercised on the Basic Object.
10. The set of Annotations including, for each Annotation:
 1. Space-Time information of the Annotation.
 2. Rights to perform Actions on the Annotation.

8.72.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/BasicSpeechObject.json>

8.72.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Speech Object Header
– Standard-BasicSpeechObject	9 Bytes	The characters “OSD-BSO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
BasicObjectID	N5 Bytes	Identifier of the Basic Object.
BasicObjectSpaceTime	N6 Bytes	Space-Time info of the Basic Object.
Qualifier	N7 Bytes	Qualifier of Basic Data.
BasicObjectAnnotations[]	N8 Bytes	Annotations of Basic Object.
– Annotation	N9 Bytes	ID of Annotation
– AnnotationSpaceTime	N10 Bytes	Where/when Annotation is attached.
Rights	N11 Bytes	Rights to perform Actions of the Basic Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.72.5 Conformance Testing

A Data instance Conforms with Basic Object V1.3 if:

1. The Data validates against the Basic Object’s JSON Schema.
2. All Data in the Basic Object’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers, if present.

8.73 Basic Speech Scene Descriptors

8.73.1 Definition

A Data Type including the Speech Objects of a scene, their time and arrangement in the scene, and the Rights that may be exercised on the scene.

In the following Object and Scene are to be read as Speech Object and Speech Scene, respectively.

8.73.2 Functional Requirements

Basic Scene Descriptors include

1. Objects
2. Space-Time information.

3. Rights that may be exercised on the Scene.

The Space-Time of the Objects may be superseded by the Space-Time of the Scene.

8.73.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/BasicSpeechSceneDescriptors.json>

8.73.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Speech Scene Descriptors Header
- Standard-BasicSpeechSceneDescriptors	9 Bytes	The characters “OSD-BSS-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneDescriptorsID	N5 Bytes	Identifier of Scene Descriptors.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SceneDescriptorsSpaceTime	N7 Bytes	Space and Time of Scene Descriptors.
SceneObjects[]	N8 Bytes	Set of Objects.
- SceneObject	N9 Bytes	An Object.
- SceneObjectSpaceTime	N10 Bytes	Space Time of Object.
Rights	N11 Bytes	Rights that may be exercised on the Scene.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.73.5 Conformance Testing

A Data instance Conforms with Basic Scene Descriptors (OSD-BSS) V1.3 if:

1. The Data validates against the Scene Descriptors’ JSON Schema.
2. All Data in the Scene Descriptors’ JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.74 Basic Text Object

8.74.1 Definition

A Data Type including

1. Data representing:

1. A sound of the real world, or
 2. Synthetic Data intended for rendering as sound, or
 3. A mixture of the two types, and
2. Optionally, descriptive Data regarding Audio Sub-Types, Formats and Attributes.

In the following, Data, Qualifier, and Object should be read as Text Data, Text Qualifiers, and Text Object, respectively.

8.74.2 Functional Requirements

A Basic Object may include:

1. The ID of a Virtual Space (M-Instance) where it is or is intended to be located.
2. The ID of the Basic Object.
3. The ID(s) of Parent Object(s) supporting two cases:
 1. The Parent Object has spawned two (or more) Objects.
 2. Two (or more) Parent Objects have merged into one.
4. The Space-Time information of Parent Objects in an M-Instance.
5. The ID(s) of Child Object(s).
6. The Space-Time information of Child Objects in an M-Instance.
7. The Space-Time information of the Basic Object in an M-Instance.
8. The Qualifier of the specific Data Type.
9. The Rights that can be exercised on the Basic Object.
10. The set of Annotations including, for each Annotation:
 1. Space-Time information of the Annotation.
 2. Rights to perform Actions on the Annotation.

8.74.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/BasicTextObject.json>

8.74.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Text Object Header
– Standard-BasicTextObject	9 Bytes	The characters “OSD-BTO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
BasicObjectID	N5 Bytes	Identifier of the Basic Object.
BasicObjectSpaceTime	N6 Bytes	Space-Time info of the Basic Object.
Qualifier	N7 Bytes	Qualifier of Basic Data.
BasicObjectAnnotations[]	N8 Bytes	Annotations of Basic Object.
– Annotation	N9 Bytes	ID of Annotation

– AnnotationSpaceTime	N10 Bytes Where/when Annotation is attached.
Rights	N11 Bytes Rights to perform Actions of the Basic Object.
DescrMetadata	N12 Bytes Descriptive Metadata

8.74.5 Conformance Testing

A Data instance Conforms with Basic Object V1.3 if:

1. The Data validates against the Basic Object's JSON Schema.
2. All Data in the Basic Object's JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers, if present.

8.75 Basic Visual Object

8.75.1 Definition

A Data Type including:

1. Visual Data perceptible by a visual device or visible to a human when rendered.
2. Descriptive Data regarding Sub-Types, Formats and Attributes of the Visual Data (optionally).

In the following, Data, Qualifier, and Object should be read as Visual Data, Visual Qualifiers, and Visual Object, respectively.

8.75.2 Functional Requirements

A Basic Object may include:

1. The ID of a Virtual Space (M-Instance) where it is or is intended to be located.
2. The ID of the Basic Object.
3. The ID(s) of Parent Object(s) supporting two cases:
 1. The Parent Object has spawned two (or more) Objects.
 2. Two (or more) Parent Objects have merged into one.
4. The Space-Time information of Parent Objects in an M-Instance.
5. The ID(s) of Child Object(s).
6. The Space-Time information of Child Objects in an M-Instance.
7. The Space-Time information of the Basic Object in an M-Instance.
8. The Qualifier of the specific Data Type.
9. The Rights that can be exercised on the Basic Object.
10. The set of Annotations including, for each Annotation:
 1. Space-Time information of the Annotation.
 2. Rights to perform Actions on the Annotation.

8.75.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/BasicVisualObject.json>

8.75.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Visual Object Header
– Standard-BasicVisualObject	9 Bytes	The characters “OSD-BSO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
BasicObjectID	N5 Bytes	Identifier of the Basic Object.
BasicObjectSpaceTime	N6 Bytes	Space-Time info of the Basic Object.
Qualifier	N7 Bytes	Qualifier of Basic Data.
BasicObjectAnnotations[]	N8 Bytes	Annotations of Basic Object.
– Annotation	N9 Bytes	ID of Annotation
– AnnotationSpaceTime	N10 Bytes	Where/when Annotation is attached.
Rights	N11 Bytes	Rights to perform Actions of the Basic Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.75.5 Conformance Testing

A Data instance Conforms with Basic Object V1.3 if:

1. The Data validates against the Basic Object’s JSON Schema.
2. All Data in the Basic Object’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers, if present.

8.76 Basic Visual Scene Descriptors

8.76.1 Definition

A Data Type including the Visual Objects of a scene, their time and arrangement in the scene, and the Rights that may be exercised on the scene.

In the following Object and Scene are to be read as Visual Object and Visual Scene, respectively.

8.76.2 Functional Requirements

Basic Scene Descriptors include

1. Objects
2. Space-Time information.

3. Rights that may be exercised on the Scene.

The Space-Time of the Objects may be superseded by the Space-Time of the Scene.

8.76.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/BasicVisualSceneDescriptors.json>

8.76.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Visual Scene Descriptors Header
- Standard-BasicVisualSceneDescriptors	9 Bytes	The characters “OSD-BVS-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneDescriptorsID	N5 Bytes	Identifier of Scene Descriptors.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SceneDescriptorsSpaceTime	N7 Bytes	Space and Time of Scene Descriptors.
SceneObjects[]	N8 Bytes	Set of Objects.
- SceneObject	N9 Bytes	An Object.
- SceneObjectSpaceTime	N10 Bytes	Space Time of Object.
Rights	N11 Bytes	Rights that may be exercised on the Scene.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.76.5 Conformance Testing

A Data instance Conforms with Basic Scene Descriptors (OSD-BVS) V1.3 if:

1. The Data validates against the Scene Descriptors’ JSON Schema.
2. All Data in the Scene Descriptors’ JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.77 Coordinates

8.77.1 Definition

A set of numbers used to indicate the position of a point in a space.

8.77.2 Functional Requirements

All points in the space shall have a set of numbers representing them.

The coordinate systems supported so far are:

1. Cartesian
2. Spherical
3. Geodesic
4. Cylindrical
5. Toroidal

8.77.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/Coordinates.json>

8.77.4 Semantics

Label	Size	Description
Header	N1 Bytes	Time Header
- Standard-Object	9 Bytes	The characters “OSD-CRD-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance
CoordinatesID	N5 Bytes	Identifier of Coordinates.
CoordinateTypes	N6 Bytes	One of Cartesian, Spherical, Geodesic, Cylindrical, Toroidal.
CoordinateData	N7 Bytes	Three numbers
DescrMetadata	N8 Bytes	Descriptive Metadata

8.77.5 Conformance Testing

A Data instance Conforms with Coordinates (OSD-CRD) V1.2 if all the Data:

1. Have the specified type.
2. Validate against the Coordinates’ JSON Schema.

8.78 Instance Identifier

8.78.1 Definition

A Data Type associating a string (Identifier) with an element of a set of entities – Speech, Objects, Visual Objects, User IDs etc. – belonging to some levels in a hierarchical classification (taxonomy).

8.78.2 Functional Requirements

Instance Identifier includes:

1. ID of Virtual Space (M-Instance)
2. Instance Label
3. Confidence level of the association between Instance Label and Instance.
4. Taxonomy
5. Confidence level of the association between Taxonomy and the Instance.

8.78.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/InstanceIdentifier.json>

8.78.4 Semantics

Label	Size	Description
Header	N1 Bytes	Instance Identifier Header
– Standard-InstanceIdentifier	9 Bytes	The characters “OSD-IID-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance
InstanceID	N5 Bytes	Identifier of Instance.
InstanceSpaceTime	N6 Bytes	Data about Space-Time
InstanceIdentifierData	N7 Bytes	Data set of Instance Identifier.
InstanceLabel	N8 Bytes	Instance identified by Instance Identifier.
LabelConfidenceLevel	N9 Bytes	Confidence of Instance Label and Instance association.
TaxonomyLabel	N10 Bytes	Taxonomy Instance Identifier belongs to.
TaxonomyConfidenceLevel	N11 Bytes	Confidence of Taxonomy Label .
TaxonomyDataLength	N12 Bytes	Number of Bytes
TaxonomyDataURI	N13 Bytes	URI of Taxonomy.
DescrMetadata	N14 Bytes	Descriptive Metadata

8.78.5 Conformance Testing

A Data instance Conforms with Instance Identifier (OSD-IID) V1.3 if:

1. The Data validates against the Instance Identifier’s JSON Schema.
2. All Data in the Instance Identifier’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.79 Location

8.79.1 Definition

A region of an entity with Space-Time attributes that is further subdivided in Basic Locations.

8.79.2 Functional Requirements

A Location

1. Has an extension limited in Space and Time.
2. Is composed of Basic Locations, e.g.:
 1. A room can be a Basic Location of the Location defined as an apartment.
 2. An apartment can be a Basic Location of the Location defined as a building.

8.79.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/Location.json>

8.79.4 Semantics

Label	Size	Description
Header	N1 Bytes	Location Header
– Standard	9 Bytes	The characters “MMM-LOC-V”
– Version	N2 Bytes	Major version
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version
MInstanceID	N4 Bytes	Identifier of M-Instance.
LocationID	N5 Bytes	Identifier of Location.
LocationData	N6 Bytes	Locations and Basic-Locations composing the Location.
– BasicLocation	N7 Bytes	A Basic Location composing the Location.
– Time	N8 Bytes	Time of validity of Basic Location in Location
DescrMetadata	N9 Bytes	Descriptive Metadata.

8.79.5 Conformance Testing

A Data instance Conforms with Location (OSD-LOC) V1.2 if:

1. The Data validates against the Location’s JSON Schema.
2. All Data in the Location’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.80 Object

8.80.1 Definition

A Data Type including a collection of Basic Objects possibly of different Media Types.

An Object can have a hierarchical structure where Objects contain Basic Objects and Objects.

8.80.2 Functional Requirements

An Object may include:

1. ID of a Virtual Space (M-Instance) where it is or intended to be located.
2. ID of the Object.
3. Space-Time information of the Object.
4. Basic Object and Objects included in the Objects.
5. Annotation data set including:
 1. Annotations
 2. Space-Times of the Annotations.
 3. Rights to perform Actions on the Object.
6. The Rights that may be exercised on the Object.

Note that.

1. An Object that does not include Sub-Scenes and only one Basic Object is a Basic Object.
2. The Space-Time information of a Basic Object, Object, or Basic Scene included in an Object may be superseded by the Space-Time information of the Object containing it.

The following table gives the Media Type-specific Objects.

Object name	Acronym	Json
3DModel Object	OSD-3DO	Link
Audio Object	OSD-AUO	Link
Audio-Visual Object	OSD-AVO	Link
LiDAR Object	OSD-LIO	Link
Offline Map Object	OSD-OMO	Link
RADAR Object	OSD-RAO	Link
Speech Object	OSD-SPO	Link
Text Object	OSD-TXO	Link
Ultrasound Object	OSD-USO	Link
Visual Object	OSD-VIO	Link

8.80.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/Object.json>

8.80.4 Semantics

Label	Size	Description
Header	N1 Bytes	Object Header
– Standard-Object	9 Bytes	The characters “OSD-Obj-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
ObjectID	N5 Bytes	Identifier of the Object.
ObjectSpaceTime	N6 Bytes	Space-Time of Object.
BasicObjectCount	N7 Bytes	Set of Parent Objects.
BasicObjects[]	N8 Bytes	Set of Basic Objects.
- BasicObjectSpaceTime	N9 Bytes	Space Time of a Basic Object in the Object.
- BasicObject	N10 Bytes	A Basic Object in the Object.
ObjectCount	N11 Bytes	Set of Child Objects.
Objects[]	N12 Bytes	Set of Objects.
- ObjectSpaceTime	N13 Bytes	Space Time of an Object in the Object.
- Object	N14 Bytes	An Object in the Object
DataAnnotations[]	N14 Bytes	Set of Object Annotation.
– Annotation	N15 Bytes	An Annotation.
– AnnotationSpaceTime	N15 Bytes	Where Annotation is attached and when it will be active.
– Rights	N16 Bytes	Actions that may be performed on the Annotation
Rights	N17 Bytes	Actions that may be performed on the Object.
DescrMetadata	N17 Bytes	Descriptive Metadata

8.80.5 Conformance Testing

A Data instance Conforms with Object (OSD-Obj) V1.3 if:

1. The Data validates against the Object’s JSON Schema.
2. All Data in the Object’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.81 Orientation

8.81.1 Definition

An Item representing an Object's orientation, orientation velocity, and orientation acceleration.

8.81.2 Functional Requirements

- The Orientation of an Object is that of the principal axis of an Object.
- The following media types are supported: Speech, Audio; Visual; 3D Model; Audio-Visual; Haptic; Smell; RADAR; LiDAR; Ultrasound.
- Accuracy is the estimated absolute difference between the measures of each of Orientation, OrientVelocity, and OrientAccel and their true values.

8.81.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/Orientation.json>

8.81.4 Semantics

Table 9 provides the semantics of the components of Orientation. It should be noted that the Euler angles are indicated by (α, β, γ) .

Table 9 - Semantics of Orientation

Label	Size	Description
Header	N1 Bytes	Orientation Header
- Standard-Orientation	9 Bytes	The characters "OSD-OOR-V"
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstance	N4 Bytes	ID id Virtual space Orientation refers tu
ObjectOrientationID	N5 Bytes	Identifier of Object Orientation.
General	N6 Bytes	Set of general data
- ObjectType	N8 Bytes	One of Digital Human, Generic.
- MediaType	N9 Bytes	One of Speech, Audio, Visual, Audio-Visual, Haptic, Smell, RADAR, LiDAR, Ultrasound.
Orientation		
- Orient (α, β, γ)	N10 Bytes	Array (in degrees)
- OrientAccuracy	N11 Bytes	Array of (α, β, γ) Accuracy
Velocity of Orientation		
- OrientVelocity (α, β, γ)	N12 Bytes	Array (in degrees/s)
- OrientVelocityAccuracy	N13 Bytes	Array of (α, β, γ) Velocity Accuracy
Acceleration of Orientation		
- OrientAccel (α, β, γ)	N14 Bytes	Array (in degrees/s ²)

- OrientAcceAccuracy N15 Bytes Array of (α, β, γ) Acceleration Accuracy
- DescrMetadata** N16 Bytes Descriptive Metadata

8.81.5 Conformance Testing

A Data instance Conforms with Orientation (OSD-OOR) V1.2 if:

1. The Data validates against the Orientation's JSON Schema.
2. All Data in the Orientation's JSON Schema have the specified types.

8.82 Perceptible Entity

8.82.1 Definition

Perceptible Entity is one of

1. Text, Speech, Audio, Visual, 3D Model, and Audio-Visual Object.
2. Speech, Audio, Visual, 3D Model, and Audio-Visual Scene.
3. Audio-Visual Event.

8.82.2 Functional Requirements

A Perceptible Entity

1. Inherits the Functional requirements of Objects, Scenes, and Events listed above.
2. May include Rights that are Granted to certain Process to perform certain Actions at certain Times and Locations on the Perceptible Entity.

8.82.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/PerceptibleEntity.json>

8.82.4 Semantics

Label	Size	Description
Header	N1 Bytes	Perceptible Entity Header
- Standard-PerceptibleEntity	9 Bytes	The characters "OSD-PCE-V"
- Version	N2 Bytes	Major version – 1 or 2 Bytes
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 Bytes
PerceptibleEntityID	N4 Bytes	Identifier of Perceptible Entity.
PerceptibleEntity	N5 Bytes	Anyone of the following Objects, Scenes, or Events.
- TextObject	N6 Bytes	Intended Text Object
- SpeechObject	N7 Bytes	Intended Speech Object
- AudioObject	N8 Bytes	Intended Audio Object

- VisualObject	N9 Bytes	Intended Visual Object
- 3D Model	N10 Bytes	Intended 3D Model Object
- AudioVisualObject	N11 Bytes	Intended Audio-Visual Object
- SpeechScene	N12 Bytes	Intended Speech Scene
- AudioScene	N13 Bytes	Intended Audio Scene
- VisualScene	N14 Bytes	Intended Visual Scene
- 3D Model Scene	N15 Bytes	Intended 3D Model Scene
- AudioVisualScene	N16 Bytes	Intended Audio-Visual Scene
- AudioVisualEvent	N17 Bytes	Intended Audio-Visual Event
- RightsID	N18 Bytes	Individual Rights ID
- Rights	N19 Bytes	Individual Rights
DescrMetadata	N20 Bytes	Descriptive Metadata

8.82.5 Conformance Testing

A Data instance Conforms with Perceptible Entity (OSD-PCE) V1.3 if:

1. The Data validates against the Perceptible Entity's JSON Schema.
2. All Data in the Perceptible Entity's JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.83 Point of View

8.83.1 Definition

Position and Orientation of an Object in a Virtual Environment excluding velocity and acceleration.

8.83.2 Functional Requirements

- An Object may have one of the following attributes: Speech, Audio; Visual; 3D Model, Audio-Visual; Haptic; Smell; RADAR; LiDAR; Ultrasound.
- Accuracy is the estimated absolute difference between the measured spatial and angular values of each of CartPosition, SpherPosition, Orientation, and their true value.

8.83.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/PointOfView.json>

8.83.4 Semantics

Table 10 provides the semantics of the components of Point of View. The following should be noted:

1. Each of Position, Velocity, and Acceleration is provided either in Cartesian (X,Y,Z) or Spherical (r,φ,θ) Coordinates.
2. The Euler angles are indicated by (α,β,γ).

Table 10 - Semantics of Point of View

Label	Size	Description
Header	N1 Bytes	Point of View Header
- Standard-Point of View	9 Bytes	The characters “OSD-OPV-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstance	N4 Bytes	ID id Virtual space Orientation refers tu
PointOfViewID	N5 Bytes	Identifier of Object Point of View.
General	N6 Bytes	Set of general data.
- CoordType	N7 Bytes	One of Cartesian, Spherical, Geodesic, Toroidal.
- ObjectType	N8 Bytes	One of Digital Human, Generic.
- MediaType	N9 Bytes	One of Speech, Audio, Visual, Audio-Visual, Haptic, Smell, RADAR, LiDAR, Ultrasound.
PositionAndOrientation		
- CartPosition (X,Y,Z)	N10 Bytes	Array (in metres)
- CartPositionAccuracy (X,Y,Z)	N11 Bytes	Array Of CartPositionAccuracy
- SpherPosition (r,φ,θ)	N12 Bytes	Array (in metres and degrees)
- SpherPositionAccuracy (r,φ,θ)	N13 Bytes	Array of - SpherPositionAccuracy
- Orient (α,β,γ)	N14 Bytes	Array (in degrees)
- OrientAccuracy (α,β,γ)	N15 Bytes	Array of OrientAccuracy
DescrMetadata	N16 Bytes	Descriptive Metadata

8.83.5 Conformance Testing

A Data instance Conforms with MPAI-OSD Point of View (OSD-OPV) V1.3 if:

1. The Data validates against the Point of View’s JSON Schema.
2. All Data in the Point of View’s JSON Schema.
 1. Have the specified type.
 2. Validate against their JSON Schemas.

8.84 Position

8.84.1 Definition

A Data Type representing an Object's position, velocity, and acceleration.

8.84.2 Functional Requirements

- The Position of an Object is that of a representative point in the Object.
- Cartesian and Polar Coordinate Systems are supported.
- An Object may have one of the following attributes: Speech, Audio; Visual; 3D Model, Audio-Visual; Haptic; Smell; RADAR; LiDAR; Ultrasound.
- Accuracy is the estimated absolute difference between the measured spatial values of each of CartPosition, SpherPosition, CartVelocity, SpherVelocity, CartAccel, SpherAccel and their true value.

8.84.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/Position.json>

8.84.4 Semantics

Table 11 provides the semantics of Position. It should be noted that each of Position, Velocity, and Acceleration can be expressed either in Cartesian (X,Y,Z) or Spherical (r,φ,θ) Coordinates.

Table 11 - Semantics of the Spatial Attitude

Label	Size	Description
Header	N1 Bytes	Position Header
- Standard-Position	9 Bytes	The characters "OSD-OPS-V"
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	ID of Virtual Space Position refers to.
ObjectPositionID	N5 Bytes	Identifier of Object Position.
General	N6 Bytes	Set of general data
- CoordinateType	N7 Bytes	One of Cartesian, Spherical, Geodesic, Toroidal.
- ObjectType	N8 Bytes	One of Digital Human, Generic.
- MediaType	N9 Bytes	One of Speech, Audio, Visual, 3D Model, Audio-Visual, Haptic, Smell, RADAR, LiDAR, Ultrasound.
Position		
- CartPosition (X,Y,Z)	N10 Bytes	Array (in metres)
- CartPositionAccuracy (X,Y,Z)	N11 Bytes	Array of CartPositionAccuracy
- SpherPosition (r,φ,θ)	N12 Bytes	Array (in metres and degrees)

- SpherPositionAccuracy (r,φ,θ)	N13 Bytes	Array of SpherPositionAccuracys
Velocity of Position		
- CartVelocity (X,Y,Z)	N14 Bytes	Array (in metres)
- CartVelocityAccuracy (X,Y,Z)	N15 Bytes	Array of - CartVelocityAccuracys (X,Y,Z)
- SpherVelocity (r,φ,θ)	N16 Bytes	Array (in metres and degrees)
- SpherVelocityAccuracy (r,φ,θ)	N17 Bytes	Array of SpherVelocityAccuracys
Acceleration of Position		
- CartAccel (X,Y,Z)	N18 Bytes	Array (in metres)
- CartAccelAccuracy (X,Y,Z)	N19 Bytes	Array of CartAccelAccuracys
- SpherAccel (r,φ,θ)	N20 Bytes	Array (in metres and degrees)
- SpherAccel (r,φ,θ)	N21 Bytes	Array (in metres and degrees)
DescrMetadata	N22 Bytes	Descriptive Metadata

8.84.5 Conformance Testing

A Data instance Conforms with MPAI-OSD V1.3 Position (OSD-OPS) if:

1. The Data validates against the Position 's JSON Schema.
2. All Data in the Position 's JSON Schema have the specifies type.

