



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

MPAI Technical Specification

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

V2.1

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 [Notices & Disclaimers](#).

Technical Specification: MPAI Metaverse Model (MPAI-MMM) – Technologies (MMM-TEC) V2.1

1	Foreword	13
2	Introduction.....	15
3	Scope	16
4	Definitions.....	16
5	References	26
5.1	Normative reference	26
5.2	Informative references.....	26
6	Architecture and Operation.....	26
7	Processes	30
7.1	General	30
7.2	Process Types	30
7.2.1	App.....	30
7.2.2	Device.....	30
7.2.3	Service	30
7.2.4	User	31
7.2.5	Actions	32
8	Items.....	33
8.1	Simple Certificate.....	36
8.1.1	Definition	36
8.1.2	Functional Requirements.....	36
8.1.3	Syntax.....	36
8.1.4	Semantics	36
8.2	Simple Contract.....	36
8.2.1	Definition	36
8.2.2	Functional Requirements.....	36
8.2.3	Syntax.....	37
8.2.4	Semantics	37
8.3	Capabilities	38
8.3.1	Definition	38
8.3.2	Functional Requirements.....	38
8.3.3	Syntax.....	38
8.3.4	Semantics	38
8.4	Event.....	39
8.4.1	Definition	39
8.4.2	Functional Requirements.....	39
8.4.3	Syntax.....	40
8.4.4	Semantics	40
8.5	Identifier.....	40
8.5.1	Definition	40
8.5.2	Functional Requirements.....	40
8.5.3	Syntax.....	41
8.5.4	Semantics	41
8.6	License	41
8.6.1	Definition	41
8.6.2	Functional Requirements.....	41

8.6.3	Syntax.....	41
8.6.4	Semantics	41
8.7	M-Environment	42
8.7.1	Definition	42
8.7.2	Functional Requirements.....	42
8.7.3	Syntax.....	42
8.7.4	Semantics	42
8.8	Account	43
8.8.1	Definition	43
8.8.2	Functional Requirements.....	43
8.8.3	Syntax.....	43
8.8.4	Semantics	43
8.9	Activity Data.....	44
8.9.1	Definition	44
8.9.2	Functional Requirements.....	44
8.9.3	Syntax.....	44
8.9.4	Semantics	44
8.10	IPP Message	44
8.10.1	Definition	44
8.10.2	Functional Requirements.....	44
8.10.3	Syntax.....	45
8.10.4	Semantics	45
8.11	Message	46
8.11.1	Definition	46
8.11.2	Functional Requirements.....	46
8.11.3	Syntax.....	46
8.11.4	Semantics	46
8.12	Authentication.....	46
8.12.1	Definition	46
8.12.2	Functional Requirements.....	46
8.12.3	Syntax.....	47
8.12.4	Semantics	47
8.13	Basic Discovery.....	47
8.13.1	Definition	47
8.13.2	Functional Requirements.....	47
8.13.3	Syntax.....	48
8.13.4	Semantics	48
8.14	Basic Information.....	49
8.14.1	Definition	49
8.14.2	Functional Requirements.....	49
8.14.3	Syntax.....	49
8.14.4	Semantics	49
8.15	Basic Interpretation	50
8.15.1	Definition	50
8.15.2	Functional Requirements.....	50
8.15.3	Syntax.....	50
8.15.4	Semantics	50
8.16	Asset.....	51
8.16.1	Definition	51
8.16.2	Functional Requirements.....	51

8.16.3 Syntax.....	51
8.16.4 Semantics	51
8.17 Currency Object	51
8.17.1 Definition	51
8.17.2 Functional Requirements.....	52
8.17.3 Syntax.....	52
8.17.4 Semantics	52
8.18 Provenance	52
8.18.1 Definition	52
8.18.2 Functional Requirements.....	52
8.18.3 Syntax.....	52
8.18.4 Semantics	52
8.19 3D Model Object.....	53
8.19.1 Definition	53
8.19.2 Functional Requirements.....	53
8.19.3 Syntax.....	53
8.19.4 Semantics	53
8.19.5 Conformance Testing	54
8.20 3D Model Scene Descriptors.....	54
8.20.1 Definition	54
8.20.2 Functional Requirements.....	54
8.20.3 Syntax.....	54
8.20.4 Semantics	54
8.20.5 Conformance Testing	55
8.21 Audio Object.....	55
8.21.1 Definition	55
8.21.2 Functional Requirements.....	55
8.21.3 Syntax.....	56
8.21.4 Semantics	56
8.21.5 Conformance Testing	57
8.22 Audio Scene Descriptors.....	57
8.22.1 Definition	57
8.22.2 Functional Requirements.....	57
8.22.3 Syntax.....	57
8.22.4 Semantics	57
8.22.5 Conformance Testing	58
8.23 Audio-Visual Object	58
8.23.1 Definition	58
8.23.2 Functional Requirements.....	58
8.23.3 Syntax.....	58
8.23.4 Semantics	58
8.23.5 Conformance Testing	59
8.24 Audio-Visual Scene Descriptors	59
8.24.1 Definition	59
8.24.2 Functional Requirements.....	60
8.24.3 Syntax.....	60
8.24.4 Semantics	60
8.24.5 Conformance Testing	61
8.25 Speech Object.....	61
8.25.1 Definition	61

8.25.2	Functional Requirements.....	61
8.25.3	Syntax.....	61
8.25.4	Semantics	61
8.25.5	Conformance Testing	62
8.26	Speech Scene Descriptors	62
8.26.1	Definition	62
8.26.2	Functional Requirements.....	62
8.26.3	Syntax.....	62
8.26.4	Semantics	63
8.26.5	Conformance Testing	63
8.27	Visual Object.....	63
8.27.1	Definition	63
8.27.2	Functional Requirements.....	63
8.27.3	Syntax.....	64
8.27.4	Semantics	64
8.27.5	Conformance Testing	65
8.28	Visual Scene Descriptors	65
8.28.1	Definition	65
8.28.2	Functional Requirements.....	65
8.28.3	Syntax.....	65
8.28.4	Semantics	65
8.28.5	Conformance Testing	66
8.29	Audio Source.....	66
8.29.1	Definition	66
8.29.2	Functional Requirements.....	66
8.29.3	Syntax.....	66
8.29.4	Semantics	66
8.29.5	Conformance Testing	67
8.30	Object Audio Characteristics	67
8.30.1	Definition	67
8.30.2	Functional Requirements.....	67
8.30.3	Syntax.....	68
8.30.4	Semantics	68
8.30.5	Conformance Testing	68
8.31	Text Object.....	69
8.31.1	Definition	69
8.31.2	Functional Requirements.....	69
8.31.3	Syntax.....	69
8.31.4	Semantics	69
8.31.5	Conformance Testing	70
8.32	Cognitive State.....	70
8.32.1	Definition	70
8.32.2	Functional Requirements.....	70
8.32.3	Syntax.....	72
8.32.4	Semantics	72
8.32.5	Conformance Testing	73
8.33	Emotion.....	73
8.33.1	Definition	73
8.33.2	Functional Requirements.....	73
8.33.3	Syntax.....	76

8.33.4	Semantics	76
8.33.5	Conformance Testing	76
8.34	Basic M-Location	77
8.34.1	Definition	77
8.34.2	Functional Requirements.....	77
8.34.3	Syntax.....	77
8.34.4	Semantics	77
8.35	Simple Time	78
8.35.1	Definition	78
8.35.2	Functional Requirements.....	78
8.35.3	Syntax.....	78
8.35.4	Semantics	78
8.35.5	Conformance Testing	78
8.36	Basic U-Location.....	78
8.36.1	Definition	78
8.36.2	Functional Requirements.....	79
8.36.3	Syntax.....	79
8.36.4	Semantics	79
8.37	M-Location.....	79
8.37.1	Definition	79
8.37.2	Functional Requirements.....	79
8.37.3	Syntax.....	80
8.37.4	Semantics	80
8.38	Orientation.....	80
8.38.1	Definition	80
8.38.2	Functional Requirements.....	80
8.38.3	Syntax.....	81
8.38.4	Semantics	81
8.38.5	Conformance Testing	81
8.39	Point of View	81
8.39.1	Definition	81
8.39.2	Functional Requirements.....	82
8.39.3	Syntax.....	82
8.39.4	Semantics	82
8.39.5	Conformance Testing	82
8.40	Certificate Object	83
8.40.1	Definition	83
8.40.2	Functional Requirements.....	83
8.40.3	Syntax.....	83
8.40.4	Semantics	83
8.41	Contract Object	83
8.41.1	Definition	83
8.41.2	Functional Requirements.....	83
8.41.3	Syntax.....	84
8.41.4	Semantics	84
8.42	M-Instance.....	84
8.42.1	Definition	84
8.42.2	Functional Requirements.....	84
8.42.3	Syntax.....	84
8.42.4	Semantics	84

8.43	Program Object	85
8.43.1	Definition	85
8.43.2	Functional Requirements.....	85
8.43.3	Syntax.....	85
8.43.4	Semantics	85
8.44	Right.....	86
8.44.1	Definition	86
8.44.2	Functional Requirements.....	86
8.44.3	Syntax.....	86
8.44.4	Semantics	86
8.45	Rule	87
8.45.1	Definition	87
8.45.2	Functional Requirements.....	87
8.45.3	Syntax.....	87
8.45.4	Semantics	87
8.46	U-Environment.....	88
8.46.1	Definition	88
8.46.2	Functional Requirements.....	88
8.46.3	Syntax.....	88
8.46.4	Semantics	88
8.47	Personal Profile	88
8.47.1	Definition	88
8.47.2	Functional Requirements.....	88
8.47.3	Syntax.....	88
8.47.4	Semantics	89
8.48	Personal Data.....	89
8.48.1	Definition	89
8.48.2	Functional Requirements.....	89
8.48.3	Syntax.....	89
8.48.4	Semantics	90
8.49	Process Action.....	90
8.49.1	Definition	90
8.49.2	Functional Requirements.....	91
8.49.3	Syntax.....	91
8.49.4	Semantics	91
8.50	Resolution.....	92
8.50.1	Definition	92
8.50.2	Functional Requirements.....	93
8.50.3	Syntax.....	93
8.50.4	Semantics	93
8.51	Validation.....	94
8.51.1	Definition	94
8.51.2	Functional Requirements.....	94
8.51.3	Syntax.....	94
8.51.4	Semantics	94
8.52	Discovery Object.....	95
8.52.1	Definition	95
8.52.2	Functional Requirements.....	95
8.52.3	Syntax.....	95
8.52.4	Semantics	95

8.53	Information Object	96
8.53.1	Definition	96
8.53.2	Functional Requirements.....	96
8.53.3	Syntax.....	96
8.53.4	Semantics	96
8.54	Interpretation Object	96
8.54.1	Definition	96
8.54.2	Functional Requirements.....	97
8.54.3	Syntax.....	97
8.54.4	Semantics	97
8.55	Transaction	97
8.55.1	Definition	97
8.55.2	Functional Requirements.....	97
8.55.3	Syntax.....	98
8.55.4	Semantics	98
8.56	Value	99
8.56.1	Definition	99
8.56.2	Functional Requirements.....	99
8.56.3	Syntax.....	99
8.56.4	Semantics	99
8.57	Wallet	99
8.57.1	Definition	99
8.57.2	Functional Requirements.....	99
8.57.3	Syntax.....	100
8.57.4	Semantics	100
8.58	3D Model Event Description.....	100
8.58.1	Definition	100
8.58.2	Functional Requirements.....	100
8.58.3	Syntax.....	100
8.58.4	Semantics	100
8.58.5	Conformance Testing	101
8.58.6	Performance Assessment.....	Error! Bookmark not defined.
8.59	3D Model Scene Geometry	101
8.59.1	Definition	101
8.59.2	Functional Requirements.....	101
8.59.3	Syntax.....	101
8.59.4	Semantics	101
8.59.5	Conformance Testing	102
8.59.6	Performance Assessment.....	Error! Bookmark not defined.
8.60	Audio Event Descriptors	102
8.60.1	Definition	102
8.60.2	Functional Requirements.....	102
8.60.3	Syntax.....	102
8.60.4	Semantics	102
8.60.5	Conformance Testing	103
8.60.6	Performance Assessment.....	Error! Bookmark not defined.
8.61	Audio Scene Geometry	103
8.61.1	Definition	103
8.61.2	Functional Requirements.....	103
8.61.3	Syntax.....	103

8.61.4	Semantics	103
8.61.5	Conformance Testing	104
8.61.6	Performance Assessment.....	Error! Bookmark not defined.
8.62	Audi-Visual Event Descriptors	104
8.62.1	Definition	104
8.62.2	Functional Requirements.....	104
8.62.3	Syntax.....	104
8.62.4	Semantics	104
8.62.5	Conformance Testing	105
8.62.6	Performance Assessment.....	Error! Bookmark not defined.
8.63	Audio-Visual Scene Geometry.....	105
8.63.1	Definition	105
8.63.2	Functional Requirements.....	105
8.63.3	Syntax.....	105
8.63.4	Semantics	105
8.63.5	Conformance Testing	106
8.63.6	Performance Assessment.....	Error! Bookmark not defined.
8.64	Speech Event Descriptors.....	106
8.64.1	Definition	106
8.64.2	Functional Requirements.....	106
8.64.3	Syntax.....	106
8.64.4	Semantics	106
8.64.5	Conformance Testing	107
8.64.6	Performance Assessment.....	Error! Bookmark not defined.
8.65	Speech Scene Geometry	107
8.65.1	Definition	107
8.65.2	Functional Requirements.....	107
8.65.3	Syntax.....	107
8.65.4	Semantics	107
8.65.5	Conformance Testing	108
8.65.6	Performance Assessment.....	Error! Bookmark not defined.
8.66	Visual Event Descriptors.....	108
8.66.1	Definition	108
8.66.2	Functional Requirements.....	108
8.66.3	Syntax.....	108
8.66.4	Semantics	108
8.66.5	Conformance Testing	109
8.66.6	Performance Assessment.....	Error! Bookmark not defined.
8.67	Visual Scene Geometry	109
8.67.1	Definition	109
8.67.2	Functional Requirements.....	109
8.67.3	Syntax.....	109
8.67.4	Semantics	109
8.67.5	Conformance Testing	110
8.67.6	Performance Assessment.....	Error! Bookmark not defined.
8.68	Light Source	110
8.68.1	Definition	110
8.68.2	Functional Requirements.....	110
8.68.3	Syntax.....	110
8.68.4	Semantics	110

8.68.5	Conformance Testing	111
8.68.6	Performance Assessment.....	Error! Bookmark not defined.
8.69	R-Item.....	111
8.69.1	Definition	111
8.69.2	Functional Requirements.....	111
8.69.3	Syntax.....	112
8.69.4	Semantics	112
8.70	Personal Status	112
8.70.1	Definition	112
8.70.2	Functional Requirements.....	112
8.70.3	Syntax.....	113
8.70.4	Semantics	113
8.70.5	Conformance Testing	114
8.70.6	Performance Assessment.....	Error! Bookmark not defined.
8.71	Social Attitude.....	114
8.71.1	Definition	114
8.71.2	Functional Requirements.....	114
8.71.3	Syntax.....	120
8.71.4	Semantics	120
8.71.5	Conformance Testing	121
8.71.6	Performance Assessment.....	Error! Bookmark not defined.
8.72	Position.....	121
8.72.1	Definition	121
8.72.2	Functional Requirements.....	121
8.72.3	Syntax.....	121
8.72.4	Semantics	121
8.72.5	Conformance Testing	122
8.72.6	Performance Assessment.....	Error! Bookmark not defined.
8.73	Space-Time.....	122
8.73.1	Definition	122
8.73.2	Functional Requirements.....	122
8.73.3	Syntax.....	122
8.73.4	Semantics	123
8.73.5	Conformance Testing	123
8.73.6	Performance Assessment.....	Error! Bookmark not defined.
8.74	Spatial Attitude.....	123
8.74.1	Definition	123
8.74.2	Functional Requirements.....	123
8.74.3	Syntax.....	123
8.74.4	Semantics	123
8.74.5	Conformance Testing	124
8.74.6	Performance Assessment.....	Error! Bookmark not defined.
8.75	Time	124
8.75.1	Definition	124
8.75.2	Functional Requirements.....	124
8.75.3	Syntax.....	124
8.75.4	Semantics	124
8.75.5	Conformance Testing	125
8.75.6	Performance Assessment.....	Error! Bookmark not defined.
8.76	U-Location	125

8.76.1	Definition	125
8.76.2	Functional Requirements.....	125
8.76.3	Syntax.....	125
8.76.4	Semantics	125
8.77	Universe-Metaverse Map	126
8.77.1	Definition	126
8.77.2	Functional Requirements.....	126
8.77.3	Syntax.....	126
8.77.4	Semantics	126
9	Process Actions	127
9.1	Introduction.....	127
9.2	Process Actions	127
9.2.1	Authenticate	129
9.2.2	Discover	130
9.2.3	Identify.....	131
9.2.4	MM-Animate.....	133
9.2.5	MM-Send	134
9.2.6	MU-Animate	136
9.2.7	Definition	136
9.2.8	MU-Send	138
9.2.9	Register.....	139
9.2.10	Transact	140
9.2.11	Validate	142
9.2.12	Author.....	143
9.2.13	Execute.....	144
9.2.14	Inform.....	146
9.2.15	MM-Add.....	147
9.2.16	Modify.....	149
9.2.17	MU-Add	150
9.2.18	Post.....	152
9.2.19	Resolve	153
9.2.20	UM-Capture	154
9.2.21	Convert.....	156
9.2.22	Hide.....	157
9.2.23	Interpret.....	158
9.2.24	MM-Move	160
9.2.25	MU-Actuate.....	161
9.2.26	MU-Move.....	163
9.2.27	Property Change.....	164
9.2.28	Rights Change	166
9.2.29	UM-Send	168
9.3	Sequences of Process Actions	169
9.4	Backus Naur Form	170
10	Protocols.....	171
10.1	Inter-Process Protocol (same M-Instance)	171
10.2	Inter-Process Protocol (different M-Instances).....	172
10.3	Inter-Process Protocol (multiple M-Instances)	173
10.4	Inter-Process Protocol elements	173
10.5	Posting Protocol	174
10.6	Licensing Protocol.....	174

11	Profiles	175
11.1	Introduction.....	175
11.2	Profile structure	176
11.3	Baseline Profile	177
11.4	Finance Profile	177
11.5	Management Profile	178
11.6	igh Profile	179
12	Verification Uses Cases	180
12.1	Introduction.....	180
12.2	Friends meet in the metaverse	180
12.2.1	Description	180
12.2.2	Variables.....	181
12.2.3	Workflow	182
12.3	Virtual lecture.....	184
12.3.1	Description	184
12.3.2	Variables.....	185
12.3.3	Workflow	186
12.4	Hybrid working	188
12.4.1	Description	188
12.4.2	Variables.....	189
12.4.3	Workflow	190
12.5	eSports Tournament	191
12.5.1	Description	191
12.5.2	Variables.....	191
12.5.3	Workflow	192
12.6	Virtual Performance	193
12.6.1	Description	193
12.6.2	Variables.....	193
12.6.3	Workflow	194
12.7	AI Tourist Guide	195
12.7.1	Description	195
12.7.2	Variables.....	196
12.7.3	Workflow	197
12.8	Virtual Dance School	198
12.8.1	Description	198
12.8.2	Variables.....	198
12.8.3	Workflow	199
12.9	Virtual Car Showroom	200
12.9.1	Description	200
12.9.2	Variables.....	201
12.9.3	Workflow	202
12.10	Meeting while driving	203
12.10.1	Description	203
12.10.2	Variables.....	204
12.10.3	Workflow	205
12.11	Co-design across metaverses.....	207
12.11.1	Description	207
12.11.2	Variables.....	207
12.11.3	Workflow	208
12.12	Selling assets on a Marketplace	209

12.12.1	Description	209
12.12.2	Variables.....	209
12.12.3	Workflow	210
12.13	Emergency in Industrial Metaverse.....	211
12.13.1	Description	211
12.13.2	Variables.....	211
12.13.3	Workflow	211
13	MMM-API	212
13.1	Introduction.....	212
13.2	Description	212
13.3	Components.....	213
13.4	Paths.....	213
13.4.1	Activity Service.....	213
13.4.2	Authentication Service	213
13.4.3	Author Service.....	214
13.4.4	Communication Service	214
13.4.5	Conversion Service	215
13.4.6	Discovery Service	215
13.4.7	Location Service.....	215
13.4.8	Marketplace Service	216
13.4.9	Registration Service	216
13.4.10	Rights Service	216
13.4.11	Transaction Service	217
14	Reference Software	217

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 four 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 AI applications 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.

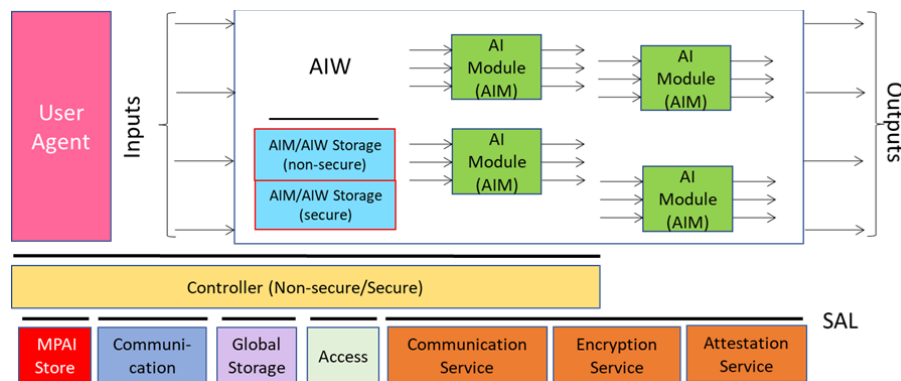


Figure 1 – The AI Framework (MPAI-AIF) V2 Reference Model

3. [Technical Specification: Data Types, Formats, and Attributes \(MPAI-TFA\) V1.4](#) 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.
4. [Technical Specification: Governance of the MPAI Ecosystem \(MPAI-GME\) V2.0](#) defines the following elements:

1. Standards, i.e., the ensemble of Technical Specifications, Reference Software, Conformance Testing, and Performance Assessment.
2. Developers of MPAI-specified AIMs and Integrators of MPAI-specified AIWS (Implementers).
3. MPAI Store in charge of making AIMs and AIWs submitted by Implementers available to Integrators and End Users.
4. Performance Assessors, independent entities assessing the performance of implementations in terms of Reliability, Replicability, Robustness, and Fairness.
5. End Users.

The interaction between and among actors of the MPAI Ecosystem are depicted in Figure 2.

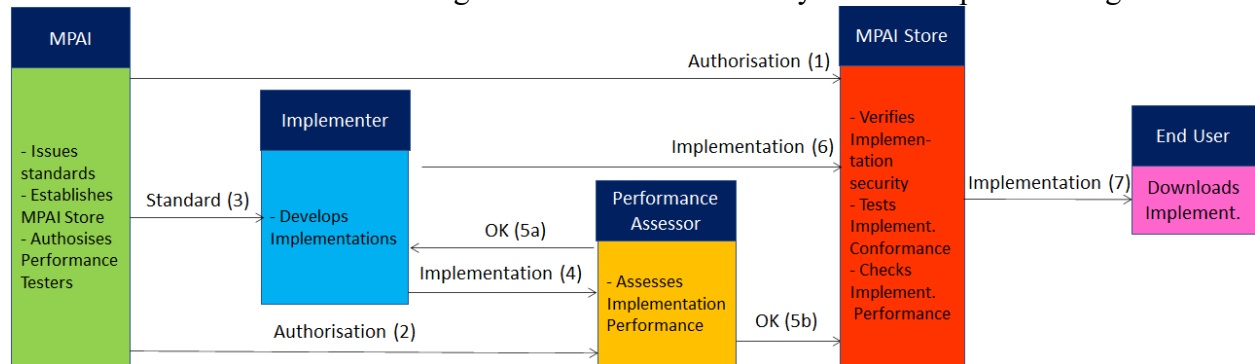


Figure 2 – The MPAI Ecosystem

2 Introduction

(Informative)

Despite being undefined and loose, many consider the "metaverse" concept as one of the most promising evolutionary steps of Information and Communication Technology. Indeed, a significant number of implementations that can be classified as "metaverse" have been developed. Unfortunately, metaverse designers and developers have generally made technology decisions that best responded to their vision and needs, with little or no consideration of the primary requirements of a communication system to interoperate with other instances. In response to concerns that such metaverse "walled gardens" do not fully exploit the opportunities offered by current and expected technologies, MPAI has published Technical Reports and developed Technical Specifications that provide exactly that functionality:

1. **Technical Report: MPAI Metaverse Model (MPAI-MMM) – Functionalities (MMM-FNC) V1.0** 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 (MMM-FPR) V1.0** revises an extended list of definitions; introduces 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, and 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) V1.1** specifies the Functional Requirements of Processes and Actions providing the means to achieve Interoperability between two or more independently

designed M-Instances executing Processes, and producing 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) V1.0** 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) V2.1** - is the latest version of the MMM-TEC standard. It provides an integrated metaverse specification enabling interoperability of Processes, Items, Process Actions, and Protocols between metaverse instances (M-Instance) and between M-Instances and clients.

In all Chapters and Sections, Terms beginning with a capital letter are defined in [Table 1](#). 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) V2.1 - in this and other MPAI documents also called MMM-TEC V2.1 or MMM-TEC:

1. Specifies the technologies enabling an M-Instance to interoperate with clients and M-Instances independently designed and implemented in conformity with MMM-TEC, namely:
 1. The *Processes* operating in an M-Instance that capture Data from the Universe and actuate Items in the Universe as R-Items.
 2. The *Items*, i.e., the Data Types with their Qualifiers recognised in an M-Instance.
 3. The *Actions* and *Process Actions* that a Process performs on Items. and Processes
 4. The *Protocols* enabling a Process to communicate with other Processes.
 5. The MPAI Metaverse Model *Profiles*.
 6. The MPAI-MMM API.
2. Makes available the MMM-TEC Reference Software demonstrating the operation of an M-Instance in one of the twelve Validation Use Cases included in MMM-TEC.

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) V1.4** 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 or 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 – 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: 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>Author</i>	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>Change</i>	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>Convert</i>	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>Discover</i>	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>Execute</i>	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>Hide</i>	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>Identify</i>	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>Inform</i>	The Action of a User requesting an Information Service to: 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>Interpret</i>	The Action of a User requesting an Interpretation Service to: 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>MM-Add</i>	The Action of a User requesting a Location Service to: 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>MM-Anim</i>	The Action of a User requesting that a Location Service: 1. Animate an MM-Added or MM-Embedded Item with a Stream or Command Item. 2. Grant Rights based on Model Rights to the MM-Animated Item.
- <i>MM-Disable</i>	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>MM-Embed</i>	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>MM-Enable</i>	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>MM-Move</i>	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>MM-PropertyChange</i>	The Action of a User requesting a Location Service to modify the characteristics of an Item: 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.
- <i>MM-Send</i>	The Action of a Process requesting a Communication Service: 1. To send a Message to other Processes. 2. To grant Rights to the receiving Process on the content of the Message.
- <i>Modify</i>	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>MU-Actuate</i>	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>MU-Send</i>	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>Post</i>	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>Register</i>	The Action of a human requesting that a Registration Service: 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.
- <i>Resolve</i>	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>Transact</i>	The Action of a User₁ ("sender") requesting that a Transaction Service: 1. Assign Rights on an Asset to User₂ ("receiver"). 2. Cause: 2.1. Wallet₁ of User₁ to be increased by Value₁. 2.2. Wallet₂ of User₂ to be decreased by Value₂. 2.3. Wallet₃ of the Service enabling/facilitating the Transaction to be increased by Value₃ (optionally).
- <i>UM-Capture</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>UM-Send</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>Validate</i>	The Action of a Process requesting a Validation Service to confirm that a Process hold claimed Rights.
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.
Command	A Item used to effect instantaneous changes to an Item or to an R-Item via an MM-Anim or MU-Anim Process Action (see Stream).
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:
- <i>Brain-Computer</i>	(BCI) by sensing and processing the electrical activity of the brain.
- <i>Haptic</i>	through bodily movements and sensations.
- <i>Speech</i>	using spoken language.
- <i>Visual</i>	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.
- <i>Account</i>	An Item that - Uniquely references a human registered with an M-Instance - Includes the IDs of the human's Personal Profile, Processes, and their Internal Rights.
- <i>Activity Data</i>	An Item that records the IPP Protocols executed by a Process.
- <i>Asset</i>	An Item that can be Transacted.
- <i>Authentication</i>	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.
- <i>Basic Certificate</i>	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.
- <i>Basic Discovery</i>	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.

- <i>Basic Information</i>	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.
- <i>Basic Interpretation</i>	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.
- <i>Basic M-Location</i>	An Item representing a region of an M-Instance with Space-Time attributes that is not exposed as further subdivided as a Location.
- <i>Basic Object</i>	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).
- <i>Basic Scene Descriptors</i>	An Item representing the Objects of a Basic Scene and their arrangement in the Scene.
- <i>Basic Scene Geometry</i>	An Item representing the spatial arrangement of the Objects in a Basic Scene.
- <i>Basic U-Location</i>	An Item representing a region of the Universe with Space-Time attributes that is not (exposed as) further subdivided.
- <i>Certificate</i>	An Item Process attesting the suitability of a Process for specific usages that is not a Basic Certificate.
- <i>Cognitive State</i>	An Item representing the Personal Status Factor representing the internal state of an Entity such as “surprised” or “interested”.
- <i>Contract</i>	An Item representing terms and conditions or a Program that is executed according to certain terms when conditions are met.
- <i>Currency</i>	An Item representing the unit of measure of an Amount of a Value.
- <i>Discovery</i>	An Item representing the description of the Item or Process (or their IDs) to be Discovered.
- <i>Emotion</i>	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”.
- <i>Event Descriptors</i>	An Item representing the series of Scene Descriptors from start and end time.
- <i>Identifier</i>	An Item that references only one Process or Item in an M-Instance.
- <i>Information</i>	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.
- <i>Interpretation</i>	An Item sent by a Process to a Service requesting interpretation of an Item (or its ID) and the response of the Service.
- <i>IPP Message</i>	An Item that a Source Process sends to a Destination Process requesting the performance of a Process Action.
- <i>M-Capabilities</i>	An Item representing the capabilities of an M-Instance or M-Environment.
- <i>M-Environment</i>	An Identified administrative subset of an M-Instance.

- <i>Message</i>	An Item that a Source Process MM-Sends to a Communication Service requesting it to deliver it to a Destination Process.
- <i>M-Instance</i>	An Item representing the Virtual Space created according to the MMM-TEC Technical Specification of the MPAI-Metaverse Model.
- <i>M-Location</i>	An Item representing a region of an M-Instance with Space-Time attributes that is (exposed as) further subdivided.
- <i>Object</i>	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.
- <i>Orientation</i>	An Item representing an Object's orientation, orientation velocity, and orientation acceleration.
- <i>P-Capabilities</i>	An Items containing the list of Processes and Actions that it can perform.
- <i>Personal Data</i>	An Item containing a human's Personal Profile and Activity Data of their Users.
- <i>Personal Profile</i>	An Item containing a human's Personal Data submitted when Registering with an M-Instance.
- <i>Personal Status</i>	An Item representing the information internal to an Entity that characterises their behaviour.
- <i>Point of View</i>	An Item representing the Position and Orientation of an Object in a Virtual Environment excluding velocity and acceleration.
- <i>Position</i>	An Item representing an Object's position, velocity, and acceleration.
- <i>Process Action</i>	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.
- <i>Program</i>	An Item containing executable code, e.g., a Process or a Contract that is a Program.
- <i>Provenance</i>	An Item containing the list of all Transactions executed on an Asset, first and last included.
- <i>Resolution</i>	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.
- <i>Rights</i>	An Item representing the set of Process Actions and the corresponding Levels that a Process may perform.
- <i>Rules</i>	An Item containing the set of Rights that Processes May, May not, or Must exercise in the M-Instance: <i>May</i> : A Process is allowed to exercise the Rights. <i>May not</i> : A Process is not allowed to exercise the Rights. <i>Must</i> : A Process must exercise the Rights.
- <i>Scene Descriptors</i>	An Item including the Objects of a Scene, the Sub-Scenes, and their arrangement in the Scene.
- <i>Scene Geometry</i>	An Item representing the arrangement of the Objects of a Scene that may include Objects and Sub-Scenes in a hierarchical fashion.

- <i>Social Attitude</i>	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”.
- <i>Space-Time</i>	An Item representing the Spatial Attitude and Time information.
- <i>Spatial Attitude</i>	An Item representing the Position and Orientation of an Object, and their velocities and accelerations.
- <i>Summary</i>	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.
- <i>Time</i>	An Item representing the start time and end time of a duration.
- <i>Transaction</i>	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.
- <i>U-Environment</i>	An Item representing an Identified subset of the Universe.
- <i>U-Location</i>	An Item representing a region of the Universe with Space-Time attributes.
- <i>Universe-Metaverse Map</i>	An Item representing a list of U-Locations and corresponding M-Locations and/or Items with their Spatial Attitudes.
- <i>Validation</i>	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.
- <i>Value</i>	An Item combining an Amount and a Currency.
- <i>Wallet</i>	A container of Values.
Layer	
- <i>Enabling Service</i>	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	
- <i>Interoperability</i>	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.
- Device	A Process able to: 1. UM-Capture Data from a U-Location 2. UM-Send Data and Metadata to a User. and/or 1. MM-Send an Entity from an M-Location to the Device. 2. MU-Render an Entity at a U-Location.
- 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.

Render	The action of creating an R-Item.
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.
Stream	A Item used to effect continuous changes to an Item or to an R-Item via an MM-Anim or MU-Anim Process Action (see Command).
Token	
- Fungible	A representation of an Asset that is interchangeable with other Assets of the same type.
- Non-Fungible	(NFT) A unique digital identifier of an Asset that: - Cannot be copied (i.e., a copy is known to be a copy), substituted, or subdivided. - Is recorded in a digital ledger. - Is used to certify Object authenticity and ownership.
Trust-less system	A system allowing a User to make reliable Transactions without trusting or knowing the parties the User makes Transactions with.
Twin	
- Analogue	The replica of an Item to the Universe as an R-Item.
- Digital	The replica of an object or scene to the Metaverse as an Item.
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	
- Decentralised	An Identifier that enables the verifiable association with a User without requiring a centralised registry.
- Self-Sovereign	A Decentralised Identifier derived from the User's Public Key owned and managed directly by the User based on the knowledge of their own Private Key, e.g., stored in the Crypto Wallet enabled by the Blockchain underpinning the Metaverse Instance.
Wallet	
- Crypto	Software or hardware holding the Public and Private Keys of a User to enable them to make Transactions by accessing their Account on a Blockchain.

