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

MPAI Technical Specification

Object and Scene Description
(MPAI-OSD)

V1.3

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 and Disclaimers.

Technical Specification
Object and Scene Description (MPAI-OSD) V1.3

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 AIWs in the standard AI Framework environment depicted in Figure 1. An AI Framework can execute AI

applications called AI Workflows (AIW) typically including interconnected AI Modules (AIM). MPAI-AIF supports small- and large-scale high-performance components and promotes solutions with improved explainability.

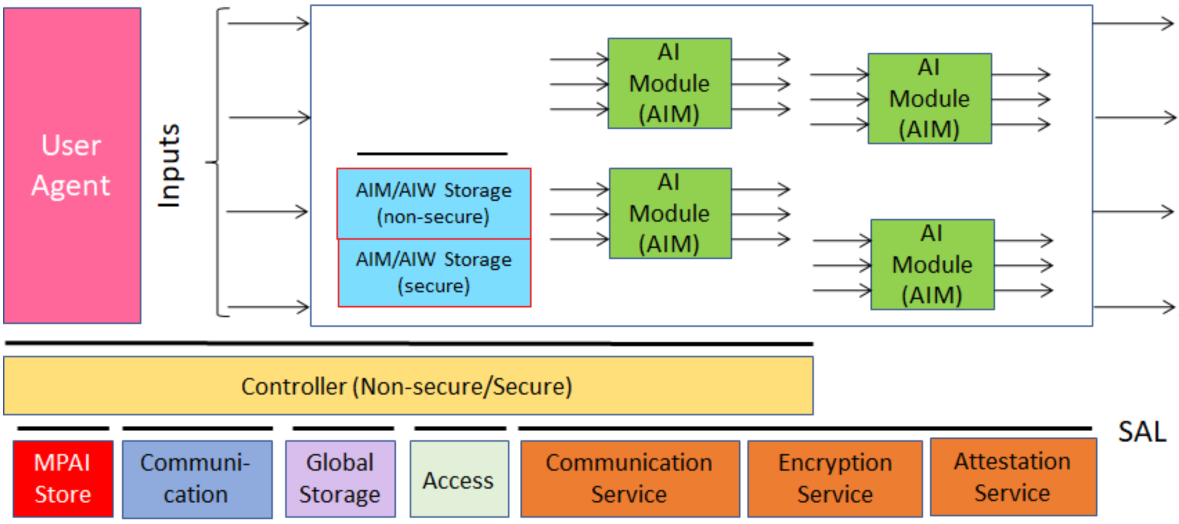


Figure 1 – The AI Framework (MPAI-AIF) V2 Reference Model

3. ***Technical Specification: Data Types, Formats, and Attributes (MPAI-TFA) V1.2*** specifies Qualifiers, a type of metadata supporting the operation of AIMs receiving data from other AIMs or from input data. Qualifiers convey information on Sub-Types (e.g., the type of colour), Formats (e.g., the type of compression and transport), and Attributes (e.g., semantic information in the Content). Although Qualifiers are human-readable, they are only intended to be used by AIMs. Therefore, Text, Speech, Audio, Visual, and other Data received by or exchanged between AIWs and AIMs should be interpreted as being composed of Content (Text, Speech, Audio, and Visual as appropriate) and associated Qualifiers. For instance, a Text Object is composed of Text Data and Text Qualifier. The specification of most MPAI Data Types reflects this point.
4. ***Technical Specification: Governance of the MPAI Ecosystem (MPAI-GME) V1.1*** 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.

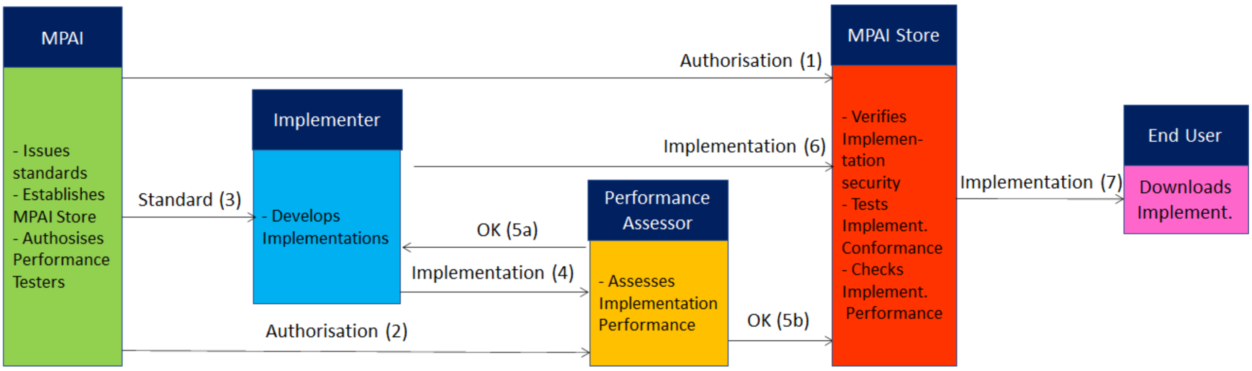


Figure 2 – The MPAI Ecosystem

2 Introduction

(Informative)

Several MPAI Technical Specifications need data types that refer to Objects and Scenes that are uni- and multimodal, refer to locations that may be in virtual spaces and include Space and Time information.

Technical Specification: Object and Scene Description (MPAI-OSD) V1.3 – in the following also called MPAI-OSD V1.3 or MPAI-OSD – has been developed to specify Time, Space, Object and Scene information for unified use across MPAI Technical Specifications and the specification to the Television Media Analysis (OSD-TMA) AI Workflow (AIW).

MPAI-OSD includes the complete OSD-TMA Reference Software Specification including the Open Source Software Implementation of the AIW and AIMs (released with BSD-3-vlause licence). The Reference Software Implementation offers users the opportunity to access Implementations that are Conforming with the Technical Specification. The AIMs used by OSD-TMA are sourced by the MPAI-CAE, MPAI-MMC, MPAI-OSD, and MPAI-PAF Technical Specifications.