8.85 Scene Descriptors

8.85.1 Definition

A Data Type including the Objects of a scene, their sub-scenes, and their arrangement in the scene.

8.85.2 Functional Requirements

Scene Descriptors include

1. Objects
2. The Descriptors of the Scenes includes in the Scene called Sub-Scenes.
3. Rights that may be exercised on the Scene.

Scenes may be hierarchical, i.e., they may contain Objects and Scenes.

Scene Descriptors name	Acronym	JSON
3DModel Scene Descriptors	OSD-3BS	Link
Audio Scene Descriptors	OSD-ABS	Link
Audio-Visual Scene Descriptors	OSD-MBS	Link
LiDAR Scene Descriptors	OSD-LBS	Link
Offline Map Scene Descriptors	OSD-OBS	Link

RADAR Scene Descriptors	OSD-RBS	Link
Speech Scene Descriptors	OSD-SBS	Link
Ultrasound Scene Descriptors	OSD-UBS	Link
Visual Basic Descriptors	OSD-VBS	Link

8.85.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/SceneDescriptors.json>

8.85.4 Semantics

Label	Size	Description
Header	N1 Bytes	Scene Descriptors Header
- Standard-SceneDescriptors	9 Bytes	The characters “OSD-SCD-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneDescriptorsID	N5 Bytes	Identifier of Scene Descriptors.
SceneDescriptorsSpaceTime	N6 Bytes	Space and Time of Scene Descriptors.
ObjectCount	N7 Bytes	Number of Objects in Scene.
Objects[]	N8 Bytes	Set of Objects.
- Object or ObjectID	N9 Bytes	Object in the Scene of its ID.
- ObjectSpaceTime	N10 Bytes	Space Time of Object.
SubSceneCount	N11 Bytes	Number of Sub-Scenes in Scene.
SubScenes[]	N12 Bytes	Set of Sub-Scenes in the Scene.
- SubScene or SubSceneID	N13 Bytes	Sub-Scene in the Scene or its ID.
- SubSceneSpaceTime	N14 Bytes	Space Time of Sub-Scene.
DescrMetadata	N15 Bytes	Descriptive Metadata

8.85.5 Conformance Testing

A Data instance Conforms with Scene Descriptors (OSD-SCD) V1.3 if:

1. The Data validates against the Scene Descriptors’ JSON Schema.
2. All Data in the Scene Descriptors’ JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.86 Space-Time

8.86.1 Definition

Data Type representing the Spatial Attitude and Time information.

8.86.2 Functional Requirements

Space-Time includes Spatial Attitude and Time.

8.86.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/SpaceTime.json>

8.86.4 Semantics

Label	Size	Description
Header	N1 Bytes	Space-Time Header
- Standard-Object	9 Bytes	The characters “OSD-SPT-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstance	N4 Bytes	Identifier of Virtual Space.
SpaceTimeID	N5 Bytes	Identifier of Space-Time.
Space	N6 Bytes	Spatial Attitudes at T ₀ and T ₁
Time	N7 Bytes	Time interval between T ₀ and T ₁
DescrMetadata	N8 Bytes	Descriptive Metadata

8.86.5 Conformance Testing

A Data instance Conforms with Space-Time (OSD-SPT) V1.3 if:

1. The Data validates against the Space-Time’s JSON Schema.
2. All Data in the Space-Time’s JSON Schema
 1. Have the specified type.
 2. Validate against their JSON Schemas.
 3. Conform with their Data Qualifiers if present.

8.87 Spatial Attitude

8.87.1 Definition

An Item representing the Position and Orientation of an Object, and their velocities and accelerations.

8.87.2 Functional Requirements

The Spatial Attitude is defined as the combination of Position and orientation, the Functional Requirements are defined by Position and Orientation.

8.87.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/SpatialAttitude.json>

8.87.4 Semantics

Table 12 provides the semantics of the components of the Spatial Attitude.

Table 12 - Semantics of the Spatial Attitude

Label	Size	Description
Header	N1 Bytes	Spatial Attitude Header
- Standard-SpatialAttitude	9 Bytes	The characters “OSD-OSA-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	ID of Virtual Space Object refers to.
ObjectSpatialAttitudeID	N5 Bytes	Identifier of Object Spatial Attitude.
General	N6 Bytes	Set of general data
- CoordinateType	N7 Bytes	One of Cartesian, Spherical, Geodesic, Toroidal.
- ObjectType	N8 Bytes	One of Digital Human, Generic.
- MediaType	N9 Bytes	One of Speech, Audio, Visual, Audio-Visual, Haptic, Smell, RADAR, LiDAR, Ultrasound.
Position	N10 Bytes	As specified by Position
Orientation	N11 Bytes	As specified by Orientation
DescrMetadata	N20 Bytes	Descriptive Metadata

8.87.5 Conformance Testing

A Data instance Conforms with V1.2 Spatial Attitude V1.3 (OSD-OSA) if:

1. The Data validates against the Spatial Attitude’s JSON Schema.
2. All Data in the Spatial Attitude ’s JSON Schema have the specified type.

8.88 Speech Event Descriptors

8.88.1 Definition

An Item including a series of Speech Scene Descriptors for a certain duration.

8.88.2 Functional Requirements

Speech Event Descriptors contains Speech Scene Descriptors for a Time.

8.88.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/SpeechEventDescriptors.json>

8.88.4 Semantics

Label	Size	Description
Header	N1 Bytes	Speech Event Descriptors Header
- Standard-SpeechEventDescriptors	9 Bytes	The characters “OSD-SPE-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
EventID	N5 Bytes	Identifier of the Event.
EventSpaceTime	17 Bytes	Data about start and end Space-Time.
SceneDescriptors[]	N6 Bytes	Set of Scene Descriptors
- SceneDescriptors	N7 Bytes	Set of AV Scene Descriptors of IDs.
DescrMetadata	N8 Bytes	Descriptive Metadata

8.88.5 Conformance Testing

A Data instance Conforms with MPAI-OSD Speech Event Descriptors V1.3 (OSD-SPE) if:

1. The Data validates against the Speech Event Descriptors’ JSON Schema.
2. All Data in the Speech Event Descriptors’ JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.89 Speech Object

8.89.1 Definition

A Data Type including a collection of Basic Speech Objects.

A Speech Object can have a hierarchical structure where Speech Objects contain Basic Speech Objects and Speech Objects.

8.89.2 Functional Requirements

A Speech Object may include:

1. ID of a Virtual Space (M-Instance) where it is or intended to be located.
2. ID of the Speech Object.
3. Space-Time information of the Speech Object.
4. Basic Speech Object and Speech Objects included in the Speech Objects.
5. Annotation data set including:
 1. Annotations
 2. Space-Times of the Annotations.
 3. Rights to perform Actions on the Speech Object.
6. The Rights that may be exercised on the Speech Object.

Note that.

1. A Speech Object that does not include Sub-Scenes and only one Basic Speech Object is a Basic Speech Object.
2. The Space-Time information of a Basic Speech Object and Speech Object included in a Speech Object may be superseded by the Space-Time information of the Speech Object containing them.

8.89.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/SpeechObject.json>

8.89.4 Semantics

Label	Size	Description
Header	N1 Bytes	Speech Object Header
– Standard-SpeechObject	9 Bytes	The characters “OSD-SPO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SpeechObjectID	N5 Bytes	Identifier of the Speech Object.
SpeechObjectSpaceTime	N6 Bytes	Space-Time of Speech Object.
BasicSpeechObjectCount	N7 Bytes	Set of Parent Speech Objects.
BasicSpeechObjects[]	N8 Bytes	Set of Basic Speech Objects.
- SpaceTime	N9 Bytes	Space Time of a Basic Speech Object in the Speech Object.
- BasicSpeechObject	N10 Bytes	A Basic Speech Object in the Speech Object.
SpeechObjectCount	N11 Bytes	Number of Speech Objects.
SpeechObjects[]	N12 Bytes	Set of Speech Objects.
- SpaceTime	N13 Bytes	Space Time of a Speech Object in the Speech Object.
- SpeechObject	N14 Bytes	A Speech Object in the Speech Object
Annotations[]	N15 Bytes	Set of Speech Object Annotation.
– Annotation	N16 Bytes	An Annotation.
– AnnotationSpaceTime	N17 Bytes	Where Annotation is attached and when it will be active.

– Rights	N18 Bytes	Actions that may be performed on the Annotation
Rights	N19 Bytes	Actions that may be performed on the Object.
DescrMetadata	N20 Bytes	Descriptive Metadata

8.89.5 Conformance Testing

A Data instance Conforms with Speech Object (OSD-SPO) V1.3 if:

1. The Data validates against the Speech Object’s JSON Schema.
2. All Data in the Speech Object’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.90 Speech Scene Descriptors

8.90.1 Definition

A Data Type including the Speech Objects of a scene, their sub-scenes, and their arrangement in the scene.

8.90.2 Functional Requirements

Speech Scene Descriptors include

1. Speech Objects
2. The Descriptors of the Speech Scenes includes in the Speech Scene called Speech Sub-Scenes.
3. Rights that may be exercised on the Speech Scene.

Scenes may be hierarchical, i.e., they may contain Speech Objects and Speech Scenes.

8.90.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/SpeechSceneDescriptors.json>

8.90.4 Semantics

	Label	Size	Description
	Header	N1 Bytes	Speech Scene Descriptors Header
-	Standard-SpeechSceneDescriptors	9 Bytes	The characters “OSD-SSD-V”
-	Version	N2 Bytes	Major version – 1 or 2 characters
-	Dot-separator	1 Byte	The character “.”
-	Subversion	N3 Bytes	Minor version – 1 or 2 characters
	MInstanceID	N4 Bytes	Identifier of M-Instance.
	SceneDescriptorsID	N5 Bytes	Identifier of Scene Descriptors.
	SceneDescriptorsSpaceTime	N6 Bytes	Space and Time of Scene Descriptors.

ObjectCount	N7 Bytes	Number of Objects in Scene.
Objects[]	N8 Bytes	Set of Objects.
- Object or ObjectID	N9 Bytes	Object in the Scene of its ID.
- ObjectSpaceTime	N10 Bytes	Space Time of Object.
SubSceneCount	N11 Bytes	Number of Sub-Scenes in Scene.
SubScenes[]	N12 Bytes	Set of Sub-Scenes in the Scene.
- SubScene or SubSceneID	N13 Bytes	Sub-Scene in the Scene or its ID.
- SubSceneSpaceTime	N14 Bytes	Space Time of Sub-Scene.
DescrMetadata	N15 Bytes	Descriptive Metadata

8.90.5 Conformance Testing

A Data instance Conforms with Speech Scene Descriptors (OSD-SSD) V1.3 if:

1. The Data validates against the Scene Descriptors' JSON Schema.
2. All Data in the Scene Descriptors' JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.91 Text Object

8.91.1 Definition

A Data Type including a collection of Basic Text Objects.

A Text Object can have a hierarchical structure where Text Objects contain Basic Text Objects and Text Objects.

8.91.2 Functional Requirements

A Text Object may include:

1. ID of a Virtual Space (M-Instance) where it is or intended to be located.
2. ID of the Text Object.
3. Space-Time information of the Text Object.
4. Basic Text Object and Text Objects included in the Text Objects.
5. Annotation data set including:
 1. Annotations
 2. Space-Times of the Annotations.
 3. Rights to perform Actions on the Text Object.
6. The Rights that may be exercised on the Text Object.

Note that.

1. A Text Object that does not include Sub-Scenes and only one Basic Text Object is a Basic Text Object.
2. The Space-Time information of a Basic Text Object and Text Object included in a Text Object may be superseded by the Space-Time information of the Text Object containing them.

8.91.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/TextObject.json>

8.91.4 Semantics

Label	Size	Description
Header	N1 Bytes	Text Object Header
– Standard-TextObject	9 Bytes	The characters “OSD-TXO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
TextObjectID	N5 Bytes	Identifier of the Text Object.
TextObjectSpaceTime	N6 Bytes	Space-Time of Text Object.
BasicTextObjectCount	N7 Bytes	Set of Parent Text Objects.
BasicTextObjects[]	N8 Bytes	Set of Basic Text Objects.
- SpaceTime	N9 Bytes	Space Time of a Basic Text Object in the Text Object.
- BasicTextObject	N10 Bytes	A Basic Text Object in the Text Object.
TextObjectCount	N11 Bytes	Number of Text Objects.
TextObjects[]	N12 Bytes	Set of Text Objects.
- SpaceTime	N13 Bytes	Space Time of a Text Object in the Text Object.
- TextObject	N14 Bytes	A Text Object in the Text Object
Annotations[]	N15 Bytes	Set of Text Object Annotation.
– Annotation	N16 Bytes	An Annotation.
– AnnotationSpaceTime	N17 Bytes	Where Annotation is attached and when it will be active.
– Rights	N18 Bytes	Actions that may be performed on the Annotation
Rights	N19 Bytes	Actions that may be performed on the Object.
DescrMetadata	N20 Bytes	Descriptive Metadata

8.91.5 Conformance Testing

A Data instance Conforms with Text Object (OSD-TXO) V1.3 if:

1. The Data validates against the Text Object’s JSON Schema.
2. All Data in the Text Object’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.92 Time

8.92.1 Definition

The start time and the end time of a duration.

8.92.2 Functional Requirements

Origin of Time can be Absolute (from 1970/01/01) or relative to a user-selected value.

8.92.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/Time.json>

8.92.4 Semantics

Label	Size	Description
Header	N1 Bytes	Time Header
- Standard-Object	9 Bytes	The characters “OSD-TIM-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance
TimeID	N5 Bytes	Identifier of M-Instance.
TimeData	17 Bytes	Data about Time
- TimeType	0 bit	0=Relative: start at 0000/00/00T00:00 1=Absolute: start at 1970/01/01T00:00.
- TimeUnit	1-5	reserved
- Reserved	6-7 bits	00=seconds, 01=milliseconds, 10=microseconds, 11=nanoseconds.
- StartTime	8 Bytes	Start of Time.
- EndTime	8 Bytes	End of Time.
DescrMetadata	N6 Bytes	Descriptive Metadata

8.92.5 Conformance Testing

A Data instance Conforms with MPAI-OSD Time V1.3 (OSD-TIM) if:

1. The Data validates against the Times’s JSON Schema.
2. All Data in the Times’s JSON Schema have the specified type.

8.93 Visual Event Descriptors

8.93.1 Definition

An Item including a series of Visual Scene Descriptors for a certain duration.

8.93.2 Functional Requirements

Visual Event Descriptors contains Visual Scene Descriptors for a Time.

8.93.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/UltrasoundEventDescriptors.json>

8.93.4 Semantics

Label	Size	Description
Header	N1 Bytes	Visual Event Descriptors Header
- Standard-VisualEventDescriptors	9 Bytes	The characters “OSD-VIEE-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
EventID	N5 Bytes	Identifier of the Event.
EventSpaceTime	17 Bytes	Data about start and end Space-Time.
SceneDescriptors[]	N6 Bytes	Set of Scene Descriptors
- SceneDescriptors	N7 Bytes	Set of AV Scene Descriptors of IDs.
DescrMetadata	N8 Bytes	Descriptive Metadata

8.93.5 Conformance Testing

A Data instance Conforms with MPAI-OSD Visual Event Descriptors V1.3 (OSD-VIE) if:

1. The Data validates against the Visual Event Descriptors’ JSON Schema.
2. All Data in the Visual Event Descriptors’ JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers if present.

8.94 Visual Object

8.94.1 Definition

A Data Type including a collection of Basic Visual Objects.

A Visual Object can have a hierarchical structure where Visual Objects contain Basic Visual Objects and Visual Objects.

8.94.2 Functional Requirements

A Visual Object may include:

1. ID of a Virtual Space (M-Instance) where it is or intended to be located.
2. ID of the Visual Object.
3. Space-Time information of the Visual Object.
4. Basic Visual Object and Visual Objects included in the Visual Objects.
5. Annotation data set including:
 1. Annotations
 2. Space-Times of the Annotations.
 3. Rights to perform Actions on the Visual Object.
6. The Rights that may be exercised on the Visual Object.

Note that.

1. A Visual Object that does not include Sub-Scenes and only one Basic Visual Object is a Basic Visual Object.
2. The Space-Time information of a Basic Visual Object and Visual Object included in a Visual Object may be superseded by the Space-Time information of the Visual Object containing them.

8.94.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/VisualObject.json>

8.94.4 Semantics

Label	Size	Description
Header	N1 Bytes	Visual Object Header
– Standard-VisualObject	9 Bytes	The characters “OSD-VIO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
VisualObjectID	N5 Bytes	Identifier of the Visual Object.
VisualObjectSpaceTime	N6 Bytes	Space-Time of Visual Object.
BasicVisualObjectCount	N7 Bytes	Set of Parent Visual Objects.
BasicVisualObjects[]	N8 Bytes	Set of Basic Visual Objects.
- SpaceTime	N9 Bytes	Space Time of a Basic Visual Object in the Visual Object.
- BasicVisualObject	N10 Bytes	A Basic Visual Object in the Visual Object.
VisualObjectCount	N11 Bytes	Number of Visual Objects.
VisualObjects[]	N12 Bytes	Set of Visual Objects.

- SpaceTime	N13 Bytes Space Time of a Visual Object in the Visual Object.
- VisualObject	N14 Bytes A Visual Object in the Visual Object
Annotations[]	N15 Bytes Set of Visual Object Annotation.
– Annotation	N16 Bytes An Annotation.
– AnnotationSpaceTime	N17 Bytes Where Annotation is attached and when it will be active.
– Rights	N18 Bytes Actions that may be performed on the Annotation
Rights	N19 Bytes Actions that may be performed on the Object.
DescrMetadata	N20 Bytes Descriptive Metadata

8.94.5 Conformance Testing

A Data instance Conforms with Visual Object (OSD-VIO) V1.3 if:

1. The Data validates against the Visual Object’s JSON Schema.
2. All Data in the Visual Object’s JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.95 Visual Scene Descriptors

8.95.1 Definition

A Data Type including the Visual Objects of a scene, their sub-scenes, and their arrangement in the scene.

8.95.2 Functional Requirements

Visual Scene Descriptors include

1. Visual Objects
2. The Descriptors of the Visual Scenes includes in the Visual Scene called Visual Sub-Scenes.
3. Rights that may be exercised on the Visual Scene.

Scenes may be hierarchical, i.e., they may contain Objects and Scenes.

8.95.3 Syntax

<https://schemas.mpai.community/OSD/V1.3/data/VisualSceneDescriptors.json>

8.95.4 Semantics

	Label	Size	Description
Header		N1 Bytes	Visual Scene Descriptors Header
- Standard-VisualSceneDescriptors		9 Bytes	The characters “OSD-VSD-V”
- Version		N2 Bytes	Major version – 1 or 2 characters
- Dot-separator		1 Byte	The character “.”

- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
VisualSceneDescriptorsID	N5 Bytes	Identifier of Visual Scene Descriptors.
ObjectCount	N6 Bytes	Number of Visual Objects in Scene.
SubSceneCount	N7 Bytes	Number of Visual Scenes in Scene.
VisualSceneDescriptorsSpaceTime	N8 Bytes	Space and Time of Visual Scene Descriptors.
VisualSceneObjects[]	N9 Bytes	Set of Visual Objects.
- VisualSceneObject	N10 Bytes	Visual Object.
- VisualSceneObjectSpaceTime	N11 Bytes	Space Time of Visual Object.
VisualSceneSubScenes[]	N12 Bytes	Set of Visual Sub-Scenes.
- VisualSceneSubScene	N13 Bytes	Visual Sub-Scene.
- VisualSceneSubSceneSpaceTime	N14 Bytes	Space Time of Visual Sub-Scene.
DescrMetadata	N15 Bytes	Descriptive Metadata

8.95.5 Conformance Testing

A Data instance Conforms with Visual Scene Descriptors (OSD-VSD) V1.3 if:

1. The Data validates against the Visual Scene Descriptors' JSON Schema.
2. All Data in the Visual Scene Descriptors' JSON Schema
 1. Have the specified type
 2. Validate against their JSON Schemas
 3. Conform with their Data Qualifiers.

8.96 Avatar

8.96.1 Definition

A Data Type that includes:

1. ID of the Virtual Space in which the Avatar id located,
2. ID of the Avatar
3. Space-Time information of the Avatar
4. 3D Model of the Avatar
5. Body Descriptors of the Avatar
6. Face Descriptors of the Avatar.

8.96.2 Functional Requirements

Avatar conveys the following information:

1. The ID of the Virtual Space.
2. The ID of the Avatar.

3. The Time and Spatial Attitude of the Avatar is in the M-Instance.
4. The set of Data characterising an Avatar:
 1. 3D Model.
 2. Face Descriptors.
 3. Body Descriptors.

An Avatar Model of a human 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.

8.96.3 Syntax

<https://schemas.mpai.community/PAF/V1.4/data/Avatar.json>

8.96.4 Semantics

Label	Size	Description
Header	N1 Bytes	Avatar Header.
– Standard-Avatar	9 Bytes	The characters “PAF-AVT-V”
– Version	N2 Bytes	Major version
– Dot-separator	1 Byte	The character "."
– Subversion	N3 Byte	Minor version
MInstanceID	N4 Bytes	ID of Virtual Space the Avatar belongs to.
AvatarSpaceTime	N5 Bytes	The inherent Space-Time info of the Avatar.
AvatarID	N6 Bytes	Identifier of Avatar.
AvatarData	N7 Bytes	Set of Data related to Avatar
- AvatarModel	N8 Bytes	Model of Avatar.
- BodyDescriptors	N9 Bytes	Avatar Body Descriptors.
- FaceDescriptors	N10 Bytes	Avatar Face Descriptors of Avatar.
DescrMetadata	N11 Bytes	Descriptive Metadata

8.96.5 Conformance Testing

A Data instance Conforms with Avatar (PAF-AVT) V1.4 if:

1. JSON Data validate against the Avatar’s JSON Schema.
2. All Data in the Avatar’s JSON Schema
 1. Have the specified type.
 2. Validate against their JSON Schemas.

8.97 Body Descriptors

8.97.1 Definition

Body Descriptors is a Data Type digitally representing a human or a humanoid.

Gesture Descriptors is a Data Type representing the subset of Body Descriptors selected by an application to convey Gesture information.

8.97.2 Functional Requirements

Body Descriptors should enable the representation of the joints of a body.

8.97.3 Syntax

Syntax is given by [Reference](#). The Body Descriptors XML Syntax is given by: <https://www.web3d.org/x3d/content/examples/X3dResources.html>

8.97.4 Semantics

The semantics of Body Descriptors is provided by <https://www.web3d.org/content/hanim-architecture-v2>.

8.97.5 Conformance Testing

A Data instance Conforms with Body Descriptors (PAF-BDD) V1.4 if the Data instance validates against the Body Descriptors XML Schema.

8.98 Face Descriptors

8.98.1 Definition

Face Descriptors is a Data Type representing the features of the Face of an Entity.

8.98.2 Functional Requirements

The Face Descriptors represent the effect of the motion of the muscles of a human face.