5 References

5.1 Normative reference

1. MPAI; Technical Specification; [Multimodal Conversation](#) (MPAI-MMC) V2.4.
2. MPAI; Technical Specification: [Object and Scene Description](#) (MPAI-OSD) V1.4.
3. MPAI; Technical Specification: [Portable Avatar Format](#) (MPAI-PAF) V1.5.
4. MPAI; Technical Specification: [Profiles](#) (MPAI-PRF) - [Modules](#) (PRF-AIM) V1.1.
5. MPAI; Technical Specification: [Data Types, Formats, and Attributes](#) (MPAI-TFA) V1.4.
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) V1.0.
10. MPAI; Technical Report – [MPAI Metaverse Model](#) (MPAI-MMM) – [Functionality Profiles](#) (MMM-FPR) V1.0.
11. MPAI; Technical Specification: [Governance of the MPAI Ecosystem](#) (MPAI-GME) V2.0.
12. MPAI; Technical Specification: [Artificial Intelligence Framework](#) (MPAI-AIF) V2.2.
13. MPAI; Technical Specification – [Connected Autonomous Vehicle](#) (MPAI-CAV) – [Technologies](#) (CAV-TEC) V1.0.
14. MPAI; [Technical Specification; Human and Machine Communication](#) (MPAI-HMC) V2.1.
15. MPAI; Framework Licence: [MPAI Metaverse Model](#) (MPAI-MMM) – [Technologies](#).

6 Architecture and Operation

(Informative)

This chapter introduces some of the basic design and operation principles on which the MMM-TEC specification is based. It is informative, but it is highly recommended to review it, as it includes elements that are important for proper understanding of the MMM-TEC specification. A metaverse instance ([M-Instance](#)) is an Information and Communication Technologies (ICT) platform implementing the MMM-TEC specification. Its main feature is to be populated by [Processes](#) that:

1. *May be imported* (UM-Sent) by a human [Registering](#) with the M-Instance. The human
 1. [Requests](#) to open an [Account](#) of a certain class.
 2. [May be requested](#) to provide their [Personal Profile](#) and perform a [Transaction](#).
 3. [Obtains](#) set of Rights that their Processes may exercise.
 4. May need to have their Processes Certified for them to be imported.
2. *Operate* with various degrees of autonomy and interactivity under the responsibility of the M-Instance Manager, Third-Party Service Providers, or humans who reside in the *Universe*, i.e., the real world.
3. *Perform* [Actions](#) on Items and/or Processes if permitted by the [Rule](#) of the M-Instance.
4. The name of an [Action](#) may begin with:

1. *MM*: to indicate Actions performed *inside* the M-Instance, e.g., [MM-Animate](#) using a stream or a command to animate a [3D Model Object](#) with a [Spatial Attitude](#) (defined as [Position](#), [Orientation](#), and their velocities and accelerations).
2. *MU*: to indicate Actions *originated in the M-Instance* but influencing the Universe, e.g., [MU-Add](#) to place a physical object ([R-Item](#)) at a [U-Location](#) with a Spatial Attitude or to [MU-Move](#) it from a U-Location to another U-Location along a [Trajectory](#).
3. *UM*: to indicate Actions originated in the Universe and influencing the M-Instance, e.g., [UM-Capture](#) to acquire Data by capturing a scene or an object at a U-Location using a Qualifier.
5. *Perform Process Action* if the Action is requested to another Process, either on their initiative, or driven by the actions of humans or machines in the Universe. A Process Action is expressed by: deontic verb, [Action](#) - set of Complements (*Nil/At/From/To/With*) - [Item](#) or ProcessID) *If Event* where:
 1. Deontic verbs are *May*, *May Not*, and *Must* corresponding to Permission, Prohibition, or Obligation.
 2. The Complements: (e.g., Nil, At, From, To With, etc.) are applied to:
 1. [Items](#) - i.e., Data that has been [Identified](#) in, and thus recognised by, the M-Instance on which the Process Action is performed or is required for the Action to be performed, such as [Asset](#), [Audio Object](#), [Audio-Visual Scene](#), and where ([M-Location](#) and/or [U-Location](#)).
 2. [Process](#) on which the Action is performed.
 3. [Items](#) are Data Type instances Identified in the M-Instance.
 4. [Event](#) is a [Process Action](#) or a logic combination thereof and Time whose performance triggers the execution of a Process Action.
6. *May hold Rights* on Items or Processes, i.e., may perform the set of Process Actions that are listed as Rights. The notion of Rights may also be applied to an Item to signal which Processes may perform which Processes Actions on it. Rights have *Levels* indicating that 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 or Item by a Process.
7. *May request* another Process to perform the Process Actions using the [Inter-Process Protocol](#), possibly after Transacting a [Value](#) (i.e., an *Amount* in a [Currency](#)) to a Wallet, in case the Process cannot (does not have the technology) or may not (does not have the Rights) to perform the Process Actions. The requested Process response consists of one of more Complements (*Nil/At/From/To/With*) - Item or ProcessID and PA Status, where:
 1. The first part conveys the result of the executed Process Action Request, and
 2. PA Status reflects success or failure in the Process Action Request execution.
8. May expose its capabilities ([Capabilities](#)). [Capabilities](#) may also be used to expose the capabilities of an M-Instance.
9. *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.

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. These are some of the functions that may be relevant to an M-Instance implementation and may be retained by an M-Instance Manager, e.g.:

1. Management of the M-Instance.
2. Establishment of M-Instance Rules.
3. Installation and maintenance of Services.
4. Definition of Account Types.
5. Monitoring of Process Actions
6. Sanctions to infringers.
7. Resolution of conflicts.
8. Certification of Processes and Data Types.

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

In some cases, an M-Instance could be wastefully too costly as an undertaking if all the MMM-TEC-specified technologies were mandatorily to be implemented, even if the particular M-Instance had limited scope. MMM-TEC specifies [Profiles](#) to facilitate the implementations of conforming M-Instances that are unduly burdened by the requirements of other application domains.

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 defines four Profiles (see Figure 1):

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.
5. 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 to verify that the currently specified Actions and Items support those Use Cases.

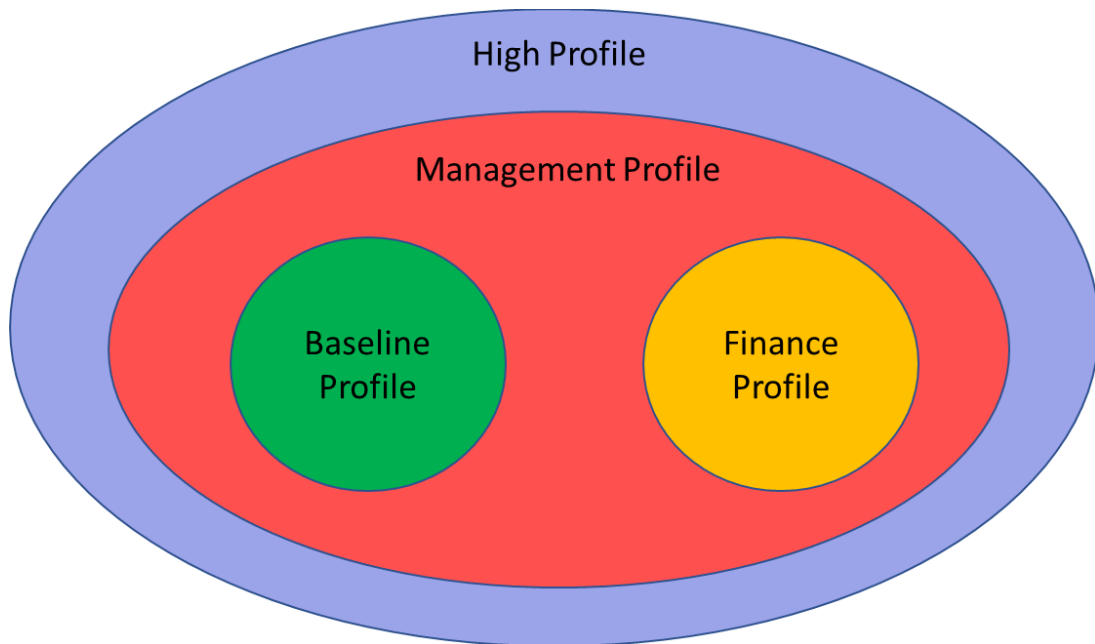


Figure 1 - MMM-TEC V2.0 Profiles

Figure 2 gives a summary view of some of the basic MMM-TEC elements.

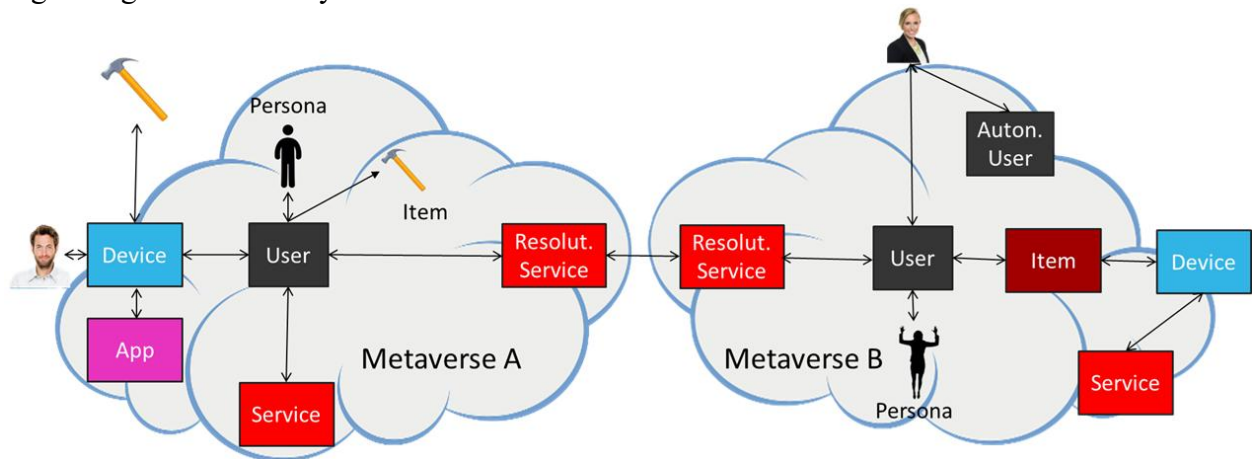


Figure 2 - Main elements of an M-Instance

x\

The fast development of certain technological areas is one of the issues that has so far prevented the development of metaverse standards. MMM-TEC deals with this issue by providing JSON Schemas and semantics for all Items. A JSON Schema may reference [Qualifiers](#), MPAI-defined Data Types that provide additional information to the Data in the form of:

1. *Sub-Type* (e.g., which colour space is used in a Visual Data Type).
2. *Format* (e.g., which compression or file/streaming format is used in a Speech Data Type).
3. *Attributes* (e.g., which Binaural Cues are used in an Audio Data Type).

For instance, a Process receiving an Object can understand from the Qualifier referenced in the Object's JSON data whether it has the required technology to process it, or else it has to Convert it to obtain a version of the Object matching its [Capabilities](#). This approach helps prolong the life of the MMM-TEC specification as in many cases only the Qualifier specification will need to be updated and not the MMM-TEC specification.

7 Processes

7.1 General

A **Process** is an instance of a [Program Object](#) running on the Information and Communication Technology (ICT) platform underpinning an M-Instance or on a device connected to it.

A Process may perform one of more MPAI-specified [Actions](#), e.g.:

1. Device "*UM-Captures scene At U-Location*" means that a named Device captures the scene at a named U-Location and represents it as Data.
2. User "*MM-Adds Persona At M-Location*" means that a named User places a named Persona at a named M-Location.
3. User "*MM-Animates Persona With Stream*" means that a named User animates a Persona MM-Added at a named M-Location with a named Stream or Command Item.

Process [Capabilities](#) describe the [Actions](#), and the [Items](#) a Process can perform, and the Metadata characterising the Process.

A Process may be unable to perform an Action on an Item because it does not:

1. Support the Item's [Qualifier](#). In this case it may request a Conversion Service to [Convert](#) the Item to a new Item with appropriate Sub-Types, Formats and Attributes.
2. Include the required technologies. In that case may request a *destination* Process to perform a [Process Action](#) by using the MMM-TEC Inter-Process [Protocols](#).
3. Hold the necessary [Rights](#).

MMM-TEC defines four varieties of Processes: App, Device, Service, and User and uses the different varieties of Process to describe their operation. However, it should be borne in mind that all Process varieties are simply Processes with different Capabilities.

Section 2 specifies the four varieties of Process. Section 3 specifies the Functional Requirements of the Action "verbs" of [Process Actions](#).

7.2 Process Types

7.2.1 App

Functions	An instance of an application-specific Program executed on a Device.
Functional Requirements	An App may need to be Certified in order to be allowed to operate in an M-Instance.

7.2.2 Device

Functions	A Device may: 1. UM-Capture Data from a U-Location. 2. Additionally, UM-Send Data and Metadata to a User. and/or 1. MU-Actuate an Item at a U-Location. 2. The Device may MU-Actuate after a User has MM-Sent an Item at 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.

7.2.3 Service

Purpose	A Process that provides specific Functionalities to other Processes.
----------------	--

Functional Requirements	A Service may be: - One of the native MMM-TEC Services (in bold in Table 1). - Hosted by the M-Instance but provided by a third party.
--------------------------------	---

Table 1 names and defines the role of Services that are part of the Process Action specification.

Table 1 - A list of Services referenced in the Verification Use Cases

Service	Acronym	Description
Authenticate	ATSrvc	Support User requests for Item Authentication
Author	AUSrvc	Enables Users to acquire or develop Items.
Communicate	COSrvc	Enables communication between Processes.
Convert	CVSrvc	Converts Data of Items to a different format based on provided Qualifier.
Discover	DSSrvc	Supports Users looking for Items of Processes with given characteristics.
Execute	EXSrvc	Executes a Program.
Export-Import	EISrvc	Manages input from and output to the Universe.
Identify	IDSrvc	Identifies Data and Metadata as Items.
Inform	IFSrvc	Supports Users looking for information about Items.
Interpret	ITSrvc	Supports users requesting Interpretation of an Item.
Locate	LCSrvc	Manages positioning and motion of Items.
Post	PTSrvc	Posts Items on which a User holds Rights.
Register	RESrvc	Enables humans to Register with an M-Instance.
RightsMng	RMSrvc	Manages the Rights of Processes and Items.
Transact	TRSrvc	Enables Users to perform Transactions.

Some [Verification Use Cases](#) reference other Services of general use. Services needed by an implementation are implemented with Process Actions. Future MMM-TEC Versions may add functionalities to existing Process Actions or add new Process Actions.

For instance, Some Verification Use Cases use the following:

Service	Acronym	Description
Parcel	PCSrvc	Offers Parcels of M-Instance land to Users.
Presence	PRSrvc	Keeps track of Processes starting a session.

7.2.4 User

Functions	A Process representing a Registered human.
Functional Requirements	- A User may perform the following functions: - Act as the interface of the human with the M-Instance. - Render the User as a Persona UM-Animated by a Stream or by an autonomous agent. - Animation results from an MM-/UM-Animate Action and enabled by a Program run by the User. - 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.
--	---

7.2.5 Actions

Table 2 specifies the Functional Requirements of the Actions that a Process may perform.

<i>Authenticate</i>	The Action of a Process producing an Authentication Response Item to confirm or not that an Item or Process is what it claims to be.
<i>Author</i>	The Action of a Process producing an Item based on provided Items and their Data and Qualifiers.
<i>Convert</i>	The Action of a Process changing the Data of an Item according to a given Qualifier
<i>Discover</i>	The Action of a Process providing a Discovery Response Item relevant to the Items or Processes that are intended to be discovered.
<i>Execute</i>	The Action of a Process executing a Program and producing Items.
<i>Hide</i>	The Action of a Process making the ID of an Item unavailable to all other Processes.
<i>Identify</i>	The Action of a Process producing and labelling an Item from Data, its Qualifier, and Model Rights.
<i>Inform</i>	The Action of a Process providing an Information Response Item about an Item or Process.
<i>Interpret</i>	The Action of a Process providing an Interpretation Response Item about an Item or Process.
<i>MM-Animate</i>	The Action of a Process animating an MM-Added Item with a Stream or Command Item.
<i>MM-Add</i>	The Action of a Process placing an Item at an M-Location with a Spatial Attitude with a specified perceptible or imperceptible state.
<i>MM-Move</i>	The Action of a Process moving an MM-Added Item from an M-Location to another M-Location along a Trajectory, preserving any change that may have been effected on the Item since it was MM-Added.
<i>MM-Send</i>	The Action of a Process sending a Message to other Processes.
<i>Modify</i>	The Action of a Process producing a new Item starting from an existing Item using new Data and possibly with a new Qualifier.
<i>MU-Actuate</i>	The Action of a Process rendering an MM-Added Item as an R-Item at a U-Location with a Spatial Attitude.
<i>MU-Add</i>	The Action of a Process placing an R-Item at a U-Location with a Spatial Attitude.
<i>MU-Animate</i>	The Action of a Process animating an MU-Added R-Item with - A stream Item equivalent to the one used to animate the Persona of which the R-Item is an Analogue Twin, or - A specific command applied to an Item that is reflected in the R-Item that is its Analogue Twin.
<i>MU-Move</i>	The Action of a Process moving an MU-Added R-Item from a U-Location to another U-Location via a Trajectory giving it an assigned Spatial Attitude when the R-Item is at the new U-Location.

<i>MU-Send</i>	The Action of a Process in an M-Instance sending a Message to Processes in the Universe.
<i>Post</i>	The Action of a Process adding an Asset to its repository and making it perceptible.
<i>Property Change</i>	The Action of a Process modifying the characteristics of an Item: 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. Change the attributes of 3.1 A Light Source: SourceType, ColourInfo, Intensity 3.2 An Audio Source: SourceType 3.3 A Basic Visual or 3D Object: Mass, Gravity Property, Texture 4. Make an Item perceptible if not already in that state.
<i>Register</i>	The Action of a human opening an Account based on the human's Personal Data.
<i>Resolve</i>	The Action of a Process setting up a multi-M-Instance Session that involves two/more than two Processes in two/more than two M-Instances.
<i>Rights Change</i>	The Action of a Process modifying the Rights of a Process or an Item.
<i>Transact</i>	The Action of a Process ₁ ("sender") 1. Assigning Rights for an Asset to Process ₂ ("receiver"). 2. Causing: 2.1. Wallet ₁ of Process ₁ to be increased by Value ₁ . 2.2. Wallet ₂ of Process ₂ to be decreased by Value ₂ . 2.3. Wallet ₃ of Process ₃ to be increased by Value ₃ (optionally).
<i>UM-Capture</i>	The Action of a Process capturing a scene at a U-Location as Data with a Qualifier.
<i>UM-Send</i>	The Action of a Process in the Universe sending a Message to Processes in an M-Instance.
<i>Validate</i>	The Action of a Process producing a Validation Response Item that confirms or not that an Item or Process has the Rights it claims to have.

8 Items

An Item is Data of a Data Type specified by MMM-TEC or other MPAI Technical Specifications It may include reference to:

1. [Rights](#), i.e., the set of [Process Actions](#) that may be exercised by certain Processes on an Item or a Process.
2. [Qualifier of the Item's Data](#) i.e., Data signaling the relevant characteristics or technologies of the Item's Data.
3. Metadata, i.e., Data describing specific characteristics of the Data.

An Item can be made available to an M-Instance by a Process that is directly driven by a human or Autonomous, e.g.:

1. By sensing Media from the Universe and producing Data, Qualifier, and Rights.
2. By producing the Item directly in the M-Instance.
3. By importing an Item from another M-Instance

Table 1 lists all Items referenced by MMM-TEC.

Table 1 - Items and Data Types used in MMM-TEC

General	MMM-SCF	Simple Certificate	X	MMM-CFO	Certificate Object	X
	MMM-SCT	Simple Contract	X	MMM-CTO	Contract Object	X
	MMM-CPB	Capabilities	X	MMM-MIN	M-Instance	X
	MMM-EVT	Event	X	MMM-PRO	Program Object	X
	MMM-IDF	Identifier	X	MMM-RGT	Rights	X
	MMM-LIC	Licence	X	MMM-RUL	Rule	X
	MMM-MEN	M-Environment	X	MMM-UEN	U-Environment	X
Human&User	MMM-ACC	Account	X	MMM-PPR	Personal Profile	X
	MMM-ACD	Activity Data	X	MMM-PDT	Personal Data	X
Process Interaction	MMM-IPP	IPP Message	X	MMM-PAC	Process Action	X
	MMM-MSG	Message	X	MMM-RES	Resolution	X
Service Access	MMM-AUT	Authentication	X	MMM-VLD	Validation	X
	MMM-BDV	Basic Discovery	X	MMM-DVO	Discovery Object	X
	MMM-BIF	Basic Information	X	MMM-IFO	Information Object	X
	MMM-BIT	Basic Interpretation	X	MMM-ITO	Interpretation Object	X
Finance	MMM-ASS	Asset	X	MMM-TRA	Transaction	X
	MMM-CUO	Currency Object	X	MMM-VAL	Value	X
	MMM-PRV	Provenance	X	MMM-WAL	Wallet	X
Perception	OSD-3DO	3D Model Object	X	OSD-3EV	3D Model Event Descriptors	X
	OSD-3SD	3D Model Scene Descriptors	X	OSD-3SG	3D Model Scene Geometry	X
	OSD-AUO	Audio Object	X	OSD-AEV	Audio Event Descriptors	X

	OSD-ASD	Audio Scene Descriptors	X	OSD-ASG	Audio Scene Geometry	X
	OSD-AVO	Audio-Visual Object	X	OSD-MEV	Audio-Visual Event Descriptors	X
	OSD-MSD	Audio-Visual Scene Descriptors	X	OSD-MSG	Audio-Visual Scene Geometry	X
	OSD-SPO	Speech Object	X	OSD-SEV	Speech Event Descriptors	X
	OSD-SSD	Speech Scene Descriptors	X	OSD-SSG	Speech Scene Geometry	X
	OSD-VIO	Visual Object	X	OSD-VSG	Visual Event Descriptors	X
	OSD-VSD	Visual Scene Descriptors	X	OSD-VSG	Visual Scene Geometry	X
	OSD-AUS	Audio Source	X	OSD-LTS	Light Source	X
	OSD-OAC	Object Audio Characteristics	X	MMM-RIT	R-Item	X
	OSD-TXO	Text Object	X			
Internal State	MMC-EGS	Cognitive State	X	MMC-EPS	Personal Status	X
	MMC-EEM	Emotion	X	MMC-ESA	Social Attitude	X
Space&Time	MMM-BML	Basic M-Location	X	OSD-OPS	Position	X
	OSD-STM	Simple Time	X	OSD-SPT	Space-Time	X
	MMM-BUL	Basic U-Location	X	OSD-SPA	Spatial Attitude	X
	MMM-MLC	M-Location	X	OSD-TIM	Time	X
	OSD-OOR	Orientation	X	MMM-ULC	U-Location	X
	OSD-OPV	Point of View	X	MMM-UMM	Universe-Metaverse Map	X

Table 1 references Object, Scene Descriptors, Scene Geometry, and Event Descriptors for 3D Model, Audio, Audio-Visual, Speech, and Visual. However, MMM-TEC implementations may also utilise the specific media versions of other Data Types: LiDAR, Offline Map, RADAR, and Ultrasound.

8.1 Simple Certificate

8.1.1 Definition

Simple Certificate is a simple form of Certificate attesting that the Actions of a Process conform with the MMM-TEC Technical Specification.

8.1.2 Functional Requirements

A Simple Certificate includes:

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

8.1.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/SimpleCertificate.json>

8.1.4 Semantics

Label	Description
Header	Simple Certificate Header
- Standard-SimpleCertificate	The characters “MMM-SCF-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SimpleCertificateID	Identifier of Simple Certificate.
SimpleCertificateTime	Validity of Simple Certificate.
CertificationAuthorityID	Identifier of Certification Authority.
CertifiedProcessActions	Process Actions for which Process is Certified.
DescrMetadata	Descriptive Metadata.

8.2 Simple Contract

8.2.1 Definition

Simple Contract is an MPAI-specified Contract providing basic functionalities for use in the MMM-TEC context.

8.2.2 Functional Requirements

A Simple Contract includes:

1. Contract ID
2. Time of Contract (Start Time and End Time).
3. Data of Seller.
4. Data of Buyer.

5. Data of Service Provider (the one that facilitates Basic Contract definition and execution).
6. Basic Contract Status (one of Model, Final)

8.2.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/SimpleContract.json>

8.2.4 Semantics

Label	Description
Header	Simple Contract Header
- Standard-SimpleContract	The characters “MMM-SCT-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SimpleContractID	Identifier of Simple Contract.
SimpleContractTime	Validity of Simple Contract.
SimpleContractData	Data contained in a Simple Contract.
- SenderData	Seller Data.
– SenderID	Seller ID.
– SenderPreValue	Value sent to SP before sale,
– SPLicence	Posting Licence sent to SP.
– SenderPostValue	Value sent to SP after sale.
– ReceiverLicence	Licence sent to buyer.
– SenderWalletID	Sender Wallet ID.
- ReceiverData	Receiver Data.
– ReceiverID	Receiver ID.
– ReceiverPostValue	Value Sent by buyer.
– ValueToSender	Rights received by buyer.
– ReceiverWalletID	Receiver Wallet ID.
- ServiceProviderData	Service Provider Data.
– ServiceProviderID	Service Provider ID.
– ServiceProviderWalletID	Service Provider Wallet ID.
- Basic Contract Status	One of Model or Final.
DescrMetadata	Descriptive Metadata

8.3 Capabilities

8.3.1 Definition

An Items containing the list of Capabilities of an Item, an M-Instance or M-Environment, or a Process.

8.3.2 Functional Requirements

The Capabilities of an Item

1. Include:
 1. The list of Processes that may perform Process Actions on the Item, and
 2. For each Process, the Process Actions the Process can perform.
2. May include Metadata that Process Actions and Rules may use to indicate classes of Items to which Process Actions and Rules may be applied.

The Capabilities of a Processes include:

1. List of Actions it can perform.
2. List of Items it can process.
3. List of Item Qualifiers.
4. List of Process Metadata.
5. List of Item Metadata.
6. The ID of the human responsible for the Process.
7. The IDs of the Apps it has onboard.

8.3.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Capabilities.json>

8.3.4 Semantics

Label	Description
Header	P-Capabilities Header
- Standard	The characters “MMM-CPB-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
CapabilitiesID	Identifier of Capabilities.
ItemCapabilities	Data in Capabilities when Item.
– ProcessActions[]	For all Process Types.
– ItemMetadata[]	The Item Metadata used by the Process.
MInstanceCapabilities	Data in Capabilities when M-Instance.
CoordinateType	Coordinate System used by the M-Instance.
Profile	One of “Baseline”, “Finance”, “Management”, or “High”.
Actions[]	Actions that can be performed in the the M-Instance or M-Environment.
Items[]	List of Item types supported by M-Instance/M-Environment.

Qualifiers[]	List of supported Item Qualifiers.
PMetadata[]	List of M-Instance-specific Process Metadata.
IMetadata[]	List of M-Instance-specific Item Metadata.
ProcessCapabilities	Data in Capabilities when Process.
- humanID	ID of human responsible for tte Process.
- SupportedApps[]	If Process is Device
- ProcessMetadata[]	List of Metadata of the Process.
- Actions[]	Actions that the Process can perform.
- Items[]	List of Items types and Data Qualifiers.
- ItemMetadata[]	List of Metadata of the Item.
- ItemQualifiers[]	The Process Actions that the Process can perform.
WalletID	The The ID of the Wallet a Process is connected to.
DescrMetadata	Descriptive Metadata

8.4 Event

8.4.1 Definition

An Event is a Process Action or a logical combination of Process Actions performed at a Time or Time Type.

8.4.2 Functional Requirements

Events involving no more than two Process Actions are listed in Table 1.

Table 1 - Basic Events

Process Action₁	Operation	Process Action₂	Basic Event happens if
Process Action			The Process Action is performed.
	NOT	Process Action	If the Process Action is not performed
Process Action ₁	AND	Process Action ₂	If both Process Actions are performed.
Process Action ₁	OR	Process Action ₂	If at least one Process Action is performed.
Process Action ₁	NAND	Process Action ₂	If at least one Process Action is not performed.
Process Action ₁	NOR	Process Action ₂	If none of the Process Actions is performed.
Process Action ₁	XOR	Process Action ₂	If only one Process Action is performed.
Process Action ₁	XNOR	Process Action ₂	If one Process Action is performed and the other is not.

and the characters *If* followed by Time, referring to a single Time, a Duration, or a recurring Time.

8.4.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Event.json>

8.4.4 Semantics

Label	Description
Header	Header of Event Item.
- Standard-Event	The characters “MMM-EVT-V”.
- Version	Major version expressed as 1 or 2 characters.
- Dot-separator	The character “.”
- Subversion	Minor version expressed as 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
EventID	The ID of the Event.
EventData[]	The set of Data defining the Event.
- BasicEvent	The collection of Process Actions preceded by May.
- ProcessAction1	As defined by Process Actions .
- LogicOperator	One of "AND", "OR", "NAND", "NOR", "XOR", "XNOR", "NOT"
- ProcessAction2	As defined by Process Actions .
- EventOrBasicEvent	Any of Event or Basic Event
- If	The characters If
- Time	A single Time, a Duration, or a recurring Time.
DescrMetadata	Descriptive Metadata.

8.5 Identifier

8.5.1 Definition

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

8.5.2 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.

8.5.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Identifier.json>

8.5.4 Semantics

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

8.6 License

8.6.1 Definition

An Item that includes Licensor, Licensee, and Rights to an Item or a Process in an M-Instance.

8.6.2 Functional Requirements

Licence includes the following Data:

1. LicenceID.
2. LicenceTime.
3. LicensorID.
4. LicenseeID.
5. Service ProviderID through which the definition of the Licence was facilitated.
6. An Identifier of a Process or an Item to which Licensee has Rights and Licensor may or may not have certain Rights.
7. Licence Status (one of Model or Final).

8.6.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Licence.json>

8.6.4 Semantics

Label	Description
Header	Licence Header
- Standard-Licence	The characters “MMM-LIC-V”
- Version	Major version – 1 or 2 Bytes
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 Bytes
MInstanceID	Identifier of M-Instance.

LicenceID	Identifier Licence
LicenceIssuanceTime	Time when Licence was issued
LicenceTime	Start Time and End Time of Licence.
LicenceData	Data included in Licence.
LicensorID	ID of Licensor
LicenseeID	ID of Licensee
ServiceProviderID	ID of Service Provider who facilitated definition of Licence.
ItemID or ProcessID	ID of Item Or Process to which Rights apply.
LicensorRights	Rights of Licensor (e.g., May Not Process Action)
LicenseeRights	Rights of Licensee
DescrMetadata	Descriptive Metadata

8.7 M-Environment

8.7.1 Definition

An Identified administrative subset of an M-Instance.

8.7.2 Functional Requirements

1. An M-Environment implements a Profile.
2. The Profile may be the same or a Profile included in the 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.

8.7.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/MEnvironment.json>

8.7.4 Semantics

Label	Description
Header	M-Environment Header
- Standard-MEnvironment	The characters “MMM-MEN-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version –1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
MCapabilities	Capabilities of M-Environment.
Rules	Rules of M-Environment.
DescrMetadata	Descriptive Metadata

8.8 Account

8.8.1 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.

8.8.2 Functional Requirements

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

8.8.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Account.json>

8.8.4 Semantics

Label	Description
Header	Account Header
- Standard - Account	The characters "MMM-ACC-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
M-EnvironmentID	Identifier of M-Environment
AccountID	Identifier of Account.
Account	N is the number of Bytes composing the Account
- humanID	Identifier of human.
- PersonalProfileID	ID of Personal Profile.
- Processes[]	Set of Process
- ProcessID	ID of a Process
- Rights	Rights of a Process
- RightsID	ID of Rights held by ProcessID.

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

8.9 Activity Data

8.9.1 Definition

An Item recording the IPP Protocols executed by a Process.

8.9.2 Functional Requirements

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

8.9.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/ActivityData.json>

8.9.4 Semantics

Label	Description
Header	Activity Data Header
- Standard - ActivityData	The characters “MMM-ACD-V”
- Version	Major version expressed as 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version expressed as 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
ProcessID	ID of Process that generated the Activity Data
ActivityDataID	Identifier of Activity Data.
ActivityData[]	The list of Process Actions.
- IPPMessageID	IDs of IPPMessage.
DescrMetadata	Descriptive Metadata.

8.10 IPP Message

8.10.1 Definition

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

8.10.2 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
Time	Provided by communication infrastructure.
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).
Resolution Service ID	Service that: 1. Receives a Message from a Process in its M-Instance and forwarding it to a peer Process in a different M-Instance, or 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.
Message Status	Ack, if successful, if failure, Error as follows: 1. “Unable” to perform request. 2. “ValueRQ” Transaction of <i>Value</i> required. 3. “ConvFail” Conversion Service failure.

8.10.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/IPPMessage.json>

8.10.4 Semantics

Label	Description
Header	IPP Message Header
- Standard-IPPMessage	The characters “MMM-IPM-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
IPPMessageID	Identifier of Message.
IPPMessage	Number of Bytes in Message Data
- ResponseID	As specified in Inter-Process Protocol .
- SourceProcessID	As specified in Inter-Process Protocol .
- ProcessAction	As specified in Inter-Process Protocol .
- ResolutionServiceID	As specified in Inter-Process Protocol .
- DestinationProcessID	As specified in Inter-Process Protocol .

- Acknowledgement	As specified in Inter-Process Protocol .
DescrMetadata	Descriptive Metadata.

8.11 Message

8.11.1 Definition

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

8.11.2 Functional Requirements

A Message may contain: Item, Data, Data and Qualifier, Data and Metadata.

8.11.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Message.json>

8.11.4 Semantics

Label	Description
Header	Message Header
- Standard-Message	The characters “MMM-MSG-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MessageID	Identifier of Message.
MessageData	Content of Message
MessageDataLength	Number of Bytes in Message Payload
MessageDataURI	URI of Message Data
DescrMetadata	Descriptive Metadata

8.12 Authentication

8.12.1 Definition

An Item generated by

1. A Process requesting a Service to confirm that an Item is what it claims to be.
2. A Service containing the response to the request.

8.12.2 Functional Requirements

An Authentication Item contains:

1. In the Authentication Request:
 1. The M-Location wrapping the Item to be Authenticated.
 2. The Media Type of the Item.

3. The claimed identity of the Item.
2. In the Authentication Response:
 1. Confidence.

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.

8.12.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Authentication.json>

8.12.4 Semantics

Label	Description
Header	Authentication Header
- Standard-Authentication	The characters “MMM-AUT-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
AuthenticationID	Identifier of Authentication Item.
AuthenticationRequest	Authentication request
- MLocation	The M-Location including the Item to be Authenticated.
- PerceptibleEntity	Item if Perceptible Entity.
- ClaimedIdentity	The claimed identity.
AuthenticationResponse	Authentication response
- Confidence	A number between 0 and 1.
DescrMetadata	Descriptive Metadata

8.13 Basic Discovery

8.13.1 Definition

An MMM-TEC-specified Item used by a Process to request that a Service discover Items or Processes. Other types of Discover Items are possible and are listed in the Discovery Qualifier.