In all Chapters and Sections, Terms beginning with a capital letter are defined in *Table 1* if they are specific to MPAI-OSD. All MPAI-defined Terms are accessible online. All Chapters, and Sections are Normative unless they are labelled as Informative.

3 Scope

Technical Specification: Object and Scenes Description (MPAI-OSD) V1.3 specifies:

1. Data Types and associated Qualifiers that are used to specify uni- and multi-modal Objects and Scenes in Virtual Environments with attributes of Space and Time.
2. The MPAI-OSD AI Modules and the Data Types.
3. The Television Media Analysis (OSD-TMA) AI Workflow using AI Modules and Data Types from MPAI-CAE, MPAI-MMC, and MPAI-PAF.

MPAI-OSD has been developed for uniform use across MPAI Technical Specifications with the cooperation of

1. The *AI Framework* (AIF-DC), *Context-based Audio Enhancement* (CAE-DC), *Multimodal Conversation* (MMC-DC), and *Portable Avatar Format* (PAF-DC).
2. The *Connected Autonomous Vehicle* (CAV) and *MPAI Metaverse Model* (MMM) groups of the Requirements Standing Committee.

In the future, MPAI may publish new Technical Specification in the MPAI-OSD scope that modify or extend the scope of this Technical Specification.

4 Definitions

Capitalised Terms have the meaning defined in *Table 1*. All MPAI-defined Terms are accessible online. Non-capitalised terms have the meaning commonly defined for the context in which they are used or represent an entity in the real world. For instance, Table 1 defines *Object, Scene*, and *User* but does not define *object*, *scene*, and *human*.

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

1. If the font is normal, the Term in Table 1 without a dash and preceding the one with a dash should be placed before that Term. The notation is used to concentrate in one place all the Terms that are composed of, e.g., the word Data followed by one of the words Format and Type.
2. If the font is *italic*, the Term in the table without a dash and preceding the one with a dash should be placed after that Term. The notation is used to concentrate in one place all the Terms that are composed of, e.g., the word Descriptor preceded by one of the words Face and Body.

Table 1 – Terms and Definitions

Term	Definition
Attitude	
- *Spatial*	Position and Orientation and their velocities and accelerations of a Human and Visual Object in a Virtual Environment.
Audio	A Data Type an instance of which represents analogue signals – or is rendered to be perceived – in the human-audible range (16 Hz - 20 kHz).
Avatar	An Data Type including the 3D Model of an Avatar and the Face and Body Descriptors.
- Model	An inanimate Avatar exposing animation interfaces.
- *Portable*	A Data Type including Avatar ID, Time, Avatar, Language, Speech, Text, Speech Model, Personal Status, Audio-Visual Scene Descriptors, and potentially an input Portable Avatar.
Centre Point	The point of an Object selected to have coordinates (0,0,0).
Coordinate System	A system where the position of a point is specified by three numbers.
- *Cartesian*	A coordinate system where the three numbers are the signed distances from the point to three mutually perpendicular planes.
- *Spherical*	A coordinate system where the three numbers are: - the radial distance of that point from a fixed origin. - the polar angle measured from a fixed zenith direction. - the azimuthal angle of its orthogonal projection on a reference plane.
Data	Information in digital form.
- Format	A specific digital representation of Data.
- *Media*	Data representing Text, Speech, Audio, Visual, 3D Model, LiDAR, RADAR, Ultrasound information.
- Object	A Data Type including Data of a given Data Type and the Qualifier of that Data Type.
- Type	A recognised instance of Data.
Descriptor	The Digital Representation of a feature of an Object.

- *Audio-Visual*	A Data Type including the digital representation of the features of an audio-visual instance.
- *Body*	A Data Type including the digital representation of the features of the body of a real or digital human.
- *Face*	A Data Type including the digital representation of a feature of the face of a real or digital human.
- *Visual*	A Data Type including the digital representation of the features of a visual instance.
Digital Representation	Data corresponding to and representing a physical entity.
Environment	A Virtual Space that may be null or may include an Audio-Visual Scene.
Human	A human being in a real space.
- *Digital*	A Digitised or a Virtual Human.
- *Digitised*	An Object that has the appearance of a specific human when rendered.
- *Virtual*	An Object created by a computer that has a human appearance when rendered but is not a Digitised Human.
Identifier	The label uniquely associated with a human or an Object.
Instance	An element of a set of entities – Scenes, Digital Humans etc. – belonging to some levels in a hierarchical classification (taxonomy).
Object	A Data Type including Media Data and an optional Qualifier.
- *3D Model*	A Data Type including 3D Model Data and Qualifier.
- *Audio*	A Data Type including Audio Data and Qualifier.
- *Audio-Visual*	A Data Type including Audio-Visual Data and Qualifier.
- *Digital*	A Digitised or a Virtual Object.
- *Digitised*	Data representing a real object.
- *Speech*	A Data Type including Speech Data and Qualifier.
- *Text*	A Data Type including Text Data and Qualifier.
- *Visual*	A Data Type including Visual Data and Qualifier.
Orientation	The 3 Euler angles of an Object in a Virtual Space.
Position	The coordinates of a representative point for an object in a Virtual Space with respect to a set of coordinate axes.
Rendering	The process of instantiating Data or a Virtual Space as a human-perceptible entity.
Scene	A composition of Objects located according to a Scene Geometry.
- *3D Model*	A Scene composed of 3D Model Objects.
- *Audio*	A Scene composed of Audio Objects.
- *Audio-Visual*	A Scene composed of Speech and Audio Objects, Visual and 3D Model Objects, and co-located Audio-Visual Objects.
- *Speech*	A Scene composed of Speech Objects.
- *Visual*	A Scene composed of Visual Objects.
Scene Descriptors	A Data Type including the Media Objects and their spatial arrangement in a Scene.