The Face Descriptors Syntax represents the Actions Units of the Facial Action Coding System (FACS) originally developed by Carl-Herman Hjortsjö, adopted by Paul Ekman and Wallace V. Friesen (1978) and updated by [Ekman, Friesen, and Joseph C. Hager](#) (2002).

8.98.3 Syntax

<https://schemas.mpai.community/PAF/V1.4/data/FaceDescriptors.json>

8.98.4 Semantics

Header	N1 Bytes	Orientation FaceDescriptors
- Standard-FaceDescriptors	9 Bytes	The characters “OSD-FCD-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
FaceDescriptorsID	N4 Bytes	Identifier of Face Descriptors.

AU	Description	N5 Bytes	Facial muscle generating the Action
1	Inner Brow Raiser	1 Byte	Frontalis, pars medialis
2	Outer Brow Raiser	1 Byte	Frontalis, pars lateralis
4	Brow Lowerer	1 Byte	Corrugator supercilii, Depressor supercilii
5	Upper Lid Raiser	1 Byte	Levator palpebrae superioris
6	Cheek Raiser	1 Byte	Orbicularis oculi, pars orbitalis
7	Lid Tightener	1 Byte	Orbicularis oculi, pars palpebralis
9	Nose Wrinkler	1 Byte	Levator labii superioris alaeque nasi
10	Upper Lip Raiser	1 Byte	Levator labii superioris
11	Nasolabial Deepener	1 Byte	Zygomaticus minor
12	Lip Corner Puller	1 Byte	Zygomaticus major
13	Cheek Puffer	1 Byte	Levator anguli oris (a.k.a. Caninus)
14	Dimpler	1 Byte	Buccinator
15	Lip Corner Depressor	1 Byte	Depressor anguli oris (a.k.a. Triangularis)
16	Lower Lip Depressor	1 Byte	Depressor labii inferioris
17	Chin Raiser	1 Byte	Mentalis
18	Lip Puckerer	1 Byte	Incisivii labii superioris and Incisivii labii inferioris
20	Lip stretcher	1 Byte	Risorius with platysma
22	Lip Funneler	1 Byte	Orbicularis oris
23	Lip Tightener	1 Byte	Orbicularis oris
24	Lip Pressor	1 Byte	Orbicularis oris
25	Lips part	1 Byte	Depressor labii inferioris or relaxation of Mentalis, or Orbicularis oris
26	Jaw Drop	1 Byte	Masseter, relaxed Temporalis and internal Pterygoid
27	Mouth Stretch	1 Byte	Pterygoids, Digastric
28	Lip Suck	1 Byte	Orbicularis oris
41	Lid droop	1 Byte	Relaxation of Levator palpebrae superioris
42	Slit	1 Byte	Orbicularis oculi
43	Eyes Closed	1 Byte	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis
44	Squint	1 Byte	Orbicularis oculi, pars palpebralis
45	Blink	1 Byte	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis
46	Wink	1 Byte	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis
61	Eyes turn left	1 Byte	Lateral rectus, medial rectus
62	Eyes turn right	1 Byte	Lateral rectus, medial rectus
63	Eyes up	1 Byte	Superior rectus, Inferior oblique
64	Eyes down	1 Byte	Inferior rectus, Superior oblique

8.98.5 Conformance Testing

A Data instance Conforms with Face Descriptors (PAF-FCD) V1.4 if:

1. The Data validates against the Face Descriptors' JSON Schema.
2. All Data in the Face Descriptors' JSON Schema
 1. Have the specified type.
 2. Validate against their JSON Schemas.

8.98.6 Mapping of AUs to Personal Status (Informative)

MPAI has defined a set of Cognitive States, Emotions, and Social Attitudes included in [Personal Status](#). The Table below offers an informative mapping of some elements of Personal Status to Action Units (from [1](#)).

Personal Status	Cognitive State	Emotion	Prototypical (and variant AUs)
Happy		17	12, 25 [6 (51%)]
Sad		32	4, 15 [1 (60%), 6 (50%), 11 (26%), 17 (67%)]
Fearful		13	1, 4, 20, 25 [2 (57%), 5 (63%), 26 (33%)]
Angry		2	4, 7, 24 [10 (26%), 17 (52%), 23 (29%)]
Surprised	18		1, 2, 25, 26 [5 (66%)]
Disgusted		11	9, 10, 17 [4 (31%), 24 (26%)]

This Table was obtained through a series of experiments with human subjects. AUs used by a subset of the subjects are shown in brackets with the percentage of the subjects using this less common AU in parentheses.

[1] [Compound facial expressions of emotion | PNAS](#)

8.99 Gesture Descriptors

8.99.1 Definition

Body Descriptors is a Data Type digitally representing a human or a humanoid.

Gesture Descriptors is a Data Type representing the subset of Body Descriptors selected by an application to convey Gesture information.

8.99.2 Functional Requirements

Body Descriptors should enable the representation of the joints of a body.

8.99.3 Syntax

Syntax is given by [Reference](#). The Body Descriptors XML Syntax is given by: <https://www.web3d.org/x3d/content/examples/X3dResources.html>

8.99.4 Semantics

The semantics of Body Descriptors is provided by <https://www.web3d.org/content/hanim-architecture-v2>.

8.99.5 Conformance Testing

A Data instance Conforms with Body Descriptors (PAF-BDD) V1.4 if the Data instance validates against the Body Descriptors XML Schema.

9 Protocols

The activities of an M-Instance involve Processes performing Actions on Items. For instance:

1. "*Device UM-Captures Media At U-Location*" means that a Device captures Media at a U-Location.
2. "*Device MM-Sends Data With Qualifier to Process*" means that a Device sends Data, Qualifier, and Rights resulting from the previous Action to a Process in the M-Instance.
3. "*User MM-Embeds Persona At M-Location and User MM-Animates Persona With Stream*" means, respectively, that User places a Persona (an avatar) at an M-Location and User animates it with a data stream.

An activity in an M-Instance may involve Process₁ who requests Process₂ to perform an Action. For instance:

1. *MM-Send* may rely on a *Communication Service* to make available Data, Qualifier, and Items of a Process to another Process. Note, however, that communication in an M-Instance may be peer-to-peer)
2. *MM-Embed* relies on a *Location Service* instantiating an Item (e.g., Persona) at an M-Location.

Process₁ may coincide with Process₂ when a Destination Process is not required, i.e., when a Process possesses the necessary functionalities.

MMM-TEC V2.0 enables a Process_A in an M-Instance_A to request a Process_B in a different M-Instance_B to perform Actions on Items in M-Instance_B. The two Processes can communicate if a Resolution session is opened by Process_A calling its Resolution Service_A.

MMM-TEC does not provide support to the establishment of business agreements based on which communication between M-Instances become possible.

Inter-Process Protocol (IPP) is the protocol enabling two Processes to communicate by sending IPP messages that can be PA Request or PA Response (PA stands for Process Action). IPP Messages include the following elements, some of which are optional. Any Time information is assumed to be added by the communication infrastructure.

IPP Message Elements	Description
Message ID	ID of PA Request or PA Response.
Response ID	Absent/Present when the Message is a PA Request/PA Response.
Source Process ID	ID of Process issuing Message.
Process Action	Combination of Action, Items/Processes and Complement (see Process Action).

	Service that
Resolution	1. Receives a Message from a Process in its M-Instance and forwarding it to a peer Process in a different M-Instance, or
Service ID	2. Sends a Message received from a peer Process in a different M-Instance to a Process in its M-Instance. (Absent if the two Processes are in the same M-Instance).
Destination	
Process ID	ID of Process to which the Message is intended to be sent.
Acknowledgement	Acknowledgement, if successful, or Error, if failure.

Figure 4 depicts the IPP steps when the two interacting Processes are in the same M-Instance.

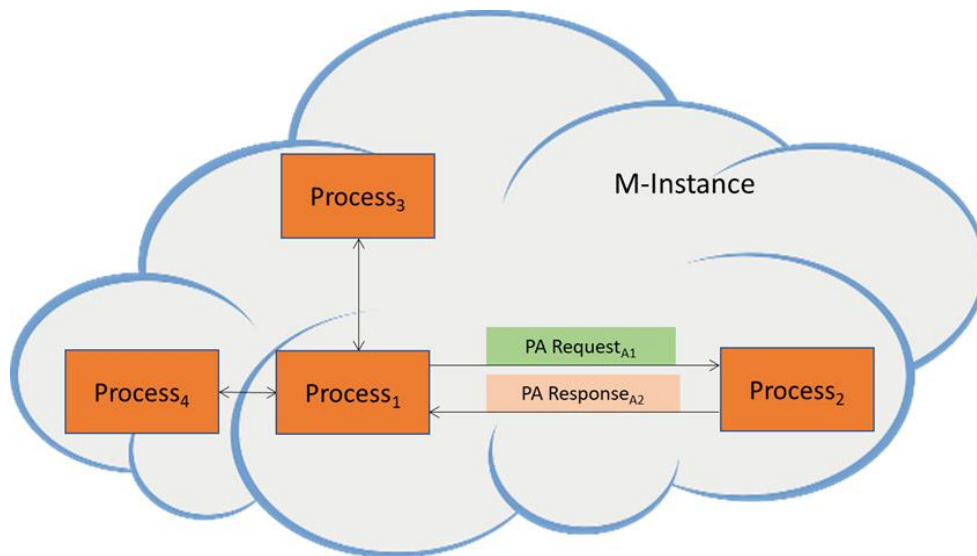


Figure 4 - The Inter-Process Protocol when the two Processes are in the same M-Instance.

The process unfolds through the following steps:

1. Process₁ sends a PA Request to Process₂.
2. Process₂ analyses the PA Request.
3. If the analysis yields an error, Process₂ sends a PA Response with an error to Process₁.
4. Else, Process₂ performs the PA Request.
5. If the execution of the PA Request fails, Process₂ sends a PA Response with an error to Process₁.
6. Else, Process₂ sends a PA Response to Process₁.

Figure 5 depicts the IPP steps when the two Processes are in different M-Instances. Here, the Process suffixes are prefixed by the letter A or B indicating the M-Instances they belong to.

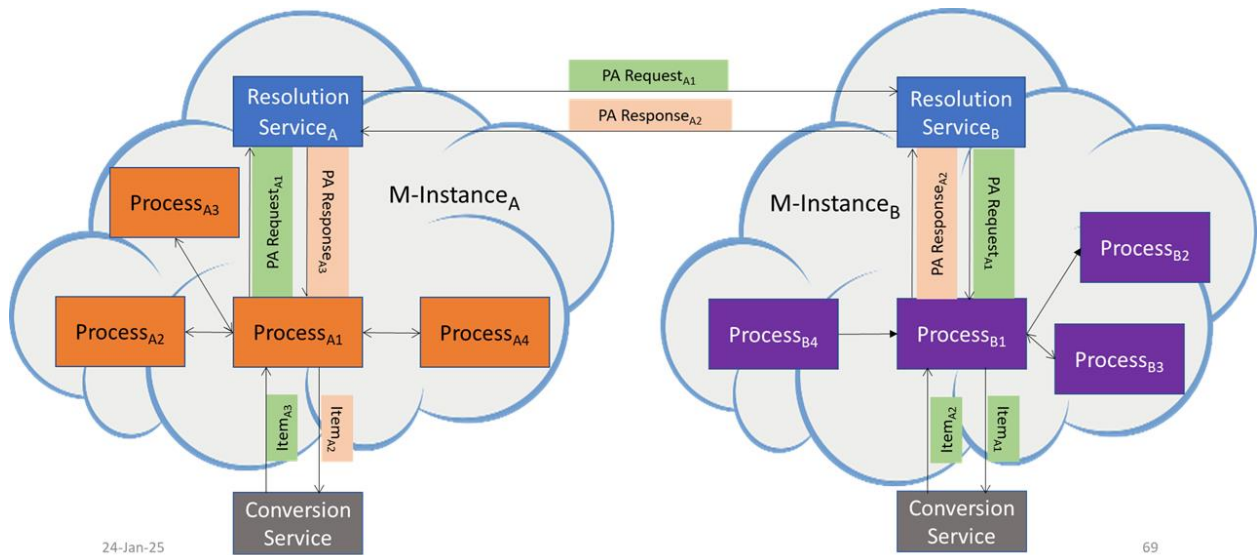


Figure 5 - The Inter-Process Protocol when the two Processes are in different M-Instances.

The process unfolds through the following steps:

1. Process_{A1} sends a PA Request to Resolution Service_A.
2. Resolution Service_A determines the Resolution Service_B it should forward the PA Request to.
3. If the determination is not reached it send an error to Process_{A1}.
4. Else it forwards the request to Resolution Service_{B1}.
5. Resolution Service_{B1} determines the Process it should send the PA Request to.
6. If the determination is not reached, it send an error to Resolution Service_{A1}.
7. Else it forwards the PA Request to Process_{B1}
8. Process_{B1} analyses the PA Request.
9. If the analysis of the request fails, it sends an error to Resolution Service_B.
10. Else, Process_{B1}
 1. May request a Conversion Service_B to make appropriate conversion of the Formats of the Data in the Items it received.
 2. Performs the request.
11. If the execution of the PA Request fails, it sends an error to Resolution Service_B.
12. Else it sends a PA Response to Resolution Service_{B1}.
13. Resolution Service_B sends error or PA Response to Resolution Service_A.
14. Resolution Service_A sends error or PA Response to Process_{A1}.
15. Process_{A1} may request a Conversion Service_A to make appropriate conversion of the Formats of the Data in the Items it received.

The Error Messages generated by either Resolution Service may be “No Rights to request Process Action to this M-Instance” or "No such ProcessID"

The Error Message generated by a receiving Process may be one of three types:

1. “Unable to perform request”.
2. “Transaction of *Value* required” (actual value provided).
3. "Conversion Service failure".
4. Process Action-specific error.

An Inter-Process Protocol session is set up by the IPP Session Setup whereby:

1. Process_{A1} requests Resolution Service_A to open a session with the M-Instances that host at least one invited Process.
2. Resolution Service_A forwards the request to all relevant Resolution Services.

3. A Resolution Service_B forwards the request to each invited Process_B's of its M-Instance.
4. An invited Process_{B1} responds to Resolution Service_B accepting or rejecting the invitation.
5. A Resolution Service_B forwards the response to Resolution Service_A.
6. Resolution Service_A forwards the responses to Process_{A1}.
7. If at least one invited Process accepts the invitation, the session starts.

The process unfolds through the following steps:

1. Process_{A1} sends a PA Request to Resolution Service_{A1}.
2. Resolution Service_{A1} determines the Resolution Service_{B1} it should forward the PA Request to.
3. If the determination is not reached it send an error to Process_{A1}.
4. Else it forwards the request to Resolution Service_{B1}.
5. Resolution Service_{B1} determines the Process it should send the PA Request to.
6. If the determination is not reached, it sends an error to Resolution Service_{A1}.
7. Else it forwards the PA Request to Process_{B1}
8. Process_{B1} analyses the PA Request.
9. If the analysis of the request fails, it sends an error to Resolution Service_B.
10. Else, Process_{B1}
 1. May request a Conversion Service_{B1} to make appropriate conversion of the Formats of the Data in the Items it received.
 2. Performs the request.
11. If the execution of the PA Request fails, it sends an error to Resolution Service_{B1}.
12. Else it sends a PA Response to Resolution Service_{B1}.
13. Resolution Service_{B1} sends error or PA Response to Resolution Service_{A1}.
14. Resolution Service_{A1} sends error or PA Response to Process_{A1}.
15. Process_{A1} may request a Conversion Service_{A1} to make appropriate conversion of the Formats of the Data in the Items it received.

The Error Messages generated by either Resolution Service may be “No Rights to request Process Action to this M-Instance” or "No such ProcessID".

The Error Message generated by a receiving Process may be one of three types:

1. “Unable to perform request”.
2. “Transaction of *Value* required” (actual value provided).
3. "Conversion Service failure".
4. Process Action-specific error.

An Inter-Process Protocol session is set up by the IPP Session Setup whereby:

1. Process_{A1} requests Resolution Service_{A1} to open a session with the M-Instances that host at least one invited Process.
2. Resolution Service_{A1} forwards the request to all relevant Resolution Services.
3. A Resolution Service_{B1} forwards the request to each invited Process_B's of its M-Instance.
4. An invited Process_{B1} responds to Resolution Service_{B1} accepting or rejecting the invitation.
5. A Resolution Service_{B1} forwards the response to Resolution Service_{A1}.
6. Resolution Service_{A1} forwards the responses to Process_{A1}.
7. If at least one invited Process accepts the invitation, the session is opened.

10 Process Actions

10.1 Components

MMM-TEC V2.0 specifies the following components of a Process Action (PA):

PA Element	Description
<i>Time</i>	Time of PA request emission and Time of PA request execution.
<i>Action</i>	One of the standard Actions specified by MMM-TEC V2.0.
<i>S-Complements</i>	(Source Complement) One or more Items or a Process participating in the execution of the Process Action at the S-Process side. Each Item/Process is preceded by at least one of <i>Nil, At, From, Of, To, or With</i> .
<i>D-Complements</i>	(Destination Complement) One or more Items or a Process participating in the execution of the Action at the D-Process. Each Item/Process is preceded by at least one of <i>Nil, At, From, Of, To, or With</i> .
<i>Error message</i>	Reasons why the requested Process does not perform PA request.

MMM-TEC V2.0 specifies the following Process Actions:

Authenticate	Author	Change	Convert	Discover
Execute	Hide	Identify	Inform	Interpret
MM-Add	MM-Anim	MM-Disable	MM-Embed	MM-Enable
MM-Move	MM-PropertyChange	MM-Send	Modify	MU-Actuate
MU-Send	Post	Register	Resolve	Track
Transact	UM-Capture	UM-Send	Validate	

General notes:

1. Process Actions are specified with the following conventions:
 1. "S-" indicates "Source" Complements.
 2. "D-" indicates "Destination" Complements.
2. Items, e.g., Rights, Transactions, etc. may be transmitted as IDs or as actual Items.
3. A Process Action Request (PA Request) may include:
 1. *With* Transaction S-Complement whereby S-Process pays for the performance of a PA Request. Therefore, *With* Transaction is not added as a Complement in the following specification.
 2. *With* Transaction Model to signal that a real Transaction that uses Transaction as a model must performed to execute the PA Request.
4. A PA Request may include a *With* Rights Model to signal that the real Rights to the Item requested should be included in the PA Response.
5. The "Insufficient Value" Error Message is given for Process Action whose performance is more likely to require a Transaction. However, other PA Requests may also be affected by this error.
6. P-Capabilities includes the Process Actions a Process can perform.
7. The specification below may make distinctions between Process types to facilitate understanding of the role typically played by a Process. However, any other type of Process may also be used.

10.2 Authenticate

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	Authenticate	The Action of a User requesting an Authentication Service to: 1. Confirm that an Item or Process is what it claims to be with an Authentication Item request. 2. Grant Rights to Authentication Item response based on Model Rights.
<i>S-Complements</i>	<i>At M-Location</i>	Where the Item to be Authenticated is located.
	<i>With Authentication</i>	The Authentication Item requesting authentication.
	<i>With ModelRights</i>	If PA Request is successfully performed, User and Authentication Item shall be granted Rights based on Model Rights.
<i>D-Complements</i>	Authentication	Authentication Item response.
	<i>With Rights</i>	Granted to Authentication Item response.
	<i>At User</i>	The requesting User receives Authentication Item response with Rights.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient
	M-Location	Out of range
	U-Location	Out of range

10.3 Author

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	Author	The Action of a User requesting an Author Service to: 1. Produce an Item based on provided Items, Data, Qualifiers. 2. Grant Rights to the Authored Item based on Model Rights.
<i>S-Complements</i>	<i>With Items</i>	Information used by Author Service may be Items and/or
	<i>With Data</i>	Data and
	<i>With Qualifier</i>	Data Qualifier.
	<i>With ModelRights</i>	If PA Request is successfully performed, Item shall be granted Rights based on Model Rights.
<i>D-Complements</i>	Item	Item produced by Author Service including Rights.
	<i>With Rights</i>	Granted to Item.
	<i>At Process</i>	A Process hosting the Authored Item.
	<i>At MLocation</i>	The M-Location where the Item is MM-Added.
<i>Error Message</i>	FaultyReq	The Request is Faulty

IDs	Incorrect
ModelRights	Insufficient
Wallet error	Insufficient Value

10.4 Change

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	Change	The Action of a User requesting a Rights Service to: 1. Modify the Rights of a Process or an Item based on Model Rights. 2. Grant the requesting User the Rights to further Change the Rights.
<i>S-Complements</i>	Process	Whose Rights are to be changed, or
	Item	Whose Rights are to be changed
	<i>With ModelRights</i>	If PA Request is successfully performed, Rights shall be granted based on Model Rights.
<i>D-Complements</i>	Rights	Granted to requesting User to further Modify Rights.
	<i>At User</i>	The User with Changed Rights receives the new Rights.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient

10.5 Convert

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	Convert	The Action of a Process requesting a Conversion Service to: 1. Change the Data of an Item according to a given Qualifier 2. Grant Rights to the Converted Item based on Model Rights.
<i>S-Complements</i>	Item	The Item to be Converted.
	<i>At Process</i>	The Item can be stored at Process, or
	<i>At M-Location</i>	Placed at M-Location.
	<i>With ModelRights</i>	If PA Request is successfully performed, Item shall be granted Rights based on Model Rights.
<i>D-Complements</i>	Item	Item produced by Conversion Service.
	<i>With Rights</i>	Granted to Item.
	<i>At Process</i>	The Item can be stored at Process, or
	<i>At M-Location</i>	MM-Embedded at M-Location.
<i>Error Message</i>	FaultyReq	The Request is Faulty

IDs	Incorrect
ModelRights	Insufficient

10.6 Discover

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	Discover	The Action of a User requesting a Discovery Service to 1. Provide Item IDs or Process IDs relevant to the Discovery Item request. 2. Grant Rights to the Discovery Item response based on Model Rights.
<i>S-Complements</i>	<i>With</i> Discovery	Discovery Item request.
	<i>With</i> ModelRights	If PA Request is successfully performed, Discovery Item shall hold Rights based on Model Rights.
<i>D-Complements</i>	Discovery	Discovery Item response including Rights.
	Rights	Granted to Discovery response Item.
	<i>At</i> User	The requesting User receives the Discovery response Item.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient

10.7 Execute

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	Execute	The Action of a Process requesting an Execution Service to: 1. Execute a Program. 2. Grant Rights to the produced Items based on Model Rights.
<i>S-Complements</i>	Program	The Program to be Executed.
	<i>At</i> Process	Program is located at Process.
	<i>With</i> ModelRights	If PA Request is successfully performed, Item produced shall be granted Rights based on Model Rights.
<i>D-Complements</i>	Items	Items produced by Program including Rights.
	Rights	Granted to Items produced by Executed Program.
	<i>At</i> Process	A Process receives the execution result.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient

10.8 Hide

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
-------------	------	---

<i>Action</i>	Hide	The Action of a Process requesting an Identification Service to: 1. Make the ID of an Item unavailable to all Processes, but the requesting Process. 2. Grant Rights to the Hidden Item based on Model Rights.
<i>S-Complements</i>	Item	Item to be Hidden.
	<i>At</i> Process	Item is at a Process or
	<i>At</i> M-Location	At an M-Location.
	<i>With</i> ModelRights	If PA Request is successfully performed, Hidden Item shall be granted Rights based on Model Rights.
<i>D-Complements</i>	Rights	Granted to the Process Hiding Items.
	<i>At</i> Process	Process receives Rights.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient

10.9 Identify

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	Identify	The Action of a Process requesting an Identification Service to: 1. Produce an Item from Data, Qualifier, and Model Rights provided by the requesting Process. 2. Grant Rights to the Item based on Model Rights.
<i>S-Complements</i>	Data	Data provided to produce and Identify Item.
	<i>With</i> Qualifier	Qualifier of Data provided.
	<i>From</i> Process	Process where Data, Qualifier, and Model Rights are stored.
	<i>With</i> ModelRights	If PA Request is successfully performed, Identifier shall be granted Rights based on Model Rights.
	<i>At</i> Process	Data and Qualifier are available at the requesting Process.
<i>D-Complements</i>	Item	Created and Identified Item including Rights.
	<i>At</i> Process	Process will receive Item ID.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient

10.10 Inform

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
-------------	------	---

		The Action of a User requesting an Information Service to:
<i>Action</i>	Inform	1. Provide information about an Item or Process as contained in the Information Item request. 2. Grant Rights to Information Item response based on Model Rights.
<i>S-Complements</i>	<i>At M-Location</i>	The location of the Item whose information is requested.
	<i>With Information</i>	Item containing the request.
	<i>With ModelRights</i>	If PA Request is successfully performed, Information Item shall be granted Rights based on Model Rights.
<i>D-Complements</i>	Information	The Information Item response including Rights.
	<i>At User</i>	Requesting User receives the Information Item response.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient

10.11 Interpret

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
		The Action of a User requesting an Interpretation Service to:
<i>Action</i>	Interpret	1. Provide information about an Item or Process as contained in the Interpretation Item request. 2. Grant Rights to Interpretation Item response based on Model Rights.
<i>S-Complements</i>	<i>At M-Location</i>	The location of the Item whose interpretation is requested.
	<i>With Interpretation</i>	Interpretation Item containing the interpretation request.
	<i>With ModelRights</i>	If PA Request is successfully performed, Interpretation Item shall be granted Rights based on Model Rights.
<i>D-Complements</i>	Interpretation	The Interpretation Item response including Rights.
	<i>At User</i>	Requesting User receives the Interpretation Item response.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient

10.12 MM-Add

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
		The Action of a User requesting a Location Service to:
<i>Action</i>	MM-Add	1. Place an Item at an M-Location with a Spatial Attitude. 2. Grant Rights based on Model Rights to the MM-Added Item.