8.13.2 Functional Requirements

Discovery Request includes

1. The IDs of a model Items or Processes.
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. The M-Location wrapping a Model Item
3. A [Perceptible Entity](#) describing the Item or the Process.
4. Model Rights to use the Basic Discovery Response.

Basic Discovery Response includes a set of:

1. Item ID, Process ID, and [Perceptible Entity](#).

8.13.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/BasicDiscovery.json>

8.13.4 Semantics

Label	Description
Header	Basic Discovery Header
- Standard-BasicDiscovery	The characters “MMM-BDV-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
BasicDiscoveryID	Identifier of Basic Discovery.
BasicDiscovery	Data of Basic Discovery.
- BasicDiscoveryRequest	Data of Request
- DiscoveryData	Data regarding what is searched
- ItemID	ID of Item provided as an example of what is searched.
- MLocation	The M-Location wrapping a Model Item.
- ProcessID	Process ID provided as an example of what is searched.
- PerceptibleEntity	A perceptible Item.
- Model Rights	Requested Rights
- BasicDiscoveryResponse	Data of Response
- Items[]	List of IDs of found Items.
- DiscoveredData	Data regarding what is discovered
- ItemID	the ID of a found Item.
- MLocation	The M-Location wrapping a Model Item.
- PerceptibleEntity	A perceptible Item.
- Rights	Acquired Rights
- Processes[]	List of IDs of found Processes.
- ProcessID	The ID of a found Process.
DescrMetadata	Descriptive Metadata

8.14 Basic Information

8.14.1 Definition

An MMM-TEC-specified Item used by a Process to request that a Service provide information on Items or Processes. Other types of Information Items are possible and are listed in the Information Qualifier.

8.14.2 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.

Information Response includes:

1. Perceptible Item providing information on the Item (e.g., Text).

8.14.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/BasicInformation.json>

8.14.4 Semantics

Label	Description
Header	Basic Information Header
- Standard-BasicInformation	The characters “MMM-BIF-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
BasicInformationID	Identifier of Basic Information .
BasicInformation[]	Data of Basic Information .
- BasicInformationRequest	Data of Request
- ItemID	Item ID
- MLocation	A wrapped Object or Scene
- PerceptibleEntity	The Perceptible Entity about which Interpretation is requested.
- BasicInformationResponse	Data of Response
- PerceptibleEntity	Perceptible Item informing about the Item.
DescrMetadata	Descriptive Metadata

8.15 Basic Interpretation

8.15.1 Definition

An MMM-TEC-specified Item used by a Process to request that a Service provide interpretation of Items or Processes. Other types of Interpretation Items are possible and are listed in the Interpretation Qualifier.

8.15.2 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:

1. Perceptible Item providing information on the Item (e.g., Text).

8.15.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/BasicInterpretation.json>

8.15.4 Semantics

Label	Description
Header	Basic Interpretation Header
- Standard-BasicInterpretation	The characters “MMM-BIT-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
BasicInterpretationID	Identifier of Basic Interpretation.
BasicInterpretationData[]	Data of Basic Interpretation.
- BasicInterpretationRequest	Data of Request.
- ItemID	The ID of an Item.
- MLocation	A wrapped Object or Scene
- PerceptibleEntity	A Perceptible Item.
- ModelRights	Requested Rights.
- BasicInterpretationResponse	Data of Response.

- PerceptibleItem	A Perceptible Item.
- Rights	Acquired Rights.
DescrMetadata	Descriptive Metadata.

8.16 Asset

8.16.1 Definition

An Item that can be Transacted.

8.16.2 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.
 - o 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.

8.16.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Asset.json>

8.16.4 Semantics

Label	Description
Header	Asset Header
- Standard-Asset	The characters “MMM-ASS-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
AssetID	Identifier of Asset.
AssetData[]	Metadata of Asset
- SourceItemID	ID of Item that spawned this Asset
- AssetDate	Time Item was Modified to Asset
- Provenance	Provenance or its ID
DescrMetadata	Descriptive Metadata

8.17 Currency Object

8.17.1 Definition

Currency Object refers to

1. Data representing unit of measure of an Amount of Value.

2. Certificate Qualifier specified by MPai-TFA providing information on Sub-Types, Formats and Attributes.

8.17.2 Functional Requirements

Amount is expressed as a number. Currency may be Real or Virtual and represented by labels whose semantics is give by a Currency Qualifier.

8.17.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/CurrencyObject.json>

8.17.4 Semantics

Label	Description
Header	Currency Header
– Standard-Currency	The characters “MMM-CUR-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
CurrencyData	The Data of the Currency (currency label).
CurrencyQualifier	The Qualifier of a Currency.
DescrMetadata	Descriptive Metadata

8.18 Provenance

8.18.1 Definition

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

8.18.2 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.

8.18.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Provenance.json>

8.18.4 Semantics

Label	Description
Header	Provenance Header
- Standard-Provenance	The characters “MMM-PRV-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters

M-InstanceID	Identifier of M-Instance.
AssetID	Identifier of the Asset.
ProvenanceID	Identifier of Provenance.
Provenance[]	Set of Provenance Data.
- TransactionID	The IDs of the Transactions in the Provenance.
DescrMetadata	Descriptive Metadata.

8.19 3D Model Object

8.19.1 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.

8.19.2 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 Objects and other 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 Annotation.
6. The Rights that may be exercised on the 3D Model Object.

Note that.

1. A 3D Model Object that does not include 3D Model Objects 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 a 3D Model Object included in an 3D Model Object may be superseded by the Space-Time information of the 3D Model Object containing them.

8.19.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/3DModelObject.json>

8.19.4 Semantics

Label	Description
Header	3D Model Object Header
- Standard-3DModelObject	The characters “OSD-3DO-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.

3DModelObjectID	Identifier of the 3D Model Object.
3DModelObjectSpaceTime	Space-Time of 3D Model Object.
Basic3DModelObjectCount	Set of Parent 3D Model Objects.
Basic3DModelObjects[]	Set of Basic 3D Model Objects.
- SpaceTime	Space Time of a Basic 3D Model Object in the 3D Model Object.
- Basic3DModelObject	A Basic 3D Model Object in the 3D Model Object.
3DModelObjectCount	Number of 3D Model Objects.
3DModelObjects[]	Set of 3D Model Objects.
- SpaceTime	Space Time of an 3D Model Object in the 3D Model Object.
- 3DModelObject	A 3D Model Object in the 3D Model Object
Annotations[]	Set of 3D Model Object Annotation.
- Annotation	An Annotation.
- AnnotationSpaceTime	Where Annotation is attached and when it will be active.
- Rights	Process Actions that may be performed on the Annotation
Rights	Process Actions that may be performed on the Object.
DescrMetadata	Descriptive Metadata

8.19.5 Conformance Testing

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

8.20 3D Model Scene Descriptors

8.20.1 Definition

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

8.20.2 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.

8.20.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/3DModelSceneDescriptors.json>

8.20.4 Semantics

Label	Description
-------	-------------

Header	3D Model Scene Descriptors Header
- Standard-3DModelSceneDescriptors	The characters “OSD-3SD-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneDescriptorsID	Identifier of Scene Descriptors.
SceneDescriptorsSpaceTime	Space and Time of Scene Descriptors.
ObjectCount	Number of Objects in Scene.
Objects[]	Set of Objects.
- Object or ObjectID	Object in the Scene of its ID.
- ObjectSpaceTime	Space Time of Object.
SubSceneCount	Number of Sub-Scenes in Scene.
SubScenes[]	Set of Sub-Scenes in the Scene.
- SubScene or SubSceneID	Sub-Scene in the Scene or its ID.
- SubSceneSpaceTime	Space Time of Sub-Scene.
DescrMetadata	Descriptive Metadata

8.20.5 Conformance Testing

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

8.21 Audio Object

8.21.1 Definition

A Data Type including Basic Audio Objects and Audio Objects.

Audio Objects have a hierarchical structure where Audio Objects contain Basic Audio Objects and Audio Objects.

8.21.2 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.

8.21.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/AudioObject.json>

8.21.4 Semantics

Label	Description
Header	Audio Object Header
– Standard-AudioObject	The characters “OSD-AUO-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
AudioObjectID	Identifier of the Audio Object.
AudioObjectSpaceTime	Space-Time of Audio Object.
BasicAudioObjectCount	Set of Parent Audio Objects.
BasicAudioObjects[]	Set of Basic Audio Objects.
- SpaceTime	Space Time of a Basic Audio Object in the Audio Object.
- BasicAudioObject	A Basic Audio Object in the Audio Object.
AudioObjectCount	Number of Audio Objects.
AudioObjects[]	Set of Audio Objects.
- SpaceTime	Space Time of an Audio Object in the Audio Object.
- AudioObject	An Audio Object in the Audio Object
Annotations[]	Set of Audio Object Annotation.
– Annotation	An Annotation.
– AnnotationSpaceTime	Where Annotation is attached and when it will be active.
– Rights	Actions that may be performed on the Annotation

Rights	Actions that may be performed on the Object.
DescrMetadata	Descriptive Metadata

8.21.5 Conformance Testing

A Data instance Conforms with Audio Object (OSD-AUO) 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.

8.22 Audio Scene Descriptors

8.22.1 Definition

Audio Scene Descriptors are defined as a Data Type including the Audio Objects of a scene, their sub-scenes, and their arrangement in the scene. Audio Scene Descriptors may be hierarchical, i.e., they may contain Objects and Audio Scene Descriptors.

8.22.2 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.

8.22.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/AudioSceneDescriptors.json>

8.22.4 Semantics

Label	Description
Header	Audio Scene Descriptors Header
- Standard-AudioSceneDescriptors	The characters "OSD-ASD-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneDescriptorsID	Identifier of Scene Descriptors.
SceneDescriptorsSpaceTime	Space and Time of Scene Descriptors.
ObjectCount	Number of Objects in Scene.
Objects[]	Set of Objects.

- Object or ObjectID	Object in the Scene of its ID.
- ObjectSpaceTime	Space Time of Object.
SubSceneCount	Number of Sub-Scenes in Scene.
SubScenes[]	Set of Sub-Scenes in the Scene.
- SubScene or SubSceneID	Sub-Scene in the Scene or its ID.
- SubSceneSpaceTime	Space Time of Sub-Scene.
DescrMetadata	Descriptive Metadata

8.22.5 Conformance Testing

A Data instance Conforms with Audio Scene Descriptors (OSD-ASD) 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.

8.23 Audio-Visual Object

8.23.1 Definition

Data whose rendering has both Audio and Visual perceptibility attributes.

8.23.2 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.

8.23.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/AudioVisualObject.json>

8.23.4 Semantics

Label	Description
Header	Audio-Visual Object Header
- Standard-AudioVisualObject	The characters “OSD-AVO-V”
- Version	Major version – 1 or 2 Bytes
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 Bytes
MInstanceID	Identifier of M-Instance.
AudioVisualObjectID	Identifier of Audio-Visual Object.

AudioVisualObjectSpaceTime	Space-Time of Audio-Visual Object
AudioVisualQualifier	Qualifier of the Audio-Visual Object
3DModelObjectData	3D Model Object Data
- 3DModelObjectID and/or 3DModelObject	3D Model Object ID and/or Object
- 3DModelObjectSpaceTime	Space-Time of Speech Object
SpeechObjectData	Speech Object Data
- SpeechObjectID and/or Speech Object	Speech Object ID and/or Object
- SpeechObjectSpaceTime	Space-Time of Speech Object
AudioObjectData	Audio Object Data
- AudioObjectID and/or Audio Object	Audio Object ID and/or Object
- AudioObjectSpaceTime	Space-Time of Audio Object
VisualObjectData	Visual Object Data
- VisualObjectID and/or Visual Object	Visual Object ID and/or Object
- VisualObjectSpaceTime	Space-Time of Visual Object
Annotations[]	Set of Audio Object Annotation.
– Annotation	An Annotation.
– AnnotationSpaceTime	Where Annotation is attached and when it will be active.
– Rights	Actions that may be performed on the Annotation
Rights	Actions that may be performed on the Object.
DescrMetadata	Descriptive Metadata

8.23.5 Conformance Testing

A Data instance Conforms with Audio-Visual Object (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.

8.24 Audio-Visual Scene Descriptors

8.24.1 Definition

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

8.24.2 Functional Requirements

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

8.24.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/AudioVisualSceneDescriptors.json>

8.24.4 Semantics

Label	Description
Header	Audio-Visual Scene Descriptors Header
- Standard-AVSceneDescriptors	The characters “OSD-AVS-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
AVBasicSceneDescriptorsID	Identifier of the AV Object.
ObjectCount	Number of Objects in Scene
AVSceneSpaceTime	Data about Space and Time
SpeechObjects[]	Set of Speech Objects
- SpeechObject	Speech Object
- SpeechObjectSpaceTime	Space-Time of Speech Object
AudioObjects[]	Set of Audio Objects
- AudioObject	ID of Audio Object
- AudioObjectSpaceTime	Space-Time of Audio Object
VisualObjects[]	Set of Visual Objects
- VisualObjectID	ID of Visual Object
- VisualObjectSpaceTime	Space-Time of Visual Object
AudioVisualObjects[]	Set of Audio-Visual Objects
- AudioVisualObjectID	ID of Audio-Visual Object
- AudioObjectSpaceTime	Space-Time of Audio-Visual Object
SubSceneCount	Number of Sub-Scenes in Scene
SpeechSubScenes[]	Set of Speech Objects
- SpeechSubScene	Speech SubScene
- SpeechSubSceneSpaceTime	Space-Time of Speech SubScene
AudioSubScenes[]	Set of Audio SubScenes
- AudioSubScene	ID of Audio SubScene
- AudioSubSceneSpaceTime	Space-Time of Audio SubScene
VisualSubScenes[]	Set of Visual SubScenes
- VisualSubSceneID	ID of Visual SubScene
- VisualSubSceneSpaceTime	Space-Time of Visual SubScene
AudioVisualSubScenes[]	Set of Audio-Visual SubScenes

- AudioVisualSubSceneID	ID of Audio-Visual SubScene
- AudioSubSceneSpaceTime	Space-Time of Audio-Visual SubScene
DescrMetadata	Descriptive Metadata

8.24.5 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.

8.25 Speech Object

8.25.1 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.

8.25.2 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.

8.25.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/SpeechObject.json>

8.25.4 Semantics

Label	Description
Header	Speech Object Header
– Standard-SpeechObject	The characters “OSD-SPO-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”

- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SpeechObjectID	Identifier of the Speech Object.
SpeechObjectSpaceTime	Space-Time of Speech Object.
BasicSpeechObjectCount	Set of Parent Speech Objects.
BasicSpeechObjects[]	Set of Basic Speech Objects.
- SpaceTime	Space Time of a Basic Speech Object in the Speech Object.
- BasicSpeechObject	A Basic Speech Object in the Speech Object.
SpeechObjectCount	Number of Speech Objects.
SpeechObjects[]	Set of Speech Objects.
- SpaceTime	Space Time of a Speech Object in the Speech Object.
- SpeechObject	A Speech Object in the Speech Object
Annotations[]	Set of Speech Object Annotation.
- Annotation	An Annotation.
- AnnotationSpaceTime	Where Annotation is attached and when it will be active.
- Rights	Actions that may be performed on the Annotation
Rights	Actions that may be performed on the Object.
DescrMetadata	Descriptive Metadata

8.25.5 Conformance Testing

A Data instance Conforms with Speech Object (OSD-SPO) 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.

8.26 Speech Scene Descriptors

8.26.1 Definition

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

8.26.2 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.

8.26.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/SpeechSceneDescriptors.json>

8.26.4 Semantics

Label	Description
Header	Speech Scene Descriptors Header
- Standard-SpeechSceneDescriptors	The characters “OSD-SSD-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneDescriptorsID	Identifier of Scene Descriptors.
SceneDescriptorsSpaceTime	Space and Time of Scene Descriptors.
ObjectCount	Number of Objects in Scene.
Objects[]	Set of Objects.
- Object or ObjectID	Object in the Scene of its ID.
- ObjectSpaceTime	Space Time of Object.
SubSceneCount	Number of Sub-Scenes in Scene.
SubScenes[]	Set of Sub-Scenes in the Scene.
- SubScene or SubSceneID	Sub-Scene in the Scene or its ID.
- SubSceneSpaceTime	Space Time of Sub-Scene.
DescrMetadata	Descriptive Metadata

8.26.5 Conformance Testing

A Data instance Conforms with Speech Scene Descriptors (OSD-SSD) 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.

8.27 Visual Object

8.27.1 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.

8.27.2 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.

8.27.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/VisualObject.json>

8.27.4 Semantics

Label	Description
Header	Visual Object Header
– Standard-VisualObject	The characters “OSD-VIO-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
VisualObjectID	Identifier of the Visual Object.
VisualObjectProperty	Properties of Visual Object - If any is present overrides the properties of component objects.
- MaterialProperty	Object has material consistence (0=No, 1=Yes)
- GravityProperty	Object is subject to gravity (0=No, 1=Yes)
VisualObjectSpaceTime	Space-Time of Visual Object.
BasicVisualObjectCount	Set of Parent Visual Objects.
BasicVisualObjects[]	Set of Basic Visual Objects.
- SpaceTime	Space Time of a Basic Visual Object in the Visual Object.
- BasicVisualObject	A Basic Visual Object in the Visual Object.
VisualObjectCount	Number of Visual Objects.
VisualObjects[]	Set of Visual Objects.
- SpaceTime	Space Time of a Visual Object in the Visual Object.
- VisualObject	A Visual Object in the Visual Object
Annotations[]	Set of Visual Object Annotation.
– Annotation	An Annotation.
– AnnotationSpaceTime	Where Annotation is attached and when it will be active.

– Rights	Actions that may be performed on the Annotation
Rights	Actions that may be performed on the Object.
DescrMetadata	Descriptive Metadata

8.27.5 Conformance Testing

A Data instance Conforms with Visual Object (OSD-VIO) 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.

8.28 Visual Scene Descriptors

8.28.1 Definition

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

8.28.2 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.

8.28.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/VisualSceneDescriptors.json>

8.28.4 Semantics

Label	Description
Header	Visual Scene Descriptors Header
- Standard-VisualSceneDescriptors	The characters “OSD-VSD-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
VisualSceneDescriptorsID	Identifier of Visual Scene Descriptors.
ObjectCount	Number of Visual Objects in Scene.
SubSceneCount	Number of Visual Scenes in Scene.
VisualSceneDescriptorsSpaceTime	Space and Time of Visual Scene Descriptors.

VisualSceneObjects[]	Set of Visual Objects.
- VisualSceneObject	Visual Object.
- VisualSceneObjectSpaceTime	Space Time of Visual Object.
VisualSceneSubScenes[]	Set of Visual Sub-Scenes.
- VisualSceneSubScene	Visual Sub-Scene.
- VisualSceneSubSceneSpaceTime	Space Time of Visual Sub-Scene.
DescrMetadata	Descriptive Metadata

8.28.5 Conformance Testing

A Data instance Conforms with Visual Scene Descriptors (OSD-VSD) 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.

8.29 Audio Source

8.29.1 Definition

A Data Type representing an audio source and its characteristics.

8.29.2 Functional Requirements

An Audio Source may include:

1. The ID of the M-Instance where it is located.
2. The Audio Source's
 1. ID.
 2. Time-Space information.
 3. Type.
 4. Annotation.
 5. Rights.

8.29.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/AudioSource.json>

8.29.4 Semantics

Label	Description
Header	Audio Source Header
- Standard-AudioSource	The characters "OSD-AUS-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.

AudioSourceID	Identifier of the Audio Source.
AudioSourceSpaceTime	Space-Time information of the Audio Source.
AudioSourceType	Type of source.
- Diffuseness	- Matrix of Frequency (Hz) - Normalised Intensity in dB.
- DirectionalPatterns	- Tensor of Frequency (Hz) - Directivity (Azimuth-Elevation in degrees) - Normalised Intensity in dB.
- ShapeAndSize	- Tensor of Frequency (Hz) - Directivity (Azimuth-Elevation in degrees) - Normalised Intensity in dB. - Shape of Source and its Size.
AudioSourceAnnotation	Annotations of Audio Source.
- Annotation	ID of Annotation.
- AnnotationSpaceTime	Where/when Annotation is attached.
- RightsID	Rights that may be exercised on the Annotation.
AudioSourceRights	Rights that may be exercised on the Audio Source.
DescrMetadata	Descriptive Metadata.

8.29.5 Conformance Testing

A Data instance Conforms with Audio Source if:

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

8.30 Object Audio Characteristics

8.30.1 Definition

Data describing the characteristics of an object as pertains Audio.

8.30.2 Functional Requirements

Object Audio Characteristics includes

1. Reflectivity
 1. Early Reflections Time (s)
 2. Late Reflections Time (s)
2. Reverberation
 1. RT60 (frequency dependent)
 2. RT30 (frequency dependent)
 3. R20 (frequency dependent)
 4. EDT (Time taken by audio from full scale to -10 dB)
3. Diffusion: Schroeder Diffusion Coefficient (number)
4. Absorption: Absorption Coefficient (number 0 to 1)

8.30.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/ObjectAudioCharacteristics.json>

8.30.4 Semantics

Label	Description
Header	Object Audio Characteristics Header
– Standard-ObjectAudioCharacteristics	The characters “OSD-OAC-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
ObjectAudioCharacteristicsID	Identifier of Texture.
ObjectAudioCharacteristics	Data of Object Audio Characteristics
- Reflectivity	The reflection of sound in the same medium after hitting a surface.
- EarlyReflectionTime	Measured in seconds
- LateReflectionTime	Measured in seconds
- Reverberation	The persistence of sound after it has been produced
- RT60	Table with frequency (Hz) and dB
- RT30	Table with frequency (Hz) and dB
- RT20	Table with frequency (Hz) and dB
- EDT	Time taken by audio from full scale to -10 dB (ms).
- Diffusion	The spreading of sound waves throughout a space, measured by the Schroeder Diffusion Coefficient (number).
- Absorption	The loss of energy affected by sound waves that have reached an absorbent material measured by the Absorption Coefficient ($1 \geq \text{number} \geq 0$).
ObjectAudioCharacteristicsRights	Rights to Act on the Object Audio Characteristics.
DescrMetadata	Descriptive Metadata

8.30.5 Conformance Testing

A Data instance Conforms with Object Audio Characteristics (OSD-OAC) if:

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

8.31 Text Object

8.31.1 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.

8.31.2 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.

8.31.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/TextObject.json>

8.31.4 Semantics

Label	Description
Header	Text Object Header
– Standard-TextObject	The characters “OSD-TXO-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
TextObjectID	Identifier of the Text Object.
TextObjectSpaceTime	Space-Time of Text Object.
BasicTextObjectCount	Set of Parent Text Objects.
BasicTextObjects[]	Set of Basic Text Objects.
- SpaceTime	Space Time of a Basic Text Object in the Text Object.
- BasicTextObject	A Basic Text Object in the Text Object.
TextObjectCount	Number of Text Objects.
TextObjects[]	Set of Text Objects.
- SpaceTime	Space Time of a Text Object in the Text Object.

- TextObject	A Text Object in the Text Object
Annotations[]	Set of Text Object Annotation.
- Annotation	An Annotation.
- AnnotationSpaceTime	Where Annotation is attached and when it will be active.
- Rights	Actions that may be performed on the Annotation
Rights	Actions that may be performed on the Object.
DescrMetadata	Descriptive Metadata

8.31.5 Conformance Testing

A Data instance Conforms with Text Object (OSD-TXO) 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.

8.32 Cognitive State

8.32.1 Definition

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

8.32.2 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., “AN-GER”).
 - The GENERAL ADJECTIVAL column gives adjectival labels for general or basic la-bels 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 1 – Basic Cognitive State Label Set

COGNITIVE CATEGORIES	GENERAL ADJECTIVAL	SPECIFIC ADJECTIVAL
AROUSAL	aroused/excited/energetic	cheerful playful lethargic sleepy
ATTENTION	attentive	expectant/anticipating 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 2 provides the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns above.

Table 2 – 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.

8.32.3 Syntax

<https://schemas.mpai.community/MMC/V2.4/data/CognitiveState.json>

8.32.4 Semantics

Label	Description
Header	Entity Cognitive State Header
- Standard-EntityCognitiveState	The characters “MMC-ECS-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.

EntityCognitiveStateID	Identifier of CogState.
EntityCognitiveStateSpaceTime	Space-Time info of CogState.
EntityCognitiveStateData	Data associated to CogState.
- FusedCogState	Integrated CogState Value.
- TextCogState	Text CogState Value.
- SpeechCogState	Speech CogState Value.
- FaceCogState	Face CogState Value.
- GestureCogState	Gesture CogState Value.
DescrMetadata	Descriptive Metadata

8.32.5 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.

8.33 Emotion

8.33.1 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”.

8.33.2 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 1 gives the standardised three-level Basic Emotion Set partly based on Paul Eckman [19].

Table 1 – 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 2 provides the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns above.

Table 2 – 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.

8.33.3 Syntax

<https://schemas.mpai.community/MMC/V2.4/data/Emotion.json>

8.33.4 Semantics

Label	Description
Header	Entity Emotion Header
- Standard-EntityEmotion	The characters “MMC-EEM-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
EntityEmotionID	Identifier of the Emotion.
EntityEmotionSpaceTime	Space-Time info of Emotion
EntityEmotionData	Data associated to Emotion.
- FusedEmotion	Integrated Emotion Value.
- TextEmotion	Text Emotion Value.
- SpeechEmotion	Speech Emotion Value.
- FaceEmotion	Face Emotion Value.
- GestureCogState	Gesture Emotion Value.
DescrMetadata	Descriptive Metadata

8.33.5 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.

8.34 Basic M-Location

8.34.1 Definition

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

8.34.2 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.

8.34.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/BasicMLocation.json>

8.34.4 Semantics

Label	Description
Header	U-Location Header
- Standard	The characters "MMM-BML-V"
- Version	Major version – 1 or 2 Bytes
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 Bytes
MLocationID	The ID of the M-Location
MEnvironmentID	The ID of an M-Environment.
BasicMLocationData	The Data of a Basic U-Location.
BasicMLocationQualifier	The Qualifier of the Basic U-Location.
BasicMLocationSpaceTime	Space-Time information of Basic U-Location
BasicMLocationRights	Rights of Basic U-Location
DescrMetadata	Descriptive Metadata.

8.35 Simple Time

8.35.1 Definition

The digital representation of (duration of) time specified by MPAI.

8.35.2 Functional Requirements

Time includes start time and the end time of a duration. The origin of Time can be Absolute (from 1970/01/01) or relative to a user-selected value.

8.35.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/SimpleTime.json>

8.35.4 Semantics

Label	Description
Header	Simple Time Header
- Standard-SimpleTime	The characters “OSD-STM-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance
SimpleTimeID	Identifier of M-Instance.
SimpleTimeData	Data about Time
- TimeType	0=Relative: start at 0000/00/00T00:00 1=Absolute: start at 1970/01/01T00:00.
- Reserved	reserved
- TimeUnit	00=seconds, 01=milliseconds, 10=microseconds, 11=nanoseconds.
- StartTime	Start of Time.
- EndTime	End of Time.
DescrMetadata	Descriptive Metadata

8.35.5 Conformance Testing

A Data instance Conforms with OSD-STM if:

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

8.36 Basic U-Location

8.36.1 Definition

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

8.36.2 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.

8.36.3 Syntax

<https://schemas.mpai.community/MMM/V2.1/data/BasicULocation.json>

8.36.4 Semantics

Label	Description
Header	U-Location Header
- Standard	The characters “MMM-BUL-V”
- Version	Major version – 1 or 2 Bytes
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 Bytes
UEnvironmentID	The ID of an administratively Identified portion of Universe
ULocationID	Identifier of U-Location.
BasicULocationData	The Data of a Basic U-Location.
BasicULocationQualifier	The Qualifier of the Basic U-Location.
BasicULocationSpaceTime	Space-Time information of Basic U-Location
BasicULocationRights	Rights of Basic U-Location
DescrMetadata	Descriptive Metadata.

8.37 M-Location

8.37.1 Definition

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

8.37.2 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 and/or Time.
3. Is an element of a hierarchy of M-Locations with a level of the hierarchy $L > 0$.
4. It is composed of Locations of Level $< L$.
5. Need not be continuous, i.e., a Level L M-Location may include space portions completely surrounded by M-Locations with Level $< L$.
6. May have Rights attached governing the Process Actions that a Process may perform in the M-Location.

7. An element of level L of an M-Location hierarchy may inherit only part of the Rights of a level L-1 element of the hierarchy.

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

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

8.37.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/MLocation.json>

8.37.4 Semantics

Label	Description
Header	M-Location Header
- Standard-MLocation	The characters “MMM-MLC-V”
- Version	Major version
- Dot-separator	The character “.”
- Subversion	Minor version
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
MLocationID	Identifier of M-Location.
MLocation	Set of Data defining M-Location.
- BasicMLocations[]	Set of Data defining M-Location.
- BasicMLocation	A Basic M-Location
- BasicMLocationSpaceTime	Spatial Attitude and Time of Basic MLocation.
- Mlocations[]	Level L-1 M-Location (L may be 0, i.e., Basic M-Location).
- MLocation	A Basic M-Location
- MLocationSpaceTime	Spatial Attitude and Time of Basic MLocation.
Rights	Rights held or granted to M-Location.
DescrMetadata	Descriptive Metadata.

8.38 Orientation

8.38.1 Definition

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

8.38.2 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.

8.38.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/Orientation.json>

8.38.4 Semantics

Table 1 provides the semantics of the components of Orientation. It should be noted that the Euler angles are indicated by (α, β, γ) .

Table 1 – Semantics of Orientation

Label	Description
Header	Orientation Header
- Standard-Orientation	The characters “OSD-OOR-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstance	ID id Virtual space Orientation refers tu
ObjectOrientationID	Identifier of Object Orientation.
General	Set of general data
- ObjectType	One of Digital Human, Generic.
- MediaType	One of Speech, Audio, Visual, Audio-Visual, Haptic, Smell, RADAR, LiDAR, Ultrasound.
Orientation	
- Orient (α, β, γ)	Array (in degrees)
- OrientAccuracy	Array of (α, β, γ) Accuracy
Velocity of Orientation	
- OrientVelocity (α, β, γ)	Array (in degrees/s)
- OrientVelocityAccuracy	Array of (α, β, γ) Velocity Accuracy
Acceleration of Orientation	
- OrientAccel (α, β, γ)	Array (in degrees/s ²)
- OrientAcceAccuracy	Array of (α, β, γ) Acceleration Accuracy
DescrMetadata	Descriptive Metadata

8.38.5 Conformance Testing

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

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

8.39 Point of View

8.39.1 Definition

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

8.39.2 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.

8.39.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/PointOfView.json>

8.39.4 Semantics

Table 1 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 1 – Semantics of Point of View

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

8.39.5 Conformance Testing

A Data instance Conforms with MPAI-OSD Point of View (OSD-OPV) 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.

8.40 Certificate Object

8.40.1 Definition

Certificate Object refers to

1. Data representing confirmation of the suitability of a Process for specific usages.
2. Certificate Qualifier specified by MPAI-TFA providing information on Sub-Types, Formats and Attributes.

8.40.2 Functional Requirements

A Certificate Object includes the Certificate Data and Qualifier.

8.40.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/CertificateObject.json>

8.40.4 Semantics

Label	Description
Header	Certificate Object Header
- Standard - CertificateObject	The characters “MMM-CFO-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
CertificateID	Identifier of Certificate.
CertificateData	Data in the Certificate,
CertificateQualifier	Qualifies of Certificate.
DescrMetadata	Descriptive Metadata

8.41 Contract Object

8.41.1 Definition

Contract Object refers to:

1. Contract Data representing the terms and conditions for executing a Program or performing an Action when an event occurs.
2. Contract Data Qualifier specified by MPAI-TFA providing information on Sub-Types, Formats and Attributes.

8.41.2 Functional Requirements

Import of a Contract Object into an M-Instance may require approval.

8.41.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/ContractObject.json>

8.41.4 Semantics

Label	Description
Header	Contract Object Header
- Standard-ContractObject	The characters “MMM-CTO-V”
- Version	Major version – 1 or 2 Bytes
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 Bytes
MInstanceID	Identifier of M-Instance.
ContractID	Identifier of Contract.
ContractData	Data of Contract
ContractDataQualifier	Qualifier of Contract.
DescrMetadata	Descriptive Metadata

8.42 M-Instance

8.42.1 Definition

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

8.42.2 Functional Requirements

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

8.42.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/MInstance.json>

8.42.4 Semantics

Label	Description
Header	MInstance Header
- Standard-MInstance	The characters “MMM-MIN-V”
- Version	Major version expressed as 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version expressed as 1 or 2 characters

MInstanceID	Identifier of M-Instance.
Coordinates	Coordinate System used by M-Instance.
MCapabilities	Capabilities of M-Instance.
Rules	Rules of M-Instance.
DescrMetadata	Descriptive Metadata.

8.43 Program Object

8.43.1 Definition

Program Object refers to

1. An Item containing executable code, e.g., a Process or a Contract that is a Program.
2. ML Model Qualifier specified by MPai-TFA providing information on Sub-Types, Formats and Attributes.

8.43.2 Functional Requirements

A Program includes the program (Data) and the Qualifier adding information to the Data and may include a Certificate information.

8.43.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/ProgramObject.json>

8.43.4 Semantics

Label	Description
Header	Program Object Header
- Standard-ProgramObject	The characters “MMM-PRG-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
ProgramObjectID	Identifier of Program.
ProgramData	Data of Program.
ProgramQualifier	Qualifier of Program.
ProgramMetadata	Metadata Characteristic of the Program.
CertificateID	Certificate ID.
DescrMetadata	Descriptive Metadata

8.44 Right

8.44.1 Definition

Rights are defined for

- A Process: The set of a Deontic Verb (one of "May", "May Not", or "Must"), a Process Action, and the Rights Level ("Internal", "Acquired", or "Granted") applicable to a Process.
- An Item: The set of a Process, a Deontic Verb, a Process Action including the Item, and the Level.

8.44.2 Functional Requirements

Process Rights include the following:

<i>Deontic Verb</i>	A Deontic Verb from "May", "May Not", or "Must".
<i>Process Action</i>	A Process Action .
<i>Level</i>	One of Internal, Acquired, or Granted.

Item Rights include a list of Processes and for each Process the three elements included in Process Rights in addition to the three elements in the Process Rights.

Note that:

<u>Process</u>	refers to a Process identified by an ID or having specific Metadata.
<u>Item</u>	refers to an Item identified by an ID or by having
	- Specific Metadata
	- A Qualifier
	- One or more parameters.
<u>Time</u>	refers to a single Time, a Duration, or a recurring Time.