- 3D Model	A Data Type including a Scene’s 3D Model Objects and Sub-Scenes, and their spatial arrangement.
- *Audio*	A Data Type including an Audio Scene’s Audio Objects and Sub-Scenes, and their spatial arrangement.
- *Audio-Visual*	A Data Type including an Audio Scene’s Speech, Audio, Visual, 3D Model, and Audio-Visual Objects, and their spatial arrangement.
- *Visual*	A Data Type including a Visual Scene’s Visual Objects and Sub-Scenes and their spatial arrangement.
Scene Geometry	A Data Type including the spatial arrangement of the Media Objects in a Scene.
3D Model	A Data Type describing the spatial arrangement of the 3D Model Objects and Sub-Scenes in a Scene.
- *Audio*	A Data Type describing the spatial arrangement of the Visual Objects and Sub-Scenes of a Scene.
- *Audio-Visual*	A Data Type describing the spatial arrangement of the Speech, Audio, Visual, 3D Model, and Audio-Visual Objects and Sub-Scenes of a Scene.
- *Speech*	A Data Type describing the Spatial arrangement of the Speech Objects and Sub-Scenes of a Scene.
- *Visual*	A Data Type describing the Spatial arrangement of the Visual Objects and Sub-Scenes of a Scene.
Selector	Input Data having the goal to set a parameter (e.g., use of Text vs Speech or Language Preference) or an operating mode of a Machine.
Speech	A Data Type an instance of which represents or is rendered to be perceived as human utterances.
Text	A series of characters drawn from a finite alphabet of a character set.
- *Recognised*	The Text at the output of an Automatic Speech Recognition AIM.
- *Refined Text*	The Text at the output of a Natural Language Understanding AIM.
Virtual Space	A space generated and maintained by a computing platform that can be rendered.

5 References

5.1 Normative Reference

1. MPAI; Technical Specification: AI Framework (MPAI-AIF) V2.1.
2. MPAI; Technical Specification: Context-based Audio Enhancement (MPAI-CAE) - Use Cases (CAE-USC) V2.3.
3. MPAI; Technical Specification: Multimodal Conversation (MPAI-MMC) V2.3.
4. MPAI; Technical Specification: Portable Avatar Format (MPAI-PAF) V1.4.
5. MPAI; Technical Specification: AI Module Profiles (MPAI-PRF) V1.0.
6. MPAI; Technical Specification: Data Types, Formats, and Attributes(MPAI-TFA) V1.3.

5.2 Informative References

7. MPAI; The MPAI Statutes; N421
8. MPAI; Patent Policy
9. MPAI; Technical Specification: Governance of the MPAI ecosystem (MPAI-GME) V1.1.

10. MPAI; Framework Licence: [Object and Scene Description]
11. MPAI; Technical Specification: [Connected Autonomous Vehicles] (MPAI-CAV) – [Technologies] (CAV-TEC) V1.0.
12. MPAI; Technical Specification: [MPAI Metaverse Model] (MPAI-MMM) – [Technologies] (MMM-TEC) V2.0.

6 AI Workflow: Television Media Analysis

6.1 Functions

The Television Media Analysis (OSD-TMA) AI Workflow produces Audio-Visual Event Descriptors (OSD-AVE) composed of a set of Audio-Visual Scene Descriptors (AVS) that include

1. Relevant Audio, Visual, or Audio-Visual Object
2. IDs of speakers and ID of faces with their spatial positions
3. Text from utterances

of a television program provided as input together with textual information that OSD-TMA may use to improve its performance.

OSD-TMA assumes that there is only one speaker at a time.

6.2 Reference Model

Figure 1 depicts the Reference Model of TV Media Analysis (OSD-TMA).

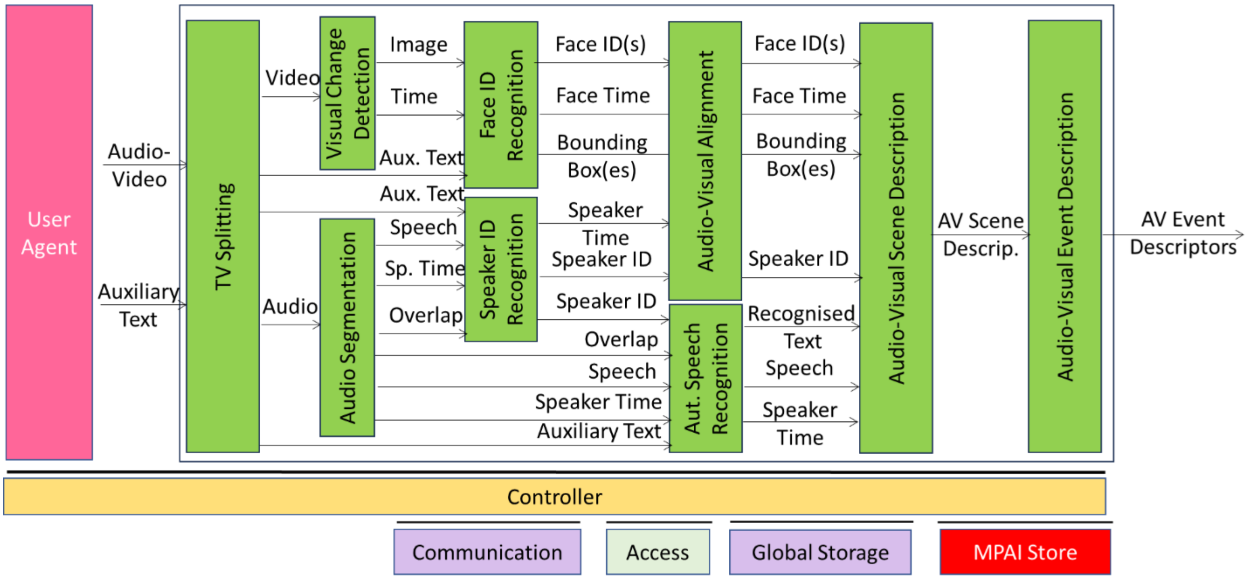


Figure 1 – Reference Model of OSD-TMA

6.3 Input/Output Data

Table 1 provides the input and output data of the TV Media Analysis Use Case:

Table 1 – I/O Data of TV Media Analysis (OSD-TMA)

Input	Descriptions
[Audio-Video]	Audio-Video to be analysed.
[Text Object]	Text helping OSD-TMA AIMs to improve their performance.