Only the User who has MM-Added the Item shall be able to perceive it. Other Users shall be able to perceive it only if the Item is MM-Enabled

<i>S-Complements</i>	Item	Item to be MM-Added.
	<i>From Process</i>	Holding Item.
	<i>With Spatial Attitude</i>	The Spatial Attitude that the MM-Added Item shall have at M-Location.
	<i>With ModelRights</i>	If PA Request is successfully performed, MM-Added Item shall be granted Rights based on Model Rights.
<i>D-Complements</i>	<i>With Rights</i>	Rights of MM-Added Item.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient
	Clash	Item clashes with another Item
	M-Location	Out of range

10.13 MM-Anim

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	MM-Anim	The Action of a User requesting that a Location Service:
		<ol style="list-style-type: none"> 1. Animate an MM-Added or MM-Embedded Item with a stream Item. 2. Grant Rights based on Model Rights to the MM-Animated Item.
<i>S-Complements</i>	Item	Item to be MM-Animated.
	<i>At M-Location</i>	Where the Item is MM-Added or MM-Embedded.
	<i>With Stream Item</i>	Stream MM-Animating Item.
	<i>With ModelRights</i>	If PA Request is successfully performed, MM-Animated Item shall be granted Rights based on Model Rights.
<i>D-Complement</i>	<i>With Rights</i>	Rights of MM-Animated Item.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient
	Clash	Item clashes with another Item
	M-Location	Out of range

10.14 MM-Disable

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	MM-Disable	The Action of a User requesting a Location Service to: 1. Stop making perceptible an MM-Embedded Item to all Users but the requesting User. 2. Preserve any change than may have been effected on the MM-Disabled Item. 3. Grant Rights to the MM-Disabled Item based on Model Rights.
<i>S-Complements</i>	Item	Item to be MM-Disabled.
	<i>At M-Location</i>	M-Location is the one currently targeted (the same Item may be located at different M-Locations simultaneously).
	<i>With ModelRights</i>	If PA Request is successfully performed, MM-Disabled Item shall be granted Rights based on Model Rights.
<i>D-Complements</i>	<i>With Rights</i>	Rights of MM-Disabled Item.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient
	M-Location	Out of range

10.15 MM-Embed

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	MM-Embed	The Action of a User requesting a Location Service to: 1. Place an Item at an M-Location with a Spatial Attitude. 2. Make the Item perceptible. 3. Grant Rights to the MM-Embedded Item based on Model Rights.
<i>S-Complements</i>	Item	Item to be MM-Embedded.
	<i>From Process</i>	Holding Item.
	<i>With Spatial Attitude</i>	The Spatial Attitude the MM-Added Item shall have at M-Location.
	<i>With ModelRights</i>	If PA Request is successfully performed, MM-Embedded Item shall hold Rights based on Model Rights.
<i>D-Complements</i>	<i>With Rights</i>	Rights to MM-Embedded Item.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient
	Clash	Item clashes with another Item
	M-Location	Out of range

10.16 MM-Enable

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	MM-Enable	The Action of a User requesting a Location Service to: 1. Add or Change the Spatial Attitude of an MM-Added Item at an M-Location: 2. Resize the Item by R_x , R_y , R_z along the Item's axes. 3. Make the Item perceptible. 4. Grant Rights to the MM-Enabled Item based on Model Rights.
<i>S-Complements</i>	Item	Item to be MM-Added.
	<i>At M-Location</i>	M-Location where the Item has been MM-Added or MM-Disabled.
	<i>With Spatial Attitude</i>	The Spatial Attitude the MM-Added Item shall have at M-Location.
	<i>With Resize</i>	The vector with coefficients R_x , R_y , R_z .
	<i>With ModelRights</i>	If PA Request is successfully performed, Rights shall be granted to the Item based on Model Rights.
<i>D-Complements</i>	<i>With Rights</i>	Rights to MM-Embedded Item.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient
	Clash	Item clashes with another Item
	M-Location	Out of range

10.17 MM-Move

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	MM-Move	The Action of a User requesting a Location Service to: 1. Move an MM-Added/MM-Embedded Item at an M-Location to another MM-Location. 2. Preserve any change that may have been effected on the Item in previous MM-Add, MM-Embed, or MM-Enable Actions. 3. Preserve the (un)perceptibility status of the Item. 4. Grant Rights to the MM-Moved Item based on Model Rights.
<i>S-Complements</i>	Item	Item to be MM-Added.
	<i>From M-Location</i>	M-Location where the Item is currently placed.
	<i>To M-Location</i>	New M-Location where the Item is moved.
	<i>With Spatial Attitude</i>	The Spatial Attitude the MM-Added Item shall have at new M-Location.
	<i>With ModelRights</i>	If PA Request is successfully performed, Rights shall be Granted to the MM-Moved Item based on Model Rights.

<i>D-Complements</i>	<i>With Rights</i>	Rights to MM-Moved Item.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient
	Clash	Item clashes with another Item
	M-Location	Out of range

10.18 MM-PropertyChange

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution. The Action of a User requesting a Location Service to modify the characteristics of an Item:
<i>Action</i>	MM-ChangeProperty	<ol style="list-style-type: none"> 1. Resize the Item by R_x, R_y, R_z along the Item's axes. 2. Display a specific Personal Status (if a Persona). 3. Makes Item perceptible in not already in that state. 4. Grant Rights to the MM-Morphed Item based on Model Rights.
<i>S-Complements</i>	Item	Item to be PropertyChanged.
	<i>At</i> M-Location	Holding Item.
	<i>With</i> ModelRights	If PA Request is successfully performed, MM-Morphed Item shall hold Rights based on Model Rights.
<i>D-Complements</i>	<i>With Rights</i>	Rights to MM-Morphed Item.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient
	M-Location	Out of range

10.19 MM-Send

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution. The Action of a Process requesting a Communication Service:
<i>Action</i>	MM-Send	<ol style="list-style-type: none"> 1. To send a Message to other Processes. 2. To grant Rights to the receiving Process on the content of the Message.
<i>S-Complements</i>	Message	The Message sent to the other Process.
	<i>To</i> Process	The Process receiving the Message.
<i>D-Complements</i>	Rights	If PA Request is successfully performed, Rights based on ModelRights shall be granted to the content of the Message.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect

ModelRights Insufficient

10.20 Modify

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	Modify	The Action of a User requesting an Identification Service to 1. Produce a new Item starting from an existing Item using new Data, Qualifier, and Model Rights to the new Item. 2. Grant Rights to the Modified Item based on Model Rights.
<i>S-Complements</i>	Item	Item to be Modified.
	<i>At Process</i>	Item is at a Process.
	<i>With Data</i>	Data, Qualifier, and Model Rights provided to Modify Item.
	<i>With Qualifier</i>	New Qualifier of Item to be Modified.
	<i>With ModelRights</i>	If PA Request is successfully performed Rights based on Model Rights shall be added to Modified Item.
<i>D-Complements</i>	Item	The Modified Item.
	<i>With Rights</i>	Rights to Modified Item.
	<i>At Process</i>	Process receives the Modified Item
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient
	DataMdata	Incorrect

10.21 MU-Actuate

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	MU-Actuate	The Action of a User requesting an ExIm Service to: 1. Render an Item at a U-Location as Media with a Spatial Attitude 2. Also render the Scene of the M-Location including the Item, if the M-Location field is present. 3. Grant Rights to the U-Location based on Model Rights. MM-Added Items preserve their (un)perceptibility attributes.
<i>S-Complements</i>	Item	Item, e.g., an M-Location to be MU-Actuated.
	<i>At M-Location</i>	The M-Location where the Item is MM-Embedded.
	<i>At U-Location</i>	U-Location where the Item shall be actuated as Media.
	<i>With Spatial Attitude</i>	The Spatial Attitude of the Item MU-Actuated as Media.
	<i>With ModelRights</i>	If ExIm Service successfully performs PA Request, it shall grant Rights on the Media to the U-Location.
<i>D-Complements</i>	Media	MU-Actuated Item.

	<i>With Rights</i>	U-Location's Rights to MU-Actuated Item.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient
	U-Location	Out of range

10.22 MU-Send

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	MU-Send	The Action of a Process requesting an ExIm Service to: 1. Send a Message to Processes in the Universe. 2. Grant Rights to Data and Qualifier based on Model Rights.
<i>S-Complements</i>	Message	Message sent to Process.
	<i>With</i>	If ExIm Service successfully performs PA Request, it shall grant
	ModelRights	Rights based on Model Rights to the content of the Message.
	<i>To Process</i>	Process receiving Message..
<i>D-Complements</i>	<i>With Rights</i>	Receiving Process's Rights to Message.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient

10.23 Post

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	Post	The Action of a User requesting that a Marketplace Service: 1. Include an Asset to its repository. 2. Grant Rights based on Model Rights if a Transaction based on Model Transaction is performed.
<i>S-Complements</i>	Asset	Asset User wishes to Post.
	<i>With</i>	If PA Request is successfully performed, Marketplace Service
	ModelRights	shall be granted Rights based on Model Rights to the Posted Asset.
	<i>With</i>	A User wishing to acquire Rights on the Asset shall use a
	ModelTransaction	Transaction Item based on the Model Transaction.
	<i>To Service</i>	Destination Marketplace Service.
<i>D-Complements</i>	<i>With Rights</i>	Granted Rights to the Asset.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect

ModelRights	Insufficient
Wallet	Insufficient Value

10.24 Register

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	Register	The Action of a human requesting that a Registration Service: 1. Open an Account based on the human's Personal Data and potentially perform a Transaction. 2. Grant the Registering human's Processes Rights to perform Process Actions in the M-Instance.
<i>S-Complements</i>	<i>With</i> PersonalProfile	Human's Personal Profile.
<i>D-Complements</i>	Account	Registered human's Account.
	<i>With</i> Rights	Rights for human's Processes.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	Wallet	Insufficient Value

10.25 Resolve

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>	Resolve	The Action of: 1. A Process requesting that a Resolution Service set up a session between/among two/more than two Processes in two/more than two M-Instances. 2. A Process or Resolution Service responding to a session request.
<i>S-Complements</i>	<i>With</i> Resolution request.	Resolution Item used to request a session or to respond to a session request.
	<i>To</i> Process	May be a Resolution Service or a User.
<i>D-Complements</i>	Resolution	Resolution Response.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	Wallet	Insufficient Value

10.26 Track

Track includes the following sequence of PA requests:

- Location Service to MM-Embed a Persona at an M-Location with a Spatial Attitude.
- ExIm Service to MM-Capture Data and Qualifier at a U-Location.
- Identification Service to Identify Data and Qualifier as Stream.
- Location Service to MM-Anim the MM-Embedded Persona.
- ExIm Service to MU-Actuate the MM-Animated Persona at a U-Location.

This is the sequence of Messages:

S-Process Action S-Complements D-Processes D-Complements

User	MM-Add	Persona At MLoc With SA	LOSrvc	
LOSrvc			User	
User	UM-Capture	Media At ULoc With SA	EISrvc	At User
EISrvc		Data With Qualifier	User	
User	Identify	Data With Qualifier	IDSrvc	At User
IDSrvc		Stream	User	
User	MM-Anim	Persona With Stream With SA	LOSrvc	
LOSrvc			User	
Process	UM-Actuate	MLoc At ULoc With SA	EISrvc	
EISrvc			User	

The Actions in Track assume that:

1. The same Device is used to UM-Capture and MU-Actuate.
2. The same Location Service is used to MM-Add and MM-Animate.

The sequence can be represented by:

User Track Persona *From* ULoc *With* SA *At* MLoc *With* SA LOSrvc, Device, IDSrvc

PA Responses are not included because, in general, they are from different Services.

10.27 Transact

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
		The Action of a User ₁ (“sender”) requesting that a Transaction Service:
		1. Assign Rights on an Asset to User ₂ (“receiver”).
<i>Action</i>	Transact	2. Cause:
		2.1. Wallet ₁ of User ₁ 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).
<i>S-Complements</i>	<i>With</i> Transaction	Transaction Item used in transaction.
<i>D-Complements</i>		Not Applicable (relevant information in Transaction).
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient
	Wallet	Insufficient Value

10.28 UM-Capture

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>		The Action of a Process holding Rights to a U-Location to request an ExIm Service to:

		1. Capture Data and Qualifier with a Spatial Attitude from Media at U-Location. 2. Grant Rights to the Process to Identify the Data and Qualifier as Item.
<i>S-Complements</i>	Media	The captured Media to be stored in Device as Data, Qualifier and Model Rights.
	At U-Location	Location of Media to be Captured
	With ModelRights	If Process successfully performs PA Request, ExIm Service shall grant Rights to Device to use UM-Captured Data and Qualifier based on Model Rights.
<i>D-Complements</i>	Data	Data residing in Device after capture of Media.
	With Qualifier	Qualifier of Data.
	With Rights	Rights of Data and Qualifier.
	At Process	Process receiving Data, Qualifier, and Rights.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient
	U-Location	Out of range

10.29 UM-Send

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
<i>Action</i>		The Action of a Process requesting an ExIm Service to: 1. Be allowed to receive a Message from a Service in the Universe. 2. Grant Rights to Act on the content of the Message.
<i>S-Complements</i>	Message	The Message sent to the Process.
	With ModelRights	If ExIm Service successfully performs PA Request, it shall grant to Process Rights to use content on Message based on Model Rights.
<i>D-Complements</i>	To Process	Process in the Universe receiving Message.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient
	U-Location	Out of range

10.30 Validate

<i>Time</i>	Time	Time of PA request emission and Time of PA request execution.
-------------	------	---

<i>Action</i>	Validate	The Action of a Process requesting a Validation Service to confirm that a Process hold claimed Rights.
<i>S-Complements</i>	<i>With Validation</i>	Validation request.
	<i>To</i> ValidationProcess	Process performing Validation.
<i>D-Complements</i>	Validation	Validation Item response.
	<i>At User</i>	The requesting User receiving the Validation Item response.
<i>Error Message</i>	FaultyReq	The Request is Faulty
	IDs	Incorrect
	ModelRights	Insufficient
	M-Location	Out of range
	U-Location	Out of range

11 Profiles

11.1 Introduction

Profiles define groups of Items and Actions that may be used to implement M-Instances where Process Actions that serve specific application areas are performed. Profiles provide a programmed level of Interoperability with M-Instances that implement different Profiles.

Tables 13 and 14 list the currently identified Actions and Items supported by a Profile.

Table 13 - MMM-TEC V2.0 Actions

General Actions	Change	Execute	Hide	Identify	Modify
	Validate				
Call a Service	Authenticate	Author	Convert	Discover	Inform
	Interpret	Post	Register	Resolve	Transact
Metaverse – Metaverse	MM-Add	MM-Anim	MM-Disable	MM-Embed	MM-Enable
	MM-Move	MM-PropertyChange	MM-Send		
Metaverse – Universe	MU-Actuate	MU-Send	Track	UM-Capture	UM-Send

Table 14 - MMM-TEC V2.0 Items

General	Basic Certificate	Certificate	Contract	Identifier
	M-Capabilities	M-Environment	M-Instance	Program
	Rights	Rules	U-Environment	
Human&User	Account	Activity Data	Personal Profile	Personal Data
Process Interaction	IPP Message	Message	P-Capabilities	Process Action

	Resolution			
Service Access	Authentication	Basic Discovery	Basic Information	Basic Interpretation
	Discovery	Information	Interpretation	Validation
Finance	Asset Value	Currency Wallet	Provenance	Transaction
Perception	Basic Object	Basic Scene Descriptors	Basic Scene Geometry	Event Descriptors
	Object	Scene Descriptors	Scene Geometry	Summary
Internal State	Cognitive State	Emotion	Personal Status	Social Attitude
Space&Time	Basic M-Location	Basic U-Location	M-Location	Orientation
	Point of View	Position	Space-Time	Spatial Attitude
	Time	U-Location	Universe-Metaverse Map	

11.2 Profile structure

The current MMM features are:

1. Specified Profiles are Baseline, Management, Finance, and High.
2. The High Profile includes the Management Profile that includes the Baseline and Finance Profiles.
3. The Baseline, Management, and High Profiles have the following Levels Audio only, Audio-Visual, and Audio-Visual-Haptic.
4. The Finance Profile does not have Levels.

This is depicted in Figure 6. The next Sections identify the Action and Items of the four Profiles.

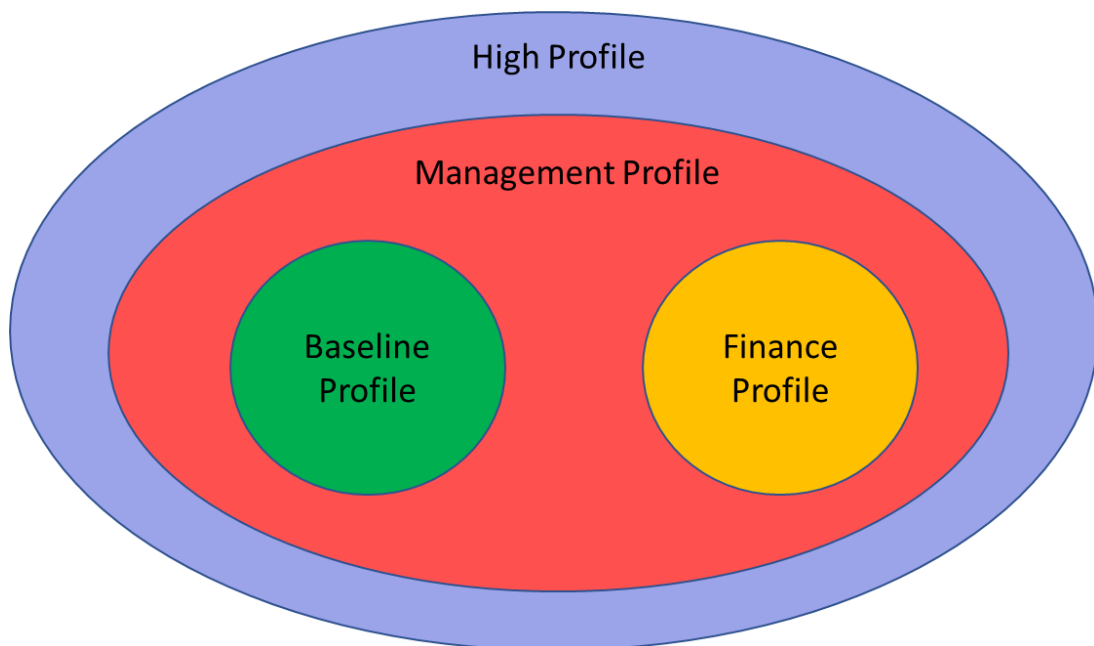


Figure 6 - MMM-TEC V2.0 Profiles

Each Profile allocates the Actions and Items that may be used in Process Actions executable in the Profile. While the identified four Profiles serve the identified needs, the consideration of more and differently grouped Functionalities may lead to an increased number of Profiles and potentially Levels in future MMM-TEC Versions.

11.3 Baseline Profile

The Baseline Profile is designed to enable Processes to perform Process Actions that include the identified Actions (Table 15) and Items (Table 17). This Profile has the following Levels: Audio only; Audio-Visual; and Audio-Visual-Haptic. The Baseline profiles is targeted to controlled environments that do not require security considerations.

Table 15 - Actions of MMM-TEC V2.0 Baseline Profile

General Actions	Identify
Metaverse to Metaverse	MM-Anim MM-Disable MM-Embed MM-Send
Metaverse - Universe	MU-Actuate Track UM-Capture

Table 16 - Items of MMM-TEC V2.0 Baseline Profile

General	Identifier	M-Instance
Process Interaction	IPP Message	Message Process Action
Perception	Basic Object	
Space&Time	Basic M-Location Basic U-Location Point of View Space-Time Time	

11.4 Finance Profile

The Finance Profile Processes to perform the functions of Table 5. The Finance Profile enables a User to Post Assets and make Transactions without avatars. As depicted in *Figure 6*, this Profile is independent of the Basic Profile, although it shares several basic Actions and Items with it. This Profile supports security and does not have Levels.

Tables 5 and 6 list the Actions and Items of the Finance Profile.

Table 17 - Actions of MMM-TEC V2.0 Finance Profile

General Actions	Change	Hide	Identify	Modify	Validate
Call a Service	Authenticate Author	Discover	Post	Register	
	Transact				
Metaverse to Metaverse	MM-Add	MM-Disable MM-Embed MM-Enable MM-Send			
Metaverse to Universe	MU-Actuate MU-Send	UM-Send			

Table 6 – Items of MMM-TEC V2.0 Finance Profile

General	Contract	Identifier	M-Instance	Rights
	Rules			
Human&User	Account	Activity Data	Personal Profile Personal Data	
Process Interaction	IPP Message	Message	Process Action Validation	
Service Access	Authentication	Basic Discovery	Discovery Validation	
Finance	Asset	Currency	Provenance Transaction	

	Value	Wallet		
Perception	Basic Object	Object		
Space&Time	Basic M-Location	Basic U-Location	Point of View	Space-Time
	Time	U-Location		

11.5 Management Profile

The Management Profile enables Processes to perform the Process Actions including the Actions and the Items of Tables 7 and 8 that include those of the Baseline and the Finance Profiles in addition to some others of its own as depicted in *Figure 6*. This Profile has the following Levels: Audio; Audio-Visual; and Audio-Visual-Haptic. This Profile supports environments where transactions are performed and security is important.