8.44.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Rights.json>

8.44.4 Semantics

Label	Description
Header	Rights Header
- Standard-Rights	The characters “MMM-RGT-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
RightsID	Identifier of Rights.
For Process Rights	
- <i>DEx</i>	One of the "May", "May Not", or "Must" Deontic Expressions.
- <i>Process Action</i>	A Process Action that may, may not, or must Perform
- <i>Level</i>	One of Internal, Acquired, or Granted.

For Item Rights	
- Processes	The list of Processes that may/may not/must Perform Process Actions followed by the element of “For Process Rights”.
RightsStatus	One of Model - used, e.g., in a PA Request, Final - used, e.g., in a PA Response.
DescrMetadata	Descriptive Metadata.

8.45 Rule

8.45.1 Definition

Rule is an Item including a Process, a deontic verb - Permission, Prohibition, or Obligation, and a Process Action.

8.45.2 Functional Requirements

A Rule is represented by the following sequence

1. Process (identified by an ID or by Metadata)
2. Deontic Verb (one of May, May Not, or Must)
3. Process Action

Here:

<u>Process</u>	refers to a Process identified by an ID or having specific Metadata.
<u>Item</u>	refers to an Item identified by an ID or by having
	- Specific Metadata
	- A Qualifier
	- One or more parameters.
<u>Time</u>	refers to a single Time, a Duration, or a recurring Time.

8.45.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Rules.json>

8.45.4 Semantics

Label	Description
Header	Header of Rule Item.
- Standard-Rule	The characters “MMM-RUL-V”.
- Version	Major version expressed as 1 or 2 characters.
- Dot-separator	The character “.”
- Subversion	Minor version expressed as 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
RuleID	The ID of the Rules

RuleData[]	The elements part of a Rule.
- Process	A Process identified by an ID or having specific Metadata.
- DeonticVerb	One of May, May Not, and Must.
- ProcessAction	As defined by Process Actions .
DescrMetadata	Descriptive Metadata.

8.46 U-Environment

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/V2.1/data/UEnvironment.json>

8.46.4 Semantics

Label	Description
Header	U-Environment Header
- Standard-UEnvironment	The characters “MMM-UEN-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version –1 or 2 characters
MInstanceID	Identifier of M-Instance.
UEnvironmentID	Identifier of U-EnvironmentID.
DescrMetadata	Descriptive Metadata

8.47 Personal Profile

8.47.1 Definition

An Item containing information about a human submitted when Registering with an M-Instance.

8.47.2 Functional Requirements

Personal Profile includes humanID and First Name, Last Name, Age, Nationality, and Email of the human, Processes and Personae to be deployed in the M-Instances. An M-Instance may extend or reduce the elements of Personal Profile.

8.47.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/PersonalProfile.json>

8.47.4 Semantics

Label	Description
Header	Personal Profile Header
- Standard - PersonalProfile	The characters “MMM-PPR-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
humanID	ID of the human the Personal Profile refers to.
PersonalProfileID	ID of Personal Profile.
PersonalProfile	The number of Bytes composing the Personal Profile.
- First Name	The human’s given name
- Last Name	The human’s family name
- Age	The human’s age
- Nationality	The human’s country
- Email	The human’s address
- Programs[]	Registered human's Programs.
- Program	A particular Program.
- Personae[]	Personae associated to a User
- Persona	A particular Persona.
DescrMetadata	Descriptive Metadata.

8.48 Personal Data

8.48.1 Definition

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

8.48.2 Functional Requirements

Personal Data includes a human's:

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

8.48.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/PersonalData.json>

8.48.4 Semantics

Label	Description
Header	Personal Data Header
- Standard PersonalData	The characters “MMM-PDT-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
PersonalDataID	Identifier of Account.
PersonalData	Personal Profile and Users' Activity Data.
- PersonalProfileID	Identifier of Personal Profile.
- Processes[]	Set of human’s Processes for which Activity is considered.
- ProcessID	Identifier of ProcessID.
- ActivityData[]	Set of ActivityData.
- ActivityDataID	Identifier of ActivityDataID.
DescrMetadata	Descriptive Metadata

8.49 Process Action

8.49.1 Definition

An Item that includes the following:

An Action.

If the Process Action is a Request:

- A series of one of *Nil*, *At*, *From*, *To*, or *With* each followed by Item or Process on which the Process Action is performed.
- Condition for performing the Request expressed by *If* followed by Event.

If the Process Item is a Response:

- A series of one of *Nil*, *At*, *From*, *To*, or *With* each followed by Item or Process on which the Process Action is performed.
- The Status of the Process Action.

Here:

<u>Nil</u>	is used to signal that the Item or Process is a direct object. The other prepositions are used as specified for each Process Action.
<u>Process</u>	refers to a Process identified by an ID or having specific Metadata.
<u>Item</u>	refers to an Item identified by an ID or by having

	- Specific Metadata
	- A Qualifier
	- One or more parameters.

8.49.2 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>RQ-Complements</i>	A series of - One of <i>Nil, At, From, Of, or With</i> - An Item or a Process on which the Action is performed at the Request side.
<i>Condition</i>	<i>If</i> Event.
<i>RS-Complements</i>	A series of - One of <i>Nil, At, From, Of, or With</i> - An Item or a Process on which the Action is performed at the Response side.
<i>PA Status</i>	"Ack" if successful, Error message otherwise, if Response.

8.49.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/ProcessAction.json>

8.49.4 Semantics

Label	Description
Header	Process Action Data Header
- Standard-ProcessAction	The characters "MMM-PAC-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
ProcessActionID	ID of Process Action.
ProcessAction	The number of Bytes composing the Process Action.
- Time	Time when the Process Action is requested and time the Process Action is executed.
- RQ-Complements[]	One <i>Nil, At, From, Of, or With</i> each followed by Item or Process.
- NilProcess	Process ID
- AtProcess	<i>At</i> followed by Process ID

- FromProcess	<i>From</i> followed by Process ID
- ToProcess	<i>To</i> followed by Process ID
- WithProcess	<i>With</i> followed by Process ID
- NilItem	Item ID
- AtItem	<i>At</i> followed by Item ID
- FromItem	<i>From</i> followed by Item ID
- ToItem	<i>To</i> followed by Item ID
- WithItem	<i>With</i> followed by Item ID
- Condition	<i>If</i> followed by Event
- RS-Complements[]	One or more Items or the Process each preceded by one of <i>Nil</i> , <i>At</i> , <i>From</i> , <i>Of</i> , or <i>With</i> .
- NilProc	Process ID
- AtProc	<i>At</i> followed by Process ID
- FromProc	<i>From</i> followed by Process ID
- ToProc	<i>To</i> followed by Process ID
- WithProc	<i>With</i> followed by Process ID
- NilItem	Item ID
- AtItem	<i>At</i> followed by Item ID
- FromItem	<i>From</i> followed by Item ID
- ToItem	<i>To</i> followed by Item ID
- WithItem	<i>With</i> followed by Item ID
- PAStatus	Process Action-specific Message specified in Process Action .
- Ack	Process Action was successfully performed.
- Request	Value requested to perform Process Action.
- Error	oneOf "Clash", "FaultyReq", "IncID", "IncDQ", "InsRights", "InsValue", "LocOOR", "PostRef", "QualNS"
DescrMetadata	Descriptive Metadata.

8.50 Resolution

8.50.1 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..

8.50.2 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

8.50.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Resolution.json>

8.50.4 Semantics

Label	Description
Header	Resolution Header
- Standard-Resolution	The characters “MMM-RSL-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
ResolutionID	Identifier of Authentication Item.
ProcessAToResolARequest	Request of Calling Process to own Resolution Service.
- MInstances[]	Set of M-Instances with a participating Process.
- MInstanceID	One M-Instance with a participating Process.
- Processes[]	Set of participating Processes in M-Instance.
- ProcessID	One Participating Process in M-Instance.
ResolAToResolBRequest	Request of Calling Process’s Resolution Service
- Processes[]	Set of participating Processes in M-Instance.
- ProcessID	One Participating Process in M-Instance.
ResolBToProcessBRequest	Response of Resolution ServiceB
- Processes[]	Set of participating Processes in M-Instance.
- ProcessID	One Participating Process in M-Instance.
ProcessBToResolBResponse	Response of ProgressB
- JoinYN	Y for confirmation, N for denial.
ResolBToResolAResponse	Response of Resolution ServiceB

- Error	Access to M-InstanceB denied.
- ProcessID	ID of Process sending Accept.
- JoinYN	Accept of Process ID (Y for confirmation, N for denial).
ResolAToProcessAResponse	Response of Resolution ServiceA
- Error	Access to M-InstanceA denied.
- MInstanceID	ID of M-Instance Process of ProcessID belongs to.
- ProcessID	ID of Process sending Accept.
- JoinYN	Accept of Process ID
DescrMetadata	Descriptive Metadata

8.51 Validation

8.51.1 Definition

An Item generated by

1. A Process requesting a Service to confirm that a Process has specified Rights that it claims to have.
2. A Service containing the response to the request.

8.51.2 Functional Requirements

An Validation Item contains:

1. In the Validation Request:
 1. The Process ID.
 2. The Rights.
2. In the Authentication Response:
 1. A Confirmation (Yes or No).

8.51.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Validation.json>

8.51.4 Semantics

Label	Size	Description
Header	N1 Bytes	Validation Header
- Standard-Validation	9 Bytes	The characters “MMM-VAL-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
- ProcessID	N7 Bytes	The Process whose Rights are questioned.

- Rights	N8 Bytes	The specific Rights.
ValidationResponse	N10 Bytes	Validation response
- Confirmation	N11 Bytes	Y for confirmation, N for denial.
DescrMetadata	N13 Bytes	Descriptive Metadata

8.52 Discovery Object

8.52.1 Definition

Discovery Object refers to

1. An Item representing the description of the Items or Processes to be Discovered by a Discover Service.
2. ML Model Qualifier specified by MPAI-TFA providing information on Sub-Types, Formats and Attributes.

8.52.2 Functional Requirements

The Discovery Item is composed of a Discovery Request Item and Discovery Response Item.

The Discovery Request Item is used, e.g., discover:

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

The Discovery Request Item may include the ID of one or more Items or Process the the Discover Service deems to comply with the Discovery Request Item.

8.52.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/DiscoveryObject.json>

8.52.4 Semantics

Label	Description
Header	Discovery Object Header
- Standard-DiscoveryObject	The characters “MMM-DSO-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
DiscoveryID	Discovery as identified by its Qualifier.
DiscoveryData	Data contained in the Discovery Item.
DiscoveryQualifier	The Qualifier of Dat in the Discovery Item.
DescrMetadata	Descriptive Metadata.

8.53 Information Object

8.53.1 Definition

Information Object refers to

1. An Item sent by a Process to a Service as an Information Request Item requesting information about an Item or Process to obtain additional information on that Item or Process that is not otherwise available.
2. ML Model Qualifier specified by MP AI-TFA providing information on Sub-Types, Formats and Attributes.

8.53.2 Functional Requirements

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

8.53.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/InformationObject.json>

8.53.4 Semantics

Label	Description
Header	Information Header
- Standard-Information	The characters “MMM-IFO-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
InformationID	Identifier of Information Item.
InformationRequestData	Set of Data in Information Request Item.
InformationResponseData	Set of Data in Information Response Item.
InformationQualifier	Qualifier of Information Item
DescrMetadata	Descriptive Metadata

8.54 Interpretation Object

8.54.1 Definition

Interpretation Object refers to

1. An Item sent by a Process to a Service as an Interpretation Request Item requesting that an Item be Interpreted.
2. ML Model Qualifier specified by MP AI-TFA providing information on Sub-Types, Formats and Attributes.

8.54.2 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

8.54.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/InterpretationObject.json>

8.54.4 Semantics

Label	Description
Header	Interpretation Header
- Standard-Interpretation	The characters “MMM-ITO-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
InterpretationID	Identifier of Interpretation Item.
InterpretationData	Interpretation Data
InterpretationQualifier	Interpretation Qualifier
InterpretationObjectTime	Time of Interpretation Object
DescrMetadata	Descriptive Metadata

8.55 Transaction

8.55.1 Definition

An Item representing:

1. The Amount, the WalletID, and the Rights on an Asset of a User (Sender) that transfers Rights to another User (Receiver).
2. The Amount, the WalletID and the Rights on on the Asset of the Receiver.
3. Optionally the Amount and the WalletID of the Service Provider facilitating/enabling the Transaction.

8.55.2 Functional Requirements

The Transaction shall represent:

1. The Time the Transaction is performed.
2. The ID of the Sender's Wallet.
3. The Value moving from the Receiver's Wallet into the Sender's Wallet.

4. The ID of the Sender's Rights to Act on the Asset after Transaction Time.
5. The ID of the Receiver's Wallet.
6. The ID of the Receiver's Rights to Act on the Asset after Transaction Time.
7. The ID of the Service Provider's Wallet.
8. The Value moving from the Sender or Receiver Wallet or both into the Service Provider's Wallet.
9. ID of the Service's Rights after Transaction Time.
10. The Status of the Transaction.

8.55.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Transaction.json>

8.55.4 Semantics

Label	Description
Header	Transaction Header
- Standard-Transaction	The characters “MMM-TRA-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
TrasactionID	Identifier of Transaction.
TransactionData	Set of Data related to Transaction
- AssetID	The ID of the Asset the Transaction refers to.
- TransactionTime	Time the Transaction is performed.
- SenderData	Sender dataset.
- SenderID	The ID of Sender “who grants the Rights” after Time.
- SToSPValue	The Value entering the Service Provider's Wallet from the Sender's Wallet.
- SenderRightsID	The ID of the Rights held by Sender after Transaction Time.
- SenderWalletID	The ID of the Sender's Wallet.
- ReceiverData	Receiver dataset.
- ReceiverID	The ID of Receiver “who is granted the Rights”.
- RToSValue	The Value entering the Sender's Wallet from the Receivers Wallet.
- RToSPValue	The Value entering the Service Provider's Wallet from the Receiver's Wallet.
- ReceiverRightsID	The ID of the Rights “granted to Receiver” after Transaction Time.
- ReceiverWalletID	The ID of the Receiver's Wallet.

- ServiceProviderData	Service Provider's dataset.
- ServiceProviderID	The ID of the Service Provider.
- ServiceRightsID	The ID of the Rights “granted to Service” after Transaction Time.
- ServiceProviderWalletID	The ID of the Service Provider’s Wallet.
- TransactionStatus	One of Model - used, e.g., in a negotiation
DescrMetadata	Any description of the Transaction.

8.56 Value

8.56.1 Definition

An Item combining an Amount and a Currency.

8.56.2 Functional Requirements

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

8.56.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Value.json>

8.56.4 Semantics

Label	Description
Header	Value Header
- Standard - Wallet	The characters “MMM-VAL-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
ValueID	Identifier of Value.
Value	N is the number of Bytes composing the Value
- Amount	The number of Currency Units of the Amount
- Currency	The 3-character code of the Currency
DescrMetadata	Descriptive Metadata

8.57 Wallet

8.57.1 Definition

A container of Values.

8.57.2 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.

8.57.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Wallet.json>

8.57.4 Semantics

Label	Description
Header	Wallet Header
- Standard - Wallet	The characters “MMM-WAL-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
M-InstanceID	Identifier of M-Instance.
WalletID	Identifier of Wallet.
AccountID	AccountID of the Wallet Holder.
Wallet	Wallet Values organised by Currency.
- ValueID[]	ID of Values for Currencies.
- TransactionID[]	ID of Transactions affecting a Value.
DescrMetadata	Descriptive Metadata

8.58 3D Model Event Description

8.58.1 Definition

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

8.58.2 Functional Requirements

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

8.58.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/3DModelEventDescriptors.json>

8.58.4 Semantics

Label	Description
Header	3D Model Event Descriptors Header
· Standard-3DModelEventDescriptors	The characters “OSD-3DE-V”
· Version	Major version – 1 or 2 characters

· Dot-separator	The character “.”
· Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
EventID	Identifier of the Event.
EventSpaceTime	Data about start and end Space-Time.
SceneDescriptors[]	Set of Scene Descriptors
- SceneDescriptors	Set of AV Scene Descriptors of IDs.
DescrMetadata	Descriptive Metadata

8.58.5 Conformance Testing

A Data instance Conforms with MPAI-OSD 3D Model Event Descriptors (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.

8.59 3D Model Scene Geometry

8.59.1 Definition

A 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

8.59.2 Functional Requirements

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

8.59.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/3DModelSceneGeometry.json>

8.59.4 Semantics

Label	Description
Header	3D Model Scene Geometry Header
- Standard-3DModelSceneGeometry	The characters “OSD-3SG-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.

SceneGeometryID	Identifier of Scene Geometry.
ObjectCount	Number of Objects in Scene.
SubSceneCount	Number of Sub-Scenes in Scene.
SceneGeometrySpaceTime	Space and Time of Scene Geometry.
SceneObjects[]	Set of Data related to Objects.
- SceneObjectID	ID of Object.
- SceneObjectSpaceTime	Space Time of Object.
SceneSubScenes[]	Set of Sub-Scenes.
- SceneSubSceneID	ID of Sub-Scene.
- SceneSubSceneSpaceTime	Space Time of Sub-Scene.
DescrMetadata	Descriptive Metadata

8.59.5 Conformance Testing

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

8.60 Audio Event Descriptors

8.60.1 Definition

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

8.60.2 Functional Requirements

Audio Event Descriptors contains Audio Scene Descriptors for a Time.

8.60.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/AudioEventDescriptors.json>

8.60.4 Semantics

Label	Description
Header	Audio Event Descriptors Header
· Standard	The characters "OSD-AUE-V"
· Version	Major version – 1 or 2 characters
· Dot-separator	The character "."
· Subversion	Minor version – 1 or 2 characters

MInstanceID	Identifier of M-Instance.
EventID	Identifier of the Event.
EventSpaceTime	Data about start and end Space-Time.
SceneDescriptors[]	Set of Scene Descriptors
- SceneDescriptors	Set of AV Scene Descriptors of IDs.
DescrMetadata	Descriptive Metadata

8.60.5 Conformance Testing

A Data instance Conforms with MPAI-OSD Audio Event Descriptors (OSD-AUE) if:

1. The Data validates against the Audio Event Descriptors' JSON Schema.
2. All Data in the Audio 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 Scene Geometry

8.61.1 Definition

A Data Type including the space-time arrangement of the Audio Objects in a scene.

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

8.61.2 Functional Requirements

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

8.61.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/AudioSceneGeometry.json>

8.61.4 Semantics

Label	Description
Header	Audio Scene Geometry Header
- Standard-AudioSceneGeometry	The characters "OSD-ASG-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneGeometryID	Identifier of Scene Geometry.
ObjectCount	Number of Objects in Scene.

SubSceneCount	Number of Sub-Scenes in Scene.
SceneGeometrySpaceTime	Space and Time of Scene Geometry.
SceneObjects[]	Set of Data related to Objects.
- SceneObjectID	ID of Object.
- SceneObjectSpaceTime	Space Time of Object.
SceneSubScenes[]	Set of Sub-Scenes.
- SceneSubSceneID	ID of Sub-Scene.
- SceneSubSceneSpaceTime	Space Time of Sub-Scene.
DescrMetadata	Descriptive Metadata.

8.61.5 Conformance Testing

A Data instance Conforms with Audio Scene Geometry (OSD-ASG) 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.

8.62 Audi-Visual Event Descriptors

8.62.1 Definition

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

8.62.2 Functional Requirements

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

8.62.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/AudioVisualEventDescriptors.json>

8.62.4 Semantics

Label	Description
Header	Audio-Visual Event Descriptors Header
- Standard-AudioVisualEventDescriptors	The characters "OSD-AVE-V"
· Version	Major version – 1 or 2 characters
· Dot-separator	The character "."
· Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
EventID	Identifier of the Event.

EventSpaceTime	Data about start and end Space-Time.
SceneDescriptors[]	Set of Scene Descriptors
- SceneDescriptors	Set of AV Scene Descriptors of IDs.
DescrMetadata	Descriptive Metadata

8.62.5 Conformance Testing

A Data instance Conforms with MPAA-OSD Audio-Visual Event Descriptors (OSD-AVE) 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.

8.63 Audio-Visual Scene Geometry

8.63.1 Definition

A Data Type including the arrangement of the Audio-Visual Objects in a scene with their Audio-Visual Qualifiers.

In the following, Data, Objects, Qualifiers, and (Sub-)Scenes should be read as Audio-Visual Data, Audio-Visual Objects, Audio-Visual Qualifiers, and Audio-Visual (Sub-)Scenes

8.63.2 Functional Requirements

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

8.63.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/AudioVisualSceneGeometry.json>

8.63.4 Semantics

Label	Description
Header	Audio-Visual Scene Geometry Header
- Standard-AudioVisualSceneGeometry	The characters "OSD-AVG-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneGeometryID	Identifier of Scene Geometry.
ObjectCount	Number of Objects in Scene.
SubSceneCount	Number of Sub-Scenes in Scene.

SceneGeometrySpaceTime	Space and Time of Scene Geometry.
SceneObjects[]	Set of Data related to Objects.
– SceneObjectID	ID of Object.
– SceneObjectSpaceTime	Space Time of Object.
SceneSubScenes[]	Set of Sub-Scenes.
– SceneSubSceneID	ID of Sub-Scene.
– SceneSubSceneSpaceTime	Space Time of Sub-Scene.
DescrMetadata	Descriptive Metadata.

8.63.5 Conformance Testing

A Data instance Conforms with Audio-Visual Scene Geometry (OSD-AVG) 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.

8.64 Speech Event Descriptors

8.64.1 Definition

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

8.64.2 Functional Requirements

Speech Event Descriptors contains Speech Scene Descriptors for a Time.

8.64.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/SpeechEventDescriptors.json>

8.64.4 Semantics

Label	Description
Header	Speech Event Descriptors Header
- Standard-SpeechEventDescriptors	The characters "OSD-SPE-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
EventID	Identifier of the Event.
EventSpaceTime	Data about start and end Space-Time.
SceneDescriptors[]	Set of Scene Descriptors

- SceneDescriptors	Set of AV Scene Descriptors of IDs.
DescrMetadata	Descriptive Metadata

8.64.5 Conformance Testing

A Data instance Conforms with MPai-OSD Speech Event Descriptors (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.

8.65 Speech Scene Geometry

8.65.1 Definition

A Data Type including the arrangement of the Speech Objects in a scene with their Speech Qualifiers.

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

8.65.2 Functional Requirements

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

8.65.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/SpeechSceneGeometry.json>

8.65.4 Semantics

Label	Description
Header	Speech Scene Geometry Header
- Standard-SpeechSceneGeometry	The characters “OSD-SSG-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneGeometryID	Identifier of Scene Geometry.
ObjectCount	Number of Objects in Scene.
SubSceneCount	Number of Sub-Scenes in Scene.
SceneGeometrySpaceTime	Space and Time of Scene Geometry.
SceneObjects[]	Set of Data related to Objects.
– SceneObjectID	ID of Object.

– SceneObjectSpaceTime	Space Time of Object.
SceneSubScenes[]	Set of Sub-Scenes.
– SceneSubSceneID	ID of Sub-Scene.
– SceneSubSceneSpaceTime	Space Time of Sub-Scene.
DescrMetadata	Descriptive Metadata.

8.65.5 Conformance Testing

A Data instance Conforms with Speech Scene Geometry (OSD-SSG) 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.

8.66 Visual Event Descriptors

8.66.1 Definition

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

8.66.2 Functional Requirements

Visual Event Descriptors contains Visual Scene Descriptors for a Time.

8.66.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/UltrasoundEventDescriptors.json>

8.66.4 Semantics

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

8.66.5 Conformance Testing

A Data instance Conforms with MPAI-OSD Visual Event Descriptors (OSD-VIE) 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.

8.67 Visual Scene Geometry

8.67.1 Definition

A Data Type including the arrangement of the Visual Objects in a scene with their Visual Qualifiers.

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

8.67.2 Functional Requirements

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

8.67.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/VisualSceneGeometry.json>

8.67.4 Semantics

Label	Description
Header	Visual Scene Geometry Header
- Standard-VisualSceneGeometry	The characters "OSD-VSG-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SceneGeometryID	Identifier of Scene Geometry.
ObjectCount	Number of Objects in Scene.
SubSceneCount	Number of Sub-Scenes in Scene.
SceneGeometrySpaceTime	Space and Time of Scene Geometry.
SceneObjects[]	Set of Data related to Objects.
– SceneObjectID	ID of Object.
– SceneObjectSpaceTime	Space Time of Object.
SceneSubScenes[]	Set of Sub-Scenes.
– SceneSubSceneID	ID of Sub-Scene.

– SceneSubSceneSpaceTime	Space Time of Sub-Scene.
DescrMetadata	Descriptive Metadata.

8.67.5 Conformance Testing

A Data instance Conforms with Visual Scene Geometry (OSD-VSG) 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.

8.68 Light Source

8.68.1 Definition

A Data Type representing a light source and its characteristics.

8.68.2 Functional Requirements

A Light Source may include:

1. The ID of the M-Instance where it is located.
2. The Light Source's
 1. ID
 2. Time-Space information
 3. Type (shape)
 4. Colour information
 5. Intensity
 6. Annotations
 7. Rights

8.68.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/LightSource.json>

8.68.4 Semantics

Label	Description
Header	Light Source Header
– Standard-LightSource	The characters “OSD-LTS-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
LightSourceID	Identifier of the Light Source.
LightSourceSpaceTime	Space-Time information of the Light Source.
LightSourceType	One of the following:

- Point	Point-like source
- Directional	- Azimuth - Elevation
- SkyLight	- Azimuth - Elevation
- Spotlight	- Azimuth - Elevation - Angle - Blend ($0 \leq 1$) representing the fall out rate of the light at the border of the basis of the cone.
- Area	Characterised by Shape
ColourInformation	Selected from Colour Space.
Intensity	The light source's intensity measured in Watt.
LightSourceAnnotation	Annotations of Visual Data
- Annotation	ID of Annotation
- AnnotationSpaceTime	Where/when Annotation is attached.
- RightsID	Rights that may be exercised on the Annotation
LightSourceRights	Rights that may be exercised on the Light Source.
DescrMetadata	Descriptive Metadata

8.68.5 Conformance Testing

A Data instance Conforms with Light Source if:

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

8.69 R-Item

8.69.1 Definition

A perceptible object in the Universe or the Analogue Twin in the Universe of an Item in the Metaverse.

8.69.2 Functional Requirements

An R-Item can be of one of 3 Types:

1. An existing real object that is placed at a U-Location (R-Object).
2. A new real object that is synthesised with a 3D printer rendering an Item (S-Object).
3. A perceptible 2D or 3D media item in the Universe that renders an Item in an M-Instance (M-Object).

8.69.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/RItem.json>

8.69.4 Semantics

Label	Description
Header	Value Header
- Standard - Wallet	The characters “MMM-RIT-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
UEnvironmentID	Identifier of a U-Environment
ItemID	Identifier of Item in the M-Instance.
RItemID	Identifier of R-Item.
RItemType	One of R-Object, S-Object, M-Object.
ULocation	U-Location of R-Item.
DescrMetadata	Descriptive Metadata.

8.70 Personal Status

8.70.1 Definition

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

8.70.2 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 MPAA 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. b. 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.

8.70.3 Syntax

<https://schemas.mpai.community/MMC/V2.4/data/PersonalStatus.json>

8.70.4 Semantics

Label	Description
Header	Personal Status Header
- Standard-PersonalStatus	The characters “MMC-EPS-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
PersonalStatusID	Identifier of Meaning.
PersonalStatusSpaceTime	Space-Time info of PersonalStatus
PersonalStatus	Personal Status
- CognitiveState	Cognitive State component of Personal Status

- Emotion	Emotion component of Personal Status
- SocialAttitude	Social Attitude component of Personal Status
DescrMetadata	Descriptive Metadata

8.70.5 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.

8.71 Social Attitude

8.71.1 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”.

8.71.2 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 la-bels 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. b. 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 1 gives the standardised three-level Basic Social Attitude Set.

Table 1 – Basic Social Attitude Label Set

SOCIAL ATTITUDE CATEGORIES	GENERAL ADJECTIVAL	SPECIFIC 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
COOPERATION	cooperative/agreeable uncooperative	flexible subversive/undermining uncommunicative stubborn disagreeable
RESPONSIVENESS	responsive/demonstrative emotional/passionate unresponsive/undemonstrative unemotional/detached	enthusiastic unenthusiastic passionate dispassionate
EMPATHY	empathetic/caring kind uncaring/callous	sympathetic merciful 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/dependabl e 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/snobbis h pedantic unaffected servile/obsequious

Table 56 provides the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns above.

Table 2 – 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	evinced willingness to communicate as needed
18	conceited/vain	evinced 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	evinced 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/cliquish	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	evincing willingness to resolve issues through reasoning
81	rebellious/defiant	evincing unwillingness to obey
82	responsible/trustworthy/ dependable	evincing 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.

8.71.3 Syntax

<https://schemas.mpai.community/MMC/V2.4/data/SocialAttitude.json>

8.71.4 Semantics

Label	Description
Header	Entity Social Attitude Header
- Standard-SocialAttitude	The characters “MMC-ESA-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
SocialAttitudeID	Identifier of the Social Attitude.
SocialAttitudeSpaceTime	Space-Time info of Social Attitude.
SocialAttitudeData	Data associated to Social Attitude.
- FusedSocAtt	Integrated Social Attitude Value.

- TextSocAtt	Text Social Attitude Value.
- SpeechSocAtt	Speech Social Attitude Value.
- FaceSocAtt	Face Social Attitude Value.
- GestureSocAtt	Gesture Social Attitude Value.
DescrMetadata	Descriptive Metadata

8.71.5 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.

8.72 Position

8.72.1 Definition

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

8.72.2 Functional Requirements

- The Position of an Object is given by the Coordinates of a representative point of 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.

8.72.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/Position.json>

8.72.4 Semantics

Table 1 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 1 – Semantics of the Spatial Attitude

Label	Description
Header	Position Header
- Standard-Position	The characters "OSD-OPS-V"
- Version	Major version – 1 or 2 characters
- Dot-separator	The character "."
- Subversion	Minor version – 1 or 2 characters
MInstanceId	ID of Virtual Space Position refers to.

ObjectPositionID	Identifier of Object Position.
General	Set of general data
- CoordinateType	One of Cartesian, Spherical, Geodesic, Toroidal.
- ObjectType	One of Digital Human, Generic.
- MediaType	One of Speech, Audio, Visual, 3D Model, Audio-Visual, Haptic, Smell, RADAR, LiDAR, Ultrasound.
Position	
- CartPosition (X,Y,Z)	Array (in metres)
- CartPositionAccuracy (X,Y,Z)	Array of CartPositionAccuracy
- SpherPosition (r,φ,θ)	Array (in metres and degrees)
- SpherPositionAccuracy (r,φ,θ)	Array of SpherPositionAccuracys
Velocity of Position	
- CartVelocity (X,Y,Z)	Array (in metres)
- CartVelocityAccuracy (X,Y,Z)	Array of - CartVelocityAccuracys (X,Y,Z)
- SpherVelocity (r,φ,θ)	Array (in metres and degrees)
- SpherVelocityAccuracy (r,φ,θ)	Array of SpherVelocityAccuracys
Acceleration of Position	
- CartAccel (X,Y,Z)	Array (in metres)
- CartAccelAccuracy (X,Y,Z)	Array of CartAccelAccuracys
- SpherAccel (r,φ,θ)	Array (in metres and degrees)
- SpherAccel (r,φ,θ)	Array (in metres and degrees)
DescrMetadata	Descriptive Metadata

8.72.5 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.

8.73 Space-Time

8.73.1 Definition

Data Type representing the Spatial Attitude and Time information.

8.73.2 Functional Requirements

Space-Time includes Spatial Attitude and Time.

8.73.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/SpaceTime.json>

8.73.4 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

8.73.5 Conformance Testing

A Data instance Conforms with Space-Time (OSD-SPT) 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.

8.74 Spatial Attitude

8.74.1 Definition

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

8.74.2 Functional Requirements

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

8.74.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/SpatialAttitude.json>

8.74.4 Semantics

Table 1 provides the semantics of the components of the Spatial Attitude.

Table 1 – Semantics of the Spatial Attitude

Label	Description
Header	Spatial Attitude Header
- Standard-SpatialAttitude	The characters “OSD-OSA-V”

- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	ID of Virtual Space Object refers to.
ObjectSpatialAttitudeID	Identifier of Object Spatial Attitude.
General	Set of general data
- CoordinateType	One of Cartesian, Spherical, Geodesic, Toroidal.
- ObjectType	One of Digital Human, Generic.
- MediaType	One of Speech, Audio, Visual, Audio-Visual, Haptic, Smell, RADAR, LiDAR, Ultrasound.
Position	As specified by Position
Orientation	As specified by Orientation
DescrMetadata	Descriptive Metadata

8.74.5 Conformance Testing

A Data instance Conforms with V1.2 Spatial Attitude (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.

8.75 Time

8.75.1 Definition

The digital representation of duration of time.

8.75.2 Functional Requirements

Time includes the digital representation of time specified by MPAI or other Standard Setting Organisations as indicated by Time Qualifier

8.75.3 Syntax

<https://schemas.mpai.community/OSD/V1.4/data/Time.json>

8.75.4 Semantics

Label	Description
Header	Time Header
- Standard-Object	The characters “OSD-TIM-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance
TimeID	Identifier of M-Instance.

TimeData	Data about Time
TimeQualifier	End of Time.
DescrMetadata	Descriptive Metadata

8.75.5 Conformance Testing

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

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

8.76 U-Location

8.76.1 Definition

A region of the Universe with Space-Time attributes.

8.76.2 Functional Requirements

A U-Location is a portion of the Universe space with the following characteristics:

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

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

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

8.76.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/ULocation.json>

8.76.4 Semantics

Label	Description
Header	M-Location Header
– Standard-ULocation	The characters “MMM-ULC-V”
– Version	Major version
– Dot-separator	The character “.”
– Subversion	Minor version
MInstanceID	Identifier of M-Instance.

MEnvironmentID	Identifier of M-Environment.
ULocationID	Identifier of U-Location.
ULocation	Set of Data defining U-Location.
– BasicULocations[]	Set of Data defining U-Location.
– BasicULocation	A Basic U-Location
– BasicULocationSpaceTime	Spatial Attitude and Time of Basic U-Location.
– Ulocations[]	Level L-1 U-Location (L may be 0, i.e., Basic U-Location).
– ULocation	An M-Location
– ULocationSpaceTime	Spatial Attitude and Time of Basic U-Location.
Rights	Rights held by or granted to U-Location.
DescrMetadata	Descriptive Metadata.

8.77 Universe-Metaverse Map

8.77.1 Definition

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

8.77.2 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.