Output	Descriptions
Audio-Visual Event Descriptors	Resulting analysis of Input Audio-Video-Text.

6.4 Functions of AI Modules

Table 2 provides the functions of the TV Media Analysis Use Case. Note that processing proceeds asynchronously, e.g., TV Splitter separates audio and video for the entire duration of the file and passes the entire audio and video files.

Table 2 – Functions of AI Modules of TV Media Analysis (OSD-TMA)

AIM	Function
TelevisionSplitting	Receives Audio-Visual File composed of: - An Audio-Video component. - A Text component. Produces - Video file - Audio file - Text file When all the files of the full duration of the video are ready, AV Splitter informs the following AIMs.
VisualChangeDetection	Receives Video file. Iteratively: - Looks for a video frame that conveys a scene changed from the preceding scene (depends on threshold). - Assigns a video clip identifier to the video clip. - Produces a set of images with StartTime and EndTime.
AudioSegmentation	Receives Audio file. Iteratively: - For each audio segment (from one change to the next): - Becomes aware that there is speech. - Assigns a speech segment ID and anonymous speaker ID (i.e., the identity is unknown) to the segment. - Decides whether: - The existing speaker has stopped. - A new speaker has started a speech segment. - If a speaker has started a speech: - Assigns a new speech segment ID. - Check whether the speaker is new or old in the session. - If old retain old anonymous speaker ID. - If new assign a new anonymous speaker ID. - Produces a series of audio sequences each of which contains: - A speech segment. - Start and end time. - Anonymous Speaker ID. - Overlap information
Face Identity Recognition	- Receives a Text file. - Extracts semantic information from the Text file. - Receives a set of images per video clip. - For each image identifies the bounding boxes.

	- Extracts faces from the bounding boxes. - Extracts the embeddings that represent a face. - Compares the embeddings with those stored in the face recognition data base. - Associates the embeddings with a face ID taking into account the semantic information from the Text.
Speaker Identity Recognition	1. Receives a Text file. 2. Extracts semantic information from the Text file. 3. Receives a Speech Object and Speech Overlap information. 4. Extracts the embeddings that represent the speech segment. 5. Compares the embeddings with those stored in the speaker recognition data base. 6. Associates the embeddings with a Speaker ID taking into account the semantic information from the Text.
Audio-Visual Alignment	Receives: - Face ID - Bounding Box - Face Time - Speaker ID - Speaker Time Associates Speaker ID and Face ID
Automatic Speech Recognition	- Receives a Text file. - Extracts semantic information from the Text file. - Receives a Speech Object. - Produces the transcription of the speech payload taking into account the semantic information from the Text. - Attaches time stamps to specific portions of the transcription.
Audio-Visual Scene Description	Receives - Bounding box coordinates, Face ID, and time stamps - Speaker ID and time stamps. - Reconciles Face ID and Speaker ID. - Text and time stamps Produces Audio-Visual Scene Descriptors
Audio-Visual Event Description	Receives Audio-Visual Scene Descriptors Produces Audio-Visual Event Descriptors

6.5 I/O Data of AI Modules

Table 3 provides the I/O Data of the AI Modules of the TV Media Analysis Use Case.

Table 3 – I/O Data of AI Modules of TV Media Analysis (OSD-TMA)

AIM	Receives	Produces
TelevisionSplitting	Audio - Video - AuxiliaryText	- Audio - Video - AuxiliaryText
VisualChangeDetection	- Video	- Image
AudioSegmentation	- Speech	- SpeechObjects - SpeechOverlap

Face Identity Recognition	- Image - Time - AuxiliaryText	VisualSceneDescriptors with: - FaceID - FaceTime - BoundingBox
Speaker Identity Recognition	- SpeechObject - SpeakerTime	- SpeechSceneDescriptors: with: - Speaker ID - SpeakerTime
Audio-Visual Alignment	- SpeechOverlap - SpeechObject - SpeakerTime - AuxiliaryText	- Recognised Text - SpeechObject - SpeakerTime
Automatic Speech Recognition	- SpeechSceneDescriptors - VisualSceneDescriptors	- AVSceneDescriptor with: - AlignedFaceID - BoundingBox
Audio-Visual Scene Description	- BoundingBox - AlignedFaceID - SceneTime - SpeakerID - RecognisedText	- AVSceneDescriptors
Audio-Visual Event Description	- AVSceneDescriptors:	- AVEventDescriptors

6.6 AIW, AIMs, and JSON Metadata

Table 4 – AIW, AIMs, and JSON Metadata of TV Media Analysis (OSD-TMA)

AIW	AIM	Name	JSON
OSD-TMA		Television Media Analysis	X
	MMC-AUS	Audio Segmentation	X
	OSD-AVA	Audio-Visual Alignment	X
	OSD-AVE	Audio-Visual Event Description	X
	OSD-AVS	Audio-Visual Scene Description	X
	MMC-ASR	Automatic Speech Recognition	X
	PAF-FIR	Face Identity Recognition	X
	MMC-SIR	Speaker Identity Recognition	X
	OSD-TVS	Television Splitting	X
	OSD-VCD	Visual Change Detection	X

6.7 Reference Software

6.7.1 Disclaimers

1. This OSD-TMA Reference Software Implementation is released with the BSD-3-Clause licence.
2. The purpose of this Reference Software is to provide a working Implementation of OSD-TMA, not to provide a ready-to-use product.
3. MPAI disclaims the suitability of the Software for any other purposes and does not guarantee that it is secure.