Table 18 - Actions of MMM-TEC V2.0 Management Profile

General Actions	Change	Execute	Hide	Identify	Modify
	Validate				
Call a Service	Authenticate	Author	Discover	Inform	Interpret
	Post	Register	Transact		
Metaverse to Metaverse	MM-Add	MM-Anim	MM-Disable	MM-Embed	MM-Enable
	MM-Move	MM-Send			
Metaverse - Universe	MU-Actuate	MU-Send	Track	UM-Capture	UM-Send

Table 19 - Items of MMM-TEC V2.0 Management Profile

General	Basic Certificate	Certificate	Contract	Identifier
	M-Capabilities	M-Environment	M-Instance	Program
	Rights	Rules	U-Environment	
Human&User	Account	Activity Data	Personal Profile	Personal Data
Process Interaction	IPP Message	Message	P-Capabilities	Process Action
	Validation			
Service Access	Authentication	Basic Discovery	Basic Information	Basic Interpretation
	Discovery	Information	Interpretation	Validation
Finance	Asset	Currency	Provenance	Transaction
	Value	Wallet		
Perception	Basic Object	Basic Scene Descriptors	Basic Scene Geometry	Event Descriptors
	Object	Scene Descriptors	Scene Geometry	Summary
Space&Time	Basic M-Location	Basic U-Location	M-Location	Orientation
	Point of View	Position	Space-Time	Spatial Attitude

[Time](#)

[U-Location](#)

[Universe-Metaverse
Map](#)

11.6 High Profile

This Profile includes all other Profiles. Tables 20 and 21 gives the list of Actions and Items not included in the Management Profile. This Profile has the following Levels: Audio; Audio-Visual; and Audio-Visual-Haptic.

Table 20 - Actions specific of MMM-TEC V2.0 High Profile

Call a Service [Convert](#) [MM-PropertyChange](#) [Resolve](#)

Table 21 - Items specific of MMM-TEC V2.0 High Profile

Internal State [Cognitive State](#) [Emotion](#) [Personal Status](#) [Social Attitude](#)
Space&Time [Universe-Metaverse Map](#)

12 Verification Use Cases

(Informative)

[1 Introduction](#)

[2 Use Case Description Language](#)

[3 Friends meet in the metaverse](#)

[3.1
Description](#)

[3.2 Workflow](#)

[3.3 Elementary
Messages](#)

[4 Virtual Lecture](#)

[4.1
Description](#)

[4.2 Workflow](#)

[4.3 Elementary
Messages](#)

[5 Virtual Meeting](#)

[5.1
Description](#)

[5.2 Workflow](#)

[5.3 Elementary
Messages](#)

[6 Hybrid working](#)

[6.1
Description](#)

[6.2 Workflow](#)

[6.3 Elementary
Messages](#)

[7 eSports Tournament](#)

[7.1
Description](#)

[7.2 Workflow](#)

[7.3 Elementary
Messages](#)

[8 Virtual Performance](#)

[8.1
Description](#)

[8.2 Workflow](#)

[8.3 Elementary
Messages](#)

[9 AR Tourist Guide](#)

[9.1
Description](#)

[9.2 Workflow](#)

[9.3 Elementary
Messages](#)

[10 Virtual Dance School](#)

[10.1
Description](#)

[10.2 Workflow](#)

[10.3 Elementary
Messages](#)

[11 Virtual Car Showroom](#)

[11.1
Description](#)

[11.2 Workflow](#)

[11.3 Elementary
Messages](#)

[12 Meeting in Connected Autonomous
Vehicles](#)

[12.1
Description](#)

[12.2 Workflow](#)

[12.3 Elementary
Messages](#)

[13 Co-design across metaverses](#)

[13.1
Description](#)

[13.2 Workflow](#)

[13.3 Elementary
Messages](#)

[14 Selling assets on a Marketplace](#)

[14.1
Description](#)

[14.2 Workflow](#)

[14.3 Elementary
Messages](#)

12.1 Introduction

The purpose of this Informative Chapter is to collect a variety of Use Cases for M-Instances in different contexts to verify that the MMM-TEC V2.0 technologies supports such 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*” is to be read as “*A User representing a student in an M-Instance MM-Embedded at an M-Location attends a virtual lecture in a virtual classroom Authored by a User representing a school manager*”.

12.2 Use Case Description Language

Metaverse Use Cases involve a plurality of Processes – Users, Devices, Services, Apps – requesting other Processes to perform Process Actions involving a variety of Items and Processes.

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 the Action followed by a sequential number of the Item.
3. The *Location* Item 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 have a sequential capital letter as first suffix to indicate in which M-Instance the Action is performed.

For instance:

1. User_i MM-Embeds Person_j at M-Location_k.
2. User_i MU-Actuates Item_j at U-Location_k.
3. User_{A,i} MM-Sends Message_j to User_{B,k}.

All Use Cases assume that Actions are performed within an M-Instance. Action specified in the Universe are specifically noted.

The following conventions are used throughout:

MLoc	M-Location
MData	Model Data
MRights	Model Rights
MTransaction	Model Transaction
SA	Spatial Attitude
ULoc	U-Location

The following Service abbreviations are used to decrease the density of the tables describing the workflows of the Use Cases:

Activity	ACSrvc
Author	AUSrvc
Communication	COSrvc
Execution	EXSrvc
Identification	IDSrvc
Interpretation	ITSrvc

Location	LOSrvc
Parcel	PASrvc
Registration	RGSrvc
Resolution	RESrvc
Rights	RISrvc
Transact	TRSrvc

Use Cases include the following:

1. Description of Use Case
2. Workflow in two parts:
 1. Declarations of all Process Actions and Items involved. (Model Items are not declared as Items are declared).
 2. Operation of Use Case using the Inter-Process Protocol.
3. Elementary Messages used by the Use Case.

The Operation of a Use Case is described by a table whose columns have headers corresponding to the elements of a Process Action. Each Process Action is split in two lines, the first describing the Source sending the request and the second describing the Destination sending a response back to the Source.

12.3 Friends meet in the metaverse

12.3.1 Description

1. Human₁ registers with Registration Service.
2. User₁
 1. Buys land parcel.
 2. Buys room.
 3. Places room on land parcel.
 4. Tracks at Metaverse Square.
 5. Moves to its room.
3. Human₂ registers with Registration Service.
4. User₂
 1. Registers with M-Instance.
 2. Tracks at Metaverse Square.
5. User₁
 1. Invites User₂ to its room.
 2. Receives acceptance message.
 3. Adds User₂'s Rights to enter its room to the room.
6. User₂ moves to User₁'s room.
7. (Nothing is said of what happens inside the room)
8. User₂ leaves User₁'s room.
9. User₁ Changes room Rights back to initial value.

12.3.2 Workflow

12.3.2.1 Declarations

Declare human₁ Friend#1.

Declare User ₁	Friend#1's User.
Declare RGSrv ₁	Registration Service.
Declare ACSrv ₁	Activity Service
Declare AUSrv ₁	Author Service
Declare COSrv ₁	Communication Service
Declare Device ₁	Friend#1's Device
Declare IDSrv ₁	Identify Service
LOSrv ₁	Location Service
PCSrv ₁	Parcel Service
Account _{1,1}	Friend#1's Account.
PersonalProfile _{1,1}	Friend#1's Personal Profile.
ULoc _{1,1}	Place in Universe where Friend#1 resides.
MLoc _{1,1}	Place in Metaverse Square.
MLoc _{1,2}	Place of Room.
MLoc _{1,3}	Place close to Parcel.
MLoc _{1,4}	Place in Room.
Message _{1,1}	User ₁ 's Message to ACSrv ₁ "Is Friend#2 on?".
Message _{1,2}	ACSrv ₁ 's Message to Friend#1 "Yes".
Message _{1,3}	Message to Friend#2 "Come to my room".
Message _{1,4}	Message to Friend#2 "Looking forward to seeing you here"
Transaction _{1,1}	Transaction to Parcel Service.
Transaction _{1,2}	Transaction to Author Service.
Rights _{1,1}	Rights to Parcel.
Rights _{1,2}	Rights to Room.
Room _{1,1}	Purchased Room.
Declare human ₂	Friend#2.
Declare User ₂	Friend#2's User.
ULoc _{2,1}	Place in Universe where Friend#2 resides.
MLoc _{2,1}	Place in Metaverse Square where Friend#2 stays.
Message _{2,1}	Friend#2's Response to Friend#1's invitation "OK"
Rights _{2,1}	Friend#2's Rights to enter Friend#1's Room
Rights _{2,2}	No Rights to Friend#2 to enter Friend#1's Room

12.3.2.2 Operation

S-Process	Action	S-Complements	D-Process	D-Compl.	Comments
human ₁	Register	<i>With</i> PersonalProfile _{1,1}	RGSrv ₁	<i>To</i> human ₁	human ₁ registers with M-Instance
RGSrv ₁		Account _{1,1}	human ₁		
User ₁	Transact	<i>With</i> Transaction _{1,1}	TRSRvc ₁	<i>To</i> User ₁	User ₁ buys Parcel
PCSrv ₁		Rights _{1,1}	User ₁		

User ₁	Transact	With Transaction _{1,2}	TRSrcv ₁	To User ₁	User ₁ buys Room
AUSrcv ₁		Rights _{1,2}	User ₁		
User ₁	Track	Persona _{1,1} From ULoc _{1,1} With SA At MLoc _{1,1} With SA	LOSrcv ₁ , Device ₁ , IDSrcv ₁		Persona _{1,1} represents human ₁ (invisible)
User ₁	MM-Move	Persona _{1,1} From MLoc _{1,1} To MLoc _{1,3} With SA	LOSrcv ₁		Persona _{1,1} moved close to Parcel
LOSrcv ₁			User ₁		
User ₁	MM-Add	Room _{1,1} At MLoc _{1,2} With SA	LOSrcv ₁		User ₁ places Room on Parcel
LOSrcv ₁			User ₁		
User ₁	MM-Embed	Persona _{1,1} At MLoc _{1,4} With SA	LOSrcv ₁		Persona _{1,1} moved inside the Room (visible)
LOSrcv ₁			User ₁		
User ₂	Track	Persona _{2,1} From Uloc _{2,1} With SA At MLoc _{2,1} With SA	LOSrcv ₁ , Device ₁ , IDSrcv ₁		Persona _{2,1} represents human ₂
User ₁	MM-Sends	Message _{1,1}	ACSrcv ₁		User ₁ checks User ₂ 's whereabouts
ACSrcv ₁		Message _{1,2}	User ₁		
User ₁	MM-Sends	Message _{1,3}	COSrcv ₁	To User ₂	User ₁ invites User ₂
COSrcv ₁			User ₁		
User ₂	MM-Sends	Message _{2,1}	COSrcv ₁	To User ₁	User ₂ accepts
COSrcv ₁			User ₂		
User ₁	MM-Sends	Message _{1,4}	COSrcv ₁	To User ₂	User ₁ waits for User ₂
COSrcv ₁			User ₁		
User ₁	Changes	User ₂ With MRights _{2,1}	RISrcv ₁	To User ₂	User ₁ gives User ₂ Rights to enter Room
RISrcv ₁		Rights _{2,1}	User ₁		
User ₂	MM-Embed	Persona _{2,1} At MLoc _{2,2} With SA	LOSrcv ₁		Persona ₂ placed in Room
LOSrcv ₁			User ₂		
User ₂	MM-Move	Persona _{2,1} From MLoc _{2,2} To MLoc _{2,1} With SA	LOSrcv ₁		Persona ₂ leaves Room
LOSrcv ₁			User ₂		
User ₁	Changes	User ₂ With MRights _{2,2}	RISrcv ₁	To User ₂	Persona ₂ 's Rights restored
RISrcv ₁		Rights _{2,1}	User ₁		

12.3.3 Elementary Messages

S-Process	Action	S-Complements	D-Process	D-Compl	Comments
human	Register	<i>With</i> PersonalProfile	RGSrvc	<i>To</i> human	human registers with M-Instance
RGSrvc		Account	human		
User	Transact	<i>With</i> Transaction	TRSRvc	<i>To</i> User	User buys Parcel
PCSrvc		Rights	User		
User	Track	Persona <i>From</i> ULoc <i>With</i> SA <i>At</i> MLoc <i>With</i> SA	LOSrvc, Device, IDSRvc		Persona represents human (invisible)
User	MM-Move	Persona <i>From</i> MLoc <i>To</i> MLoc ₃ <i>With</i> SA	LOSrvc		Persona moved close to Parcel
LOSrvc			User		
User	MM-Add	Room <i>At</i> MLoc <i>With</i> SA	LOSrvc		User places Room on Parcel
LOSrvc			User		
User	MM-Embed	Persona <i>At</i> MLoc ₄ <i>With</i> SA	LOSrvc		Persona moved inside the Room (visible)
LOSrvc			User		
User	MM-Sends	Message	ACSRvc		User checks User's whereabouts
ACSRvc		Message	User		
User	Changes	User <i>With</i> MRights	RISRvc	<i>To</i> User	User gives User Rights to enter Room
RISRvc		Rights	User		

12.4 Virtual Lecture

12.4.1 Description

1. School Manager
 1. Authors a virtual classroom
 2. Makes transaction for the virtual classroom
 3. Embeds the virtual classroom in an M-Location on which it has Rights.
 4. Changes Rights to Teacher on Classroom.
2. Teacher
 1. Is at home.
 2. Embeds their Persona at the classroom desk.
 3. Embeds and animates a 3D Object for the lecture.
3. Student
 1. Is at home.
 2. Pays lecture fees with the right to make a copy of the Audio-Visual Event.
4. School Manager
 1. Changes Rights to Students on Classroom after receiving lecture fees.
5. Student
 1. Embeds their persona in the classroom.

2. Approaches teacher's desk to handle 3D Object with haptic gloves.
3. Stores the lecture's Audio-Visual Event.
4. Returns Persona to home
6. School manager pays Teacher.
7. Teacher returns Persona to home.

12.4.2 Workflow

12.4.2.1 Declarations

Declare human ₁	School manager
Declare User ₁	School manager's User
Room _{1.1}	Classroom
MLoc _{1.1}	Metaverse Location in which User ₁ holds Rights.
MLoc _{1.2}	MLoc _{1.1} 's Sub M-Location where classroom is located.
Transaction _{1.1}	Payment for meeting room
Transaction _{1.2}	Lecture consideration
Rights _{1.1}	User ₁ 's Rights in Room _{1.1} after Transaction _{1.1}
Rights _{1.2}	Teacher's Rights in MLoc _{1.2}
Rights _{1.3}	Student's Rights in MLoc _{1.2}
Declare AUSvc ₁	Author Service
Declare IDService	Identification Service
Declare LOSvc ₁	Location Service
Declare RISvc ₁	Rights Service
Declare TRSvc ₁	Transaction Service
Declare human ₂	Teacher
Declare User ₂	Teacher's User
Persona _{2.1}	Teacher's Persona
MLoc _{2.1}	Teacher's home
MLoc _{2.2}	Place of classroom desk
MLoc _{2.3}	Place of object being experimented
Object _{2.1}	Object being experimented
ULoc _{2.1}	Location of Universe where the Teacher stays.
Declare human ₃	Student
Declare User ₃	Student's User
Persona _{3.1}	Student's Persona
MLoc _{3.1}	Student's home
MLoc _{3.2}	Classroom seat
MLoc _{3.3}	Location close to Object being experimented
Transaction _{3.1}	Lecture fees
Event _{3.1}	Lecture
Address _{3.1}	Address where Lecture Event Item is stored.

12.4.2.2 Operation

S-Process	Action	S-Complements	D-Process	D-Compl.	Comments
User ₁	Author	<i>With</i> (Data _{1,1} , Qualifier, MRights _{1,1})	AUSrvc ₁	<i>At</i> AUSrvc ₁	User ₁ Authors Room
AUSrvc ₁		Room _{1,1}	User ₁		
User ₁	Transact	<i>With</i> Transaction _{1,1}	TRSrvc ₁	<i>At</i> User ₁	User ₁ pays for Room
TRSrvc ₁		Rights _{1,1}	User ₁		
User ₁	MM-Add	Room _{1,1} <i>From</i> AUSrvc ₁ <i>At</i> MLoc _{1,2}	LOSrvc ₁		Places Room (invisible)
LOSrvc ₁			User ₁		
User ₁	MM-Enable	Room _{1,1}	LOSrvc ₁		Displays Room
LOSrvc ₁			User ₁		
User ₁	Change	User ₂ <i>With</i> MRights _{1,2}	RISrvc	<i>At</i> User ₂	User ₁ allows User ₂ to enter Room
RISrvc		Rights _{1,2}	User ₁		
User ₂	Track	Persona _{2,1} <i>From</i> Uloc _{2,1} <i>With</i> SA <i>At</i> MLoc _{2,1} <i>With</i> SA	LOSrvc, IDSrvc		Persona _{2,1} represents User ₂ in Room (invisible)
User ₂	MM-Embed	Persona _{2,1} <i>At</i> MLoc _{2,2} <i>With</i> SA	LOSrvc		Persona _{2,1} (visible) placed at desk
LOSrvc			User ₂		
User ₂	MM-Embed	Object _{2,1} <i>At</i> MLoc _{2,3}	LOSrvc ₁		User ₂ places experimental object.
LOSrvc ₁			User ₂		
User ₁	Change	User _{3,1} <i>With</i> MRights _{1,3}	RISrvc	<i>At</i> User ₃	User ₁ allows Student to enter Room
RISrvc		Rights _{1,3}	User ₁		
User ₃	Transact	<i>With</i> Transaction _{3,1}	User ₁	<i>To</i> User ₃	User ₃ pays for lecture and recording.
TRSrvc ₁		Rights _{3,1}	User ₃		
User ₃	Track	Persona _{3,1} <i>From</i> Uloc _{3,1} <i>With</i> SA <i>At</i> MLoc _{3,1} <i>With</i> SA	LOSrvc, IDSrvc		Persona _{3,1} represents User ₃ in Room (invisible)
User ₃	MM-PropertyChange	Persona _{3,1} <i>With</i> Personal Status	LOSrvc		Persona _{3,1} displays Personal Status
LOSrvc					
User ₃	MM-Move	Persona _{3,1} <i>To</i> MLoc _{3,2} <i>With</i> SA	LOSrvc		Persona _{3,1} moves to seat.
LOSrvc ₁			User ₃		
User ₂	MM-Embed	Speech _{2,1} <i>At</i> MLoc _{2,2}	LOSrvc ₁		User ₂ lectures.
LOSrvc ₁			User ₂		
User ₃	MM-Move	Persona _{3,1} <i>From</i> MLoc _{3,2} <i>To</i> MLoc _{3,3} <i>With</i> SA	LSrvc ₁		User ₃ gets closer to experimental object.

LOSrv _{c1}			User ₃		
User ₃	MU-Send	Event _{3,1} At URI _{3,1}	COSrv _{c1}		User ₃ stores lecture.
COSrv _{c1}			User ₃		
User ₃	MM-Move	Persona _{3,1} From MLoc _{3,3} To MLoc _{3,1} With SA	LSrv _{c1}		User ₃ returns Persona to home.
LOSrv _{c1}			User ₃		
User ₁	Transact	With Transaction _{1,2}	TRSrv _{c1}	At User ₂	User ₁ pays teacher.
TRSrv _{c1}		Rights _{1,2}	User ₁		
User ₂	MM-Move	Persona _{2,1} From MLoc _{2,3} To MLoc _{2,1} With SA	LSrv _{c1}		User ₂ returns Persona to home.
LOSrv _{c1}			User ₂	LOSrv _c	

12.4.3 Elementary Messages

S-Process	Action	S-Complements	D-Processes	D-Complements
User	Author	With (Data, Qualifier, MRights)	AUSrv _c	At AUSrv _c
AUSrv _c		Room	User	
User	Transact	With Transaction	TRSrv _c	At User
TRSrv _c		Rights	User	
User	MM-Add	Room From AUSrv _c At MLoc	LOSrv _c	
LOSrv _c			User	
User	MM-Enable	Room	LOSrv _c	
LOSrv _c			User	
User	Change	User With MRights	RISrv _c	At User
RISrv _c		Rights	User	
User	UM-Capture	Media At ULoc	Device	At Device
Device		Data With Qualifier	User	
User	Identify	Data With Qualifier	IDrv _c	At User
IDSrv _c		Stream With Rights	User	
User	MM-Animate	Persona With Stream	LOSrv _c	
LOSrv _c		Rights	User	
User	UM-Actuate	MLoc At ULoc With Spatial Attitude	Device ₃	
Device			User	
User	MM-Move	Persona From MLoc At MLoc With SA	LOSrv _c	
LOSrv _c			User	
User	MM-Enable	Persona At MLoc	LOSrv _c	
LOSrv _c			User	
User	MM-Embed	Object At MLoc	LOSrv _c	
LOSrv _c			User	
User	MM-Embed	Speech At MLoc	LOSrv _c	
LOSrv _c			User	
User	MU-Send	Event At URI	COSrv _c	
COSrv _c			User	

12.5 Virtual Meeting

12.5.1 Description

1. A meeting manager:
 1. Authors a meeting room.
 2. Makes a Transaction to the authoring service.
 3. Places the room at a location which is a subset of a larger location that meeting manager has Rights in.
 4. Grants participants the Rights to Act in meeting room.
 5. Places its Persona at a location in the room to act as virtual meeting secretary.
 6. Executes a virtual meeting secretary program tasked to produce a summary of the conversations.
 7. Summary is displayed in the meeting room for participants to comment.
2. Participant #1 joins the meeting.
3. Participant #2 joins the meeting.
4. Participant #1 speaks.
5. Participant #2
 1. Does not understand the speech.
 2. Makes a Transaction to an interpretation service to get a translation.
 3. Responds to Participant #1 speech.
6. Virtual meeting secretary displays meeting summary.

12.5.2 Workflow

12.5.2.1 Declarations

Declare human ₁	Meeting manager
Declare User ₁	Meeting manager's User
Room _{1.1}	Meeting room
Transaction _{1.1}	Transaction effected by Meeting Manager to Authoring Service
MLoc _{1.1}	M-Instance estate Meeting Manager has Rights in.
MLoc _{1.2}	Location of Meeting Manager's estate portion where meeting room is placed.
Persona _{1.1}	Avatar of Virtual Meeting Secretary.
Anim _{1.1}	Program animating Virtual Meeting Secretary.
MLoc _{1.3}	Place in Room where Virtual Meeting Secretary sits.
MLoc _{1.4}	Place where Summary is displayed.
Rights _{1.1}	User ₁ 's Rights in meeting room.
Summary _{1.1}	Summary posted by Executed Amim _{1.1}
Declare AUSrvc ₁	Authoring Service.
Declare EXsrvc ₁	Program Execution Service.
Declare IDSrvc	Identification Service
Declare TRSrvc ₁	Transaction Service.
Declare human ₂	Participant #1.
Declare User ₂	Participant #1' User.
Persona _{2.1}	User ₂ 's Persona.
MLoc _{2.1}	Participant #1's location in meeting room.