8.77.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/UniverseMetaverseMap.json>

8.77.4 Semantics

Label	Description
Header	Universe-Metaverse Map Header
- Standard - UniverseMetaverseMap	The characters “MMM-UMM-V”
- Version	Major version
- Dot-separator	The character “.”
- Subversion	Minor version
MInstanceID	Identifier of M-Instance.
UEnvironmentID	Identifier of U-Environment.

MapData[]	Collection of Map elements.
- ULocationID	ID of specific U-Environment.
- MLocations[]	Corresponding M-Locations.
- MLocationID	ID of Corresponding M-Location
- Items[]	Items at M-LocationID
- ItemID	ID of an Item at M-LocationID
DescrMetadata	Descriptive Metadata.

9 Process Actions

9.1 Introduction

If a Process does not have the necessary Capabilities and/or Rights to perform an Action, the Process (now called Process₁) can request Process₂, that has the necessary Capabilities and Rights. to perform the Action. For instance:

1. Device "*Identifies Data With Qualifier With Model Rights*" means that a named Device requests an Identify Service to provide an Identifier to the Item that includes Data, its Qualifier, and Model Rights (requested Rights) to the Item.
2. User₁ "*MM-Sends Message With Model Rights*" to User₂ means that a named User₁ requests a *Communicate Service* to make available a Message to a named User₂, with a request to grant it Rights represented by Model Rights.

9.2 Process Actions

Process Action is the payload used as Process Action Request or Process Action Response that includes:

1. The Request of Process₁ to Process₂ or the Response of Process₂ to the request of Process₁.
2. A series of Complements *Nil*, *At*, *From*, *To*, or *With* each followed by an Item or a Process.
3. Condition expressed by *If* [Event](#), if the Process Action is a Request.
4. Process Action (PA) Status if the Process Action if a Response.

Table 1 - Component of a Request/Response Process Action

Process Action Request

PA Element	Description
<i>Time</i>	Time of PA request emission and Time of PA request execution.
<i>RQ-Complements</i>	The characters of one or more prepositions: <i>Nil</i> , <i>At</i> , <i>From</i> , <i>To</i> , or <i>With</i> each followed by Item (ID or the Item) or ProcessID.
<i>Condition</i>	The characters <i>If</i> followed by Event .

Process Action Response

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

<i>RS-Complements</i>	The characters of one or more prepositions: <i>Nil, At, From, To, or With</i> each followed by Item (ID or the Item) or ProcessID.
<i>PA Status</i>	If the Process Action was successfully performed, the characters "Ack". If the Process Action was not performed, one of the characters enclosed in "": 1. "Clash", the PA Request causes a clash with an existing object. 2. "FaultyReq", the PA Request is not syntactically correct. 3. "IncID", no Item or Process with such an ID. 4. "IncDQ", the Data and/or its Qualifier are/is incorrect 5. "InsRights", the requesting Process may not obtain the requested Rights. 6. "InsValue", a Wallet referenced in the Transaction has insufficient funds. 7. "LocOOR", no such Location exists. 8. "PostRef", Post refused by Service. 9. "QualNS": Qualifier not supported.

Each linked Action in Table 2 provides Definition, Functional Requirements, Syntax and Semantics of the corresponding Process Actions.

Table 2 - List of Actions linked to Process Action specifications

MMM-AUP	Authenticate	X	MMM-ATP	Author	X	MMM-CVP	Convert	X
MMM-DVP	Discover	X	MMM-EXP	Execute	X	MMM-HDP	Hide	X
MMM-IDP	Identify	X	MMM-IFP	Inform	X	MMM-ITP	Interpret	X
MMM-2AP	MM-Animate	X	MMM-2DP	MM-Add	X	MMM-2MP	MM-Move	X
MMM-2SP	MM-Send	X	MMM-MDP	Modify	X	MMM-3CP	MU-Actuate	X
MMM-3AP	MU-Animate	X	MMM-3DP	MU-Add	X	MMM-3MP	MU-Move	X
MMM-3SP	MU-Send	X	MMM-PTP	Post	X	MMM-PCP	Property Change	X
MMM-RGP	Register	X	MMM-RVP	Resolve	X	MMM-RCP	Rights Change	X
MMM-TRP	Transact	X	MMM-1CP	UM-Capture	X	MMM-1SP	UM-Send	X
MMM-VLP	Validate	X						

General notes:

1. Items, e.g., Rights, Transactions, etc. may be transmitted as IDs or as actual Items.
2. A Process Action Request (PA Request) may include:
 1. Model Rights to request that a Rights Service produce actual Rights based on Model Rights for a non-errored PA Status.
 2. Model Transaction to request that the Transaction service produce an actual Transaction based on Model Transaction for a non-errored PA Status.

3. The JSON Schemas of the Process Actions specify which component of an RQ-Complement and a RS-Complement are required.
4. The Process Action specification 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 instead of, e.g., User.

9.2.1 Authenticate

9.2.1.1 Definition

The Process Action of a User sending an Authentication Item to an Authenticate Service requesting to:

1. Confirm that an Item or Process is what it claims to be.
2. Grant Rights to the Authentication Response Item based on Model Rights.

If the request is accepted, the Authentication Service responds with an Authentication Response Item and actual Rights to the Authentication Response Item.

9.2.1.2 Functional Requirements

The Authenticate Process Action includes the following elements:

1. In the request
 1. The Authentication Request Item.
 2. The Model Rights requested to the Authenticate Service.
 3. A Transaction if requested by the Authenticate Service.
2. In the response
 1. The Authentication Response Item.
 2. The Rights to the Authentication Response Item.

9.2.1.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/AuthenticatePA.json>

9.2.1.4 Semantics

Label	Description
Header	Authenticate Process Action Header
– Standard-AuthenticatePA	The characters “OSD-AUP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
AuthenticatePAData	Data of the Authenticate PA Item.
- Time	Times of PA request emission and PA request execution.
- RQ-Complements	Source Complements.
- <i>With Authentication</i>	The Authentication Item Request.
- <i>With ModelRights</i>	Requested Rights on the Authentication Response.
- <i>With Transaction</i>	Transaction, if requested by Service.
- RS-Complements	Destination Complements.

- <i>With Authentication</i>	Authentication Response Item .
- <i>With Rights</i>	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With PAStatus</i>	PA Request Status.
- Ack	If PA Request could be performed.
- Err	If PA Request could not be performed.
- FaultyPA	Faulty PA Request.
- IncID	Incorrect ID.
- MLocOOR	M-Location out of range.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.2 Discover

9.2.2.1 Definition

The Action of a User requesting a Discover Service to

1. Provide information on Items or Processes relevant to the Discovery Request Item.
2. Grant Rights to the Discovery Response Item based on Model Rights.

If the request is accepted, the Discover Service responds with an Discovery Response Item and actual Rights to the Discovery Response Item.

9.2.2.2 Functional Requirements

The Discover Process Action includes:

In the request

- The Discovery Request Item.
- The Model Rights to the Discovery Response Item.
- The Process that should receive the Discovery Response Item.
- A Transaction if requested by the Discover Service.

In the response

- The Discovery Response Item.
- The Rights to the Discovery Response Item per Model Rights if request is accepted.

9.2.2.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/DiscoverPA.json>

9.2.2.4 Semantics

Label	Description
Header	Discover Process Action Header
- Standard-DiscoverPA	The characters “MMM-DSP-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters

MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
DiscoverPAData	Data included in the Discover Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in PA Tequest.
- <i>With</i> DiscoveryID	ID of Discovery Request Item.
- <i>With</i> ModelRights	Rights that are requested to the Discovery Response Item.
- <i>At</i> ProcessID	The Process receiving the Discovery Response Item.
- <i>With</i> Transaction	Transaction made to Discover Service if requested.
- RS-Complements	Complements in PA Response.
- <i>With</i> Discovery	ID of Discovery Response Item.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With</i> PAStatus	Status of PA Request.
- Ack	If PA Request was performed.
- Err	If PA Request was not performed.
- FaultyPA	Faulty PA Request.
- IncID	Incorrect ID.
- MLocOOR	M-Location out of range.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.3 Identify

9.2.3.1 Definition

The Action of a Process requesting an Identify 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.

If the request is accepted, the Identify Service responds with the ItemID and the actual Rights to the Item.

9.2.3.2 Functional Requirements

The Identify Process Action includes:

In the request

- The Data to be Identified and the Qualifier of the Data.
- The Model Rights to the Identified Item.

- The Process that holds the Data.
- A Transaction if requested by the Identify Service.

In the response

- The Item produced.
- The Rights to the Item per the Model Rights if request is successful.

9.2.3.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/data/Identify.json>

9.2.3.4 Semantics

Label	Description
Header	Identify Process Action Header
- Standard-IdentifyPA	The characters “MMM-IDF-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
IdentifyPAData	Data included in the Identify Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in the PA Requests.
- Nil Data	Data to be Identified.
- With Qualifier	Qualifier of Data.
- At ProcessID	Process holding Data to be Identified.
- With ModelRights	Requested Rights to the Identify Item.
- With Transaction	Transaction made to Identify Service if requested.
- RS-Complements	Complements in the Response Item.
- Nil ItemID	Item produced.
- With Rights	Actual Rights based on Model Rights if PA Request is accepted.
- With PAStatus	Possible Error Messages.
- Ack	If PA Request could be performed
- Err	If Error was found

– FaultyPA	Faulty PA Request.
– IncDQ	Incorrect Data and/or Qualifier.
– InsRights	Insufficient Rights.
– InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.4 MM-Animate

9.2.4.1 Definition

The Action of a Process requesting that a Locate Service:

1. Animate an MM-Embedded Item with a Stream or Command Item. A Command is an atomic element of a Stream.
2. Grant Rights based on Model Rights to the MM-Animated Item.

If the request is accepted, the Locate Service responds with actual Rights to the MM-Animated Item.

9.2.4.2 Functional Requirements

The MM-Animate Process Action includes:

In the request

- The Item to be MM-Animated.
- The M-Location where the Item to be MM-Animated is MM-Added.
- The Stream used to MM-Animate Item.
- The Model Rights to the MM-Animated Item.
- A Transaction if requested by the Locate Service.

In the response

- The Rights per the Model Rights if request is successful.

9.2.4.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/MMAnimatePA.json>

9.2.4.4 Semantics

Label	Description
Header	Convert Process Action Header

– Standard-ConvertPA	The characters “MMM-MAP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
MMAnimPAData	Data included in the MM-Animate Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in MM-Animate Request.
- <i>Nil</i> ItemID	Item to be MM-Animated.
- <i>At</i> MLocationID	M-Location where the Item is MM-Embedded.
- <i>With</i> Stream	Stream Item used to MM-Animate Item.
- <i>With</i> ModelRights	Requested Rights to the MM-Animated Item.
- <i>With</i> Transaction	Transaction made to Locate Service if requested.
- RS-Complements	Complements in MM-Animate Response.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
– <i>With</i> PAStatus	Possible Error Messages.
– Ack	If PA Request could be performed
– Err	If Error was found
– Clash	Clashes with other Items.
– FaultyPA	Faulty PA Request.
– IncID	Incorrect ID.
– InsRights	Insufficient Rights.
– InsValue	Insufficient Value in Wallet.
- LocOOR	No such Location.
DescMetadata	Descriptive Metadata.

9.2.5 MM-Send

9.2.5.1 Definition

The Action of a Process requesting a Communicate Service:

1. To send a Message to other Processes.
2. To grant to the receiving Processes Rights to the content of the Message.

If the request is accepted, the Communicate Service delivers the Message providing actual Rights to the MM-Sent Item.

9.2.5.2 Functional Requirements

The MM-Send Process Action includes:

In the request

- The Message or MessageID to be sent to the other Processes.
- The Processes receiving the Message.
- The Model Rights to the Message granted to the other Processes.
- A Transaction if requested by the Communicate Service.

In the response

- The Rights to the Message granted to the other Processes per the Model Rights if the request is successful.

9.2.5.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/MMSendPA.json>

9.2.5.4 Semantics

Label	Description
Header	MM-Sens Process Action Header
– Standard-MMSendPA	The characters “MMM-MSP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
MMSendPAData	Data included in the MM-Send Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in the PA Request.
- <i>Nil</i> MessageID or Message	Item to be MM-Sent.
- <i>To</i> ProcessIDs	ID of Process receiving the Message where the Item is to be MM-Sent.
- <i>With</i> ModelRights	Rights that are requested to be granted to the other Process.
- <i>With</i> Transaction	Transaction made to Communication Service if requested.
- D-Complements	Complements in the PA Response.

- <i>With Rights</i>	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With PAStatus</i>	PA Request Status.
- <i>Ack</i>	If PA Request could be performed.
- <i>Err</i>	If PA Request could not be performed.
- <i>FaultyPA</i>	Faulty PA Request.
- <i>InclID</i>	Incorrect ID.
- <i>InsRights</i>	Insufficient Rights.
- <i>InsValue</i>	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.6 MU-Animate

9.2.7 Definition

The Action of a User requesting that an Export-Import Service:

1. Animate an MU-Added R-Item with

1.1 A stream Item equivalent to the one that is used to animate the Item of which the R-Item is an Analogue Twin, or

1.2 Specific commands applied to an Item in the M-Instance that should be reflected to the R-Item that is its Analogue Twin.

2. Grant Rights based on Model Rights to the MU-Animated R-Item.

If the request is accepted, the Export-Import Service responds MU-Animates the the Item providing actual Rights to the requesting User.

9.2.7.1 Functional Requirements

The MU-Animate Process Action includes:

In the request

- The R-Item to be MU-Animated.

- The U-Location where the R-Item is to be MU-Animated.

- A Stream or a Command used to MU-Animate the R-Item. A Command is an atomic element of a Stream.

- The Model Rights to the MM-Animated Item.

- A Transaction if requested by the Export-Import Service.

In the response

- The Rights per the Model Rights if request is successful.

9.2.7.2 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/MUAnimatePA.json>

9.2.7.3 Semantics

Label	Description
Header	MU-Animate Process Action Header
– Standard-MUAnimatePA	The characters “MMM-3AP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
UEnvironmentID	Identifier of U-Environment.
MUAnimatePAData	Data included in the MU-Animate Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in the PA Request.
- <i>Nil</i> RItem	R-Item to be MU-Animated.
- <i>At</i> ULocationID	U-Location where the R-Item is to be MU-Animated.
- <i>With</i> Stream	Stream Item used to MU-Animate R-Item.
- <i>With</i> ModelRights	Requested Rights to the MU-Animated R-Item.
- <i>With</i> Transaction	Transaction made to Export-Import Service if requested.
- RS-Complements	Complements in the PA-Response.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
– <i>With</i> PAStatus	Status of PA Request.
– Ack	If PA Request could be performed
– Err	If Error was found
– Clash	MU-Animated Item clashes with an existing R-Item,
– FaultyPA	Faulty PA Request.
– IncID	Incorrect ID.
– ULocOOR	M-Location out of range.
– InsRights	Insufficient Rights.
– InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.8 MU-Send

9.2.8.1 Definition

The Action of a Process requesting an Export-Import Service:

1. To send a Message to a Process in the U-Environment.
2. To grant Rights to the receiving Process to the content of the Message.

If the request is accepted, the Export-Import Service delivers the Message to the Process in the U-Environment and grants actual Rights to the content of the Message.

9.2.8.2 Functional Requirements

The MU-Send Process Action includes:

In the request

- The Message sent to the Processes in the U-Environment.
- The ID of the Processes receiving the Message.
- The Model Rights to the Message.
- A Transaction if requested by the Export-Import Service.

In the response

- The Rights to the Message granted to the Processes in the U-Environment per the Model Rights if request is successful.

9.2.8.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/MUSendPA.json>

9.2.8.4 Semantics

Label	Description
Header	MU-Send Process Action Header
– Standard-MUSendPA	The characters “MMM-3SP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
UEnvironmentID	Identifier of U-Environment.

MUSendPAData	Data included in the MU-Send Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in the PA Request.
- <i>Nil</i> MessageID or Message	Item to be MU-Sent.
- <i>To</i> ProcessIDs	U-Location where the Item is to be MU-Sent.
- <i>With</i> ModelRights	Rights that are requested to be granted to the Process in the U-Environment.
- <i>With</i> Transaction	Transaction made to Import-Export Service if requested.
- D-Complements	Complements in the PA Response.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With</i> PAStatus	Possible Error Messages.
- Ack	If PA Request could be performed.
- Err	If PA Request could not be performed.
- FaultyPA	Faulty PA Request.
- IncID	Incorrect ID.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.

9.2.9 Register

9.2.9.1 Definition

The Action of a human requesting that a Register Service:

1. Open an Account based on the human's Personal Data.
2. Grant Rights to the Registering human's Processes and Items.

If the request is accepted, the human will receive an Account.

9.2.9.2 Functional Requirements

The Register Process Action includes:

In the request

- Registering Human's Personal Profile.
- A Transaction if requested by the Import-Export Service.

In the response

- Account.

9.2.9.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/RegisterPA.json>

9.2.9.4 Semantics

Label	Description
Header	Register Process Action Header
– Standard-RegisterPA	The characters “USP-RGP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
RegisterPAData	Data included in the Register Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- S-Complements	Complements in PA Request.
- <i>Nil</i> PersonalProfile	Registering Human's Personal Profile.
- <i>With</i> Transaction	Transaction made to Registration Service if requested.
- D-Complements	Complements in PA Response.
- <i>Nil</i> Account	Registering Human's Account.
– <i>With</i> PAStatus	Possible Error Messages.
– Ack	If PA Request could be performed
– Err	If Error was found
– FaultyPA	Faulty PA Request.
– IncID	Incorrect ID.
– InsRights	Insufficient Rights.
– InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.10 Transact

9.2.10.1 Definition

The Action of a User₁ (“sender”) requesting that a Transaction Service:

1. Assign Rights to an Asset to User₂ (“receiver”).
2. Cause:

- 2.1. Wallet₁ of User₁ to be increased by Value₁.
- 2.2. Wallet₂ of User₂ to be decreased by Value₂.
- 2.3. Wallet₃ of the Service enabling/facilitating the Transaction to be increased by Value₃ (optionally).

9.2.10.2 Functional Requirements

The Transact Process Action includes:

In the request

- Transaction Item used in transaction.

In the response

- No response elements (Transaction Service notifies).

9.2.10.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/TransactPA.json>

9.2.10.4 Semantics

Label	Description
Header	Transact Process Action Header
– Standard-TransactPA	The characters “USP-TRP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
TransactPAData	Data included in the Transact Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in PA Request.
- <i>With</i> Transaction	A Transaction Item.
- RS-Complements	Complements in PA Response.
– <i>With</i> PAStatus	Status of PA Request.
– Ack	If PA Request could be performed.
– Err	If PA Request could not be performed.
– FaultyPA	Faulty PA Request.
– IncID	Incorrect ID.

– InsRights	Insufficient Rights.
– InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.11 Validate

9.2.11.1 Definition

The Action of a Process requesting a Rights Service to:

1. Confirm that an Item or Process has the Rights it claims to have with a Validation Request Item.
2. Grant Rights to the Validation Response Item based on Model Rights.

If the request is accepted, the Rights Service responds with an Validation Response Item and grants actual Rights to the Validation Response Item.

9.2.11.2 Functional Requirements

The Validate Process Action includes:

In the request

- The Validation Request Item.
- The requested Model Rights to the Validation Response Item.
- A Transaction if requested by the Rights Service.

In the response

- The Validation Item Response.
- The Rights to the Validation Item Response if request is successful.

9.2.11.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/ValidatePA.json>

9.2.11.4 Semantics

Label	Description
Header	Validate Process Action Header
– Standard-ValidatePA	The characters “USP-VLP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
ValidatePAData	Data included in the Validate Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in PA Request.
- <i>With</i> Validation	Validate Request Item.

- <i>With</i> ModelRights	Rights requested to Validation Response.
- <i>With</i> Transaction	Transaction made to Import-Export Service if requested.
- RS-Complements	Complements in PA Response.
- <i>With</i> Validation	Validate Request Item.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With</i> PAStatus	Status of PA Request.
- Ack	If PA Request could be performed.
- Err	If PA Request could not be performed.
- FaultyPA	Faulty PA Request.
- IncID	Incorrect ID.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.

9.2.12 Author

9.2.12.1 *Definition*

The Action of a User requesting an Author Service to:

1. Produce an Item based on provided Items.
2. Grant Rights to the User and the Authored Item based on Model Rights.

If the request is accepted, the User will receive Items and Rights to Items based on Model Licence.

9.2.12.2 *Functional Requirements*

The Author Process Action includes:

In the request:

- Items used by the Author Service.
- The Model Rights requested for the Authored Items
- Process that will host the Authored Items.
- A Transaction if requested by the Authenticate Service.

In the response

- The Authored Item.
- The Rights to the Authored Item per the Model Rights.

9.2.12.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/AuthorPA.json>

9.2.12.4 Semantics

Label	Description
Header	Author Process Action Header
- Standard-AuthorPA	The characters “OSD-ATP-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
AuthorPAData	Data of Author Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in PA Request.
- <i>With</i> ItemIDs	Items requested by Author Service to Author the Items.
- <i>With</i> ModelRights	The Rights requested to the Authored Item.
- <i>At</i> Process	Process that will receive the Authored Items.
- <i>With</i> Transaction	Transaction to be made to Author Service if requested.
- RS-Complements	Complements in PA Response.
- <i>Nil</i> ItemIDs	Items produced by Author Service.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With</i> PAStatus	Status of PA Request.
- Ack	If PA Request could be performed.
- Err	If PA Request could not be performed.
- FaultyPA	Faulty PA Request.
- IncID	Incorrect ID.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.13 Execute

9.2.13.1 Definition

The Action of a Process requesting an Execution Service to:

1. Execute a Program.
2. Grant Rights to the Items produced by the Executed Program based on Model Rights.

If the request is accepted, the Execute Service executes the Program and grants Rights to the Items produced by it.

9.2.13.2 Functional Requirements

The Execute Process Action includes:

In the request

- The ID of the Program to be Executed or the Process holding the Program to be Executed.
- The Model Rights requested to the Items produced by the Executed Program.
- The Process that should receive the Items produced by the Executed Program.
- A Transaction if requested by the Execution Service.

In the response

- The Items produced by the Executed Program.
- The Rights to the Items per the Model Rights if request is successful.

9.2.13.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/ExecutePA.json>

9.2.13.4 Semantics

Label	Description
Header	Execute Process Action Header
- Standard-ExecutePA	The characters “MMM-EXP-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
ExecutePAData	Data included in the Execute Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Source Complements.
- <i>Nil</i> ProgramID	Program to be Executed.
- <i>With</i> ItemIDs	Items requested by Execute Program.

- <i>With</i> ModelRights	Requested Rights to the Items produced by the Executed Program.
- <i>At</i> ProcessID	Process holding Items resulting from Execution.
- <i>With</i> Transaction	Transaction made to Execute Service if requested.
- RS-Complements	Destination Complements.
- <i>Nil</i> ItemIDs	Items produced by the Executed Program.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With</i> PAStatus	Possible Error Messages.
- Ack	If PA Request could be performed
- Err	If Error was found
- FaultyPA	Faulty PA Request.
- InclID	Incorrect ID.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.14 Inform

9.2.14.1 Definition

The Action of a User requesting an Inform Service to:

1. Provide information about an Item or a Process relevant to the Information Request Item.
2. Grant Rights to the Information Response Item based on Model Rights.

If the request is accepted, the Inform Service responds with an Information Item Response.

9.2.14.2 Functional Requirements

The Inform Process Action includes the following elements:

1. In the request
 1. The Information Request Item.
 2. The Model Rights to the Inform Response Item.
 3. The ID of the User that should receive the Information Response Item.
 4. A Transaction if requested by the Inform Service.
2. In the response
 1. The Information Response Item.
 2. The Rights to the Information Response Item per the Model Rights.

9.2.14.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/InformPA.json>

9.2.14.4 Semantics

Label	Description
Header	Inform Process Action Header
- Standard-InformPA	The characters “OSD-IFP-V”
- Version	Major version – 1 or 2 characters

– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
InformPAData	Data included in the Inform Item.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in PA Request.
- <i>With</i> Information	The Information Request Item.
- <i>With</i> ModelRights	Requested Rights on the Information Response Item.
- <i>At</i> User	User receiving Information Response Item with Rights.
- <i>With</i> Transaction	If Inform Service requests it.
- RS-Complements	Complements in PA Response.
- <i>With</i> Information	Information Response Item.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted..
– <i>With</i> PAStatus	Possible Error Messages.
– Ack	If PA Request could be performed
– Err	If Error was found
– FaultyPA	Faulty PA Request.
– IncID	Incorrect ID.
– InsRights	Insufficient Rights.
– InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.15 MM-Add

9.2.15.1 Definition

The Action of a User requesting a Locate Service to

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

If the request is accepted, the Locate Service responds places the Item at the requested M-Location providing actual Rights to the requesting User..

9.2.15.2 Functional Requirements

The MM-Add Process Action includes:

In the request

- The Item to be MM-Added.
- The Spatial Attitude of the Item to be MM-Added.
- The M-Location where the Item is to be MM-Added.
- The Perceptibility State
- The Model Rights to the MM-Added Item.
- A Transaction if requested by the Locate Service.

In the response

- The Rights per the Model Rights if request is successful.

9.2.15.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/MMAddPA.json>

9.2.15.4 Semantics

Label	Description
Header	MM-Add Process Action Header
– Standard-MMAddPA	The characters “MMM-2DP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
MMAddPAData	Data included in the MM-Add Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Source Complements.
- <i>Nil</i> Item	Item to be MM-Added.
- <i>At</i> MLocationID	M-Location where the Item is to be MM-Added.
- <i>With</i> SpatialAttitude	The Spatial Attitude of the Item to be MM-Added.
- <i>With</i> PerceptibilityState	0=Imperceptible, 1=Perceptible.
- <i>With</i> ModelRights	Rights that are requested to the MM-Added Item.
- <i>With</i> Transaction	Transaction made to Locate Service if requested.
- RS-Complements	Destination Complements.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
– <i>With</i> PAStatus	Possible Error Messages.
– Ack	If PA Request could be performed
– Err	If Error was found
- Clash	MM-Added Item clashes with an existing Item.
– FaultyPA	Faulty PA Request.
– InclID	Incorrect ID.
– MLocOOR	M-Location out of range.
– InsRights	Insufficient Rights.
– InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.16 Modify

9.2.16.1 Definition

The Action of a User requesting an Identify Service to:

1. Produce a new Item starting from an existing Item using new Data and Qualifier.
2. Grant Rights to the Modified Item based on Model Rights.

If the request is accepted, the Identify Service provides the ID of the Modified Item and grants actual Rights to the Modified Item.

9.2.16.2 Functional Requirements

The MM-Send Process Action includes:

In the request

- The Item to be Modified.
- The Process holding the Item.
- Data and Qualifier to Modify the Item
- The Model Rights to the Modified Item requested.
- The Process receiving the Modified Item.
- A Transaction if requested by the Identify Service.

In the response

- The Modified Item.
- The Rights to the Modified Item per the Model Rights if request is successful.

9.2.16.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/ModifyPA.json>

9.2.16.4 Semantics

Label	Description
Header	Modify Process Action Header
– Standard-ModifyPA	The characters “MMM-MOP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
ModifyPAData	Data included in the Modify Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in PA Request.
- Nil ItemID	Item to be MM-Sent.
- At ProcessID	Process holding the Item.

- <i>With Data</i>	Data To Modify the Item.
- <i>With Qualifier</i>	Data Qualifier.
- <i>With ModelRights</i>	Rights that are requested to the Modified Item.
- <i>At ProcessID</i>	Process Receiving the Item.
- <i>With Transaction</i>	Transaction made to Identify Service if requested.
- D-Complements	Complements in PA Response.
- <i>Nil ItemID</i>	Item to be Modified.
- <i>With Rights</i>	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With PAStatus</i>	Possible Error Messages.
- Ack	If PA Request could be performed
- Err	If Error was found
- FaultyPA	Faulty PA Request.
- IncID	Incorrect ID.
- IncDQ	Incorrect Data and/or Qualifier.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.17 MU-Add

9.2.17.1 Definition

The Action of a User requesting an Export-Import Service to:

1. Place an R-Item at a U-Location with a Spatial Attitude.
2. Grant Rights to the MU-Added Item based on Model Rights.

If the request is accepted, the Export-Import Service places the R-Item and grants actual Rights to the R-Item.

9.2.17.2 Functional Requirements

The MU-Add Process Action includes:

In the request

- The R-Item to be MU-Added.
- The M-Location where the R-Item is to be MM-Added.
- The Spatial Attitude of the R-Item to be MU-Added.
- The Model Rights to the MU-Added Item.
- A Transaction if requested by the Export-Import Service.

In the response

- The Rights per the Model Rights if request is successful.

9.2.17.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/MUAddPA.json>

9.2.17.4 Semantics

Label	Description
Header	MU-Add Process Action Header
– Standard-MYAddPA	The characters “MMM-3DP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of the M-Environment.
UEnvironmentID	Identifier of the U-Environment.
AddPAData	Data included in the MU-Add Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in the PA Request.
- <i>Nil</i> RItemID	R-Item to be MU-Added.
- <i>At</i> ULocationID	U-Location where the Item is to be MU-Added.
- <i>With</i> SpatialAttitude	The Spatial Attitude of the Item to be MU-Added.
- <i>With</i> ModelRights	Rights that are requested to the MU-Added Item.
- <i>With</i> Transaction	Transaction made to Locate Service if requested.
- D-Complements	Complements in the PA Response.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
– <i>With</i> PAMStatus	Status of PA Request.
– Ack	If PA Request could be performed
– Err	If Error was found
– Clash	R-Item clashes with another R-Item.
– FaultyPA	Faulty PA Request.
– IncID	Incorrect ID.
– ULocOOR	M-Location out of range.
– InsRights	Insufficient Rights.
– InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.18 Post

9.2.18.1 Definition

The Action of a User requesting that a Marketplace Service to:

1. Include an Asset to its repository of Assets.
2. Grant to the Marketplace Service Rights to the Asset based on Model Rights.

9.2.18.2 Functional Requirements

The Post Process Action includes:

In the request

- Asset User wishes to Post.
- The ID of the Marketplace Service.
- The Model Rights to the Asset to be granted to the Marketplace Service.
- A Transaction if requested by the Marketplace Service.

In the response

- The Rights to the Message granted to the Marketplace Service if the Marketplace Service agrees.

9.2.18.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/PostPA.json>

9.2.18.4 Semantics

Label	Description
Header	Post Process Action Header
– Standard-PostPA	The characters “USP-PTP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
PostPAData	Data included in the Post Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in PA Request.
- Nil AssetID	ID of Asset Posted.
- To ServiceID	ID of the Marketplace Service where the Asset is Posted.
- With ModelRights	Rights granted to the Marketplace Service if the Marketplace Service agrees.
- With Transaction	Transaction made to Marketplace Service if requested.
- RS-Complements	Complements in PA Response.

- <i>With Rights</i>	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With PAStatus</i>	PA Request Status.
- <i>Ack</i>	If PA Request could be performed.
- <i>Err</i>	If PA Request could not be performed.
- <i>FaultyPA</i>	Faulty PA Request.
- <i>InclID</i>	Incorrect ID.
- <i>InsRights</i>	Insufficient Rights.
- <i>InsValue</i>	Insufficient Value in Wallet.
- <i>PostRef</i>	Post refused
DescMetadata	Descriptive Metadata.

9.2.19 Resolve

9.2.19.1 *Definition*

The Action of:

1. A Process requesting that a Resolution Service set up a multi-metaverse session (Session) involving two/more than two Processes in two/more than two M-Instances.
2. A Process or Resolution Service responding to a multi-metaverse Session request.

9.2.19.2 *Functional Requirements*

The Resolve Process Action includes:

In the request

- Resolution Request Item used to request a Session.
- Resolution Service ID.
- A Transaction if requested by the Resolution Service.

In the response

- A Resolution Response Item.

9.2.19.3 *Syntax*

<https://schemas.mpai.community/MMM4/V2.1/actions/ResolvePA.json>

9.2.19.4 *Semantics*

Label	Description
Header	Resolve Process Action Header
- Standard-ResolvePA	The characters “USP-RSP-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters

MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
ResolutionPAData	Data included in the Resolution Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- S-Complements	Source Complements.
- <i>Nil</i> Resolution	Resolution Request.
- <i>To</i> ServiceID	ID of the Resolution Service.
- <i>With</i> Transaction	Transaction made to Resolution Service if requested.
- D-Complements	Destination Complements.
- <i>Nil</i> Resolution	Resolution Response.
- <i>With</i> PAStatus	Possible Error Messages.
- Ack	If PA Request could be performed.
- Err	If PA Request could not be performed.
- FaultyPA	Faulty PA Request.
- IncID	Incorrect ID.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.20 UM-Capture

9.2.20.1 Definition

The Action of a Device requesting an Export-Import Service to capture a scene at a U-Location with a Spatial Attitude based on the Rights to that U-Location. The MM-Captured Data reside in the Device.

If the request is accepted, the Device digitally represent the scene with a Qualifier.

9.2.20.2 Functional Requirements

MU-Capture Process Action includes:

In the request

- The U-Location wrapping the object or scene to be UM-Captured.
- The Qualifier(s) of the UM-Captured Data.
- The Model Rights to the MU-Captured Data.
- A Transaction if requested by the Export-Import Service.

In the response

- The MU-Captured Data.
- The Spatial Attitude of the UM-Captured Data.
- The Rights to the UM-Captured Data per the Model Rights if request is successful.

9.2.20.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/UMCapturePA.json>

9.2.20.4 Semantics

Label	Description
Header	UM-Capture Process Action Header
– Standard-UMCapturePA	The characters “MMM-1CP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
UMCapturePAData	Data included in the UM-Capture Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in the PA Request.
- <i>Nil</i> ULocation	U-Location wrapping object or scene.
- <i>With</i> Qualifier(s)	The Qualifier(s) to be used to UM-Capture the Data.
- <i>With</i> ModelRights	Requested Rights to the UM-Captured Data.
- <i>With</i> Transaction	Transaction made to the Import-Export Service if requested.
- RS-Complements	Complements in the PA Response.
- <i>Nil</i> Data	Data produced by the UM-Capture Process Action.
- <i>With</i> SpatialAttitude	The Spatial Attitude of the UM-Captured Data.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With</i> PAStatus	Status of PA Request.
- Ack	If PA Request could be performed.
- Err	If PA Request could not be performed.
- FaultyPA	Faulty PA Request.
- IncID	Incorrect ID.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.
- ULocOOR	No such U-Location.

9.2.21 Convert

9.2.21.1 Definition

The Action of a Process requesting a Convert 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.

If the request is accepted, the Convert Service responds with an Item and actual Rights to the Item.

9.2.21.2 Functional Requirements

The Convert Process Action includes:

In the request

- One of
 - The ID of Item whose Data is to be Converted.
 - The Process holding the Item whose Data is to be Converted.
 - The M-Location wrapping the Item whose Data is to be Converted is MM-Embedded.
 - The Model Qualifier to be used to Convert the Item.
 - The Model Rights to the Converted Item.
 - The Process that should receive the Converted Item.
 - A Transaction if requested by the Convert Service.