4. Use of this Reference Software may require acceptance of licences from the respective repositories. Users shall verify that they have the right to use any third-party software required by this Reference Software.

6.7.2 Guide to the OSD-TMA code

The OSD-TMA Reference Software:

1. Receives any input Audio-Visual file that can be demultiplexed by FFMPEG.
2. Uses FFMPEG to extract:
    1. A WAV file (uncompressed audio).
    2. A video file.
3. Produces descriptors of the input Audio-Visual file represented by Audio-Visual Event Descriptors.

The current OSD-TMA implementation does not support Auxiliary Text.

The OSD-TMA uses the following software components:

1. RabbitMQ is an MPAI-AIF function but is adapted for use in OSD-TMA. This service starts, stops and removes all OSD-TMA AIMs automatically using Docker in Docker (DinD): the controller lets only one dockerized OSD-TMA AIM run at any time, which saves computing resources. Docker is used to automate the deployment of all OSD-TMA AIMs so that they can run in different environments.
2. Portainer is a service that helps manage Docker containers, images, volumes, networks and stacks using a Graphical User Interface (GUI). Here it is mainly used to manage two containerized services: RabbitMQ and Controller
3. Docker Compose has a yml file to build and run all OSD-TMA AIMs and the Module for the Controller
4. Controller builds the Docker images of all OSD-TMA AIMs and of the Controller by reading a YAML file called compose.yml. Docker Compose helps run RabbitMQ and Controller as Docker containers.
5. Python code manages the MPAI-AIF operation but is adapted for use in OSD-TMA.

The OSD-TMA software includes compose.yml and controller. They can be found at https://experts.mpai.community/software/mpai-aif/osd_tma/orchestrator. compose.yml starts RabbitMQ, nor Portainer. https://docs.portainer.io/start/install-ce describes how to install Portainer.

The OSD-TMA code includes compose.yml and the code for the Controller.

compose.yml starts RabbitMQ, not Portainer. Portainer installation is described.

The OSD-TMA code is released with a demo. Recorded demo: The saved JSON of OSD-AVE.

The OSD-TMA Reference Software has been developed by the MPAI *AI Framework* Development Committee (AIF-DC), in particular, Francesco Gallo (EURIX) and Mattia Bergagio (EURIX) for developing this software.

The implementation includes the AI Modules of the following table. AIMs in red achieve high performance if a GPU to used. However, the software can also operate without GPU.

AIM	Name	Library	Dataset
MMC-ASR	Speech recognition	whisper-timestamped	Labeled Faces in the Wild
MMC-SIR	Speaker identification	SpeechBrain	VoxCeleb1
MMC-AUS	Audio segmentation	pyannote.audio	
OSD-AVE	Event descriptors		
OSD-AVS	Scene descriptors		
OSD-TVS	TV splitting	ffmpeg	