Speech _{2,1}	Speech uttered by Participant #1.
ULoc _{2,1}	Universe Location where Participant #1 is located.
Declare Device ₂	Device of human ₂
Declare human ₃	Participant #2.
Declare User ₃	Participant #2's User.
Persona _{3,1}	User ₃ 's Persona.
MLoc _{3,1}	Place in meeting room assigned to Persona _{3,1}
Speech _{3,1}	Speech uttered by Participant #2.
ULoc _{3,1}	Universe Location where Participant #2 is located.
Declare ITSrv ₁	Interpretation Service

12.5.2.2 Operation

S-Proc.	Action	S-Complements	D-Proc.	D-Compl.	Comments
User ₁	Authors	<i>With</i> (Data _{1,1} , Qualifier, MRights _{1,1} , Transaction _{1,1})	AUSrv ₁	At User ₁	User ₁ authors Room paying for Rights
AUSrv ₁		Room _{1,1} <i>With</i> Rights _{1,1}	User ₁		
User ₁	MM-Embed	Room _{1,1} <i>At</i> MLoc _{1,2}	LOSrv ₁		User ₁ embeds Room
LOSrv ₁			User ₁		
User ₁	MM-Embed	Persona _{1,1} <i>At</i> MLoc _{1,2}	LSrv ₁		User ₁ embeds Virtual Secretary
LOSrv ₁			User ₁		
User ₁	Execute	Anim _{1,1} <i>With</i> Persona _{1,1} <i>At</i> MLoc _{1,2}	EXsrv ₁		User ₁ animates Virtual Secretary
EXsrv ₁			User ₁		
User ₂	Track	Persona _{2,1} <i>From</i> Uloc _{2,1} <i>With</i> SA <i>At</i> MLoc _{2,1} <i>With</i> SA	LOSrv ₁ , Device ₂ , IDSrv ₁		Persona _{2,1} represents invisible User ₂
User ₂	MM-Enable	Persona _{2,1} <i>At</i> MLoc _{2,1}	LOSrv ₁		Persona _{2,1} is made visible
LOSrv ₁			User ₁		
User ₃	Track	Persona _{3,1} <i>From</i> Uloc _{3,1} <i>With</i> SA <i>At</i> MLoc _{3,1} <i>With</i> SA	LOSrv ₁ , Device ₃ , IDSrv ₁		Persona _{3,1} represents invisible User ₃
User ₃	MM-Enable	Persona _{3,1} <i>At</i> MLoc _{3,1}	LOSrv ₁		Persona _{3,1} is made visible
LOSrv ₁			User ₁		
User ₂	MM-Embed	Speech _{2,1} <i>At</i> MLoc _{2,1}	LOSrv ₁		User ₂ speaks
User ₃	Interpret	<i>With</i> (Interpretation _{3,1} , Transaction _{3,1})	ITSrv ₁	At User ₃	User ₃ requests interpretation of speech
ITSrv ₁			User ₃		

User ₃	MM-Embed	Speech _{3.1} At MLoc _{3.1}	LOSrv ₁	User ₃ speaks
LOSrv ₁			User ₃	
User ₁	MM-Embed	Summary _{1.1} With Qualifier At MLoc _{1.3}	LOSrv ₁	Virtual Secretary displays Summary
LOSrv ₁			User ₁	

12.5.3 Elementary Messages

S-Proc	Action	S-Complements	D-Proc	D-Compl
User	Authors	With (Data, Qualifier, Mrights, Transaction)	AUSrv _c	At User
AUSrv _c		Room With Rights	User	
User	MM-Embed	Room At MLoc	LOSrv _c	
LOSrv _c		Rights	User	
User	Execute	Anim With Persona At MLoc	EXSrv _c	
EXSrv _c		Rights	User	
User	UM-Capture	MediaAt ULoc	Device	At Device
Device		Data With Qualifier	User	
User	Identify	Data With Qualifier	IDrv _c	At User
IDSrv _c		Stream With Rights	User	
User	MM-Animate	Persona With Stream	LOSrv _c	
LOSrv _c		Rights	User	
User	UM-Actuate	MLoc At ULoc With Spatial Attitude	Device ₃	
Device			User	
User	MM-Embed	Speech With Qualifier At MLoc	LOSrv _c	
LOSrv _c			User	
User ₃	Interpret	With (Interpretation, Transaction)	ITRSrv _c	At User
RISrv _c		Rights	User	
User	MM-Embed	Summary With (Qualifier, Rights) At MLoc	LOSrv _c	
LOSrv _c			User	

12.6 Hybrid working

12.6.1 Description

1. Workers of a company implementing a mixed in-presence and remote policy may attend in one of two ways:
 1. Physically (called R-Workers) attending the Real Venue.
 2. Virtually (called V-Workers) attending the Virtual Venue.
2. The Virtual Venue
 1. Is an accurate replica of the Real Venue.
 2. Includes Rights to V-Workers to Act.
3. Real Workers
 3. Are authenticated when entering the Company.
 4. Are present in the Virtual space with their Personae with the Rights to Act.
4. Virtual Workers are rendered as Avatars in the Real space of the Company.

5. Workers in the Virtual space do not need Authentication because the Company's Virtual Space includes the Rights for Workers to Act with their specified Personae.
6. All Workers share their speech objects with:
 1. The Virtual Venue, if they are in the Real Venue.
 2. The Real Venue, if they are in the Virtual Venue,
 3. Participate in virtual and real meetings with Virtual Workers rendered as Avatars in the Real space of the Company.
7. A V-Worker
 1. Sends speech message to an R-Worker.
 2. Moves their Persona to R-Worker's place.
 3. Moves their Persona to Meeting Room.
8. R-Worker Moves themselves to the Real Meeting Room and its Persona to the Virtual Meeting Room.
9. Company manager
 1. Places Whiteboard in Meeting Room
 2. Animates Whiteboard.

12.6.2 Workflow

12.6.2.1 Declarations

Declare human ₁	Company manager
Declare User ₁	Company manager's User
Office _{1,1}	Company Office
Whiteboard ₁₁	Virtual Whiteboard in Office.
MLoc _{1,1}	A location in the M-Instance where Office is located.
MLoc _{1,3}	Location of Meeting Room in Office.
MLoc _{2,4}	Place of Whiteboard in Office.
Declare Anim _{1,2}	Process animating Whiteboard.
Declare human ₂	R-Worker
Declare User ₂	R-Worker's User
Authentication _{2,1}	Authentication Item for User ₂ .
Persona _{2,1}	R-Worker's Persona (R-Persona)
MLoc _{2,1}	Place of R-Worker's desk in Office.
MLoc _{2,2}	Location close to R-Worker's virtual desk.
Declare human ₃	V-Worker
Declare User ₃	V-Worker's User
Authentication _{3,1}	Authentication Item for User ₃ .
Persona _{3,1}	V-Worker's Persona (V-Persona)
MLoc _{3,1}	Place of V-Worker's desk in Office.
Object _{3,1}	V-Worker's Speech Object.
MLoc _{3,2}	A place in Meeting Room.

12.6.2.2 Operation

S-Proc	Action	S-Complements	D-Process	D-Compl.	Comments
--------	--------	---------------	-----------	----------	----------

User ₁	MM-Embed	Office At MLoc _{1,1} With SA	LOSrvc		User ₁ places Office
LOSrvc			User ₁		
User ₂	Track	Persona _{2,1} From Uloc _{2,1} With SA At MLoc _{2,1} With SA	LOSrvc, Device ₂ , IDSrvc		Persona _{2,1} (invisible) appears
User ₂	MM-Enable	Persona _{2,1} At MLoc _{2,1}	LOSrvc		Persona _{2,1} made visible
LOSrvc			User ₁		
User ₃	Track	Persona _{3,1} From Uloc _{3,1} With SA At MLoc _{3,1} With SA	LOSrvc, Device ₃ , IDSrvc		Persona _{3,1} (invisible) appears
User ₃	MM-Enable	Persona _{3,1} At MLoc _{3,1}	LOSrvc		Persona _{3,1} made visible
LOSrvc			User ₁		
User ₃	MM-Send	Speech _{3,1}	LOSrvc	To User ₂	User ₃ privately talks to User ₂
LOSrvc			User ₃		
User ₂	MM-Send	Speech _{2,1}	LOSrvc	To User ₃	User ₂ responds to User ₃
LOSrvc			User ₂		
User ₃	MM-Move	Persona _{3,1} From MLoc _{3,1} To MLoc _{3,2} With SA	LOSrvc ₁		User ₃ moves to User ₂ 's desk
LOSrvc			User ₃		
User ₂	MM-Move	Persona _{2,1} From MLoc _{2,1} To MLoc _{1,3} With SA	LOSrvc ₁		User ₂ moves to meeting room
LOSrvc			User ₂		
User ₃	MM-Move	Persona _{3,1} From MLoc _{3,2} To MLoc _{1,3} With SA	LOSrvc ₁		User ₃ moves to meeting room
LOSrvc			User ₃		
User ₁	MM-Embed	Whiteboard _{1,1} At MLoc _{2,4} With SA	LOSrvc ₁		User ₁ places whiteboard
LOSrvc			User ₁		
User ₁	Execute	Anim _{1,2} With Whiteboard _{1,1} At MLoc _{1,4} With SA	LOSrvc ₁		User ₁ animates whiteboard
LOSrvc			User ₁		

12.6.3 Elementary Messages

S-Proc	Action	S-Complements	D-Proc	D-Compl
User	MM-Embed	Office At MLoc With SA	LOSrvc	
LOSrvc			User	

User	MM-Embed	Persona <i>At</i> MLoc <i>With</i> SA	LOSrvc	
LOSrvc			User	
User	Execute	Anim <i>With</i> Persona <i>At</i> MLoc	EXsrvc	
EXSrvc			User	
User	Track	Persona <i>From</i> Uloc <i>With</i> SA <i>At</i> MLoc <i>With</i> SA	LOSrvc, Device, IDSrvc	
User	Authenticates	Persona <i>With</i> Authentication	ATSrvc	<i>At</i> User
ATSrvc		Authentication	User	
User	MM-Send	Speech	LOSrvc	<i>To</i> User
LOSrvc			User	
User	MM-Move	Persona <i>From</i> MLoc <i>At</i> MLoc <i>With</i> SA	LOSrvc	
LOSrvc			User	
User	MM- Embed	Whiteboard <i>At</i> MLoc <i>With</i> SA	LOSrvc	
LOSrvc			User	
User	Execute	Anim _{1,1} <i>With</i> Whiteboard <i>At</i> MLoc <i>With</i> SA	LOSrvc	
LOSrvc			User	

12.7 eSports Tournament

12.7.1 Description

1. Site manager's User:
 - Develops a game landscape.
 - Places it at a virtual location where it has Rights.
2. Game manager's User:
 - Acquires Rights to game landscape on location.
 - Places and animates autonomous characters on game landscape.
 - Places and animates virtual cameras and microphones in the landscape.
3. Players' Users
 - Place their Personae in the game landscape.
 - Track themselves in the game landscape.

4. Spectators have the rights to be present in the game landscape without being seen and without interfering with the game and to view captured AV from game landscape onto a dome screen and streamed online.

12.7.2 Workflow

12.7.2.1 Declarations

Declare human ₁	Site Manager.
Declare User ₁	Site Manager's User.
Declare AUSrvc ₁	Author Service.
Declare LOSrvc ₁	Location Service.
Declare TRSrvc ₁	Transaction Service.
Landscape _{1,1}	Game Landscape.
MLoc _{1,1}	Game Landscape Location in M-Instance.
Rights _{1,1}	Rights to Landscape
Declare human ₂	Game Manager.
Declare Anim _{2,i}	Processes animating Autonomous Characters.
Declare Device ₁	Microphone/Camera control.
Declare Device ₂	Dome Screen.
Declare MediaAnim _k	Processes operating Microphones/Cameras.
Declare User ₂	Game Manager's User.
M-Loc _{2,i}	Place in Game Landscape for Persona _{2,i} .
Persona _{2,i}	Autonomous characters deployed by Game Manager.
M-Loc _{2,k}	Place in Game Landscape for MediaDevice _{2,k}
MediaDevice _{2,k}	Visual Object operated by MediaAnim _k
Rights _{2,1}	Acquired Rights to Landscape.
Transaction _{2,1}	Fees paid by Game Manager to rent Game Landscape.
Scene _{2,1}	Game's Scene.
Declare User _j	An eSports Game Players.
Personae _{j,1}	A Players' character.
M-Loc _{j,1}	Location in Game Landscape for player characters.
Declare Process _i	Process Animating i-th Autonomous character.
Declare human _n	An online subscriber.
Declare User _n	An online subscriber' User.
Declare ULoc _{n,1}	Place in Universe of an online subscriber.
Declare MLoc _{n,1}	Place in Landscape of online subscriber's invisible Persona.

12.7.2.2 Operation

S-Process	Action	S-Complements	D-Process	D-Compl.	Comments
-----------	--------	---------------	-----------	----------	----------

User ₁	Author	<i>With</i> (Item, Data, Qualifier, ModelRights _{1.1})	AUSrv ₁	Landscape _{1.1} <i>At</i> MLoc _{1.1}	User ₁ Authors Landscape & places it invisible
AUSrv ₁		Landscape _{1.1}	User ₁		
User ₁	Transact	<i>With</i> Transaction _{1.1}	TRSrv ₁		User ₁ pays for the landscape
TRSrv ₁			User ₁		
User ₁	MM-Enable	Landscape _{1.1} <i>With</i> SA	LOSrv ₁		Landscape is made visible
LOSrv ₁			User ₁		
User ₂	Transact	<i>With</i> Transaction _{2.1}	TRSrv ₁	<i>At</i> User ₂	Game manager rents Landscape
TRSrv ₁		Rights _{2.1}			
User ₂	MM-Embed	Personae _{2.i} <i>At</i> MLoc _{2.i}	LOSrv ₁		Game manager places NPCs
LOSrv ₁			User ₂		
User ₂	Execute	Anim _{2.i} <i>With</i> Persona _{2.i}	EXSrv ₁		Game manager animates NPCs
EXSrv ₁			User ₂		
User ₂	MM-Embed	MediaDevice _{2.k} <i>At</i> MLoc _{2.k} <i>With</i> SA	LOSrv ₁		Game manager places media devices
LOSrv ₁			User ₂		
User ₂	Execute	MediaAnim _{2.k} MediaDevice _{2.k} <i>At</i> MLoc _{2.k}	EXSrv ₁		Game manager animates media devices
LOSrv ₁			User ₂		
User _j	Track	Personae _{j.1} <i>From</i> Uloc _{j.1} <i>With</i> SA <i>At</i> MLoc _{j.1} <i>With</i> SA	LOSrv ₁ , Device ₁ , IDSrv ₁		Invisible Personae _{j.1} placed in Landscape
User _j	MM-Embed	Personae _{2.j} <i>At</i> MLoc _{2.j}	LOSrv ₁		Personae _{j.1} made visible
LOSrv ₁			User ₂		
User _j	Track	Personae _{n.1} <i>From</i> Uloc _{n.1} <i>With</i> SA <i>At</i> MLoc _{n.1} <i>With</i> SA	LOSrv ₁ , Device ₁ , IDSrv ₁		Online human places invisible Persona

User _n	MU- Actuate	MLoc _{n.1} At ULoc _{n.1} With SA	LOSrv _{c1}	Online human views game.
LOSrv _{c1}			User _n	

12.7.3 Elementary Messages

S- Process	Action	S-Complements	D- Process	D-Compl
User	Author	<i>With</i> (Item, Data, Qualifier, ModelRights)	AUSrv _c	Landscape At MLoc
AUSrv _c		Rights	User	
User	MM- Enable	Landscape <i>With</i> SA	LOSrv _c	
LOSrv _c			User	
User	Transact	<i>With</i> Transaction	TRSrv _c	At User
TR Srv _c		Rights		
User	MM- Embed	Personae _i At MLoc _i	LOSrv _c	
LOSrv _c			User	
User	Execute	Anim _i <i>With</i> Personae _i	EXSrv _c	
EXSrv _c			User	
User	MM- Embed	MediaDevice _k At MLoc _k <i>With</i> SA	LOSrv _c	
LOSrv _c			User	
User	Execute	MediaAnim _k MediaDevice _k At MLoc _k	EXSrv _c	
LOSrv _c			User	
User _n	MM-Add	Personae _n At MLoc _n <i>With</i> SA	LOSrv _c	
LOSrv _c			User _n	

12.8 Virtual Performance

12.8.1 Description

1. Impresario's User:
 1. Holds rights in parcel.
 2. Authors auditorium.
 3. Places auditorium on parcel.
2. Performer sings.
3. Participant#1's User:
 1. Buys a ticket for an event with the Rights to stay close to the performer for 5 minutes.
 2. Moves close to Performer.
 3. Moves back to original place.
4. Participant#2's User:
 1. Utters a private speech to Participant#1.
5. Impresario's User:
 1. Collects participants' preferences.
 2. Interprets participants' mood (Participants Status).

3. Generates special effects based on preferences and Participant Status.
4. Pays Performer.

12.8.2 Workflow

12.8.2.1 Declarations

Declare User ₁	Impresario.
Declare AUSrvc ₁	Author Service.
Declare IDSrvc ₁	Identify Service.
Declare ITSrvc ₁	Interpret Service.
Declare LOSrvc ₁	Location Service.
Declare PASrvc ₁	Parcel Service.
Declare TRSrvc ₁	Transaction Service.
Auditorium _{1,1}	Auditorium for performances.
Collect _{1,1}	Program collecting preferences.
Interpretation _{1,1}	Interpretation sent to ITSrvc.
Interpretation _{1,2}	Interpretation received from ITSrvc.
MLoc _{1,1}	Impresario's Parcel.
MLoc _{1,i}	Places in Auditorium where Special FXs are placed.
PersonalStatus _{1,i}	Personal Status of i-th participants in event.
Rights _{1,1}	Rights in Auditorium.
SFX _{1,i}	Special FX.
Transaction _{1,1}	Payment for Auditorium authoring.
Transaction _{1,2}	Impresario's Consideration for Performance.
Declare User ₂	Performer
Persona _{2,1}	Performer's Persona
M-LOC _{2,1}	Stage in Auditorium
Song _{2,1}	Performance
ULoc _{2,1}	Place in Universe where Performer resides.
Declare User ₃	Participant#1
Persona _{3,1}	Participant#1's Persona
M-LOC _{3,1}	Participant#1's place in Auditorium
Rights _{3,1}	Participant#1's Rights to attend event.
Speech _{3,1}	Speech Object
Transaction _{3,1}	Ticket#1 to event
ULoc _{3,1}	Place in Universe where Participant#1 resides.
Declare User ₄	Participant#2
Persona _{4,1}	Participant#2's Persona
M-LOC _{4,1}	Participant#2's place in Auditorium
Transaction _{4,1}	Ticket#2 to event
ULoc _{4,1}	Place in Universe where Participant#2 resides.

12.8.2.2 Operation

S-Process	Action	S-Complements	D-Process	D-Compl.	Comments
User ₁	Author	<i>With</i> (Data, Qualifier, MRights _{1.1})	AUSrvc ₁	<i>At</i> User ₁	Impresario authors Auditorium
AUSrvc ₁		Auditorium _{1.1}	User ₁		
User ₁	Transact	Transaction _{1.1}	TRSrvc ₁	<i>At</i> User ₁	Impresario pays for Auditorium
TRSrvc ₁		Rights _{1.1}	User ₁		
User ₁	MM-Embeds	Auditorium _{1.1} <i>At</i> M-Loc _{1.1} <i>With</i> SA	LOSrvc ₁		Impresario places Auditorium
LOSrvc ₁			User ₁		
User ₁	Execute	Collect _{1.1}	EXSrvc ₁		Impresario executes preference collection program
EXSrvc ₁			User ₁		
User ₂	Track	Persona _{2.1} <i>From</i> ULoc _{2.1} <i>With</i> SA <i>To</i> MLoc _{2.1} <i>With</i> SA	LOSrvc ₁ , Device ₁ , IDSrvc ₁		Performer's Persona is invisible
User ₂	MM-Enable	Persona _{2.1} <i>At</i> MLoc _{2.1}	LOSrvc ₁		Performer's Persona is visible
LOSrvc ₁			User ₂		
User ₃	Transact	Transaction _{3.1}	TRSrvc ₁		User ₃ buys ticket
TRSrvc ₁		Rights _{3.1}	User ₃		
User ₃	Track	Persona _{3.1} <i>From</i> ULoc _{3.1} <i>At</i> MLoc _{3.1} <i>With</i> SA	LOSrvc ₁ , Device ₁ , IDSrvc ₁		User ₃ 's Persona is invisible
User ₃	MM-Enable	Persona _{3.1} <i>At</i> MLoc _{3.1}	LOSrvc ₁		User ₃ 's Persona is visible
LOSrvc ₁			User ₂		
User ₂	MM-Embed	Song _{2.1} <i>At</i> MLoc _{2.1} <i>With</i> SA	LOSrvc ₁		Performer sings
LOSrvc ₁			User ₂		

User ₃	MM-Move	Persona _{3.1} <i>From</i> MLoc _{3.1} <i>To</i> MLoc _{3.2} <i>With</i> SA	LOSrv ₁		User ₃ 's Persona comes close to Performer
LOSrv ₁			User ₃		
User ₃	MM-Move	Persona _{3.1} <i>From</i> MLoc _{3.2} <i>SA To</i> MLoc _{3.1} <i>With</i> SA	LOSrv ₁		User ₃ 's Persona returns to original place
LOSrv ₁			User ₃		
User ₄	Transact	Transaction _{4.1}	TRSRvc ₁		User ₄ buys ticket
TRSRvc ₁		Rights _{4.1}	User ₄		
User ₄	Track	Persona _{4.1} <i>From</i> ULoc _{4.1} <i>With</i> SA <i>At</i> MLoc _{4.1} <i>With</i> SA	LOSrv ₁ , Device ₁ , IDSrv ₁		User ₄ 's Persona is invisible
User ₄	MM-Send	Speech _{4.1}	COSrv ₁	<i>To</i> User ₃	User ₄ sends vocal message to User ₃
COSrv ₁			User ₄		
User ₁	Interprets	<i>With</i> Interpretation _{1.1}	ITSrv ₁	<i>At</i> User ₁	Impresario gets interpretation of preferences
ITSrv ₁		Interpretation _{1.2}	User ₁		
User ₁	MM-Embeds	SFX _{1.i} <i>At</i> M-Loc _{1.i} <i>With</i> SA	LOSrv ₁		Impresario places SFX's at appropriate places
LOSrv ₁			User ₃		
User ₁	Transact	<i>With</i> Transaction _{1.2}	TRSRvc ₁	<i>To</i> User ₂	Impresario pays Performer
TRSRvc ₁			User ₂		

12.8.3 Elementary Messages

S-Process	Action	S-Complements	D-Process	D-Compl.	Comments
User ₁	Author	<i>With</i> (Data, Qualifier, MRights _{1.1})	AUSrv ₁	<i>At</i> User ₁	Impresario authors auditorium
AUSrv ₁		Auditorium _{1.1}	User ₁		
User ₁	Transact	Transaction _{1.1}	TRSRvc ₁	<i>At</i> User ₁	Impresario pays for auditorium
TRSRvc ₁		Rights _{1.1}	User ₁		

User ₁	MM-Embeds	Auditorium _{1,1} At M-Loc _{1,1} With SA	LOSrv ₁		<i>Impresario lplaces auditorium</i>
LOSrv ₁			User ₁		
User ₁	Execute	Collect _{1,1}	EXSrv ₁		<i>Impresario launches preference collection</i>
EXSrv ₁			User ₁		
User ₂	Track	Persona _{2,1} From ULoc _{2,1} With SA At MLoc _{2,1} With SA	LOSrv ₁ , Device ₁ , IDSrv ₁		Persona _{2,1} is in M-Instance invisible
User ₂	MM-Enable	Persona _{2,1}	LOSrv ₁		Performer is visible
User ₃	Transact	Transaction _{3,1}	TRSrv ₁		User ₃ buys ticket
TRSrv ₁		Rights _{3,1}	User ₃		
User ₃	Track	Persona _{3,1} From ULoc _{3,1} With SA At MLoc _{3,1} With SA	LOSrv ₁ , Device ₁ , IDSrv ₁		Persona _{3,1} is in M-Instance invisible
User ₃	MM-Enable	Persona _{3,1}	LOSrv ₁		Participant#1 is visible
User ₃	MM-Move	Persona _{3,1} From MLoc _{3,1} At MLoc _{3,2} With SA	LOSrv ₁		Participant#1 moves close to Performer
LOSrv ₁			User ₃		
User ₄	Transact	Transaction _{4,1}	TRSrv ₁		User ₄ buys ticket
TRSrv ₁		Rights _{4,1}	User ₄		
User ₄	Track	Persona _{4,1} From ULoc _{4,1} With SA At MLoc _{4,1} With SA	LOSrv ₁ , Device ₁ , IDSrv ₁		Persona _{4,1} is in M-Instance invisible
User ₄	MM-Enable	Persona _{4,1}	LOSrv ₁		Participant#2 is visible
User ₄	MM-Send	Speech _{4,1}	COSrv ₁	To User ₃	Participant#2 sends speech to Participant#1
COSrv ₁			User ₄		
User ₁	Interpret	With Interpretation _{1,1}	ITSrv ₁	At User ₁	Impresario interprets participant mood
ITSrv ₁		Interpretation _{1,2}	User ₁		

12.9 AR Tourist Guide

12.9.1 Description

1. User₁ pays for and gets Rights in a land parcel.
2. User₂ develops a landscape with Author Service.
3. User₁
 1. Acquires rights in the Landscape.
 2. Places the landscape on its land parcel.