In the response

- The Converted Item.
- The Rights per the Model Rights if the request is successful.

9.2.21.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/ConvertPA.json>

9.2.21.4 Semantics

Label	Description
Header	Convert Process Action Header
- Standard-ConvertPA	The characters “MMM-CVP-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
ConvertPAData	Data included in the Convert Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Source Complements.
- <i>Nil</i> ItemID	Item whose Data is requested to be Converted.
- <i>At</i> ProcessID	Process holding Item whose Data is requested to be Converted.
- <i>At</i> MLocation	M-Location where the Item is MM-Embedded.
- <i>With</i> ModelQualifier	Qualifier to be used to Convert the Data of the Item.
- <i>At</i> ProcessID	The Process that should receive the Converted Item.

- <i>With</i> ModelRights	Rights that are requested to the Converted Item.
- <i>With</i> Transaction	Transaction made to Convert Service if requested.
- RS-Complements	Destination Complements.
- <i>Nil</i> ItemIDs	Item whose Data have been Convertedd.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With</i> PAStatus	Possible Error Messages.
- Ack	If PA Request could be performed
- Err	If Error was found
- FaultyPA	Faulty PA Request.
- InclID	Incorrect ID.
- MLocOOR	M-Location out of range.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.
- QualNS	Qualifier not supported.
DescMetadata	Descriptive Metadata.

9.2.22 Hide

9.2.22.1 Definition

The Action of a Process requesting an Identify Service to:

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

If the request is accepted, the Identify Service responds with the actual Rights to the Hidden Item.

9.2.22.2 Functional Requirements

The Hide Process Action includes:

In the request

- One of
- The ID of the Item to be Hidden.
- The M-Location wrapping the Item to be Hidden.
- The requested Model Rights to the Hidden Item.
- A Transaction if requested by the Execution Service.

In the response

- The Rights per the Model Rights if request is successful.

9.2.22.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/HidePA.json>

9.2.22.4 Semantics

Label	Description
Header	Hide Process Action Header

– Standard-HidePA	The characters “MMM-HDP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
HidePAData	Data included in the Hide Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in the PA Request.
- One of	
- <i>Nil</i> HideID	Item to be Hidden.
- <i>At</i> MLocationID	M-Location where the Item to be Hidden is MM-Embedded,
- <i>With</i> ModelRights	Requested Rights to the Hidden Item.
- <i>With</i> Transaction	Transaction made to Identify Service if requested.
- RS-Complements	Complements in the PA Request.
- <i>Nil</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
– <i>With</i> PAStatus	Possible Error Messages.
– Ack	If PA Request could be performed
– Err	If Error was found
– FaultyPA	Faulty PA Request.
– IncID	Incorrect ID.
– InsRights	Insufficient Rights.
– InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.23 Interpret

9.2.23.1 Definition

The Action of a User requesting an Interpret Service to:

1. Interpret an Item relevant to the Interpretation Request Item.
2. Grant Rights to Interpretation Response Item based on Model Rights.

If the request is accepted, the Interpret Service responds with an Interpretation Item.

9.2.23.2 Functional Requirements

The Interpret Process Action includes the following elements:

1. In the request
 1. The Interpretation Request Item.
 2. The Model Rights requested to the Interpretation Response Item.

3. The ID of the User that should receive the Interpretation Response Item.
4. A Transaction if requested by the Interpret Service.
2. In the response
 1. The Interpretation Response Item.
 2. The Rights to the Interpretation Response Item per the Model Rights.

9.2.23.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/InterpretPA.json>

9.2.23.4 Semantics

Label	Description
Header	Interpret Process Action Header
– Standard-InterpretPA	The characters “OSD-ITP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
InterpretPAData	Data included in Interpretation Item.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in PA Request.
- <i>With</i> Interpretation	The Interpretation Request Item.
- <i>With</i> ModelRights	Requested Rights on the Interpretation Response Item.
- <i>At</i> User	User receiving Interpretation Response Item with Rights.
- With Transaction	If Interpret Service requests it.
- RS-Complements	Complements in PA Response.
- <i>With</i> Interpretation	Interpretation Response Item.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted..
– <i>With</i> PAStatus	Possible Error Messages.
– Ack	If PA Request could be performed
– Err	If Error was found
– FaultyPA	Faulty PA Request.
– IncDQ	Incorrect Data and/or Qualifier.
– InsRights	Insufficient Rights.
– InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.24 MM-Move

9.2.24.1 Definition

The Action of a User requesting a Location Service to:

1. Move an MM-Added Item from an M-Location to another M-Location along a Trajectory.
2. Preserve any PropertyChange that may have been applied to the Item.
3. Grant Rights to the MM-Moved Item based on Model Rights.

If the request is accepted, the Locate Service responds MM-Moves the Item and provides actual Rights to the requesting User.

9.2.24.2 Functional Requirements

The MM-Move Process Action includes:

In the request

- The Item to be MM-Moved.
- The M-Location where the Item is currently MM-Added.
- The M-Location where the Item is to be MM-Moved.
- Trajectory the Item should follow in the motion.
- The Spatial Attitude the Item shall have at the new M-Location.
- The Model Rights to the MM-Moved Item.
- A Transaction if requested by the Locate Service.

In the response

- The Rights per the Model Rights if request is successful.

9.2.24.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/MMMovePA.json>

9.2.24.4 Semantics

Label	Description
Header	MM-Move Process Action Header
- Standard-MMMovePA	The characters “MMM-2MP-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
MMMovePAData	Data included in the MM-Move Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in the PA Request.
- Nil ItemID	Item to be MM-Moved.
- From MLocationID	M-Location where the Item is MM-Added.

- <i>To</i> MLocationID	M-Location where the Item is to be MM-Moved.
- <i>With</i> Trajectory	Trajectory the Item should follow in moving. If absent, Item is tele-transported.
- <i>With</i> SpatialAttitude	The Spatial Attitude of the Item at the new M-Location.
- <i>With</i> ModelRights	Rights that are requested to the MM-Moved Item.
- <i>With</i> Transaction	Transaction made to Locate Service if requested.
- RS-Complements	Complements in the PA Response
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With</i> PAStatus	Possible Error Messages.
- Ack	If PA Request could be performed.
- Err	If Error was found.
- Clash	MM-Moved Item clashes with existing Item.
- FaultyPA	Faulty PA Request.
- IncID	Incorrect ID.
- MLocOOR	M-Location out of range.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.25 MU-Actuate

9.2.25.1 Definition

The Action of a User requesting an Export-Import Service to:

1. Render an Item MM-Embedded at an M-Location with a Spatial Attitude as an R-Item at a U-Location with a Spatial Attitude.
2. Additionally render the Scene of the M-Location that includes the Item, if the M-Location field is present.
3. Grant Rights to the U-Location based on Model Rights.

If the request is accepted, the Export-Import Service places the R-Item at the requested U-Location providing actual Rights to the requesting User.

9.2.25.2 Functional Requirements

The MU-Actuate Process Action includes:

In the request

- The Item to be MU-Actuated or the M-Location where the Item is MM-Embedded.
- The Spatial Attitude of the Item.
- The U-Location where the R-Item is to be MU-Actuated.
- The R-Item Type (one of the three Types of R-Item) used to render the Item.
- The Spatial Attitude of the R-Item to be MU-Actuated at the U-Location.
- The Model Rights to the MU-Actuated Item.
- A Transaction if requested by the Export-Import Service.

In the response

- The MU-Actuated R-Item and its ID (if any is provided).
- The Rights to the MU-Actuated R-Item per the Model Rights if request is successful.

9.2.25.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/MUActuatePA.json>

9.2.25.4 Semantics

Label	Description
Header	MuActuate Process Action Header
– Standard-MUActuatePA	The characters “MMM-3CP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”
– Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
MUActuatePAData	Data included in the MU-Actuate Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in the PA Request.
- oneOf	One of
- Nil ItemID	Item to be MU-Actuated.
- At MLocation	M-Location where Item to be MU-Actuated is MM-Embedded.
- At ULocation	The U-Location where the Item is MU-Actuated.
- With RItemType	The Type of R-Item used to MU-Actuate the Item.
- With SpatialAttitude	The Spatial Attitude of the Item MU-Actuated as an R-Item.
- With ModelRights	Rights that are requested to the Modified Item.
- With Transaction	Transaction made to Import-Export Service if requested.
- RS-Complements	Complements in the PA Response.
- Nil RItem	R-Item produced by the MU-Actuate Process Action.
- With Rights	Actual Rights based on Model Rights if PA Request is accepted.
– With PAStatus	Possible Error Messages.
– Ack	If PA Request could be performed
– Err	If Error was found
– Clash	MM-Embedded Item clashes with an existing object,
– FaultyPA	Faulty PA Request.
– InclID	Incorrect ID.

– MLocOOR	M-Location out of range.
– ULocOOR	U-Location out of range.
– InsRights	Insufficient Rights.
– InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.26 MU-Move

9.2.26.1 Definition

The Action of a User requesting a Location Service to:

1. Move an MU-Added R-Item from a U-Location to another U-Location via a Trajectory. When at the new U-Location, the R-Item shall have an assigned Spatial Attitude.
2. Grant Rights to the MU-Moved Item based on Model Rights.

If the request is accepted, the Export-Import Service moves the R-Item to the new U-Location with the assigned Spatial Attitude via a Trajectory and grants actual Rights to the R-Item.

9.2.26.2 Functional Requirements

The MU-Move Process Action includes:

In the request

- The R-Item to be MU-Moved.
- The U-Location where the Item is currently MU-Added.
- The U-Location where the Item is to be MU-Moved.
- The Trajectory that the R-Item should follow to reach the destination U-Location.
- The Spatial Attitude the Item shall have at the new U-Location.
- The Model Rights to the MU-Moved R-Item.
- A Transaction if requested by the Export-Import Service.

In the response

- The Rights per the Model Rights if request is successful.

Note: the U-Location client of the Export-Import Service converts the original U-Location, the destination U-Location, the Trajectory and Spatial Attitude into a Stream that moves the R-Item to the destination U-Location. If no Trajectory is provided, the Export-Import Service or the R-Item organise the movement from the source to the destination U-Location.

9.2.26.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/MUMovePA.json>

9.2.26.4 Semantics

Label	Description
Header	MU-Move Process Action Header
– Standard-MUMovePA	The characters “MMM-3MP-V”
– Version	Major version – 1 or 2 characters
– Dot-separator	The character “.”

- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
UEnvironmentID	Identifier of U-Environment.
MUMovePAData	Data included in the MU-Move Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in the PA Request.
- <i>Nil</i> RItemID	R-Item to be MU-Moved.
- <i>From</i> ULocationID	U-Location where the Item is MU-Added.
- <i>To</i> ULocationID	M-Location where the Item is to be MU-Moved.
- <i>With</i> Trajectory	The Trajectory to be used to move the R-Item.
- <i>With</i> SpatialAttitude	The Spatial Attitude of the Item at the new U-Location.
- <i>With</i> ModelRights	Rights that are requested to the MU-Moved Item.
- <i>With</i> Transaction	Transaction made to Export-Import Service if requested.
- D-Complements	Complements in the PA Response.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With</i> PAStatus	Status of PA Request.
- Ack	If PA Request could be performed
- Err	If Error was found
- Clash	R-Item clashes with another R-Item.
- FaultyPA	Faulty PA Request.
- IncID	Incorrect ID.
- ULocOOR	M-Location out of range.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.27 Property Change

9.2.27.1 Definition

The Action of a User requesting that a Locate Service modify the characteristics of an Item:

1. Change the Perceptibility of the Item.
2. Apply an affine transformation to the Item.
3. Change the attributes of

- 4.1 SourceType, ColourInfo, Intensity (if a Light Source)
- 4.2 SourceType (If an Audio Source)
- 4.3 Mass, Material, Gravity Property, TextureMap (if a Basic Visual Object or a Basic 3D Object)
5. Set the perceptibility status of an Item.
6. Display a specific Personal Status (if a Persona).
7. Grant Rights to the MM-PropertyChanged Item based on Model Rights.

If the request is accepted, the Locate Service responds changes the properties as requested and provides actual Rights to the requesting User.

9.2.27.2 Functional Requirements

The MM-Embed Process Action includes:

In the request

- One of
- The ID Item to be Property Changed.
- The M-Location wrapping the Item.
- A set of parameters selected from Property Change Parameters.
- The Model Rights to the Property Changed Item.
- A Transaction if requested by the Locate Service.

In the response

- The Rights per the Model Rights if request is successful.

9.2.27.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/PropertyChangePA.json>

9.2.27.4 Semantics

Label	Description
Header	MM-Embed Process Action Header
- Standard-PropertyChangePA	The characters “MMM-PCP-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
PropertyChangePAData	Data included in the MM-Embed Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in the PA Request.
- Nil ItemID	Item whose Properties are to be Changed.
- At MLocationID	M-Location where the Item is MM-Added.

- With PerceptibilityStatus	0= non perceptible, 1 = perceptible.
- With Resize	Resize the Item by R_x , R_y , R_z where x, y, and z are the Objects axes rotated according to an Orientation.
- With 3DModelObject	Change Mass, Material Property, and Gravity Property, TextureMap, Object Audio Characteristics.
- With VisualObject	Change Mass, Gravity and Material Property, TextureMap, Object Audio Characteristics.
- With ObjectAudioCharacteristics	(OAC) Change the audio characteristics of an Item.
- With LightSource	Change any of SourceType, ColourInfo, Intensity.
- With AudioSource	Change Source Type.
- With PersonalStatus	Display a specific Personal Status (if a Persona).
- With ModelRights	Rights that are requested to the Property Changed Item.
- With Transaction	Transaction made to Locate Service if requested.
- RS-Complements	Complements in the PA Request.
- With Rights	Actual Rights based on Model Rights if PA Request is accepted.
- With PAStatus	Possible Error Messages.
- Ack	If PA Request could be performed.
- Err	If PA Request could not be performed.
- FaultyPA	Faulty PA Request.
- IncID	Incorrect ID.
- MLocOOR	M-Location out of range.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.
DescMetadata	Descriptive Metadata.

9.2.28 Rights Change

9.2.28.1 Definition

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 RightsChange the Rights to the Process or Item.

If the request is accepted, the User will receive Rights to Items based on Model Licence and the RightsChanged Process the new Rights.

9.2.28.2 Functional Requirements

The RightsChange Process Action includes:

In the request

- The ID of the Item or Process whose Rights are to be changed.
- The M-Location wrapping the Item.
- The Model Rights to be granted to or revoked from the Item or Process.
- The User that should receive notification of Rights Change.
- A Transaction if requested by the Rights Service.

In the response

- The Rights granted or revoked from the Item or process per the Model Rights.

9.2.28.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/RightsChangePA.json>

9.2.28.4 Semantics

Label	Description
Header	RightsChange Process Action Header
- Standard-RightsChangePA	The characters “MMM-RCP-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters
MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
RightsChangePAData	Data included in the RightsChange Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in the PA Request.
- <i>Nil</i> ProcessID or	Process whose Rights are requested to be Changed.
- <i>oneOf</i>	One of
- <i>Nil</i> ProcessID	- Process
- <i>oneOf</i>	- One of
- <i>Nil</i> ItemIDs	- Item whose Rights are requested to be Changed.
- <i>At</i> MLocationID	-M-Location wrapping the Item.
- <i>With</i> ModelRights	Rights that are requested to be Changed.
- <i>To</i> ProcessID	Process that should receive Rights.
- <i>With</i> Transaction	Transaction made to Rights Service if requested.
- RS-Complements	Complements in the PA Response.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With</i> PAMStatus	Possible Error Messages.
- Ack	If PA Request could be performed.

- Err	If PA Request could not be performed.
- FaultyPA	Faulty PA Request.
- IncID	Incorrect ID.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.
- MLocOOR	No such M-Location.
DescMetadata	Descriptive Metadata.

9.2.29 UM-Send

9.2.29.1 Definition

The Action of a Process in the Universe requesting an Export-Import Service to:

1. Send a Message to a Process in the M-Instance on its own initiative or on request by that Process.

2. Grant Model Rights to the Message to the Process in the M-Instance.

If the request is accepted, the Process in the M-Instance received the Message and is granted Rights to it.

9.2.29.2 Functional Requirements

The UM-Send Process Action includes:

In the request

- The ID of the Message sent to the Processes in the M-Environment.
- The ID of Processes in the M-Environment receiving the Message.
- The Model Rights to the Message granted to the Processes in the M-Environment.
- A Transaction if requested by the Export-Import Service.

In the response

- The Rights to the Message granted to the Processes in the M-Environment per the Model Rights.

9.2.29.3 Syntax

<https://schemas.mpai.community/MMM4/V2.1/actions/UMSendPA.json>

9.2.29.4 Semantics

Label	Description
Header	MU-Send Process Action Header
- Standard-MUSendPA	The characters “USP-ISP-V”
- Version	Major version – 1 or 2 characters
- Dot-separator	The character “.”
- Subversion	Minor version – 1 or 2 characters

MInstanceID	Identifier of M-Instance.
MEnvironmentID	Identifier of M-Environment.
UEnvironmentID	Identifier of U-Environment.
UMSendPAData	Data included in the UM-Send Process Action.
- Time	Time of PA request emission and Time of PA request execution.
- RQ-Complements	Complements in the PA Request.
- <i>Nil</i> Message or MessageID	Message to be MU-Sent to Process in M-Environment.
- <i>To</i> ProcessIDs	Process where the Item is to be UM-Sent.
- <i>With</i> ModelRights	Rights requested to be granted to the Process in the M-Environment.
- <i>With</i> Transaction	Transaction made to Import-Export Service if requested.
- D-Complements	Complements in the PA Response.
- <i>With</i> Rights	Actual Rights based on Model Rights if PA Request is accepted.
- <i>With</i> PAStatus	Status of PA Request.
- Ack	If PA Request could be performed.
- Err	If PA Request could not be performed.
- FaultyPA	Faulty PA Request.
- IncID	Incorrect ID.
- InsRights	Insufficient Rights.
- InsValue	Insufficient Value in Wallet.

9.3 Sequences of Process Actions

It is useful to specify standards sequences of IPP messages that are widely used, such as **Track** that includes the following workflow where a User requests:

- *Locate* Service to MM-Add a Persona at an M-Location with a Spatial Attitude.
- *Export-Import* Service to MM-Capture Data and Qualifier at a U-Location.
- *Identify* Service to Identify Data and Qualifier as Stream.
- *Locate* Service to MM-Animate the MM-Added Persona.
- *Export-Import* Service to MU-Actuate the MM-Animated Persona at a U-Location.

This is the sequence of Messages:

Action	RQ-Complements	RS-Complements
MM-Add	<i>Nil</i> Persona At MLoc <i>With</i> SA <i>With</i> Model Rights	<i>With</i> Rights
UM-Capture	<i>Nil</i> Data <i>With</i> Qualifier At ULoc	<i>Nil</i> Data <i>With</i> Qualifier <i>With</i> SA
Identify	<i>Nil</i> Data <i>With</i> Qualifier	<i>Nil</i> ItemID

MM- Animate	<i>Nil Persona With Stream With Model Rights</i>	<i>With Rights</i>
UM- Actuate	<i>Nil MLoc At ULoc With SA With Model Rights</i>	<i>With Rights With R-Item Type.</i>

The Process 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.

9.4 Backus Naur Form

```

program :=
| /* empty */
| one_or_more_statements
one_or_more_statements :=
| statement
| statement one_or_more_statements
statement :=
| id action_keyword id modifiers
action_keyword :=
"Authenticate"
"Author"
"Convert"
"Discover"
"Execute"
"Hide"
"Identify"
"Inform"
"Interpret"
"MM-Animate"
"MM-Add"
"MM-Move"
"MM-Send"
"Modify"
"MU-Actuate"
"MU-Animate"
"MU-Add"
"MU-Move"
"MU-Send"
"Post"
"Property Change"
"Register"
"Resolve"
"Rights Change"
"Transact"
"UM-Capture"
"UM-Send"
"Validate"
modifiers :=
| /* empty */
| one_or_more_modifiers

```

```

one_or_more_modifiers :=
| modifier
| modifier one_or_more_modifiers
modifier :=
| modifier_keyword id
| "At" "(" modifier_subkeyword_At "," id ")"
| "With" "(" modifier_subkeyword_With "," id ")"
modifier_keyword :=
| "At"
| "From"
| "Nil"
| "Of"
| "To"
| "With"
modifier_subkeyword_At :=
| "M-Location"
| "Item"
modifier_subkeyword_With :=
| "Rights"
| "Transaction"
| "Item"
id :=
| STRING
| STRING "@" TIME "," TIME

```

10 Protocols

10.1 Inter-Process Protocol (same M-Instance)

As depicted in Figure 1, Process₁ uses the simple form of the Inter-Process Protocol (IP Protocol or IPP) to request Process₂ in the same M-Instance to perform a Process Action.

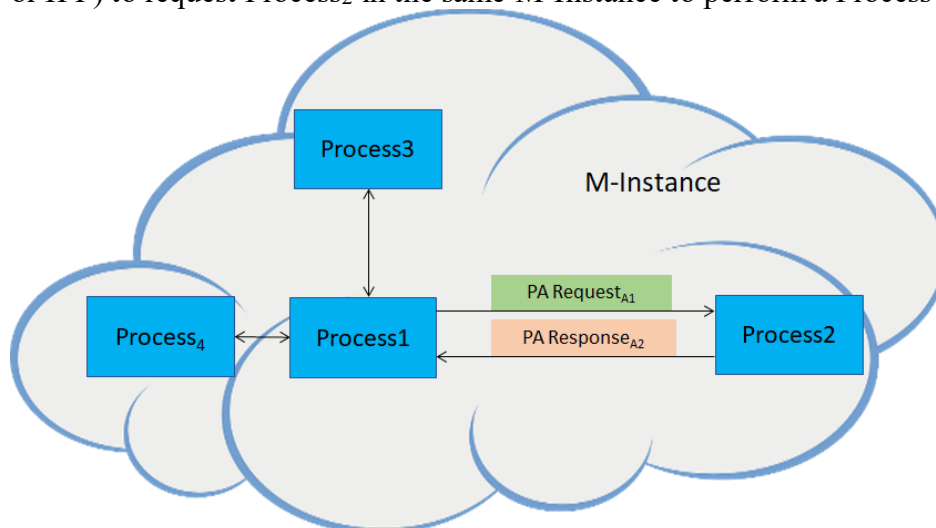


Figure 1 - Process₁ requests Process₂ to perform a Process Action (same M-Instance).

1. Process₁ sends a PA Request to Process₂.
2. If Process₂ finds that the PA Request can be performed,
 1. Then Process₂ performs the PA Request and sends a PA Response to Process₁.
 2. Else, Process₂, sends a PA Response with an error code to Process₁.

10.2 Inter-Process Protocol (different M-Instances)

Figure 2 depicts the case of a Process_A in an M-Instance_A requesting a Process_B in a different M-Instance_B to perform a Process Action in M-Instance_B.

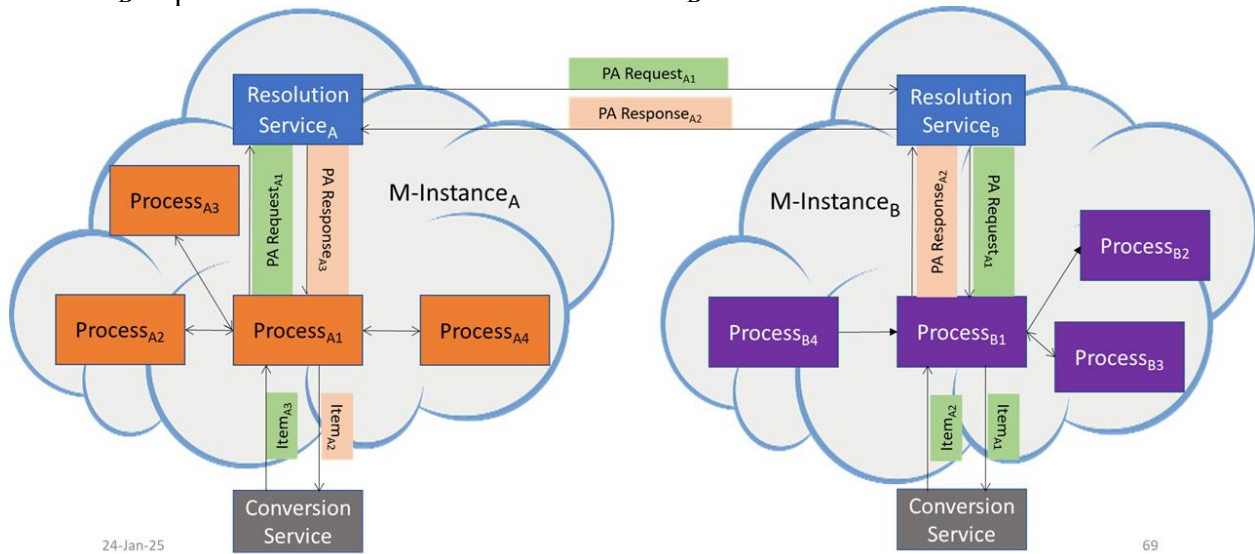


Figure 2 - Process_A requests Process_B to perform a Process Action (different M-Instances).

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 sends 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 and may find that the PA Request may be performed,
 1. Then
 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.
 2. Else, Process_{B1} sends an error to Resolution Service_B.
9. If the execution of the PA Request fails, it sends an error to Resolution Service_B.
10. Resolution Service_B sends error or PA Response to Resolution Service_A.
11. Resolution Service_A sends error or PA Response to Process_{A1}.
12. 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 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)."Conversion Service failure".

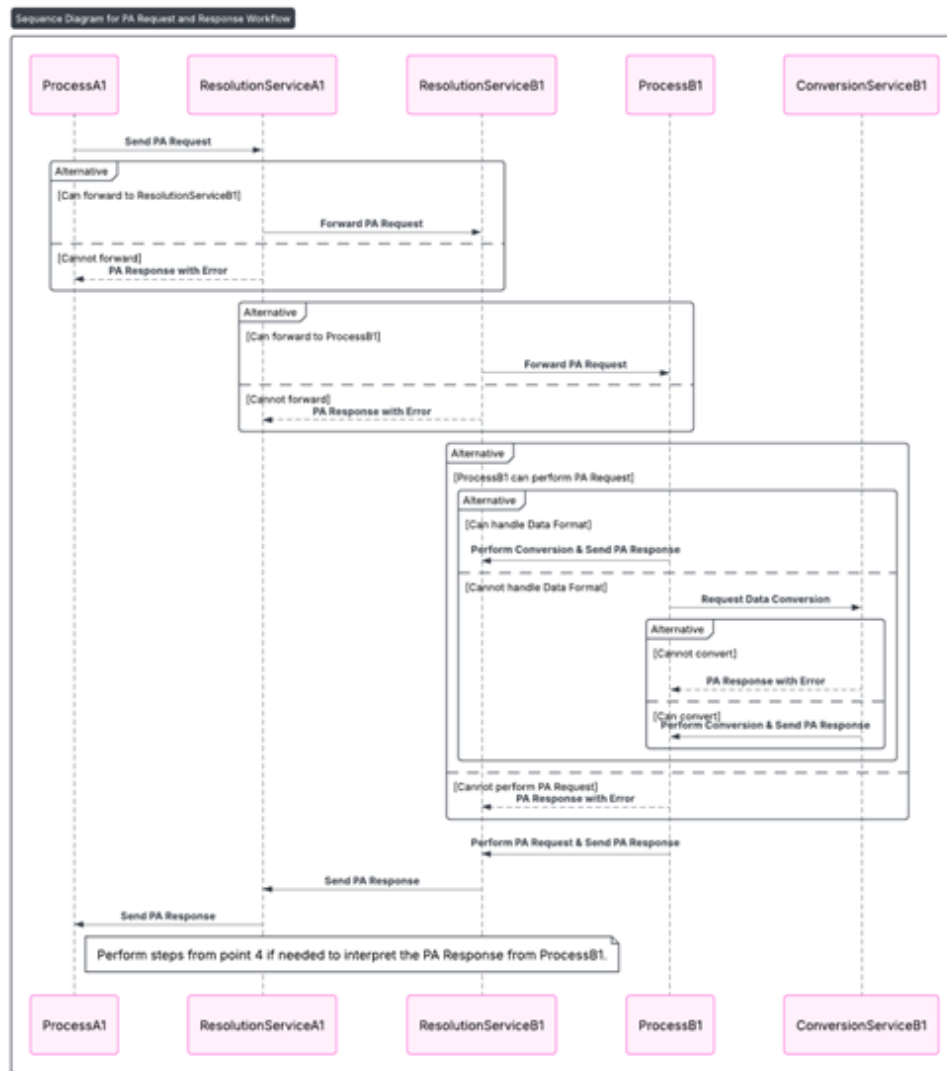


Figure 3 - IP Protocol Sequence Diagram when Processes are in different M-Instances
 Note: MMM-TEC does not provide support to the establishment of business agreements based on which communication between different M-Instances become possible.

10.3 Inter-Process Protocol (multiple M-Instances)

In the case of a multi-M-Instance session (more than 2 M-Instances), an IPP session is set up in the following way:

1. ProcessA₁ requests Resolution ServiceA₁ to open a session with the M-Instances that host at least one invited Process.
2. Resolution ServiceA₁ forwards the request to all relevant Resolution Services.
3. A Resolution ServiceB₁ forwards the request to each invited ProcessB's of its M-Instance.
4. An invited ProcessB₁ responds to Resolution ServiceB₁ accepting or rejecting the invitation.
5. A Resolution ServiceB₁ forwards the response to Resolution ServiceA₁.
6. Resolution ServiceA₁ forwards the responses to ProcessA₁.
7. If at least one invited Process accepts the invitation, the session is opened.

10.4 Inter-Process Protocol elements

Table 1 provides the elements of the IP Protocol (IPP).

Table 1 - IPP Message elements

IPP Message Elements	Description
Time	Provided by communication infrastructure.
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).
Resolution Service ID	Service that: 1. Receives a Message from a Process in its M-Instance and forwarding it to a peer Process in a different M-Instance, or 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.
Message Status	“Ack”, if delivery of payload successful, if delivery fails, Error.

In case of multiple Destination Processes, the Source Process receives the Message Status for each Destination Process.

The JSON Schema of the Inter-Process Protocol is [here](#).

10.5 Posting Protocol

To request a Service Provider (marketplace) to Post an Item, a Process (User₁) invokes the Posting Protocol:

1. User₁ MM-Sends Model Simple Contract To Service Provider.
2. Service Provider MM-Sends Response To User₁.
 1. If response=No, goto End
 2. Else User₁
 1. Transacts SenderPreValue (Service Provider's request to Post the Item) To Service Provider.
 2. Licenses *Nil* Service Provider *With* Service Provider Licence (giving Rights to Post Item)
3. Service Provider operates per Service Provider Licence.
4. When Time=Time₂ (End Time of Licence)
 1. Service Provider stops operating per Service Provider Licence.
 2. User₁ recovers full Rights on Item.
5. End

10.6 Licensing Protocol

The Licensing Protocol assumes that a Process (User₁) has Posted an Item at Service Provider and that User₂ wishes to obtain a Licence to the Posted Item per the Model Basic Contract:

1. User₂ MM-Sends *Nil* Model Basic Contract with its own data *To* Service Provider.
2. Service Provider MM-Sends *Nil* "Ack" (the characters Ack) *To* User₂.
3. User₂ Transacts (ReceiverPostValue, the Value that Service Provider requests to User₂ for its Service) *To* Service Provider.
4. User₂ Transacts (ValueToSend, the Value in the Model Basic Contract) *To* User₁.
5. User₁ Transacts (SenderPostValue, the Value that Service Provider requests to User₁ for its Service) *To* Service Provider.
6. User₁ MM-Sends Model Licence to Licence Service.
7. Licence Service MM-Sends Nil Model Licence to User₂.
8. User MM-Sends "Ack" (the characters Ack) to Licence Service.
9. Licence Service MM-Sends Licence *To* User₁.
10. Licence Service MM-Sends Licence *To* User₂.

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 1 and 2 list the currently identified Actions and Items supported by a Profile.

Table 1 – MMM-TEC V2.1 Actions

General Actions	Execute	Hide	Identify	Modify	Rights Change
	Validate				
Call a Service	Authenticate	Author	Convert	Discover	Inform
	Interpret	Post	Register	Resolve	Transact
Metaverse – Metaverse	MM-Add	MM-Animate	MM-Move	MM-Send	Property Change
Metaverse – Universe	MU-Actuate	MU-Add	MU-Animate	MU-Move	MU-Send
	UM-Capture	UM-Send			

Table 2 – MMM-TEC V2.1 Items

General	Certificate	M-Instance	Contract Object
	Program Object	Identifier	Rights
	M-Capabilities	Rules	M-Environment
	U-Environment		
Human&User	Account	Personal Profile	Activity Data
	Personal Data		
Process Interaction	IPP Message	Process Action	Message
	Resolution	P-Capabilities	
Service Access	Authentication	Validation	Basic Discovery
	Discovery Object	Basic Information	Information Object

	Basic Interpretation	Interpretation Object	
Finance	Asset	Transaction	Currency Object
	Value	Provenance	Wallet
Perception	3D Model Event Descriptors	3D Model Object	3D Model Scene Descriptors
	3D Model Scene Geometry	Audio Event Descriptors	Audio Object
	Audio Scene Descriptors	Audio Scene Geometry	Audio Source
	Audio-Visual Event Descriptors	Audio-Visual Object	Audio-Visual Scene Descriptors
	Audio-Visual Scene Geometry	Light Source	Object Audio Characteristics
	R-Item	Speech Event Descriptors	Speech Object
	Speech Scene Descriptors	Speech Scene Geometry	Text Object
	Visual Event Descriptors	Visual Object	Visual Scene Descriptors
	Visual Scene Geometry		
Internal State	Cognitive State	Emotion	Personal Status
	Social Attitude		
Space&Time	Basic M-Location	Basic Time	Basic U-Location
	M-Location	Orientation	Point of View
	Position	Spatial Attitude	Space-Time
	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 1. The next Sections identify the Action and Items of the four 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.