OSD-VCD	Visual change	PySceneDetect
PAF-FIR	Face recognition	DeepFace

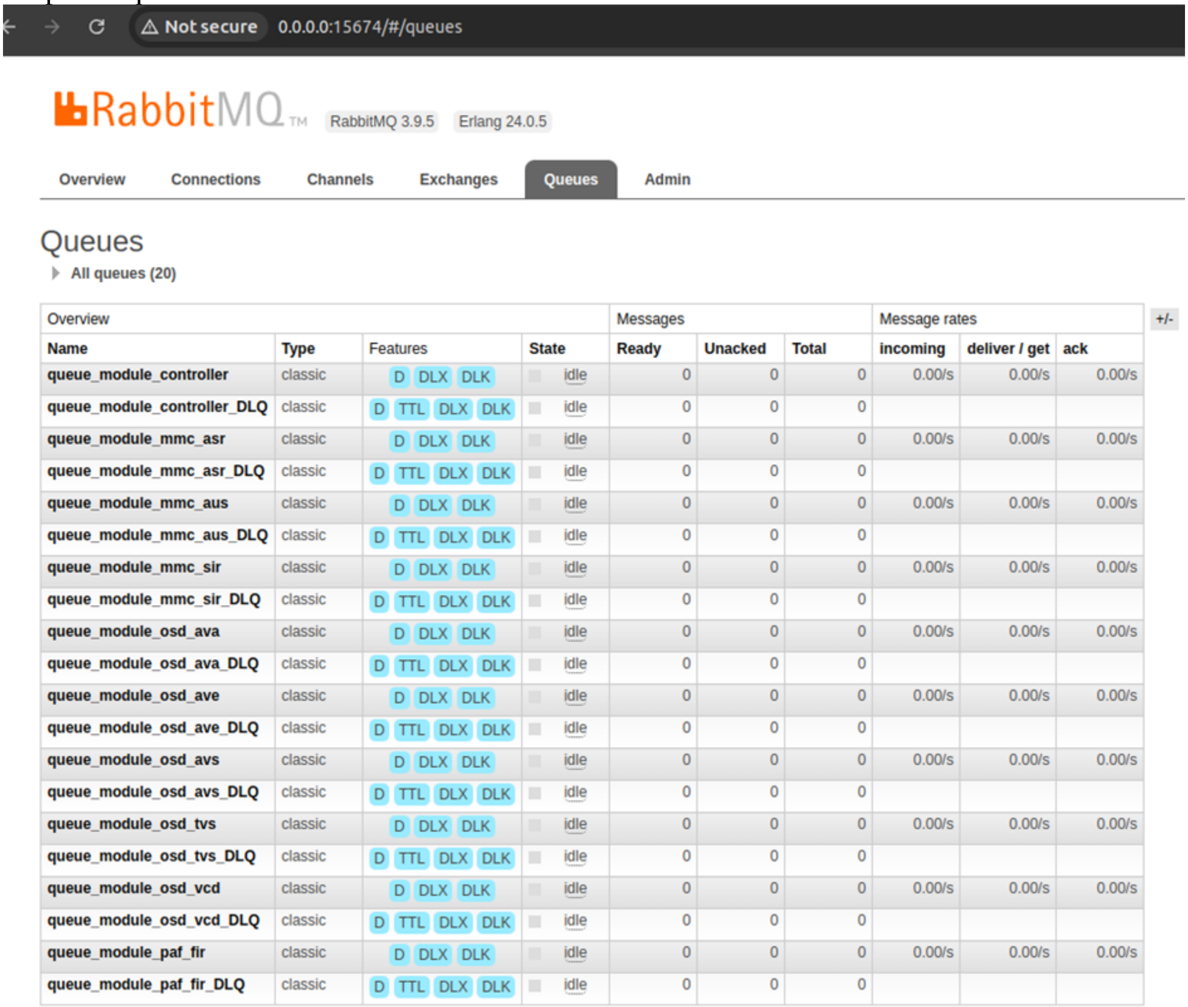
6.7.3 RabbitMQ

RabbitMQ is an open-source message broker that implements AMQP (Advanced Message Queuing Protocol). It enables asynchronous communication among the modules, allows them to be decoupled and improves the scalability and resilience of the system.

RabbitMQ provides a number of key features:

- Asynchronous messaging: RabbitMQ supports asynchronous messaging, which decouples senders from receivers and allows applications to communicate in a non-blocking manner.
- Queue management: RabbitMQ implements message queuing, which helps distribute messages among consumers.
- Reliability: RabbitMQ provides mechanisms to ensure message reliability, such as delivery acknowledgement and message persistence.
- Flexible routing: RabbitMQ supports different types of exchanges for message routing, so messages can be routed according to various criteria.
- Clustering: RabbitMQ supports clustering, which allows multiple RabbitMQ servers to be grouped into a single logical broker, thereby improving reliability and availability.
  RabbitMQ is usable in a wide range of applications, from simple data transfer among processes to more complex messaging patterns such as publish/subscribe and work queues.

Example of queues:



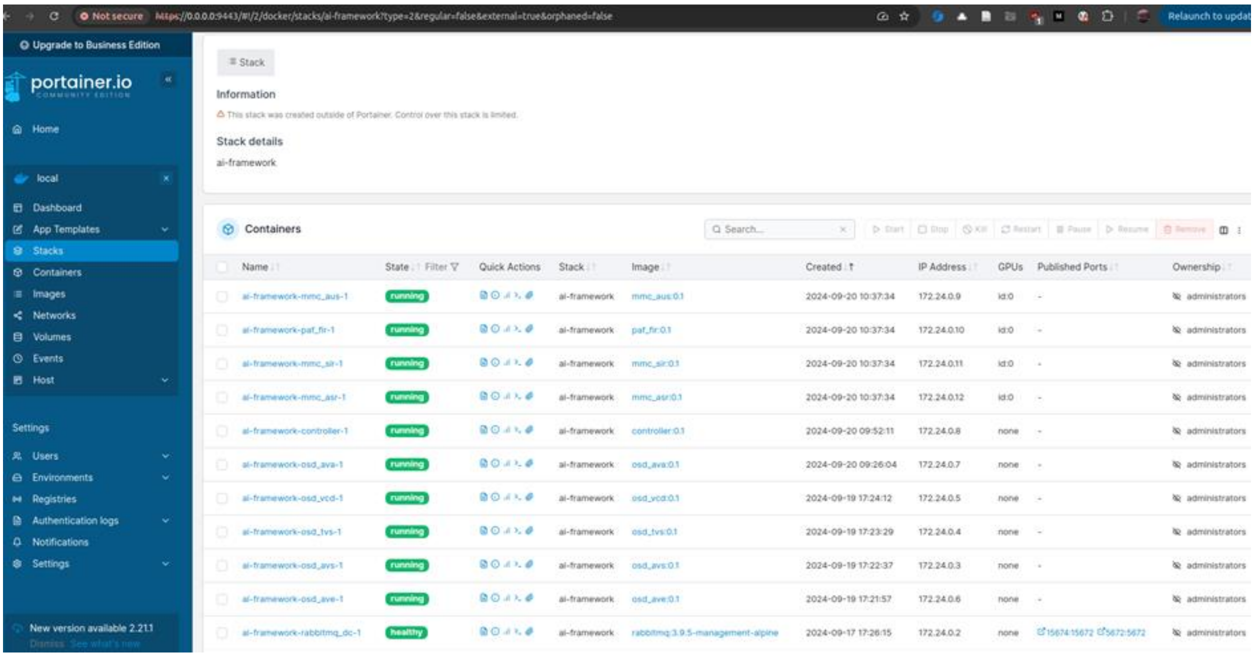
Queues

All queues (20)