4. User₃ develops objects with Author Service.
5. User₁
 1. Acquires rights in the objects.
 2. Places objects in the landscape.
 3. Places personae at the different locations.
 4. Animates personae.
 5. Develops Universe-Metaverse Map.
6. human₁ develops and distributes AR app.
7. human₄
 1. Installs app on device.
 2. Moves around in the Universe.
 3. Upon reaching a U-Location, human₄ is alerted.
8. User₄ interacts with User₁'s Personae at the M-Locations that correspond with the U-Location.

12.9.2 Workflow

12.9.2.1 Declarations

Declare human ₁	AR app developer and distributor.
Declare User ₁	AR app developer and distributor's User.
Declare AUSvc ₁	Author Service
Declare EXSvc ₁	Execution Service
Declare IDSvc ₁	Identify Service
Declare LOSvc ₁	LocationService
Declare PASvc ₁	Parcel Service.
Declare TRSvc ₁	Transaction Service
Persona _{1,i}	User ₁ 's Persona _{1,i} .
Anim _{1,1}	
MLoc _{1,1}	Land parcel.
MLoc _{1,j}	Place where User ₁ places Objects in Landscape _{2,1} .
MLoc _{1,k}	M-Locations correspondent to ULoc _{1,k} .
Rights _{1,1}	Rights in MLoc _{1,1}
Rights _{1,j}	Rights in MLoc _{1,1}
ULoc _{1,k}	U-Locations where App reacts.
Map _{1,1}	Universe-Metaverse Map for the mobile app.
Transaction _{1,1}	Payment for Land Parcel to PASvc.
Transaction _{1,2}	Payment for Landscape.
Transaction _{1,3}	Payment for Objects _{3,j}
Declare human ₂	Landscape developer.
Declare User ₂	Landscape developer's User.
Landscape _{2,1}	Landscape developed for Land Parcel.
Declare human ₃	Object developer.
Declare User ₃	Object developer.
Object _{3,j}	Objects for landscape.
Declare human ₄	human installing and running mobile app.

Declare User ₄	User of human ₄ (End User)
Declare Device ₄	Device of human ₄
Declare App ₄	Installed on Device ₄
MLoc _{4,k}	M-Location corresponding to U-Location _{4,k}

12.9.2.2 Operation

S-Process	Action	S-Complements	D-Process	D-Compl.	Comments
User ₁	Transact	<i>With</i> Transaction _{1,1}	TRSrcv ₁	<i>At</i> User ₁	AR app developer pays for Land parcel
TRSrcv ₁		Rights _{1,1}	User ₁		
User ₂	Author	<i>With</i> (Item _{2,1} , Data, Qualifier, ModelRights _{1,1})	AUSrcv ₁	<i>At</i> User ₁	Developer authors Landscape
AUSrcv ₁		Landscape _{2,1}	User ₂		
User ₁	Transact	<i>With</i> Transaction _{1,2}	TRSrcv ₁	<i>At</i> User ₁	<i>Gets Rights in</i> Landscape
TRSrcv ₁		Rights _{1,1}	User ₁		
User ₁	MM-Embed	Landscape _{2,1} <i>At</i> MLoc _{1,1} <i>With</i> SA	LOSrcv ₁		AR app developer places Landscape
LOSrcv ₁			User ₁		
User ₃	Author	<i>With</i> (Item, Data, Qualifier, ModelRights _{1,j})	AUSrcv ₁	<i>At</i> User ₁	Object developer authors objects for Landscape
AUSrcv ₁		Object _{3,j}	User ₂		
User ₁	Transact	<i>With</i> Transaction _{1,2}	TRSrcv ₁	<i>At</i> User ₁	AR app developer buys objects.
TRSrcv ₁		Rights _{1,j}	User ₁		
User ₁	MM-Embed	Objects _{3,j} <i>At</i> MLoc _{1,j} <i>With</i> SA	LOSrcv ₁		AR app developer places objects in Landscape.
LOSrcv ₁			User ₁		
User ₁	MM-Embed	Persona _{1,i} <i>At</i> MLoc _{1,i} <i>With</i> SA	LOSrcv ₁		AR app developer places NPCs in Landscape.
LOSrcv ₁			User ₁		
User ₁	Execute	Anim _{1,i} <i>At</i> MLoc _{1,t} <i>With</i> SA	LOSrcv ₁		AR app developer animates NPCs.
LOSrcv ₁			User ₁		
App ₄	MU-Actuate	MLoc _{1,k} <i>At</i> ULoc _{1,k}	LOSrcv ₁		End User's App renders M-Instance.
LOSrcv ₁		Media _{1,k}	App ₄		

12.9.3 Elementary Messages

S-Process	Action	S-Complements	D-Process	D-Compl.
User ₁	Transact	<i>With</i> Transaction _{1,1}	TRSrcv ₁	<i>At</i> User ₁
TRSrcv ₁		Rights _{1,1}	User ₁	
User ₁	MM-Embed	Landscape _{2,1} <i>at</i> MLoc _{1,1} <i>With</i> SA	LOSrcv ₁	

LOSrvC ₁			User ₁
User ₂	Author	<i>With</i> (Item, Data, Qualifier, ModelRights _{1.1})	AUSrvC ₁ At User ₁
AUSrvC ₁		Landscape _{2.1}	User ₂
User ₁	Execute	Anim _{1.i} At MLoc _{1.t} With SA	LOSrvC ₁
LOSrvC ₁			User ₁
App ₄	MU-Actuate	MLoc _{1.k} At ULoc _{1.k}	LOSrvC ₁
LOSrvC ₁		Media _{1.k}	App ₄

12.10 Virtual Dance School

12.10.1 Description

1. Dance teacher:
 1. Places their Persona#1 at the reception area of the dance school as virtual receptionist.
 2. Animates Persona#1 with an autonomous agent.
2. Student #1:
 1. Shows up at school.
 2. Greets receptionist.
3. Virtual receptionist reciprocates greetings.
4. Dance teacher:
 1. Places their Persona#2 (with haptic capabilities) in dance area#1.
 2. Dances with student #1's Persona (with haptic capabilities).
5. Student #2 shows up at school.
6. Teacher:
 1. Places their Persona#3 (with haptic capabilities) in dance area #2.
 2. Keeps on dancing with student #1 by animating Persona#2 with an autonomous agent.
 3. Dances with student #2's Persona (with haptic capabilities).

12.10.2 Workflow

12.10.2.1 Declarations

Declare human₁ Dance teacher.

Declare User₁ Dance teacher's User.

Persona_{1.1} School Receptionist persona.

MLoc_{1.1} Teacher's Office.

Anim_{1.1} Receptionist animation program.

Persona_{1.2} Dancing persona#1

Anim_{1.2} Dancer animation program.

MLoc_{1.2} Dancing area#1

Persona_{1.3} Dancing persona#2 (with autonomous agent)

MLoc_{1.3} Dancing area#1

Speech_{1.1} Speech Object#2 (Greetings)

Persona_{1.4} Dancing persona#3 (with autonomous agent)

MLoc_{1.4} Dancing area#2

Declare Device₁ Dance teacher's Device

Declare human₂ Dance student #1

Declare User₂ Dance student #1's User
 Persona_{2.1} Dance student#1's Persona
 ULoc_{2.1} Dance student#1's home.
 MLoc_{2.1} Place close to reception area.
 MLoc_{2.2} Place at dancing area#1
 Declare Device₂ Dance student#1's Device.
 Declare human₃ Dance student#2
 Declare User₃ Dance student#2#1's User.
 Persona_{3.1} Dance student#2#1's Persona.
 ULoc_{3.1} Dance student#2#1's home.
 MLoc_{3.1} Dance student#2's landing place.
 Mloc_{3.2} Dance student#2's dancing place.
 Declare Device₃ Dance teacher's Device.

12.10.2.2 Operation

S-Process	Action	S-Complements	D-Process	D-Compl.	Comments
User ₁ LOSrvC ₁	MM-Embed	Persona _{1.1} At MLoc _{1.1} With SA	LOSrvC ₁ User ₁		Teacher places NPC
User ₁ EXSrvC ₁	Execute	Anim _{1.1} With Persona _{1.1} At MLOC _{1.1}	EXSrvC ₁ User ₁		Teacher animates NPC
User ₂ LOSrvC ₁	Track	Persona _{2.1} From Uloc _{2.1} With SA At MLoc _{2.1} With SA	LOSrvC ₁ , Device ₁ , IDSrvC ₁		Student#1 represented as Persona (invisible)
User ₂ LOSrvC ₁	MM-Embed	Persona _{2.1} At MLoc _{2.1} With SA	LOSrvC ₁ User ₂		Persona _{2.1} invisible
User ₂ LOSrvC ₁	MM-Embed	Speech _{2.1} At MLOC _{2.1} With SA	LOSrvC ₁ User ₂		Persona _{2.1} greets
User ₁ LOSrvC ₁	MM-Embed	Speech _{1.1} At MLOC _{1.1} With SA	LOSrvC ₁ User ₁		NPC reciprocates
User ₁ LOSrvC ₁	Track	Persona _{1.2} From Uloc _{1.1} With SA At MLoc _{1.2} With SA	LOSrvC ₁ , Device ₁ , IDSrvC ₁		Teacher represented as Persona (invisible)
User ₁ LOSrvC ₁	MM-Embed	Persona _{1.2} At MLoc _{1.2} With SA	LOSrvC ₁ User ₂		Persona _{1.2} is visible
User ₂ LOSrvC ₁	MM-Move	Persona _{2.1} From MLoc _{2.1} To MLoc _{2.2} With SA	LOSrvC ₁ User ₂		Persona _{2.1} moves to dancing area #1

User ₃	Tracks	Persona _{3.1} <i>From</i> Uloc _{3.1} <i>With SA At MLoc_{3.1} With SA</i>	LOSrv ₁ , Device ₁ , IDSrv ₁	Student#2 represented as Persona (invisible)
User ₃	MM-Embed	Persona _{3.1} <i>At MLoc_{3.1} With SA</i>	LOSrv ₁	Student#2 is visible
LOSrv ₁			User ₃	
User ₃	MM-Move	Persona _{3.1} <i>To MLoc_{1.4} With SA</i>	LOSrv ₁	Student#2 moves to dancing place #2
LOSrv ₁			User ₃	
User ₁	MM-Disable	Persona _{1.2} <i>At MLoc_{1.2}</i>	LOSrv ₁	Teacher makes its dancing persona#1 invisible
LOSrv ₁			User ₁	
User ₁	MM-Embeds	Persona _{1.3} <i>At MLoc_{1.2} With SA</i>	LOSrv ₁	Teacher places its dancing persona#2 at dancing place #1
LOSrv ₁			User ₁	
User ₁	Execute	Anim _{1.2} <i>With Persona_{1.3} At MLoc_{1.2}</i>	EXSrv ₁	Teacher animates its dancing persona#2
EXSrv ₁			User ₁	
User ₁	MM-Move	Persona _{1.2} <i>From MLoc_{1.2} With SA At MLoc_{1.3} With SA</i>	LOSrv ₁	Teacher moves its dancing persona#1 at dancing place #2
LOSrv ₁			User ₁	

12.10.3 Elementary Messages

S-Process	Action	S-Complements	D-Process	D-Compl
User	MM-Embed	Persona <i>At MLoc With SA</i>	LOSrv	
LOSrv			User	
User	Execute	Anim <i>With Persona At MLoc</i>	EXSrv	
EXSrv			User	
User	Track	Persona <i>From Uloc With SA At MLoc With SA</i>	LOSrv, Device, IDSrv	
User	MM-Move	Persona <i>From MLoc At MLoc With SA</i>	LOSrv	
LOSrv			User	
User	MM-Disable	Persona <i>At MLoc</i>	LOSrv	
LOSrv			User	

12.11 Virtual Car Showroom

12.11.1 Description

1. A car dealer

1. Places its Persona in the car showroom.
 2. Animates Persona as showroom attendant.
 3. Places another Persona in the car showroom.
 4. Animates second Persona as showroom salesman.
2. A customer:
 1. Moves its Persona in the car showroom.
 2. Greets showroom attendant.
 3. Showroom attendant reciprocates the greeting.
 4. Dealer (as salesman):
 1. Converses with Customer.
 2. Embeds a model of the car being discussed.
 5. Customer
 1. Has a virtual test drive.
 2. Buys the car.
 3. Returns home.

12.11.2 Workflow

12.11.2.1 *Declarations*

Declare human₁ Car dealer
 Declare User₁ Car dealer's User.
 AVPersona_{1,1} User₁'s Persona (showroom attendant).
 AVPersona_{1,2} User₁'s Persona (salesman).
 MLoc_{1,1} Showroom attendant's place.
 AVPersona_{1,2} Showroom attendant.
 ULoc_{1,1} U-Location of Car dealer.
 MLoc_{1,2} Salesman's place.
 Speech_{1,1} Showroom attendant's greetings.
 CarModel_{1,1} 3D A-V-H model of car.
 MLoc_{1,3} Car model's place.
 Anim_{1,1} Program to animate showroom attendant.
 Anim_{1,2} Program to animate car model.
 Declare IDSrvc
 Declare LOSrvc₁
 Declare RISrvc₁
 Declare COSrvc₁
 Declare Device₁ Car dealer's Device
 Declare human₂ Customer.
 Declare User₂ Customer's User.
 Persona_{2,1} User₂'s Persona.
 ULoc_{2,1} U-Location of Customer.
 MLoc_{2,1} Place in showroom where Customer lands.
 MLoc_{2,2} Customer's home.
 Transaction_{2,4} Payment for car.
 Declare Device₂ Customer's Device.

Declare TRSrv₁ Transaction Service.

12.11.2.2 Operation

S-Process	Action	S-Complements	D-Process	D-Compl.	Comments
User ₁	Track	Persona _{1.1} <i>From</i> Uloc _{1.1} <i>With SA At</i> MLoc _{1.1} <i>With</i> SA	LOSrv ₁ , Device ₁ , IDSrv ₁		Car dealer's Persona _{1.1} is invisible
User ₁	MM- Embed	Persona _{1.1} <i>At</i> MLoc _{1.1} <i>With</i> SA	LOSrv ₁		Persona is visible
LOSrv ₁			User ₁		
User ₁	Execute	Anim _{1.1} <i>With</i> Persona _{1.1} <i>At</i> MLoc _{1.1}	EXSrv ₁		Persona (NPC) is animated
EXSrv ₁			User ₁		
User ₂	Tracks	Persona _{2.1} <i>From</i> Uloc _{2.1} <i>With SA At</i> MLoc _{2.1} <i>With</i> SA	LOSrv ₁ , Device ₁ , IDSrv ₁		Customer's Persona _{1.2} is invisible
User ₂	MM- Embed	Persona _{1.1} <i>At</i> MLoc _{2.1} <i>With</i> SA	LOSrv ₁		Persona is visible
LOSrv ₁			User ₂		
User ₁	MM- Embed	Speech _{1.1} <i>At</i> MLoc _{1.1} <i>With</i> SA	LOSrv ₁		NPC greets Customer
COSrv ₁			User ₁		
User ₂	MM- Embed	Speech _{2.1} <i>At</i> MLoc _{2.1} <i>With</i> SA	LOSrv ₁	<i>To</i> User ₁	Customer reciprocates
COSrv ₁			User ₂		
User ₁	MM- Embed	CarModel _{1.1} <i>At</i> MLoc _{1.3} <i>With SA</i>	LOSrv ₁		Dealer places visible Car Model
LOSrv ₁			User ₁		
User ₁	Execute	Anim _{1.2} <i>With</i> CarModel _{1.1} <i>At</i> MLoc _{1.3}	EXSrv ₁		Dealer animates Car Model
EXSrv ₁			User ₁		
User ₂	MM- Move	Persona _{2.1} <i>From</i> MLoc _{2.1} <i>To</i> MLoc _{1.3} <i>With SA</i>	<i>At</i> MLoc _{2.3} <i>With SA</i>		Customer gets in the Car Model
LOSrv ₁			User ₂		
User ₂	MM- Move	Persona _{2.1} <i>From</i> MLoc _{1.3} <i>To</i> MLoc _{2.1} <i>With SA</i>	<i>At</i> MLoc _{2.3} <i>With SA</i>		Customer returns to original place
LOSrv ₁			User ₂		
User ₂	Transacts	<i>With</i> Transaction ₁ Rights _{1.1}	<i>To</i> User ₁	<i>At</i> User ₁	Customer pays
TRSrv ₁			User ₂		
User ₂	MM- Moves	Persona _{2.1} <i>To</i> MLoc _{2.2}	<i>From</i> MLoc _{2.1}		Customer returns to home
LOSrv ₁			User ₂		

12.11.3 Elementary Messages

S-Process	Action	S-Complements	D-Process	D-Compl
User	Track	Persona <i>From</i> Uloc <i>With</i> SA <i>At</i> MLoc <i>With</i> SA	LOSrvc, Device, IDSrvc	
User	MM-Embed	Persona <i>At</i> MLoc <i>With</i> SA	LOSrvc	
LOSrvc			User	
User	Execute	Anim <i>With</i> Persona <i>At</i> MLoc	EXSrvc	
EXSrvc			User	
User	MM-Move	Persona <i>From</i> MLoc <i>At</i> MLoc ₃ <i>With</i> SA	<i>At</i> MLoc ₃ <i>With</i> SA	
LOSrvc			User	
User	Transacts	<i>With</i> Transaction	<i>To</i> User	<i>At</i> User
TRSrvc		Rights	User	
LOSrvc	MM-Disable	Persona <i>At</i> MLoc ₃	<i>From</i> MLoc	
LOSrvc			User	

12.12 Meeting in Connected Autonomous Vehicles

12.12.1 Description

A human_{A.1} traveling in CAV_A wishes to have a meeting with a friend who is travelling in CAV_B. The metaverses generated by the two CAVs comply with the MPAI-CAV specifications but the metaverses they generate are independent but conform with the CAV-TEC 2.0 Technical Specification.

1. Human₁

1. Registers with CAV_A's Registration Service by providing their Personal Profile.

2. Registers

1. Their own User_{A.1} to interact with CAV_A via the CAV_A's Human-CAV Interaction Subsystem.
2. User_{A.1}'s Persona_{A.1.1},
3. Their own User_{A.2}, to access the CAV_A M-Instance created by the Autonomous Motion Subsystem.

2. Requests a Route to User₁.

2. User₁ requests and obtains proposed Routes to the destination from User_{A.2} of CAV_A, communicates the proposed Routes, and puts CAV_A in motion.

3. While travelling, human₁ wishes to meet in the virtual cabin of CAV_B with a friend travelling on CAV_B.

12.12.2 Workflow

12.12.2.1 Declarations

Declare human_{A.1}

CAV_A's Passenger.

Declare User_{A.1}

CAV_A User implemented as Human-CAV Interaction Subsystem.

Declare User_{A.2}

CAV_A User implemented as Autonomous Motion Subsystem.

Declare COSrvc _{A.1}	CAV _A 's Communication Service.
Declare IDsrvc _{A.1}	CAV _A 's Identification Service.
Declare LOSrvc _{A.1}	CAV _A 's Location Service.
Declare RESrvc _{A.1}	CAV _A 's Resolution Service
Declare RGSrvc _{A.1}	CAV _A 's Registration Service
Declare RPSrvc _{A.1}	CAV _A 's Route Selection Planning AIM.
Persona _{A.1.1}	User _{A.1} 's avatar.
BED _{A.1.1}	Initial CAV _A 's Basic Environment Descriptors
EgoRemoteAMSMMessage _{A.2.1}	CAV _A 's request for U-Location details
EgoRemoteAMSMMessage _{A.2.1}	CAV _B 's response
Interpretation _{A.1.1}	Request to Interpret Speech _{A.1.1}
Interpretation _{A.1.2}	Response to Interpretation _{A.1.1}
Interpretation _{A.1.3}	Request to Interpret Speech _{A.1.2}
Interpretation _{A.1.4}	Response to Interpretation _{A.1.3}
AMSHCIMessage _{A.1.1}	Message#1 from User _{A.1} to User _{A.2}
AMSHCIMessage _{A.1.2}	Message#2 from User _{A.2} to User _{A.1}
AMSHCIMessage _{A.1.3}	Message#3 from User _{A.1} to User _{A.2}
PersonaID _{A.1}	Identifier of Persona _{A.1}
Resolution _{A.1.1}	Resolution Item sent to CAV _B by CAV _A
Resolution _{A.1.2}	Resolution Item sent to CAV _A by CAV _B
ULoc _{A.1.1}	Place where human _{A.1} boards CAV _A .
UserID _{A.1}	Identifier of User _{A.1}
UserID _{A.2}	Identifier of User _{A.2}
Declare User _{B.1}	User of CAV _B implemented as Human-CAV Interaction Subsystem.
Declare RESrvc _{B.1}	CAV _B 's Resolution Service
MLOC _{B.1.1}	CAV _B 's Virtual Cabin.