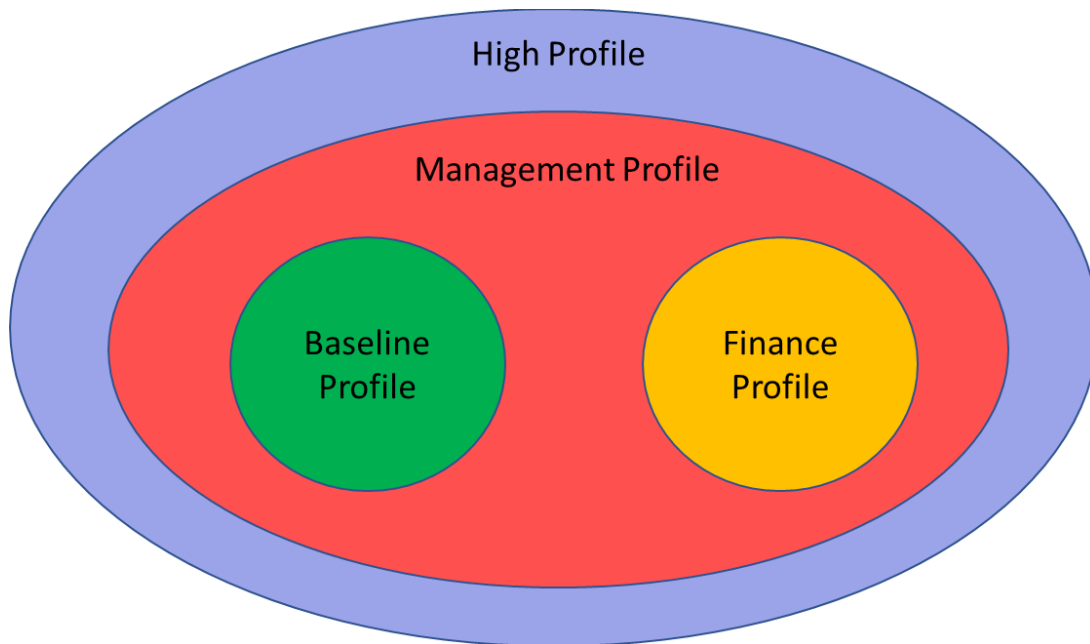


Figure 1 - MMM-TEC V2.1 Profiles

11.3 Baseline Profile

The Baseline Profile is designed to enable Processes to perform Process Actions that include the identified Actions (Table 3) and Items (Table 4). 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 3 – Actions of MMM-TEC V2.1 Baseline Profile

General Actions	Identify			
Metaverse to Metaverse	MM-Add	MM-Animate	MM-Send	
Metaverse - Universe	MU-Actuate	UM-Capture		

Table 4 – Items of MMM-TEC V2.1 Baseline Profile

General	Identifier	M-Instance			
Process Interaction	IPP Message	Message	Process Action		
Perception	Basic 3D Model Object	Basic Audio Object	Basic Audio-Visual Object	Basic Speech Object	Basic Visual Object
	Basic 3D Model Scene Descriptors	Basic Audio Scene Descriptors	Basic Audio-Visual Scene Descriptors	Basic Speech Scene Descriptors	Basic Visual Scene Descriptors
Space&Time	Basic M-Location	Basic U-Location	Point of View	Spatial Attitude	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 1*, 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 5 – Actions of MMM-TEC V2.1 Finance Profile

General Actions	Hide	Identify	Modify	Rights Change	Validate
Call a Service	Authenticate	Author	Discover	Post	Register
	Transact				
Metaverse to Metaverse	MM-Add	MM-Send	Property Change		
Metaverse to Universe	MU-Actuate	MU-Send	UM-Send		

Table 6 – Items of MMM-TEC V2.1 Finance Profile

General	Certificate	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	Discovery Object	Validation		
Finance	Asset	Currency Object	Provenance	Transaction	Value
	Wallet				
Perception	3D Model Object	Audio Object	Audio-Visual Object	Speech Object	Visual Object
Space&Time	M-Location	U-Location	Point of View	Time	

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 1*. 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 7 – Actions of MMM-TEC V2.1 Management Profile

General Actions	Execute	Hide	Identify	Modify	Rights Change
	Validate				
Call a Service	Authenticate	Author	Discover	Inform	Interpret
	Post	Register	Transact		
Metaverse – Metaverse	MM-Add	MM-Animate	MM-Move	MM-Send	Property Change
Metaverse – Universe	MU-Actuate	MU-Add	MU-Animate	MU-Move	MU-Send
	UM-Capture	UM-Send			

Table 8 – Items of MMM-TEC V2.1 Management Profile

General	Certificate	M-Instance	Contract Object
	Program Object	Identifier	Rights
	M-Capabilities	Rules	M-Environment
	U-Environment		
Human&User	Account	Personal Profile	Activity Data
	Personal Data		
Process Interaction	IPP Message	Process Action	Message
	Resolution	P-Capabilities	
Service Access	Authentication	Validation	Basic Discovery
	Discovery Object	Basic Information	Information Object
	Basic Interpretation	Interpretation Object	
Finance	Asset	Transaction	Currency Object
	Value	Provenance	Wallet
Perception	3D Model Event Descriptors	3D Model Object	3D Model Scene Descriptors
	3D Model Scene Geometry	Audio Event Descriptors	Audio Object
	Audio Scene Descriptors	Audio Scene Geometry	Audio Source
	Audio-Visual Event Descriptors	Audio-Visual Object	Audio-Visual Scene Descriptors
	Audio-Visual Scene Geometry	Light Source	Object Audio Characteristics
	R-Item	Speech Event Descriptors	Speech Object
	Speech Scene Descriptors	Speech Scene Geometry	Text Object
	Visual Event Descriptors	Visual Object	Visual Scene Descriptors
	Visual Scene Geometry		
Space&Time	Basic M-Location	Basic Time	Basic U-Location
	M-Location	Orientation	Point of View
	Position	Spatial Attitude	Space-Time
	Time	U-Location	

11.6 igh Profile

This Profile includes all other Profiles. Table 9 and 10 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 9 – Actions specific of MMM-TEC V2.1 High Profile

Call a Service	Convert	Resolve
----------------	---------	---------

Table 10 – Items specific of MMM-TEC V2.1 High Profile

Internal State	Cognitive State	Emotion	Personal Status	Social Attitude
Space&Time	Universe-Metaverse Map			

12 Verification Uses Cases

(Informative)

12.1 Introduction

This Informative Chapter verifies that MMM-TEC supports a variety of M-Instance context Use Cases.

The following conventions are used throughout:

Acronym	Meaning	Acronym	Meaning	Acronym	Meaning
MLoc	M-Location	MTransaction	Model Transaction	MData	Model Data
SA	Spatial Attitude	MRights	Model Rights	ULoc	U-Location

The following Service abbreviations are used to increase the readability of the tables describing Use Case workflows:

Service Name	Acronym	Service Name	Acronym	Service Name	Acronym	Service Name	Acronym
Author	AUSrvc	Identify	IDSrvc	Land	LDSrvc	Resolve	RESrvc
Communicate	COSrvc	Interpret	ITSrvc	Presence	PRSrvc	Rights	RTSrvc
Execute	EXSrvc	Locate	LCSrvc	Register	RGSrvc	Transact	TRSrvc

All Use Cases include the following:

1. Description – summary description of the Use Case.
2. Variables – the main Processes and Items referenced in the Process Actions.
3. Workflow – expressed by the main elements of Process Actions

MMM-TEC is based on the notions of:

1. human – operating in the Universe
2. User – representing a human operating in an M-Instance as a replica of the human or as an autonomous agent.
3. Persona – an avatar animated by a User.

To make use case descriptions less congested, the same name may be used to indicate a human, a User, or a Persona. The context will clearly identify to which of the three entities the name refers to in a specific context.

12.2 Friends meet in the metaverse

12.2.1 Description

1. human₁ Registers with an M-Instance.

2. Friend₁ (User of human₁):
 - Buys a land parcel from a Land Service.
 - Buys a room from an Author Service.
 - Tracks its Persona at Metaverse Square (MVS).
 - Signals its presence to the Presence Service.
 - Moves its Persona close to the land parcel.
 - Places the room on the land parcel.
 - Moves inside the room.
 - Invites Friend₂ already Tracked at the at the MVS to its room.
3. Friend₂ accepts invitation.
4. Friend₁ changes Rights to Friend₂ to allow it to enter its room.
5. Friend₂ moves to Friend₁'s room.
6. (After a while)
7. Friend₂ leaves Friend₁'s room.
8. Friend₁ Changes its room's Rights back to their initial value.

12.2.2 Variables

Processes	Name	Function
	ATSrv	Author Service
	COSrv	Communication Service
	EISrv	Export/Import Service
	Friend ₁	Friend of Friend ₂
	Friend ₂	Friend of Friend ₁
	human ₁	Human, whose User is Friend ₁
	IDSrv	Identify Service
	LCSrv	Locate Service
	LDSrv	Land Service
	RGSrv	Register Service
	RTSrv	Rights Service
Items	Name	Function
	Data	Representing the capture of human ₁ at home
	home	The physical home of Friend ₁ .
	human ₁	Human, whose User is Friend ₁
	Friend ₁	Friend of Friend ₂
	Friend ₂	Friend of Friend ₁
	Message ₁₁	Message sent to Presence Service
	Message ₁₂	Message sent by Friend ₁ inviting Friend ₂
	Message ₂₁	Message of Friend ₂ accepting invitation
	MV Square	A reference place to MM-Add Personae
	Persona ₁	Persona of Friend ₁
	Persona ₂	Persona of Friend ₂
	Qualifier	The Qualifier used to represent the MM-Captured Data

	Rights ₁	Rights to Persona ₁ after it has been MM-Added
	Rights ₂	Rights to Animation Stream Data
	Rights ₃	Rights to Animation Stream Item
	Rights ₄	Rights to Animated Persona ₁
	Rights _{F5}	Rights to MM-Actuated Animated Persona ₁
	Rights _{F6}	Rights granted to PRSrv to Use Message ₁₁
	Rights _{F7}	Rights to MM-Move Persona ₁ at MV Square
	Rights ₈	Rights to use Room MM-Added to Land Parcel
	Rights _{F9}	Rights granted to Friend ₁ to MM-Move Persona ₁ inside the Room
	Rights _{F1.10}	Rights granted to Friend ₂ to Use Message ₁₂
	Rights ₁₁	Rights granted to Friend ₁ to Use Message ₂₁
	Rights _{F1.11}	Rights granted to Friend ₂ to come into Room
	Rights _{F1.12}	Friend ₂ 's Rights restored to original value.
	Transaction _{F11}	For Friend ₁ to get Rights to Land Parcel
	Transaction _{F11}	For Friend ₁ to get Rights to Room

12.2.3 Workflow

S-Process	Action	RQ-Complement	D-Process	RS-Complement	Comment
human ₁	Registers	<i>With</i> Personal Profile	RGSrv	<i>Nil</i> Account	human Registers
Friend ₁	Transact	<i>With</i> Transaction ₁	LDSrv	<i>With</i> Ack	Friend ₁ buys Parcel
	Transact	<i>With</i> Transaction ₂	ATSrv	<i>With</i> Ack	Friend ₁ buys Room
	MM-Add	<i>Nil</i> Persona ₁	LCSrv	<i>With</i> Rights ₁₁	Persona ₁ is MM-Added
		<i>With</i> SA		<i>With</i> Ack	
		<i>At</i> MV Square			
		<i>With</i> MRights			
	UM-Capture	<i>Nil</i> Scene	EISrv	<i>Nil</i> Data	Animation Stream is captured
		<i>From</i> home ₁		<i>With</i> SA	
		<i>With</i> Qualifier		<i>With</i> Rights _{F12}	
		<i>With</i> MRights		<i>With</i> Ack	
	Identify	<i>Nil</i> Data	IDSrv	<i>Nil</i> Stream	Stream Data is Identified

		<i>With Qualifier</i>		<i>With Rights</i> ₁₃	
		<i>At Friend</i> ₁		<i>With Ack</i>	
		<i>With MRights</i>			
	MM-Anim	<i>Nil Persona</i> _{F1}	LCSrv	<i>With Rights</i> _{F14}	<i>Persona</i> _{F1} is MM-Animated
		<i>At MV Square</i>		<i>With Ack</i>	
		<i>With Stream</i> ₁			
		<i>With MRights</i>			
	MU-Actuate	<i>Nil Persona</i> ₁		<i>Nil Media</i>	Actuate Animated Persona
		<i>At MV Square</i>		<i>With Rights</i> ₁₅	
		<i>At home</i> ₁		<i>With Ack</i>	
		<i>With RItemType</i>			As Media
		<i>With SA</i>			
	Property Change	<i>Nil Persona</i> ₁	LCSrv	<i>With Ack</i>	<i>Persona</i> ₁ is made perceptible.
		<i>At MV Square</i>			
		<i>With Perceptibility</i>			
	MM-Send	<i>Nil Message</i> ₁₁	COSrv	<i>With Rights</i> ₁₆	Message to Presence Service
		<i>With MRights</i>		<i>With Ack</i>	
		<i>To PRSrv</i>			
	MM-Move	<i>Nil Persona</i> ₁	LCSrv	<i>With Rights</i> ₁₇	<i>Persona</i> ₁ is moved to Parcel
		<i>From MV Square</i>		<i>With Ack</i>	
		<i>To Parcel</i>			
		<i>With MRights</i>			
	MM-Add	<i>Nil Room</i>		<i>With Rights</i> ₁₈	Room is MM-Added to Parcel
		<i>At Parcel</i>		<i>With Ack</i>	
		<i>With SA</i>			
		<i>With MRights</i>			

		<i>With Perceptibility</i>			Room is invisible
	MM-Move	<i>Nil Persona_{F1}</i>	LCSrvc	<i>With Rights₁₉</i>	Persona ₁ is moved into Room
		<i>From Parcel</i>		<i>With Ack</i>	
		<i>To Room</i>			
		<i>With SA</i>			
	MM-Send	<i>Nil Message₁₂</i>	COSrvc	<i>With Rights₁₀</i>	Friend ₁ sends message to Friend ₂
		<i>With MRights</i>		<i>With Ack</i>	
		<i>To Friend₂</i>			
Friend ₂	MM-Send	<i>Nil Message₂₁</i>	COSrvc	<i>With Rights_{F11}</i>	Friend ₂ accepts invitation
		<i>With MRights</i>		<i>With Ack</i>	
		<i>To Friend₁</i>			
Friend ₁	Rights Change	<i>Nil Friend₂</i>	RTSrv	<i>With Rights_{F1.11}</i>	Friend ₂ may come to Room
		<i>With MRights</i>		<i>With Ack</i>	
Friend ₂	MM-Move	<i>Nil Persona₂</i>	LCSrvc	<i>With Ack</i>	Friend ₂ comes to Room
		<i>From Home₂</i>			
		<i>To Room</i>			
		<i>With SA</i>			
<i>Activities</i>	<i>Not</i>	<i>Relevant</i>	<i>To</i>	<i>Use</i>	<i>Case</i>
Friend ₂	MM-Move	<i>Nil Persona₂</i>	LCSrvc	<i>With Ack</i>	Friend ₂ leaves Room
		<i>From Room</i>			
		<i>With SA</i>			
		<i>To Home₂</i>			
Friend ₁	Rights Change	<i>Nil Friend₂</i>	RTSrv	<i>With Rights₁₃</i>	Friend ₂ cannot come to Room
		<i>With MRights</i>		<i>With Ack</i>	

12.3 Virtual lecture

12.3.1 Description

1. Manager

1. Creates a room at the Author Service (AUSrvc).
2. Makes a Transaction via the Transaction Service (TRSrvc) to get the Rights to use it.
3. Places the room at an M-Location for which it has Rights via Location Service (LCSrvc).
4. Grants Rights to Teacher to enter the room via the Rights Service (RTSrvc).
2. Teacher
 1. Shows up in the Room from its virtual home.
 2. Places an object close to the desk for use during the lecture.
3. Student
 1. Is Tracked at Home
 2. Makes a Transaction to attend lecture, change Personal Status, approach Teacher's desk, and record lecture.
4. Manager grants attendance Rights to Student.
5. Student
 1. Shows up in the lecture room.
 2. Change its Persona's Property to give it a Personal Status appropriate to the lecture.
 3. Stores the recorded lecture.
 4. Returns to its virtual home.
6. Teacher delivers Lecture.
7. Student
 1. Approaches Teacher's desk.
 2. Returns to its virtual home.
8. Manager makes a Transaction to Teacher for the lecture.
9. Teacher returns to its virtual home.

12.3.2 Variables

Processes	Name	Function
	ATSrvc	Author Service
	EISrvc	Export/Import Service
	IDSrvc	Identify Service
	LCSrvc	Locate Service
	RGSrvc	Register Service
	RTSrvc	RightsService
	TRSrvc	Transaction Service
	Manager	School Manager
	Student	Student attending lecture
	Teacher	Teacher
Items	Name	Function
	Classroom	Classroom edited by manager
	Data	Data captured by Teacher at home _T
	Homes	Virtual Home of Student
	home _T	Real home of Teacher
	Items	Items used to author Classroom

	Lecture	Lecture recorded by Student
	Media	The rendering on Person _T at home _T
	MLoc _M	M-Location of Classroom
	MLoc _{S1}	Landing M-Location of Student
	MLoc _{S2}	M-Location close to Teacher
	MLoc _T	Landing M-Location of Teacher
	Personal Status	Imposed by Student to Personas
	Personas	Persona of Student
	Person _T	Persona of Teacher
	Rights _{M1}	Rights to Manager (Classroom)
	Rights _{M2}	Rights to Manager (Classroom placement)
	Rights _{S1}	Rights to Student (attend & record lecture)
	Rights _{T1}	Rights to Teacher (enter classroom)
	Rights _{T2}	Rights to Teacher (capture streaming data)
	Rights _{T3}	Rights to Teacher (to Stream)
	Rights _{T4}	Rights to Teacher (to animated Person _T)
	Rights _{T5}	Rights to Teacher (to render at home _T)
	Stream	Stream animating Person _T
	Transaction _{M1}	To get Rights to Classroom
	Transaction _{M2}	To get Rights to
	Transaction _{M3}	To get Rights to
	Transactions _{S1}	To get Rights to attend & record lecture
	Utterance	Teacher's lecture utterances

12.3.3 Workflow

S-Process	Action	PA-Request	D-Process	PA-Response	Comment
Manager	Authors	<i>With</i> Items	ATSvc	<i>Nil</i> Classroom	Authors Classroom
		<i>With</i> MRights		<i>With</i> Rights _{M1}	
		<i>At</i> Manager			
		<i>With</i> Transaction _{M1}			
Manager	MM-Add	<i>Nil</i> Classroom	LCSvc	<i>With</i> Rights _{M2}	Places Classroom
		<i>At</i> MLoc _M			
		<i>With</i> SA			
		<i>With</i> MRights			

		<i>With Transaction</i> _{M2}			
Manager	Rights Change	<i>Nil</i> Teacher	RTSrv	<i>With Rights</i> _{SM3}	Rights to Teacher
		<i>With MRights</i>			
Teacher	MM-Add	<i>Nil</i> Person _{AT}	LCSrv	<i>With Rights</i> _{ST1}	Teacher MM-Adds Person _{AT}
		<i>With SA</i>		<i>With Ack</i>	
		<i>At MLoc</i> _T			
		<i>With MRights</i>			
	UM-Capture	<i>From</i> home _T	EISrv	<i>Nil</i> Data	Teacher captures Animation Stream
		<i>With Qualifier</i>		<i>With SA</i> _{T1}	
		<i>With MRights</i>		<i>With Rights</i> _{ST2}	
				<i>With Ack</i>	
	Identify	<i>Nil</i> Data	IDSrv	<i>Nil</i> StreamID	Teacher Identifies Stream Data
		<i>With Qualifier</i>		<i>With Rights</i> _{ST3}	
		<i>At Teacher</i>		<i>With Ack</i>	
		<i>With MRights</i>			
	MM-Anim	<i>Nil</i> Person _{AT}	LCSrv	<i>With Rights</i> _{ST4}	Teacher animates Person _{AT}
		<i>At Loc</i> _T		<i>With Ack</i>	
		<i>With Stream</i>			
		<i>With MRights</i>			
	MU-Actuate	<i>Nil</i> Person _{AT}		<i>Nil</i> Media	Teacher renders animated Person _{AT}
		<i>From MLoc</i> _T		<i>With Rights</i> _{ST5}	
		<i>To</i> home _T		<i>With Ack</i>	
		<i>With RItemType</i>			As media.
		<i>With SA</i>			
		<i>With MRights</i>			
		<i>At URL</i>			
Student	Transact	<i>With Transactions</i> ₁	TRSrvs		Pays for attending

Manager	Rights Change	<i>Nil</i> Student	RTSrvc	<i>With</i> Rightss ₁	Rights to Student
		<i>With</i> MRights			
Student	MM-Move	<i>Nil</i> Personas		<i>With</i> Ack	Student moves to Classroom.
		<i>From</i> Homes			
		<i>To</i> MLocs ₁			
	Property Change	<i>Nil</i> Personas			
		<i>With</i> Personal Status			
Teacher	MM-Add	<i>Nil</i> Utterance		<i>With</i> Ack	Teacher starts lecture
		<i>At</i> MLoc _T			
		<i>With</i> Perceptibility			Lecture is audible
Student	MM-Move	<i>Nil</i> Personas	LCSrvc		Moves close to Teacher
		<i>From</i> Homes			
		<i>At</i> MLocs ₂			
Student	MU-Send	<i>Nil</i> Lecture			Stores recorded lecture
		<i>At</i> URL			
		<i>With</i> MRights			
Student	MM-Move	<i>Nil</i> Personas			Returns virtual home
		<i>From</i> MLocs ₂			
		<i>To</i> Homes			
Manager	Transact	<i>With</i> Transaction _{M3}	TRSrvc		Pays for lecture
Teacher	MM-Move	<i>Nil</i> Person _T			Returns virtual home
		<i>From</i> MLoc _T			
		<i>To</i> home _T			

12.4 Hybrid working

12.4.1 Description

1. Company workers may go to work:
 1. Physically as R-Workers if in the Company (R-Company).
 2. Virtually as V-Workers if in the V-Company, the Real Company's Digital Twin.
2. V-Workers can perform the Process Actions proper of the work.

3. R-Workers' Personae are automatically present in the V-Company with the same Rights as V-Workers.
4. V-Workers are automatically rendered as Avatars in the R-Company.
5. All Workers share their speech objects with:
 1. The V-Company/R-Company, if they are in the R-Company/ V-Company.
 2. Participate in V-Meetings with their Personae if R-Workers.
 3. Participate in R-Meeting as rendered Personae if V-Workers.
6. V-Worker
 1. Sends a speech message to an R-Worker.
 2. Moves its Persona to R-Worker's desk.
 3. Has a conversation at the R-Worker's desk.
 4. Moves its Persona to a V-Meeting Room.
 5. Renders its Persona in the R-Meeting Room.
7. R-Worker
 1.
 1. Goes to the R-Meeting Room.
 2. Their Persona is moved to the V-Meeting Room.
9. Manager (A V-Worker)
 1. Places a V-Whiteboard in the V-Meeting Room.
 2. Animates the V-Whiteboard.
 3. Causes an R-Whiteboard to be placed (MU-Actuates) in the R-Meeting Room.

12.4.2 Variables

Processes	Name	Function
	eWhiteboard	Program animating Whiteboard
	COSrvc	Communicate Service
	EXSrvc	Execute Service
	IESrvc	Import/Export Service
	LCSrvc	Locate Service
	RTSrvc	RightsService
	Manager	Company Manager
	R-Worker	Worker in R-Company
	V-Worker	Worker in V-Company
Items	Name	Function
	Classroom	Classroom edited by manager
	Desk _{R1}	Desk of R-Worker in V-Company
	Desk _{V1}	Desk of V-Worker
	Input	Inputs to Whiteboard
	Messenger _{R1}	R-Worker's Message

	Message _{V1}	V-Worker's Message
	MtgRoom	Virtual Meeting Room
	Persona _{R1}	Persona of R-Worker
	ULoc	Place in R-Company's meeting room
	Utterance _{R1}	R-Worker's Conversation
	Utterance _{V1}	V-Worker's Conversation

12.4.3 Workflow

S-Process	Action	PA-Request	D-Process	PA-Response	Comment
VWorker	MM-Send	<i>Nil</i> Message _{V1}	IESrvc	<i>With Ack</i>	Sends utterance to R-Worker
	MM-Move	<i>Nil</i> Persona _{V1}	COSrvc	<i>With Ack</i>	Moves Persona to Desk _R
		<i>From</i> Desk _{V1}			
		<i>To</i> Desk _{R1}			
		<i>With SA</i>			
	MM-MEmbed	<i>Nil</i> Utterance _{V1}	LCSrvc	<i>With Ack</i>	Converses to RWorker
		<i>At</i> Desk _{R1}			
RWorker	MM-MEmbed	<i>Nil</i> Utterance _{R1}	LCSrvc	<i>With Ack</i>	Converses to VWorker
		<i>At</i> Desk _{R1}			
VWorker	MM-Move	<i>Nil</i> Persona _{V1}	COSrvc	<i>With Ack</i>	Moves Personav to Meeting Room
		<i>From</i> Desk _{R1}			
		<i>To</i> MtgRoom			
		<i>With SA</i>			
RWorker	MM-Move	<i>Nil</i> Persona _{R1}	COSrvc	<i>With Ack</i>	Moves Persona _R to Meeting Room
		<i>From</i> Desk _{R1}			
		<i>To</i> MtgRoom			
		<i>With SA</i>			
Manager	MU-Embed	<i>Nil</i> Whiteboard	IESrvc	<i>With Ack</i>	Places Whiteboard in R-Meeting Room

		<i>At</i> ULoc			
		<i>With</i> SA			
	Execute	<i>Nil</i> eWhiteboard	EXSrv	<i>With</i> Ack	Activates eWhiteboard in V-Meeting Room
		<i>With</i> Input			Participants' input to the eWhiteboard

12.5 eSports Tournament

12.5.1 Description

1. Site manager (S-Manager):
 - Develops a game landscape.
 - Places game landscape at an M-Location where it has Rights.
2. Game manager (G-Manager):
 - Acquires Rights to place game landscape on M-Location.
 - Places and animates
 - Autonomous characters in the game landscape.
 - Virtual cameras and microphones in game landscape.
3. Players move to their M-Locations in the game landscape.
4. Game manager (G-Manager) renders the output of virtual cameras and microphones onto the dome.

12.5.2 Variables

Processes	Name	Function
	AVAnim _{G,j}	Program animating j-th AV Device
	COSrv	Communicate Service
	EXSrv	Execute Service
	IESrv	Import/Export Service
	LCSrv	Locate Service
	GManager	Game Manager
	PAnim _{G,i}	Programs to animate i-th Character Persona
	RTSrv	RightsService
	SManager	Site Manager
	R-Worker	Worker in R-Company
	V-Worker	Worker in V-Company
Items	Name	Function
	AVDevice _{G,j}	j-th AV Device in the Landscape
	AVDeviceOutput _G	Output of j-th AV Device in the Landscape
	Items	To author Landscape
	Landscape	For placing Characters and Players
	MLoc _{S1}	Location where Landscape is placed

	Persona _{G,i}	Persona of i-th NPC
	Personap _k	Persona of k-th Player
	MLoc _{G,i}	Game Manager's Rights to use Landscape
	MLoc _{Pk}	Location where k-th Player in MM-Added
	Rights _{S1}	Rights to Landscape
	Rights _{S2}	Rights to place Landscape at MLoc _{S1}
	Transaction _{G1}	Game Manager's Transaction to get Rights to use Landscape
	Transactions _{S1}	To get Rights to Landscape

12.5.3 Workflow

S-Process	Action	PA-Request	D-Process	PA-Response	Comment
SManager	Author	<i>With</i> Items	ATSrvc	<i>Nil</i> Landscape	Authors Landscape
		<i>With</i> MRights		<i>With</i> Rights _{S1}	
		<i>At</i> SManager		<i>With</i> Ack	
		<i>With</i> Transactions _{S1}			
	MM-Add	<i>Nil</i> Landscape	LCSrvc	<i>With</i> Rights _{S2}	Places GLandscape
		<i>At</i> MLoc _{S1}		<i>With</i> Ack	
		<i>With</i> SA			
		<i>With</i> Perceptibility			Perceptible
		<i>With</i> MRights			
GManager	Rights Change	<i>Nil</i> Landscape	RT Srvc	<i>With</i> Rights _{G1}	Buys Rights to GLandscape
		<i>With</i> MRights		<i>With</i> Ack	
		<i>With</i> Transaction _{G1}			
	<i>Do i=1,N</i>				
	MM-Add	<i>Nil</i> Persona _{G,i}	LCSrvc	<i>With</i> Ack	Places Autonomous Characters
		<i>At</i> MLoc _{G,i}			
		<i>With</i> SA			
	Execute	<i>Nil</i> PAnim _{G,i}			Animates Autonomous Characters
	<i>End</i>				
	<i>Do i=1,M</i>				
	MM-Add	<i>Nil</i> AVDevice _{G,j}	LCSrvc	<i>With</i> Ack	
		<i>At</i> MLoc _{G,j}			
		<i>With</i> SA			
	Execute	<i>Nil</i> AVAnim _{G,j}	EXSrvc		Animates AV Devices
<i>Do k=1,K</i>					

Player _k	MM-Move	Nil Personap _k	LCSrv		Players take their positions
		At MLocP _k			
		With SA			
End					
GManager	Do i=1,M				
	MU-Actuate	Nil AVDeviceOutputG _i	IESrv	With Ack	Displays selected outputs of AV Devices
		At Dome			
End					

12.6 Virtual Performance

12.6.1 Description

1. Impresario:
 1. Holds rights to parcel.
 2. Authors auditorium.
 3. Places auditorium on parcel.
2. Participant₁
 1. Buys ticket with the Rights to stay close to the performer for 5 minutes.
 2. Takes seat
3. Participant₂
 1. Buys ticket.
 2. Takes seat
4. Performer sings.
5. Impresario:
 1. Interprets participants' mood (Performance Status).
 2. Generates special effects based on Participant Status.
6. Participant₁:
 1. Moves close to Performer.
 2. Moves back to original place after 5 minutes.
7. Participant₂ utters a private speech to Participant₁.
8. Impresario pays Performer.

12.6.2 Variables

Processes	Name	Function
	ATSrv	Author Service
	COSrv	Communicate Service
	Impresario	Organising event
	ITSrv	Interpret Service
	LCSrv	Locate Service
	Participant ₁	Participant#1 in event
	Participant ₂	Participant#2 in event
	Performer	Event's Performer
	RTSrv	Rights Service

	TRSRVC	Transact Service
Items	Name	Function
	Auditorium	Where Virtual Performance is help
	FX	Special Effects
	Interpretation	Sent to Interpret Service
	Items	Used to Author Auditorium
	Message _{P2}	Participant#1's Message to Participant#1
	MLoc _{I1}	M-Location where Auditorium is MM-Added
	MLocIFX	M-Location in Auditorium where FX is MM-Added
	MLoc _{P1}	Participant#1's M-Location in Auditorium
	MLoc _{P1New}	Participant#1's new M-Location
	MLoc _{P2}	Participant#2's M-Location in Auditorium
	Personap	Performer's Persona
	Personap ₁	Participant#1's Persona
	Personap ₂	Participant#1's Persona
	Rights _{I1}	Impresario's Rights to Auditorium
	Rights _{I2}	Impresario's Rights to Auditorium MM-Added at MLoc _{I1}
	Rights _{I3}	Rights granted to Performer
	Song	Audio of Performance
	Transaction _{I1}	Impresario's Transaction to get Rights to Auditorium
	Transaction _{P1.1}	Participant#1's Transaction to buy ticket
	Transaction _{P2.1}	Participant#2's Transaction to buy ticket

12.6.3 Workflow

S-Process	Action	PA-Request	D-Process	PA-Response	Comment
Impresario	Author	<i>With</i> Items	ATSrvc	<i>Nil</i> Auditorium	Authors Auditorium
		<i>With</i> MRights		<i>With</i> Rights _{I1}	
		<i>At</i> Impresario		<i>With</i> Ack	
		<i>With</i> Transaction _{I1}			
	MM-Add	<i>Nil</i> Auditorium	LCSrvc	<i>With</i> Rights _{I2}	Places Auditorium
		<i>At</i> MLoc _{I1}		<i>With</i> Ack	
		<i>With</i> SA			
		<i>With</i> Perceptibility			Perceptible
		<i>With</i> MRights			
Impresario	Rights Change	<i>Nil</i> Performer	RTSrvc	<i>With</i> Rights _{I3}	Rights in Transaction
		<i>With</i> ModelRights		<i>With</i> Ack	
		<i>To</i> Performer			Performer receives Rights

Participant ₁	Transact	<i>With</i> Transaction _{P1.1}	TRSrv	<i>With</i> Ack	Rights in Transaction
	MM-Moves	<i>Nil</i> Personap ₁	LCSrv	<i>With</i> Ack	Takes seat
		<i>To</i> MLoc _{P1}			
		<i>With</i> SA			
Participant ₂	Transact	<i>With</i> Transaction _{P2.1}	TRSrv	<i>With</i> Ack	Rights in Transaction
	MM-Moves	<i>Nil</i> Personap ₂	LCSrv	<i>With</i> Ack	Takes seat
		<i>To</i> MLoc _{P2}			
		<i>With</i> SA _{P2}			
Performer	MM-Moves	<i>Nil</i> Personap	LCSrv	<i>With</i> Ack	Comes to the stage
		<i>To</i> MLoc _P			
		<i>With</i> SA _P			
	MM-Add	<i>Nil</i> Song	LCSrv	<i>With</i> Ack	Starts performance
		<i>At</i> MLoc _P			
		<i>With</i> Perceptibility			
Impresario	Interpret	<i>With</i> Interpretation	ITSrv	<i>Nil</i> Interpretation	
				<i>With</i> Ack	
	MM-Add	<i>Nil</i> FX	LCSrv	<i>With</i> Ack	Adds a Special Effect
		<i>At</i> MLoc _{IFX}			
		<i>With</i> SA			
		<i>With</i> Perceptibility			
Participant ₁	MM-Moves	<i>Nil</i> Personap ₁	LCSrv	<i>With</i> Ack	Moves close to performer
		<i>To</i> MLoc _{P1New}			
		<i>With</i> SA			
	MM-Moves	<i>Nil</i> Personap ₁	LCSrv	<i>With</i> Ack	Returns after 5 minutes
		<i>To</i> MLoc _{P1}			
		<i>With</i> SA			
Participant ₂	MM-Send	<i>Nil</i> Message _{P2}	COSrv	<i>With</i> Ack	Sends msg w/ utterance
		<i>To</i> Participant _{P1}			
Impresario	Transact	<i>With</i> Transaction _{I2}	TRSrv	<i>With</i> Ack	Pays performer

12.7 AI Tourist Guide

12.7.1 Description

1. Developer uses Author Service to develop
 1. Landscape
 2. Tourist objects
 3. AR app (for mobile handsets) (in the Universe).

4. Universe-Metaverse Map (in the Universe).
2. Guide
 1. Acquires rights to land parcel.
 2. Acquires rights to landscape.
 3. Places landscape in land parcel.
 4. Acquires rights to objects.
 5. Places objects in the landscape.
 6. Places Autonomous Personae at appropriate landscape locations.
 7. Animates Personae.
 8. Distributes AR app (in Universe).
3. Tourist
 1. Installs app on their handset.
 2. Registers with the service offered by Guide.
 3. Moves around in the Universe.
 4. Is alerted when a U-Location is reached.
 5. Interacts with Guide's Personae at the M-Locations paired with the U-Location.