Overview				Messages			Message rates		
Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack
queue_module_controller	classic	D DLX DLK	idle	0	0	0	0.00/s	0.00/s	0.00/s
queue_module_controller_DLQ	classic	D TTL DLX DLK	idle	0	0	0			
queue_module_mmc_asr	classic	D DLX DLK	idle	0	0	0	0.00/s	0.00/s	0.00/s
queue_module_mmc_asr_DLQ	classic	D TTL DLX DLK	idle	0	0	0			
queue_module_mmc_aus	classic	D DLX DLK	idle	0	0	0	0.00/s	0.00/s	0.00/s
queue_module_mmc_aus_DLQ	classic	D TTL DLX DLK	idle	0	0	0			
queue_module_mmc_sir	classic	D DLX DLK	idle	0	0	0	0.00/s	0.00/s	0.00/s
queue_module_mmc_sir_DLQ	classic	D TTL DLX DLK	idle	0	0	0			
queue_module_osd_ava	classic	D DLX DLK	idle	0	0	0	0.00/s	0.00/s	0.00/s
queue_module_osd_ava_DLQ	classic	D TTL DLX DLK	idle	0	0	0			
queue_module_osd_ave	classic	D DLX DLK	idle	0	0	0	0.00/s	0.00/s	0.00/s
queue_module_osd_ave_DLQ	classic	D TTL DLX DLK	idle	0	0	0			
queue_module_osd_avs	classic	D DLX DLK	idle	0	0	0	0.00/s	0.00/s	0.00/s
queue_module_osd_avs_DLQ	classic	D TTL DLX DLK	idle	0	0	0			
queue_module_osd_tvs	classic	D DLX DLK	idle	0	0	0	0.00/s	0.00/s	0.00/s
queue_module_osd_tvs_DLQ	classic	D TTL DLX DLK	idle	0	0	0			
queue_module_osd_vcd	classic	D DLX DLK	idle	0	0	0	0.00/s	0.00/s	0.00/s
queue_module_osd_vcd_DLQ	classic	D TTL DLX DLK	idle	0	0	0			
queue_module_paf_fir	classic	D DLX DLK	idle	0	0	0	0.00/s	0.00/s	0.00/s
queue_module_paf_fir_DLQ	classic	D TTL DLX DLK	idle	0	0	0			

6.7.4 Portainer

Portainer is an open-source user interface that streamlines the management of Docker containers. It allows you to manage Docker containers, images, stacks, networks and volumes from a web interface; no need for a command-line interface. You can use Portainer to perform tasks such as building images, adding containers or stacks, and monitoring resource usage.
Example of containers in one of our stacks:



Information

This stack was created outside of Portainer. Control over this stack is limited.

Stack details

ai-framework

Containers

Name	State	Quick Actions	Stack	Image	Created	IP Address	GPUs	Published Ports	Ownership
ai-framework-mmc_aus-1	running		ai-framework	mmc_aus:0.1	2024-09-20 10:37:34	172.24.0.9	id:0	-	administrators
ai-framework-paf_fir-1	running		ai-framework	paf_fir:0.1	2024-09-20 10:37:34	172.24.0.10	id:0	-	administrators
ai-framework-mmc_sir-1	running		ai-framework	mmc_sir:0.1	2024-09-20 10:37:34	172.24.0.11	id:0	-	administrators
ai-framework-mmc_asr-1	running		ai-framework	mmc_asr:0.1	2024-09-20 10:37:34	172.24.0.12	id:0	-	administrators
ai-framework-controller-1	running		ai-framework	controller:0.1	2024-09-20 09:52:11	172.24.0.8	none	-	administrators
ai-framework-osd_ava-1	running		ai-framework	osd_ava:0.1	2024-09-20 09:26:04	172.24.0.7	none	-	administrators
ai-framework-osd_vcd-1	running		ai-framework	osd_vcd:0.1	2024-09-19 17:24:12	172.24.0.5	none	-	administrators
ai-framework-osd_tvs-1	running		ai-framework	osd_tvs:0.1	2024-09-19 17:23:29	172.24.0.4	none	-	administrators
ai-framework-osd_avs-1	running		ai-framework	osd_avs:0.1	2024-09-19 17:22:37	172.24.0.3	none	-	administrators
ai-framework-osd_ave-1	running		ai-framework	osd_ave:0.1	2024-09-19 17:21:57	172.24.0.6	none	-	administrators
ai-framework-rabbitmq_dc-1	healthy		ai-framework	rabbitmq:3.9.5-management-alpine	2024-09-17 17:26:15	172.24.0.2	none	15674:15672 5672:5672	administrators

You can build and start all modules by running
docker compose --env-file .env -f compose.yml up
compose.yml is in repo.
.env lists the environment variables below:
TAG=0.1
LOG_LEVEL: log level. Can be CRITICAL, ERROR, WARNING, INFO, DEBUG, or NOTSET
PATH_SHARED: path of storage on disk
AI_FW_DIR: path of storage in container
HUGGINGFACE_TOKEN: HuggingFace token. Used by MMC-AUS
GIT_NAME: Git username
GIT_TOKEN: Git token
MIDDLEWARE_USER: RabbitMQ username
MIDDLEWARE_PASSWORD: RabbitMQ password
MIDDLEWARE_PORT: RabbitMQ port
MIDDLEWARE_EXTERNAL_PORT: outer RabbitMQ port
MIDDLEWARE_VIRTUALHOST: RabbitMQ virtual host. Default:
MIDDLEWARE_VIRTUALHOST=/
EXCHANGE_NAME: RabbitMQ exchange name
To start module x run
docker compose --env-file .env -f compose.yml up controller rabbitmq_dc x
Send a suitable payload to the queue of module x, as shown for the controller here:

6.7.5 Controller

The controller is the module that orchestrates the workflow. It sends a message to the right module for processing.

The controller receives the input message from queue queue_module_controller. Then, the controller sends a tweaked message to OSD-TVS. This message contains the input message. OSD-TVS sends its output message to the controller, which sends a tweaked output message to MMC-AUS.

Modules are called in this order:

1. OSD-TVS
2. MMC-AUS
3. OSD-VCD
4. MMC-SIR
5. MMC-ASR
6. PAF-FIR
7. OSD-AVA
8. OSD-AVS
9. OSD-AVE.

6.7.6 Examples/README.md

examples/README.md details how to run examples are here.

6.7.7 Acknowledgements

This version of the OSD-TMA Reference Software has been developed by the MPAI *AI Framework* Development Committee (AIF-DC).

6.8 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-TMA AIW. Conformance Testing of the individual AIMs of the OSD-TMA Composite AIM are given by the individual AIM Specification.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-TMA AIW

Receives	Audio-Video	Shall validate against Audio-Visual Object schema. Also, Audio-Visual Data shall conform with Audio-Visual Qualifier.
	Text Object	Shall validate against Text Object schema. Also, Text Data shall conform with Text Qualifier.
Produces	Audio-Visual Event Descriptors	Shall validate against Audio-Visual Event Descriptors schema.