12.12.2.2 *Operation*

S-Process	Action	S-Complements	D-Process	D-Comp.	Comments
human _{A.1}	Registers	With PersonalProfile _{A.1.1}	RGSrvc _{A.1}	To human _{A.1}	human _{A.1} registers with RGSrvc _{A.1}
RGSrvc _{A.1}		Account _{A.1.1}	human _{A.1}		human _{A.1} receives Account _{A.1.1}
human _{A.1}	UM-Send	User _{A.1}	IDSrvc _{A.1}	To human _{A.1}	human _{A.1} registers User _{A.1} to CAV _A
IDSrvc _{A.1}		UserID _{A.1}	human _{A.1}		
human _{A.1}	UM-Send	User _{A.2}	IDSrvc _{A.1}	To human _{A.1}	human _{A.1} registers User _{A.2} to CAV _A

IDSrvCA.1		UserIDA.2	humanA.1		
humanA.1	UM-Send	PersonaA.1	IDSrvCA.1	To humanA.1	humanA.1 uploads PersonaA.1 to CAVA
IDSrvCA.1		PersonaIDA.1	humanA.1		
UserA.1	Track	PersonaA.1.1 From ULocA.1.1 With SA At MLocA.1.1 With SA	LOSrvCA.1, DeviceA.1, IDSrvCA.1		PersonaA.1.1 shows up (invisible).
UserA.1	MM-Send	AMSHCIMessageA.1.1	COSrvCA.1	To UserA.2	UserA.1 sends route request
COSrvCA.1			UserA.1		
UserA.2	Execute	ESSA.1	EXSrvCA.1		UserA.2 activates ESSA.1
EXSrvCA.1		BEDA.1.1	UserA.2		UserA.2 receives Basic Environment Description
UserA.2	Execute	RoutePlanA.1	EXSrvCA.1		UserA.2 activates RoutePlanA.1
EXSrvCA.1			UserA.2		
UserA.2	MM-Send	AMSHCIMessageA.1.1	COSrvCA.1	To RoutePlanA.1	RoutePlanA.1 sends route request
COSrvCA.1			UserA.2		
RoutePlanA.1	MM-Send	AMSHCIMessageA.1.2	COSrvCA.1	To UserA.1	UserA.2 sends proposed routes
COSrvCA.1			RoutePlanA.1		
UserA.1	MM-Send	AMSHCIMessageA.1.3	COSrvCA.1	To UserA.2	With selected Route and Start command
COSrvCA.1			UserA.1		
UserA.1	Resolve	With ResolutionA.1.1	RESrvCA.1		UserA.1 requests session with CAVB
RESrvCA.1	Resolve	With ResolutionA.1.1	RESrvCB.1		RESrvCA.1 contacts RESrvCB.1
RESrvCB.1	Resolve	With ResolutionA.1.1	UserB.1		RESrvCB.1 contacts UserB.1
UserB.1	Resolve	With ResolutionA.1.2	RESrvCB.1		UserB.1 responds to RESrvCB.1
RESrvCB.1	Resolve	With ResolutionA.1.2	RESrvCA.1		RESrvCB.1 responds to RESrvCA.1

RESrvc _{A.1}	Resolve	<i>With Resolution_{A.1.2}</i>	User _{A.1}	RESrvc _{A.1} responds to User _{A.1}
User _{A.1}	MM-Embed	<i>At MLoc_{B.1.1} With SA</i>	LOSrvc _{A.1}	Persona _{A.1.1} is visible at MLoc _{B.1.1}
LOSrvc _{A.1}			User _{A.1}	
User _{A.1}	MM-Move	<i>From MLoc_{B.1.3} To MLoc_{A.1.3} With SA</i>	LOSrvc _{A.1}	Persona _{A.1.1} leaves MLoc _{B.1.1}
LOSrvc _{A.1}			User _{A.1}	
User _{A.1}	Resolve	<i>With Resolution_{A.1.3}</i>	RESrvc _{A.1}	User _{A.1} leaves Multi- Metaverse session
RESrvc _{A.1}	Resolve	<i>With Resolution_{A.1.3}</i>	RESrvc _{B.1}	RESrvc _{A.1} contacts RESrvc _{B.1}
RESrvc _{B.1}	Resolve	<i>With Resolution_{A.1.4}</i>	RESrvc _{A.1}	RESrvc _{B.1} responds to RESrvc _{A.1}
RESrvc _{A.1}	Resolve	<i>With Resolution_{A.1.4}</i>	User _{A.1}	RESrvc _{A.1} responds to User _{A.1}

12.12.3 Elementary Messages

S-Proc.	Action	S-Complements	D-Proc.	D-Compl.
human _{A.1}	Registers	<i>With PersonalProfile_{A.1.1}</i>	RGSrvc _{A.1}	<i>To human_{A.1}</i>
RGSrvc _{A.1}		Account _{A.1.1}	human _{A.1}	
human _{A.1}	UM-Send	User _{A.1}	IDSrvc _{A.1}	<i>To human_{A.1}</i>
IDSrvc _{A.1}		UserID _{A.1}	human _{A.1}	
human _{A.1}	UM-Send	Persona _{A.1}	IDSrvc _{A.1}	<i>To human_{A.1}</i>
IDSrvc _{A.1}		UserIID	human _{A.1}	
User _{A.1}	Track	<i>Persona_{A.1.1} From Uloc_{A.1.1} With SA At MLoc_{A.1.1} With SA</i>	LOSrvc _{A.1} , Device _{A.1} , IDSrvc _{A.1}	
User _{A.1}	MM-Send	AMSHCIMessage _{A.1.1}	COSrvc _{A.1}	<i>To User_{A.2}</i>
COSrvc _{A.1}			User _{A.1}	
User _{A.2}	Execute	ESS _{A.1}	EXSrvc _{A.1}	
EXSrvc _{A.1}		BED _{A.1.1}	User _{A.2}	
User _{A.2}	Execute	RoutePlan _{A.1}	EXSrvc _{A.1}	
EXSrvc _{A.1}			User _{A.2}	
User _{A.2}	MM-Send	AMSHCIMessage _{A.1.1}	COSrvc _{A.1}	<i>To RoutePlan_{A.1}</i>
COSrvc _{A.1}			User _{A.2}	

RoutePlan _{A.1}	MM-Send	AMSHCIMessage _{A.1.2}	COSrvc _{A.1}	To User _{A.1}
COSrvc _{A.1}			User _{A.2}	
User _{A.1}	MM-Send	AMSHCIMessage _{A.1.3}	COSrvc _{A.1}	To User _{A.2}
COSrvc _{A.1}			User _{A.1}	
User _{A.1}	Resolve	With Resolution _{A.1.1}	RESrvc _{A.1}	
RESrvc _{A.1}	Resolve	With Resolution _{A.1.1}	RESrvc _{B.1}	
RESrvc _{B.1}	Resolve	With Resolution _{A.1.1}	User _{B.1}	
User _{B.1}	Resolve	With Resolution _{A.1.2}	RESrvc _{B.1}	
RESrvc _{B.1}	Resolve	With Resolution _{A.1.2}	RESrvc _{A.1}	
RESrvc _{A.1}	Resolve	With Resolution _{A.1.2}	User _{A.1}	
User _{A.1}	MM-Embed	At MLoc _{B.1.1} With SA	LOSrvc _{A.1}	
LOSrvc _{A.1}			User _{A.1}	
User _{A.1}	MM-Move	From MLoc _{B.1.3} To MLoc _{A.1.3} With SA	LOSrvc _{A.1}	
LOSrvc _{A.1}			User _{A.1}	
User _{A.1}	Resolve	With Resolution _{A.1.3}	RESrvc _{A.1}	
RESrvc _{A.1}	Resolve	With Resolution _{A.1.3}	RESrvc _{B.1}	
RESrvc _{B.1}	Resolve	With Resolution _{A.1.4}	RESrvc _{A.1}	
RESrvc _{A.1}	Resolve	With Resolution _{A.1.4}	User _{A.1}	

12.13 Co-design across metaverses

12.13.1 Description

1. Company_A based in M-Instance_A has commissioned the design of their new car to Company_B based in M-Instance_B.
2. User_{B.1}, a Designer of Company_B, produces a 3D Model Object composed of different 3D Model Objects, and 3D Model Basic Objects.
3. A 3D Model Object may include different versions of car elements, eg, rearview mirrors, the dashboard, cabin lights etc.
4. The progress meeting has two participants: Customer (User_{A.1}) and Designer (User_A),
 1. User₁
 1. Has full Rights on Car Model_{1.1}.
 2. MM-Embeds Car Model at MLoc₁.
 3. Grants Rights to User₁ to Act on MLoc_{B.1.1}, i.e., Room_{B.1.1} (mirror of room in M-Instance_B), CarModel_{B.1.1} (Car Model) and User_{B.1}
 2. User₁ initiates a session with Resolution Service_{A.1} to be able to interact with MLoc_{B.1.1} and perform Actions on its Items on
 3. User₁ assesses the results of selecting different options, e.g.,
 1. MM-Disables RearviewMirror_{1.1}
 2. MM-Enables RearviewMirror_{1.2}
 3. Resizes RearviewMirror_{1.2} by R_x, R_y, R_z

12.13.2 Workflow

12.13.2.1 Declarations

Declare human _{A.1}	Car Model customer
Declare User _{A.1}	Car Model customer's User (in M-Instance _A)
Persona _{A.1.1}	User _{A.1} 's avatar.
M-Location _{A.1.1}	User _{A.1.1} 's location in M-Instance _A
U-Location _{A.1.1}	Portion of Universe where human _{A.1} is located.
RoomRights _{A.1.1}	Rights on Room _{B.1.1} granted to User _{A.1}
ModelRights _{A.1.1}	Rights on CarModel _{B.1.1} granted to User _{A.1}
Resolution _{A.1.1}	Resolution Item sent by User _{A.1} to Resolution Service _{A.1}
Declare Device _{A.1}	Device used by human _{A.1}
Declare LOSrvc _{A.1}	Location Service of M-Instance _A
Declare IDSrvc _{A.1}	Identification Service of M-Instance _A
Declare RESrvc _{A.1}	Resolution Service of M-Instance _A
Declare human _{B.1}	Car Model designer.
Declare User _{B.1}	Car Model designer's User (in M-Instance _B)
Persona _{B.1.1}	User _{B.1} 's avatar.
M-Location _{B.1.1}	User _{B.1} 's location in M-Instance _B
U-Location _{B.1.1}	human _{B.1} 's location in Universe
CarModel _{B.1.1}	3D-Model Object of Car Model
RVMirror _{B.1.1}	First Rearview Mirror
RVMirror _{B.1.2}	Second Rearview Mirror
Declare Device _{B.1}	Device used by human _{B.1}
Declare LOSrvc _{B.1.1}	Location Service of Metaverse _B
Declare IDSrvc _{B.1.1}	Identification Service of Metaverse _B

12.13.2.2 Operation

S-Proc	Action	S-Complements	D-Process	D-Compl	Comments
User _{A.1}	Track	Persona _{A.1.1} <i>From</i> Uloc _{A.1.1} <i>With SA At</i> MLoc _{A.1.1} <i>With SA</i>	LOSrvc _{A.1} , Device _{A.1} , IDSrvc _{A.1}		Persona _{A.1.1} shows up (invisible).
User _{A.1}	MM-Embed	Persona _{A.1.1} <i>At</i> MLoc _{A.1.1}	LOSrvc _{A.1}		Persona _{A.1.1} is visible.
LOSrvc _{A.1}			User _{A.1}		
User _{B.1}	Track	Persona _{B.1.1} <i>From</i> Uloc _{B.1.1} <i>With SA At</i> MLoc _{B.1.1} <i>With SA</i>	LOSrvc _{B.1} , Device _{B.1} , IDSrvc _{B.1}		Persona _{B.1} shows up (invisible)
User _{B.1}	MM-Embed	Persona _{B.1.1} <i>At</i> MLoc _{B.1.1}	LOSrvc _{B.1}		Persona _{B.1.1} is visible.
LOSrvc _{B.1}			User _{A.1}		

User _{A.1}	Resolve	<i>With Resolution_{A.1.1}</i>	RESrvc _{A.1}	User _{A.1} requests multi- metaverse session
RESrvc _{A.1}	Resolve	<i>With Resolution_{A.1.1}</i>	RESrvc _{B.1}	RESrvc _{A.1} contacts RESrvc _{B.1}
RESrvc _{B.1}	Resolve	<i>With Resolution_{A.1.1}</i>	User _{B.1}	RESrvc _{B.1} contacts User _{B.1}
User _{B.1}	Resolve	<i>With Resolution_{A.1.2}</i>	RESrvc _{B.1}	User _{B.1} responds to RESrvc _{B.1}
RESrvc _{B.1}	Resolve	<i>With Resolution_{A.1.2}</i>	RESrvc _{A.1}	RESrvc _{B.1} responds to RESrvc _{A.1}
RESrvc _{A.1}	Resolve	<i>With Resolution_{A.1.2}</i>	User _{A.1}	RESrvc _{A.1} responds to User _{A.1}
User _{B.1}	MM- Embed	CarModel _{B.1.1} <i>At</i> MLoc _{B.1.1} <i>With SA</i>	LOServ _{B.1}	User _{B.1} places CarModel _{A.1.1}
LOServ _{B.1}			User _{B.1}	
User _{A.1}	MM- Disable	RVMirror _{A.1}	LOServ _{A.1}	User _{A.1} makes current RVMirror _{A.1} invisible
LOServ _{A.1}			User _{A.1}	
User _{A.1}	MM- Enable	RVMirror _{A.2}	LOServ _{A.1}	User _{A.1} makes new RVMirror _{A.2} visible
LServ _{A.1}			User _{A.1}	
User _{A.1}	MM- Disable	RVMirror _{A.2}	LOServ _{A.1}	User _{A.1} makes new RVMirror _{A.2} invisible
LOServ _{A.1}			User _{A.1}	
User _{A.1}	MM- Embed	RVMirror _{A.2} <i>With</i> (R _x ,R _y ,R _z ,)	LOSrvc _{A.1}	User _{A.1} changes RVMirror _{A.2} 's size
LOSrvc _{A.1}			User _{A.1}	
User _{A.1}	Resolve	<i>With Resolution_{A.1.3}</i>	RESrvc _{A.1}	User _{A.1} leaves Multi-MV session
RESrvc _{A.1}	Resolve	<i>With Resolution_{A.1.3}</i>	RESrvc _{B.1}	RESrvc _{A.1} contacts RESrvc _{B.1}
RESrvc _{B.1}	Resolve	<i>With Resolution_{A.1.4}</i>	RESrvc _{A.1}	RESrvc _{B.1} responds to RESrvc _{A.1}
RESrvc _{A.1}	Resolve	<i>With Resolution_{A.1.4}</i>	User _{A.1}	RESrvc _{A.1} responds to User _{A.1}

12.13.3 Elementary Messages

S-Proc	Action	S-Complements	D-Proc	D- Compl
User	Track	Persona <i>From Uloc With SA At MLoc With SA</i>	LOSrvc, Device , IDSrvc	
User	Resolve	<i>With Resolution</i>	RESrvc	
RESrvc	Resolve	<i>With Resolution</i>	RESrvc	
RESrvc	Resolve	<i>With Resolution</i>	User	
User	Resolve	<i>With Resolution</i>	RESrvc	

RESrvc	Resolve	With Resolution	RESrvc
RESrvc	Resolve	With Resolution	RESrvc
RESrvc	Resolve	With Resolution	User
User	MM-Embed	CarModel At MLoc With SA	LOServ
LServ			User
User	MM-Disable	RVMirror	LOServ
LServ			User
User	MM-Enable	RVMirror	LOServ
LServ			User
User	MM-Embed	CarModel With (R _x ,R _y ,R _z ,)	LOSrvc
LOSrvc			User

12.14 Selling assets on a Marketplace

12.14.1 Description

User₁ develops a Product that is Posted to a Marketplace Service, which sells it to User₂.

12.14.2 Workflow

12.14.2.1 Declarations

Declare human ₁	Vendor.
human ₂	Purchaser.
User ₁	Vendor's User.
User ₂	Purchaser's User.
AUSrvc ₁	Author Service.
LOSrvc ₁	Location Service.
TRSrvc ₁	Transaction Service.
Product _{1,1}	Item to be sold out.
MKSrvc ₁	A Marketplace service.
Transaction _{1,1}	Payment for the Product
Transaction _{1,2}	Fees paid by Vendor to post the Product on the market.
MLoc _{1,1}	Product Location in M-Instance.
Rights _{1,1}	Rights to Product.
Right _{1,2}	Acquired Rights to the Vendor.
Transaction _{2,1}	Fees paid by Purchaser to purchase the Product.

12.14.2.2 Operation

S-Process	Action	S-Complements	D-Process	D-Compl.	Comments
-----------	--------	---------------	-----------	----------	----------

User ₁	Author	<i>With (Item, Data, Qualifier, MRights_{1.1})</i>	AUSrvc ₁	Product _{1.1} At MLoc _{1.1}	User ₁ authors Product & places it invisible
AUSrvc ₁		Product _{1.1}	User ₁		
User ₁	Transact	<i>With Transaction_{1.1}</i>	TRSrvc ₁		User ₁ pays for the Product _{1.1}
TRSrvc ₁		Rights _{1.1}	User ₁		
User ₁	MM-Enable	Product _{1.1} <i>With SA</i>	LOSrvc ₁		Product is made visible
LOSrvc ₁			User ₁		
User ₁	Transact	<i>With Transaction_{1.2}</i>	TRSrvc ₁		User ₁ pays to Post Product _{1.1} to marketplace service
LOSrvc ₁		Rights _{1.2}	User ₁		
User ₁	Post	<i>With MTransaction_{1.3}</i>	MKSrvc ₁		User ₁ posts Product to the marketplace
MKSrvc ₁			User ₁		
User ₂	Transact	<i>With Transaction_{2.1}</i>	TRSrvc ₁	At User ₁	Purchaser buys Product & pays Vendor
TRSrvc ₁			User ₂		

12.14.3 Elementary Messages

S-Process	Action	S-Complements	D-Process	D-Compl	Comments
User	Author	<i>With (Item, Data, Qualifier, MRights)</i>	AUSrvc	Product At MLoc	User authors Product & places it invisible
AUSrvc		Product	User		
User	Transact	<i>With Transaction</i>	TRSrvc		User pays for the Product
TRSrvc		Rights	User		
User	MM-Enable	Product <i>With SA</i>	LOSrvc		Product is made visible
LOSrvc			User		
User	Post	<i>With MTransaction₃</i>	MKSrvc		User posts Product to the marketplace
MKSrvc			User		

13 MPAI-MMM API

13.1 MMM-API

The MMM-API is a collection of endpoints and objects to support the execution of Process Actions performed by Processes (e.g. Users and Device) involving Items or other Processes. The list of Process Actions is reported in MMM-TEC V2.0 Process Actions.

The MMM-API can also be leveraged in order to provide Users specific functionalities through Services, e.g., Registration or Right service. Possible interactions envisioned in MMM have been reported in Verification Use Cases.

13.2 Description

The APIs were designed to run Create, Read, Update and Delete (CRUD) operations on a MongoDB NoSQL database. The base path is: /api. The documentation of the API relies on the OpenAPI 3.0.1 specification. More specifically, the documentation is structured as follows:

Section	Description
openapi	OpenAPI version used for the documentation
info	Metadata about the API (title, version, description, contact info).
paths	Endpoints and operations (GET, POST, PUT and DELETE).
components	Reusable objects like schemas, responses and parameters.

The comprehensive list including all the paths and components is [available](#) in the JSON format.

13.3 Components

Components were developed by following the definition of the MMM's Item available at MMM-TEC V2.0 Items (<https://mpai.community/standards/mpai-mmm/tec/v2-0/data-types/>).

13.4 Paths

Paths are structured to match the MMM's Services.

13.4.1 Activity Service

Base Route: api/Activity

Description: The Activity Service exposes endpoints to manage Process Actions and User information.

Sample Endpoint

To memorize a Process Action, it is possible to send a POST to the api/Activity/process-action endpoint. The Process Action to be added is specified in the request body.

```
curl -X 'POST' \
  'https://localhost:7099/api/Activity/process-actions' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d ' <ProcessAction> '
```

13.4.2 Authentication Service

Base Route: /api/authentication

Description: The Authentication Service provides endpoints for managing Authentication item. Endpoint are provided to create, retrieve, and delete Authentication item.

Sample Endpoint

To retrieve a list of all Authentication items it is possible to send a GET request.

```
curl -X 'GET' \
  'https://localhost:7099/api/Authentication/authentications' \
```

```
-H 'accept: text/plain'
```

13.4.3 Author Service

Base Route: /api/author

Description: This controller handles the creation, retrieval, and deletion of Item. It supports dynamic deserialization and conditional logic for polymorphic data insertion via query parameter format. Dedicated endpoints are available for specific items such as BasicObject, and MObject.

Sample Endpoint:

To create a new Item based on the provide format query parameter, a POST request can be sent to the endpoint `api/Author/item?format=item`. Depending on the specified format, the web server automatically deserialize the raw JSON input provided in the request body.

```
curl -X 'POST' \
  'https://localhost:7099/api/Author/items?format=item' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "itemType": <Item_Data_type>,
  "itemContent": <Json_Item>
}'
```

13.4.4 Communication Service

Base Route: /api/communication

Description: The Communication Service manages Message Items used for communication within Processes.

Sample Endpoint: The GET request can be used to retrieve all the messages.

```
curl -X 'GET' \
  'https://localhost:7099/api/Communication/messages' \
  -H 'accept: text/plain'
```

13.4.5 Conversion Service

Base Route: /api/conversion

Description: The Conversion service provides endpoints for converting Items by updating their definition

Sample Endpoint: To update an existing BasicObject Item by its ID, a PUT request can be sent to the endpoint `/api/conversion/basic-objects/{basicObjectId}`. The `basicObjectId` is provided as a path parameter, and the updated BasicObject data is included in the request body.

```
url -X 'PUT' \
  'https://localhost:7099/api/Conversion/basic-objects/<Basic_object_ID>' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{<Modified_basic_object>}'
```

13.4.6 Discovery Service

Base Route: api/discovery

Description: This Service manages BasicDiscovery Item and provides access to discovery data.

Sample Endpoint: A GET request can be sent to retrieve a list of all available BasicDiscovery Items.

```
curl -X 'GET' \
  'https://localhost:7099/api/Discovery/basic-discoveries' \
  -H 'accept: text/plain'
```

13.4.7 Location Service

Base Route: /api/location

Description: The Location Service provides endpoints for managing both BasicMLocations and MLocations

Sample Endpoint: To delete a MLocation using its Id, a DELETE request can be sent to the api/Location/mlocations/<MLocationID> endpoint. The MLocationID of the MLocation to be removed is provided as a path parameter.

```
curl -X 'DELETE' \
  'https://localhost:7099/api/Location/mlocations/<MLocationID>' \
  -H 'accept: */*'
```

13.4.8 Marketplace Service

Base Route: /api/marketplace

Description: This Service manages marketplace assets.

Sample Endpoint:

A POST request can be sent to the api/Marketplace/asset endpoint. The new Asset Item is specified in the request body.

```
curl -X 'POST' \
  'https://localhost:7099/api/Marketplace/assets' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{<Asset>}'
```

13.4.9 Registration Service

Base Route: api/Registration

Description: This Service provides endpoints for managing user registration, account details, profiles, and avatars (i.e., personae). It supports operations like retrieving, creating, and updating accounts, personal profiles, and users. Additionally, it supports fetching and serving avatar files in different formats (GLB, PNG, JSON).

Sample Endpoint: A POST request can be sent to the api/Registration/accounts endpoint. The new Account Item is specified in the request body.

```
curl -X 'POST' \
  'https://localhost:7099/api/Registration/accounts' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{ <Account> }'
```

13.4.10 Right Service

Base Route: /api/Right

Description: The Right Service manages the rights associated with various Items such as Accounts, Basic Objects, MLocations, etc. It offers endpoints for CRUD operations on rights, as well as specific endpoints for associating rights with Processes, Items, and MLocations.

Sample Endpoint:

To retrieve all the rights associated with a User of a specific Account, it is possible to send a GET request to api/Right/accounts/<AccountId>/<UserId>/rights. The accountId and UserId are provided as path parameters.

```
curl -X 'GET' \  
  
'https://localhost:7099/api/Right/accounts/<AccountId>/<UserId>/rights' \  
  -H 'accept: */*' 
```

13.5 Transaction Service

Base Route: /api/Transaction

Description: The Transaction service handles the operations related to the Transaction and Wallet Items.

Sample Endpoint:

To update an existing Wallet item by its Id, a PUT request can be sent to the api/Transaction/wallets/<walletId> endpoint.

The walletId is provided as a path parameter, and the updated Wallet data is included in the request body.

```
curl -X 'PUT' \  
  'https://localhost:7099/api/Transaction/wallets/<walletId>' \  
  -H 'accept: */*' \  
  -H 'Content-Type: application/json' \  
  -d '{<Wallet>}' 
```

14 Reference Software

The MMM-TEC V2.0 Reference Software implements the [Friends meet in the metaverse](#) use case. It is released as Open-Source Software with the 3-Clause BSD licence. It relies on Unity and is composed of two software packages:

1. The [MPAI- MMM](#) implements the services and provides the MMM-API.
2. The [MPAI-MMM Unity](#) implements the MMM Server and Client.

Please send an email to [MPAI Secretariat](#) to access the code.