12.7.2 Variables

Processes	Name	Function
	ATSrv	Author Service
	EXSrv	Resolution Service of M-Instance _A
	LCServ	Locate Service of M-Instance _A
	RGSrv	Student#1
	TRSrv	Student#2
Items	Name	Function
	Account _{Ti}	Account of i-th Tourist
	Anim _j	Program animating Persona _j
	Items	Used to Author Landscape and Objects
	Landscaped	Landscape Authored by Developer
	MLoc _{GL}	Location where Landscape is MM-Added
	MLoc _{Gi}	Location where i-th Tourist Object is MM-Added
	Object _{Oi}	i-th Tourist Object
	Persona _j	Persona of i-th Tourist
	Persona _j	Persona
	P-Profile _{Ti}	Personal Profile of i-th Tourist
	Rights _{Tij}	Rights of i-th Tourist to conversation with Persona _j at j-th Tourist Object

	Transaction _{D1}	To get Rights to Landscaped
	Transaction _{D2}	To get Rights to Tourist Objects _D
	Transaction _{G1}	To get Rights to Land
	Transaction _{G2}	To get Rights to Landscaped
	Transaction _{G3}	To get Rights to Tourist Objects _D
	Transaction _{Ti}	Transaction of i-th Tourist
	Utterance _{Tij}	Conversation of i-th Tourist with Persona at j-th Tourist Object

12.7.3 Workflow

S-Process	Action	PA-Request	D-Process	PA-Response	Comment
Developer	Author	<i>With Items</i>	ATSvc	<i>Nil Landscaped</i>	Authors Landscape
		<i>With Transaction_{D1}</i>	TRSvc	<i>With Ack</i>	
	Author	<i>With Items</i>	ATSvc	<i>Nil Tourist Objects_D</i>	Authors Tourist Objects
		<i>With Transaction_{D2}</i>	TRSvc	<i>With Ack</i>	
Guide	Transact	<i>With Transaction_{G1}</i>	TRSvc	<i>With Ack</i>	Gets Rights to land
	Transact	<i>With Transaction_{G2}</i>	TRSvc	<i>With Ack</i>	Gets Rights to landscape
	MM-Add	<i>Nil Landscaped</i>	LCSvc	<i>With Ack</i>	Places landscape in land parcel
		<i>At MLoc_{GL}</i>			
		<i>With SA</i>			
	Transact	<i>With Transaction_{G3}</i>	TRSvc	<i>With Ack</i>	Gets Rights to Tourist Objects
	<i>Do i=1,N</i>				
	MM-Add	<i>Nil Object_{oi}</i>	LCSvc	<i>With Ack</i>	Places T-Objects in landscape
		<i>At MLoc_{Gi}</i>			
		<i>With SA</i>			
	<i>End</i>				
	<i>Do j=1,M</i>				
	MM-Add	<i>Nil Persona_j</i>	LCSvc	<i>With Ack</i>	Places A-Personae in landscape
		<i>At MLoc_{Gj}</i>			
		<i>With SA</i>			
	Execute	<i>Nil Anim_j</i>	EISvc	<i>With Ack</i>	Animates Persona _j
	<i>End</i>				

Tourist	Registers	<i>With</i> P-Profile _{Ti}	RGSrv	<i>Nil</i> Account _{Ti}	Tourist Registers
		<i>With</i> Transaction _{Ti}	TRSrv	<i>With</i> Ack	To open Account
	MM-Add	<i>Nil</i> Utterance _{Tij}	LCSrv	<i>With</i> Rights _{Tij}	Converses with Persona _j
		<i>At</i> MLoc _{Gj}		<i>With</i> Ack	
		<i>With</i> SA			
		<i>With</i> Perceptibility			
		<i>With</i> MRights			

12.8 Virtual Dance School

12.8.1 Description

1. Teacher:
 1. Places Persona₁ at the reception area of the dance school as virtual receptionist.
 2. Animates Persona₁ with an autonomous agent.
2. Student₁:
 1. Shows up at school.
 2. Greets receptionist.
3. Virtual receptionist reciprocates greetings.
4. Teacher:
 1. Places their Persona₂ (with haptic capabilities) in dance area₁.
 2. Dances with Student₁'s Persona (with haptic capabilities).
5. Student₂ shows up at school.
6. Teacher:
 1. Places Persona₃ (with haptic capabilities) in dance area₂.
 2. Keeps on dancing with Student₁ by animating Persona₂ with an autonomous agent.
7. Student₂ moved to Moves to dancing area₂.
8. Teacher dances with Student₂'s Persona (with haptic capabilities).

12.8.2 Variables

Processes	Name	Function
	Anim _{T1}	Process to animate Receptionist
	Anim _{T2}	Process to animate Persona#2 for Student#1
	LCServ	Locate Service of M-Instance _A
	EXSrv	Resolution Service of M-Instance _A
	Student ₁	Student#1
	Student ₂	Student#2
Items	Name	Function
	Homes ₁	Student#1's Home in the M-Instance
	Homes ₂	Student#2's Home in the M-Instance

	MLoc _{S1}	Student#1's landing position at Dance School
	MLoc _{S2}	Student#2's landing position at Dance School
	MLoc _{S3}	Student#1's position in dance area#1
	MLoc _{S3}	Student#2's position in dance area#2
	MLoc _{T1}	Receptionist's location
	MLoc _{T2}	Dancing area#1
	MLoc _{T3}	Dancing area#2
	Personas ₁	Student#1's Persona
	Personas ₂	Student#2's Persona
	Persona _{T1}	Dance Teacher's Persona#1 (Receptionist)
	Persona _{T2}	Dance Teacher's Persona#2 (Teacher#1)
	Persona _{T2}	Dance Teacher's Persona#2 (Teacher#2)
	Rights _{S1}	Student#1 Rights at Dance School
	Rights _{S2}	Student#2 Rights at Dance School
	MLoc _{C1}	Location in showroom where Customer moves its Persona
	Office _{T1}	Teacher's office in the M-Instance
	Utterances ₁	Student#1's utterance to Receptionist
	Utterance _{T1}	Receptionist utterance to Student#1

12.8.3 Workflow

S-Process	Action	PA-Request	D-Process	PA-Response	Comment
Teacher	MM-Add	<i>Nil</i> Persona _{T1}	LCS _{svc}	<i>With</i> Ack	Places Receptionist
		<i>At</i> MLoc _{T1}			
		<i>With</i> SA			
		<i>With</i> Perceptibility			
	Execute	<i>Nil</i> Anim _{T1}	EXS _{svc}	<i>With</i> Ack	Animates Receptionist
Student ₁	MM-Move	<i>Nil</i> Personas ₁	LCS _{svc}	<i>With</i> Rights _{S1}	Shows up at School
		<i>From</i> Home _{S1}		<i>With</i> Ack	
		<i>To</i> MLoc _{S1}			
		<i>With</i> SA			

		<i>With Perceptibility</i>			
		<i>With MRights</i>			
	MM-Add	<i>Nil Utterances_I</i>	LCSrvc	<i>With Ack</i>	Greets Receptionist
		<i>At MLoc_{S1}</i>			
		<i>With SA</i>			
Teacher (Persona _{T1})	MM-Add	<i>Nil Utterance_{T1}</i>	LCSrvc	<i>With Ack</i>	Reciprocates Student ₁
		<i>At MLoc_{T1}</i>			
		<i>With SA</i>			
		<i>With Perceptibility</i>			
Teacher (Persona _{T2})	MM-Move	<i>Nil Persona_{T2}</i>		<i>With Ack</i>	Moves to dancing area ₁
		<i>From Office_{T1}</i>			
		<i>To MLoc_{T2}</i>			
		<i>With SA</i>			
Student ₁	MM-Move	<i>Nil Persona₁</i>	LCSrvc	<i>With Ack</i>	Moves close to Teacher
		<i>From MLoc_{S1}</i>			
		<i>To MLoc_{S3}</i>			
		<i>With SA</i>			
Student ₂	MM-Move	<i>Nil Personas₂</i>	LCSrvc	<i>With Ack</i>	Shows up at School
		<i>From Homes₂</i>			
		<i>To MLoc_{S4}</i>			
		<i>With SA</i>			
Teacher	MM-Add	<i>Nil Persona_{T3}</i>	LCSrvc	<i>With Ack</i>	Placed at dancing area ₂
	MM-Animate	<i>Nil Persona_{T3}</i>			
		<i>At MLoc_{T3}</i>			
		<i>With Stream_{T3}</i>			
	Execute	<i>Nil Anim_{T2}</i>	ExSrv	<i>With Ack</i>	Animates Persona _{T2}
Student ₂	MM-Move	<i>Nil Persona₂</i>	LCSrvc	<i>With Ack</i>	Moves to dancing area ₂
		<i>From MLoc_{S2}</i>			
		<i>To MLoc_{S4}</i>			
		<i>With SA</i>			

12.9 Virtual Car Showroom

12.9.1 Description

1. A 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. Dealer (as showroom attendant) reciprocates the greeting.
4. Dealer (as salesman):
 1. Converses with Customer.
 2. Embeds a model of the car being considered.
5. Customer
 1. Has a virtual test drive.
 2. Buys the car.
 3. Returns home.

12.9.2 Variables

Processes	Name	Function
	AnimD1	Process to animate Dealers Persona _{D1}
	AnimD3	Process to animate Car Model
	LCServ	Locate Service of M-Instance _A
	EXSrv	Resolution Service of M-Instance _A
	TRSrv	Resolution Service of M-Instance _B
	Customer	CAV _A Passenger's User for Human-CAV Interface
	Dealer	CAV _A Passenger's User for interfacing with CAV
	User _{B,1}	CAV _B Passenger's User for Human-CAV Interface
Items	Name	Function
	CarModel _D	Model of Car being considered
	Home _C	Customer's Home in the M-Instance
	MLoc _{C1}	Location in showroom where Customer moves its Persona
	MLoc _{D1}	Location in showroom where Receptionist's Persona is MM-Added
	MLoc _{D2}	Location in showroom where Dealer's Persona is MM-Added
	MLoc _{D3}	Location in showroom where Car Model is MM-Added
	Office	Dealer's office in the M-Instance
	Person _C	Customer's Persona
	Persona _{D1}	Dealer's first Persona
	Persona _{D2}	Dealer's second Persona
	Rights _{C1}	Customer's Rights to its Persona MM-Moved to Showroom
	Rights _{C1}	Customer's Rights to its Utterance to Receptionist
	Rights _{C3}	Customer's Rights to its Conversation with Dealer (Persona _{D2})
	Transaction	Customer's Transaction to buy car
	Utterance _C	Customer's Utterance to Receptionist
	Utterance _D	Receptionist's Utterance to Customer

12.9.3 Workflow

S-Process	Action	PA-Request	D-Process	PA-Response	Comment
Dealer	MM-Add	<i>Nil</i> Person _{D1}	LCSrv	<i>With</i> Ack	Places Receptionist
		<i>At</i> MLoc _{D1}			
		<i>With</i> SA			
		<i>With</i> Perceptibility			
	Execute	<i>Nil</i> Anim _{D1}	EXSrv	<i>With</i> Ack	Animates Receptionist
Customer	MM-Move	<i>Nil</i> Person _{ac}	LCSrv	<i>With</i> Rights _{sc1}	Shows up at Showroom
		<i>From</i> Home _c		<i>With</i> Ack	
		<i>To</i> MLoc _{C1}			
		<i>With</i> SA			
		<i>With</i> MRights			
	MM-Add	<i>Nil</i> Utterance _c	LCSrv	<i>With</i> Rights _{sc2}	Greets Receptionist
		<i>At</i> MLoc _{C1}		<i>With</i> Ack	
		<i>With</i> SA			
		<i>With</i> Perceptibility			
		<i>With</i> MRights			
Dealer (Persona _{D1})	MM-Add	<i>Nil</i> Utterance _{D1}	LCSrv	<i>With</i> Ack	Reciprocates Customer
		<i>At</i> MLoc _{D1}			
		<i>With</i> SA _{D1}			
		<i>With</i> Perceptibility			
Dealer (Persona _{D2})	MM-Move	<i>Nil</i> Person _{D2}	LCSrv	<i>With</i> Ack	Moves close to Customer
		<i>From</i> Office			
		<i>To</i> MLoc _{D2}			
		<i>With</i> SA			
Dealer (Persona _{D2})	MM-Add	<i>Nil</i> Utterance _{D2}	LCSrv	<i>With</i> Ack	Converses with Customer
		<i>At</i> MLoc _{C1}			
		<i>With</i> SA			
		<i>With</i> MRights			

		<i>With</i> Perceptibility			
Customer	MM-Add	<i>Nil</i> Person _{ac}	LCSrv _c	<i>With</i> Rights _{c3}	Converses with Customer
		<i>At</i> MLoc _{c2}		<i>With</i> Ack	
		<i>With</i> SA _{c2}			
		<i>With</i> MRights			
		<i>With</i> Perceptibility			
Dealer (Persona D ₂)	MM-Add	<i>Nil</i> CarModel _D	LCSrv _c	<i>With</i> Ack	Converses with Customer
		<i>At</i> MLoc _{D3}			
		<i>With</i> SA _{D3}			
		<i>With</i> Perceptibility			
Customer	MM-Move	<i>Nil</i> Person _{ac}	LCSrv _c	<i>With</i> Ack	Sits in Car Model
		<i>From</i> MLoc _{c2}			
		<i>To</i> MLoc _{D3}			
		<i>With</i> SA _{c3}			
	MM-Move	<i>Nil</i> Person _{aD2}	LCSrv _c	<i>With</i> Ack	Back to previous MLoc
		<i>From</i> MLoc _{D3}			
		<i>To</i> MLoc _{C1}			
		<i>With</i> SA			
	Transact	<i>With</i> Transaction	TRSrv _c	<i>With</i> Ack	Pays for car
Customer	MM-Move	<i>Nil</i> Person _{ac}	LCSrv _c	<i>With</i> Ack	Returns home
		<i>From</i> MLoc _{D3}			
		<i>To</i> Home _c			
		<i>With</i> SA			

12.10 Meeting while driving

12.10.1Description

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 are independent but conform with the CAV-TEC Technical Specification.

1. human_{A1}

1. Registers with CAV_A's Registration Service with their Personal Profile containing:
 1. User₁ interacting with the CAV_A's Human-CAV Interaction Subsystem.
 2. User₂ interacting with the CAV_AM-Instance created by the Autonomous Motion Subsystem.
 3. User₁'s Persona_{A.1.1}, used to Track human_{A1}.
2. Requests User₁ to provide a Route.
2. User₁ requests and obtains proposed Routes to the destination from User_{A.2} of CAV_A, communicates the proposed Route, 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.

12.10.2 Variables

Processes	Name	Function
	COSrv _{CA}	Communicate Service of M-Instance _A
	EISrv _{CA}	Export/Import Service of M-Instance _A
	IDSrv _{CA}	Identify Service of M-Instance _A
	LCSrv _A	Locate Service of M-Instance _A
	RESrv _{CA}	Resolution Service of M-Instance _A
	RESrv _{CB}	Resolution Service of M-Instance _B
	RGSrv _{CA}	Service of M-Instance _A
	User _{A.1}	CAV _A Passenger's User for Human-CAV Interface
	User _{A.2}	CAV _A Passenger's User for interfacing with CAV
	User _{B.1}	CAV _B Passenger's User for Human-CAV Interface
Items	Name	Function
	Account _{A.1}	Account of Registering human
	AMSHCIMessage _{A.1}	User _{A.1} 's Message to User _{A.2} (request to be taken somewhere)
	AMSHCIMessage _{A.2}	User _{A.2} 's Response to User _{A.1}
	Cabin _A	Location in CAV _A 's cabin
	Cabin _B	Location in CAV _B 's cabin
	Data	MM-Captured by CAV _A
	Message _{A.1}	User _A 's Message requesting meeting
	Message _{B.1}	User _B 's Message accepting meeting
	Persona _A	User _A 's Persona with human's appearance.
	Qualifier	Of Data (Stream) MM-Captured by CAV _A
	MV _A Square	Reference location in CAV _A 's MV _A where Personae are MM-Added
	Personal Profile _A	CAV _A Passenger Personal Profile
	Resolution _{A.2}	RESrv _{CA} to RESrv _{CB} Resolution Request (request session)
	Resolution _{A.3}	RESrv _{CB} to User _B Resolution Request to User _B (request session)
	Resolution _{A.4}	Resolution Request to RESrv _{CB} (request session)

	Resolution _{A.5}	User _A to RESrv _{CA} Resolution Response (request session)
	Resolution _{A.6}	RESrv _{CA} to RESrv _{CB} Resolution Response (request session)
	Resolution _{A.7}	RESrv _{CB} to User _B Resolution Response to User _B (request session)
	Resolution _{A.8}	Resolution Response to RESrv _{CB} (leave session)
	Resolution _{A.9}	User _A to RESrv _{CA} Resolution Request (leave session)
	Resolution _{A.10}	RESrv _{CA} to RESrv _{CB} Resolution Request (leave session)
	Resolution _{A.11}	RESrv _{CB} to User _B Resolution Request to User _B (leave session)
	Resolution _{A.12}	Resolution Request to RESrv _{CB} (leave session)
	Rights _{A.1}	User _A 's Rights to Persona _A MM-Added in Cabin _A
	Rights _{A.2}	User _A 's Rights to Message _{A.1} sent to User _B
	Rights _{A.3}	User _A 's Rights to Cabin _B MU-Actuated in Cabin _B
	Rights _{B.1}	User _B 's Rights Message _{B.1} sent to User _A
	Scene	Scene captured by Cabin _A from by CAV _A
	Stream	human's animation stream

12.10.3 Workflow

S-Process	Action	PA-Request	D-Process	PA-Response	Comments
human _{A.1}	Registers	<i>With</i> PersonalProfile	RGSrv _{CA}	<i>Nil</i> Account	human _{A.1} registers with RGSrv _{CA.1}
	MM-Add	<i>Nil</i> Persona _A	LCSrv _{CA}	<i>With</i> Rights _A	Persona ₁ is MM-Added
		<i>With</i> SA		<i>With</i> Ack	
		<i>At</i> Cabin _A			
		<i>With</i> Perceptibility			Non perceptible
	UM-Capture	<i>From</i> Scene	EISrv _{CA}	<i>Nil</i> Data	Animation Stream is captured
		<i>With</i> Qualifier		<i>With</i> SA	
				<i>With</i> Ack	
	Identify	<i>Nil</i> Data	IDSrv _{CA}	<i>Nil</i> Stream	Stream Data is Identified
		<i>With</i> Qualifier		<i>With</i> Ack	
		<i>At</i> User _A			
User _{A.1}	MM-Send	AMSHCIMessage _{A.1}	COSrv _{CA}	<i>Nil</i> Route _A	Requests Route
				<i>With</i> Ack	
User _{A.2}	MM-Send	AMSHCIMessage _{A.2}	COSrv _{CA}	<i>Nil</i> Route _A	Route Response

User _{A.1}	Resolve	<i>With Resolution</i> _{A.1}	RESrv _{CA}	<i>With Ack</i>	User _{A.1} requests session with User _{B.1}
RESrv _{CA}	Resolve	<i>With Resolution</i> _{A.2}	RESrv _{CB}	<i>With Ack</i>	RESrv _{CA} contacts RESrv _{CB}
RESrv _{CB}	Resolve	<i>With Resolution</i> _{A.3}	User _B	<i>With Ack</i>	RESrv _{CB} contacts User _{B.1}
User _B	Resolve	<i>With Resolution</i> _{A.4}	RESrv _{CB}	<i>With Ack</i>	User _B responds to RESrv _{CB}
RESrv _{CB}	Resolve	<i>With Resolution</i> _{A.5}	RESrv _{CA}	<i>With Ack</i>	RESrv _{CB} responds to RESrv _{CA}
RESrv _{CA}	Resolve	<i>With Resolution</i> _{A.6}	User _A	<i>With Ack</i>	RESrv _{CA} responds to User _{A.1}
User _{A.1}	MM-Send	<i>Nil Message</i> _{A.1}	User _B	<i>With Rights</i> _{A.2}	Requests meeting with User _{B.1}
		<i>With MRights</i>		<i>With Ack</i>	
User _{B.1}	Rights Change	<i>Nil User</i> _{A.1}	RTSrv _{CB}	<i>With Rights</i> _{B.1}	Changes Rights of User _{A.1}
		<i>With MRights</i>		<i>With Ack</i>	
		<i>To User</i> _{A.1}		<i>With Ack</i>	
User _{A.1}	MM-Move	<i>Nil Persona</i> _A	LCSrv _{CA}	<i>With Ack</i>	Moves to CAV _B
		<i>From Cabin</i> _A		<i>With Rights</i> _{A.3}	
		<i>To Cabin</i> _B			
		<i>With SA</i>			
		<i>With Perceptibility</i>			Persona _A is perceptible
	MM-Anim	<i>Nil Persona</i> _A	LCSrv _{CA}	<i>With Ack</i>	Persona is MM-Animated
		<i>At Cabin</i> _B			
		<i>With Stream</i>			
	MU-Actuate	<i>Nil Cabin</i> _B		<i>Nil Media</i>	Actuate Cabin _B
		<i>At Cabin</i> _A		<i>With Rights</i> _{A.3}	The real cabin of CAV _A
		<i>With SA</i>		<i>With Ack</i>	
		<i>With MRights</i>			
User _{A.1}	Resolve	<i>With Resolution</i> _{A.7}	RESrv _{CA}	<i>With Ack</i>	Leaves Multi-MV session
RESrv _{CA}	Resolve	<i>With Resolution</i> _{A.8}	RESrv _{CB}	<i>With Ack</i>	RESrv _{CA} contacts RESrv _{CB.1}
RESrv _{CB}	Resolve	<i>With Resolution</i> _{A.9}	User _B	<i>With Ack</i>	RESrv _{CB} contacts User _B

User _B	Resolve	<i>With Resolution</i> _{A.10}	RESrv _{CB}	<i>With Ack</i>	User _B responds to RESrv _{CB.1}
RESrv _{CB}	Resolve	<i>With Resolution</i> _{A.11}	RESrv _{CA}	<i>With Ack</i>	RESrv _{CB} responds to RESrv _{CA.1}
RESrv _{CA}	Resolve	<i>With Resolution</i> _{A.12}	User _A	<i>With Ack</i>	RESrv _{CA} responds to User _{A.1}

12.11 Co-design across metaverses

12.11.1 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, a Company_B Designer, 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, e.g., rearview mirrors, dashboards, cabin lights etc.
4. The progress meeting is held at Room_B attended by Customer (User_A) and Designer (User_B).
5. User_B has full Rights to CarModel_B and and Room_B.
6. User_A initiates a session by requesting Resolution Service_A to open contact with User_B.
7. Resolution Service_A forwards the request to Resolution Service_B.
8. Resolution Service_B forwards the request to User_B.
9. User_B responds to Resolution Service_B accepting the invitation.
10. Resolution Service_B forwards the response to Resolution Service_A.
11. Resolution Service_A forwards the responses to User_A.
12. User_B
 1. MM-Adds Car Model_B at MLoc_B with perceptibility status on.
 2. Grants Rights_B to User_A to Act on CarModel_B and Room_B.
13. User_A assesses the results of the design made so far as follows:
 1. PropertyChanges RearviewMirror₁ making it imperceptible.
 2. PropertyChanges RearviewMirror₂ making it perceptible (both Rearview Mirrors are co-located).
 3. PropertyChanges RearviewMirror₂ giving it material consistency.
 4. PropertyChanges RearviewMirror₂ resizing it by R_x, R_y, R_z

12.11.2 Variables

Processes	Name	Function
	LCServ _A	Locate Service of M-Instance _A
	LCServ _B	Locate Service of M-Instance _B
	RESrv _{CA}	Resolution Service of M-Instance _A
	RESrv _{CB}	Resolution Service of M-Instance _B
	User _A	Company _A 's User
	User _B	Company _B 's User
Items	Name	Function
	Intensity	Light Source _B 's intensity
	LightSource _B	Source illuminating meeting room
	MLoc _A	User _A 's location in M-Instance _B meeting room

	MLoc _B	User _B 's location in M-Instance _B meeting room
	Persona _A	User _A 's Persona
	Person _B	User _B 's Persona
	Resolution _{A.1}	User _A to RESrv _{CA} Resolution Request (request session)
	Resolution _{A.2}	RESrv _{CA} to RESrv _{CB} Resolution Request (request session)
	Resolution _{A.3}	RESrv _{CB} to User _B Resolution Request to User _B (request session)
	Resolution _{A.4}	Resolution Request to RESrv _{CB} (request session)
	Resolution _{A.5}	User _A to RESrv _{CA} Resolution Response (request session)
	Resolution _{A.6}	RESrv _{CA} to RESrv _{CB} Resolution Response (request session)
	Resolution _{A.7}	RESrv _{CB} to User _B Resolution Response to User _B (request session)
	Resolution _{A.8}	Resolution Response to RESrv _{CB} (leave session)
	Resolution _{A.9}	User _A to RESrv _{CA} Resolution Request (leave session)
	Resolution _{A.10}	RESrv _{CA} to RESrv _{CB} Resolution Request (leave session)
	Resolution _{A.11}	RESrv _{CB} to User _B Resolution Request to User _B (leave session)
	Resolution _{A.12}	Resolution Request to RESrv _{CB} (leave session)
	RVMirror _{B.1}	Rearview mirror with default perceptibility and material
	RVMirror _{B.2}	Another rearview mirror
	LightSource _B	Light Source in MLoc _B

12.11.3 Workflow

S-Proc	Action	RA-Request	D-Process	RA-Response	Comments
User _A	Resolve	<i>With</i> Resolution _{A.1}	RESrv _{CA}	<i>With</i> Ack	User _A requests Multi-MV session
RESrv _{CA}	Resolve	<i>With</i> Resolution _{A.2}	RESrv _{CB}	<i>With</i> Ack	RESrv _{CA} contacts RESrv _{CB}
RESrv _{CB}	Resolve	<i>With</i> Resolution _{A.3}	User _B	<i>With</i> Ack	RESrv _{CB} contacts User _B
User _B	Resolve	<i>With</i> Resolution _{A.4}	RESrv _{CB}	<i>With</i> Ack	User _B responds to RESrv _{CB}
RESrv _{CB}	Resolve	<i>With</i> Resolution _{A.5}	RESrv _{CA}	<i>With</i> Ack	RESrv _{CB} responds to RESrv _{CA}
RESrv _{CA}	Resolve	<i>With</i> Resolution _{A.6}	User _A	<i>With</i> Ack	RESrv _{CA} responds to User _A
User _A	MM-Add	<i>Nil</i> Persona _A	LCSrv _{CA}	<i>With</i> Ack	Is moved to MLoc _{B.1}
		<i>At</i> MLoc _{B.1}			
		<i>With</i> SA			
		<i>With</i> Perceptibility			Is perceptible
User _{B.1}	MM-Add	<i>Nil</i> Person _B	LCSrv _{CA}	<i>With</i> Ack	Is moved to MLoc _{B.2}
		<i>At</i> MLoc _{B.2}			
		<i>With</i> SA			
		<i>With</i> Perceptibility			Is perceptible
	MM-Add	CarModel	LCSrv _B	<i>With</i> Ack	User _B places Car Model
		<i>At</i> MLoc _{B.3}			

		<i>With SA</i>			
		<i>With Perceptibility</i>			
User _A	Property Change	<i>Nil RVMirror_{B.1}</i>	LCSrv _A	<i>With Ack</i>	RVMirror _{B.1} is made invisible and immaterial
		<i>With Perceptibility</i>			
User _A	Property Change	<i>Nil RVMirror_{B.2}</i>	LCSrv _A	<i>With Ack</i>	RVMirror _{B.2} is made visible and material
		<i>With Perceptibility</i>			
User _A	Property Change	<i>Nil RVMirror_{B.2}</i>	LCSrv _A	<i>With Ack</i>	RVMirror _{B.2} is made invisible
		<i>With Perceptibility</i>			
User _A	Property Change	<i>Nil RVMirror_{B.2}</i>	LCSrv _A	<i>With Ack</i>	Changes RVMirror _{B.2} 's size
		<i>With (R_x,R_y,R_z)</i>			
User _A	Property Change	<i>Nil LightSource_B</i>	LCSrv _A	<i>With Ack</i>	Changes LightSource _B properties
		<i>With Intensity</i>			
User _A	Resolve	<i>With Resolution_{A.7}</i>	RESrv _A	<i>With Ack</i>	User _{A.1} leaves Multi-MV session
RESrv _A	Resolve	<i>With Resolution_{A.8}</i>	RESrv _B	<i>With Ack</i>	RESrv _{A.1} contacts RESrv _B
RESrv _B	Resolve	<i>With Resolution_{A.9}</i>	User _B	<i>With Ack</i>	RESrv _{B.1} contacts User _B
User _B	Resolve	<i>With Resolution_{A.10}</i>	RESrv _B	<i>With Ack</i>	User _{B.1} responds to RESrv _B
RESrv _B	Resolve	<i>With Resolution_{A.11}</i>	RESrv _A	<i>With Ack</i>	RESrv _{B.1} responds to RESrv _A
RESrv _A	Resolve	<i>With Resolution_{A.12}</i>	User _A	<i>With Ack</i>	RESrv _{A.1} responds to User _A

12.12 Selling assets on a Marketplace

12.12.1 Description

1. Seller
 1. Produces Asset
 2. Posts Asset to a Marketplace Service (Mkt Service).
2. Mkt Service places the Asset at its exhibition.
3. Buyer
 1. Is interested in the Asset.
 2. Inquires about it with the Mkt Service's Information Service.
 3. Makes a Transaction to Mkt Service.
 4. Gets Rights to the Asset.

12.12.2 Variables

Processes	Name	Function
	ATSrv	Author Service
	ITSrv	Information Service

	LCSrv	Location Service
	TRSRvc	Transaction Service
	Buyer	User making a Transaction to get Rights to Asset
	Mkt Service	Marketplace Service
	Seller	User Posting Asset
Items	Name	Function
	Asset	Asset Posted to Mkt Service by Seller.
	Information	Request to and Response from Information Service
	MLoc	Location where Asset is dispalyed
	Rightss	Seller's Rights to Asset
	Rightss	Mkt Service's Rights to Asset
	Transactions ₁	Transaction made by Seller to Author Service
	Transactions ₂	Transaction made by Seller to Mkt Service
	Transaction _{B1}	Transaction made by Buyer to Seller and Mkt Service

12.12.3Workflow

S-Process	Action	PA-Request	D-Process	PA-Response	Comments
Seller	Author	<i>With</i> Item	ATSrv	<i>Nil</i> Asset	Authors Asset
		<i>At</i> Seller		<i>With</i> Ack	
		<i>With</i> Transactions ₁			
	Post	<i>Nil</i> Asset		<i>With</i> Rightss	Posts Asset
		<i>To</i> MktService		<i>With</i> Ack	
		<i>With</i> MRights			
		<i>With</i> Transactions ₂			
MktService	MM-Add	<i>Nil</i> Asset	LCSrv	<i>With</i> Rights _M	Displays Asset
		<i>At</i> MLoc		<i>With</i> Ack	
		<i>With</i> Perceptibility			Asset is Perceptible
Buyer	Inform	<i>With</i> Information	ITSrv	<i>Nil</i> Information	Requests information on Asset
	Transact	<i>With</i> Transaction _{B1}	TRSRvc	<i>With</i> Ack	Gets Rights to Asset

12.13 Emergency in Industrial Metaverse

12.13.1 Description

1. An M-Location includes the Digital Twin of a real factory (R-Factory). In this use case the regular operation is separated from emergency operation.
2. An “emergency” User in the Digital Twin (V-Factory).
 1. Has the Rights to actuate and animate an “emergency” robot in the R-Factory.
 2. Can be rendered as a Persona having the appearance of the corresponding robot.
3. In case of an emergency, the “emergency” User:
 1. Activates an alarm in the R-Factory.
 2. Actuates its “emergency” robot in the R-Factory.
 3. Animates the robot to solve the problem.
 4. Renders its Persona so that humans can see what is happening in the R-Factory
4. When the emergency is resolved, the robot is moved to its repository.

12.13.2 Variables

Processes	Name	Function
	EISrvc	Export/Import Service
	User ₁	Process in the R-Factory’s Digital Twin
	User ₂	User representing human who has heard the bell ringing
Items	Name	Function
	Repository	Space containing R-Factory’s Robots.
	Rights ₁	Rights to MU-Actuated V-Scene of the V-Factory
	Rights ₂	Rights to Robot MM-Added at ULoc ₂
	Rights ₃	Rights to MM-Animated Robot
	Rights ₄	Rights to Robot MM-Moved back to Repository
	Robot	Analogue Twin of User’s Persona
	Scene	In the V-Factory to be MU-Actuate in the R-Factory
	Sound	Alarm sound MU-Added at ULoc ₁
	Stream	Used to MU-Animate Robot
	Trajectory	Followed by Robot to reach ULoc ₂
	ULoc ₁	Place in the R-Factory where the alarm bell is located.
	ULoc ₂	Place in the R-Factory where the Robot is MU-Added.

12.13.3 Workflow

S-Process	Action	PA-Request	D-Process	PA-Response	Comments
-----------	--------	------------	-----------	-------------	----------

User ₁	MU-Add	<i>Nil</i> Sound	EISrvc	<i>With</i> Ack	Sounds alarm bell
		<i>At</i> ULoc ₁			
User ₂	MU-Actuate	<i>Nil</i> Scene		<i>With</i> Rights ₁	
		<i>At</i> ULoc ₂		<i>With</i> Ack	
		<i>With</i> SA			
		<i>With</i> MRights			
User ₁	MU-Add	<i>Nil</i> Robot	EISrvc	<i>With</i> Rights ₂	Call Robot from repository
		<i>From</i> Repository		<i>With</i> Ack	
		<i>To</i> ULoc ₂			
		<i>With</i> Trajectory			
		<i>With</i> MRights			
	MU-Animate	<i>Nil</i> Robot	EISrvc	<i>With</i> Rights ₃	
		<i>At</i> ULoc ₂		<i>With</i> Ack	
		<i>With</i> Stream			
		<i>With</i> MRights			
	MU-Move	<i>Nil</i> Robot	EISrvc	<i>With</i> Rights ₄	Robot returns to repository
		<i>From</i> ULoc ₂		<i>With</i> Ack	
		<i>To</i> Repository			
		<i>With</i> MRights			

13 MMM-API

13.1 Introduction

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 MMM-API can also be leveraged to provide Users with specific functionalities through Services. A list of Services used in [Verification Use Cases](#), e.g., Registration or Rights.

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 follow:

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-TEC V2.1 [Item](#).

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 Rights Service

Base Route: /api/Rights

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/Rights/accounts/<AccountId>/<UserId>/rights' \

-H 'accept: */*' 
```

13.4.11 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 V2.1 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 - which is not part of the MMM-TEC V2.1 Reference Software - 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.