7 AI Modules

Table 1 provides the links to the specifications and the JSON syntax of all AIMs specified by ***Technical Specification: Object and Scene Description (MPAI-OSD) V1.3***. All previously specified MPAI-OSD AI-Modules are superseded by those specified by V1.3 but may still be used by explicitly signaling their version. AI Modules in bold are Composite.

Table 1 - Specifications and JSON syntax of AIMs used by MPAI-OSD V1.3

AIMs	Name and Specification	JSON	AIMs	Name and Specification	JSON
OSD-3ED	3D Model Event Description	X	OSD-EVD	Event Description	X
OSD-3SD	3D Model Scene Description	X	OSD-LED	LiDAR Event Description	X
OSD-AED	Audio Event Description	X	OSD-LSD	LiDAR Scene Description	X
OSD-ASD	Audio Scene Description	X	OSD-OED	Offline Map Event Description	X
OSD-AVA	Audio-Visual Alignment	X	OSD-OSD	Offline Map Scene Description	X
OSD-MED	Audio-Visual Event Description	X	OSD-RED	RADAR Event Description	X
OSD-SDX	Audio-Visual Scene Demultiplexing	X	OSD-RSD	RADAR Scene Description	X
OSD-MSD	Audio-Visual Scene Description	X	OSD-SCD	Scene Description	X
OSD-SMX	Audio-Visual Scene Multiplexing	X	OSD-SED	Speech Event Description	X
OSD-B3D	Basic 3D Model Scene Description	X	OSD-SSD	Speech Scene Description	X
OSD-BAS	Basic Audio Scene Description	X	OSD-TVS	Television Splitting	X
OSD-BMD	Basic Audio-Visual Scene Description	X	OSD-UED	Ultrasound Event Description	X
OSD-BLD	Basic LiDAR Scene Description	X	OSD-USD	Ultrasound Scene Description	X
OSD-BOD	Basic Offline Map Scene Description	X	OSD-VDI	Visual Direction Identification	X
OSD-BRD	Basic RADAR Scene Description	X	OSD-VED	Visual Event Description	X
OSD-BSC	Basic Scene Description	X	OSD-VII	Visual Instance Identification	X
OSD-BSD	Basic Speech Scene Description	X	OSD-VOE	Visual Object Extraction	X
OSD-BUD	Basic Ultrasound Scene Description	X	OSD-VOI	Visual Object Identification	X
OSD-BVD	Basic Visual Scene Description	X	OSD-VOB	Visual Object Identification by Body	X
OSD-DVI	Direct Visual Identification	X	OSD-VSD	Visual Scene Description	X

Note that:

1. All previously specified MPAI-OSD Data Types that are specified by V1.2 are superseded. Use of earlier versions of Data Types is permitted if their version is explicitly signalled.

2. MPAI-OSD AIWs and AIMs utilise Data Types specified by other MPAI Technical Specifications.
3. MPAI-OSD V1.3 specifies Basic Scene Geometry and Scene Geometry; Basic Scene Descriptors and Scene Descriptors; and Basic Scene Geometry and Scene Geometry, but not their media specific versions because they cab be obtained by referencing their Qualifiers.

7.1 3D Model Event Description

7.1.1 Functions

3D Model Event Description (OSD-3DE) V1.3 produces the Descriptors of of a 3D Model Event from a sequence of Visual Scene Descriptors:

Receives	*3D Model Scene Descriptors*.	A sequence.
Produces	*3D Model Event Descriptors*	A file

7.1.2 Reference Model

The 3D Model Event Description (OSD-3DE) AIM Reference Model is depicted in Figure 1.

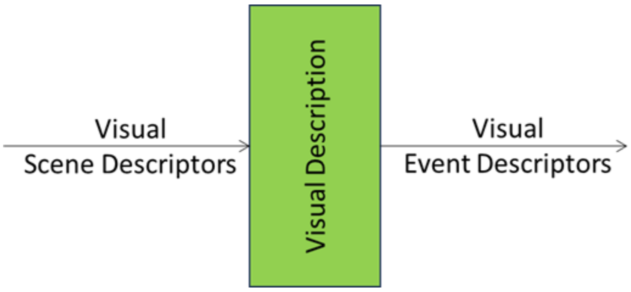


Figure 1 – The 3D Model Event Description (OSD-3DE) AIM Reference Model

7.1.3 Input/Output Data

Table 1 specifies the Input and Output Data of the 3D Model Event Description AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the 3D Model Event Description (OSD-3DE) AIM

Input	Description
3D Model Scene Descriptors	Sequence of 3D Model Scene Descriptors.
Output	**Description**
3D Model Event Descriptors	The 3D Model Event Descriptors of the 3D Model Scene.

7.1.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/3DModelEventDescription.json

7.1.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-3DE AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-3DE AIM

Receives	3D Model Scene Descriptors	Shall validate against AV Scene Descriptors schema
Produces	3D Model Event Descriptors	Shall validate against AV Event Descriptors schema

7.2 3D Model Scene Description

7.2.1 Functions

3D Model Scene Description (OSD-3SD) V1.3 produces the Descriptors of a Scene composed by 3D Model Objects and Scenes:

Receives	Space-Time	of the input Objects having the same time base.
	3D Model Objects	individual Objects.
	Scene Descriptors	Additional Descriptors of a Scene the Objects belong to.
Integrates	Space-Time and 3D Model Object	with Scene Descriptors.
Produces	3D Model Scene Descriptors	Output#1 of AIM
	Alert	Output#2 of AIM signalling potential anomalies in Object.

7.2.2 Reference Model

The Reference Architecture is depicted in Figure 1.

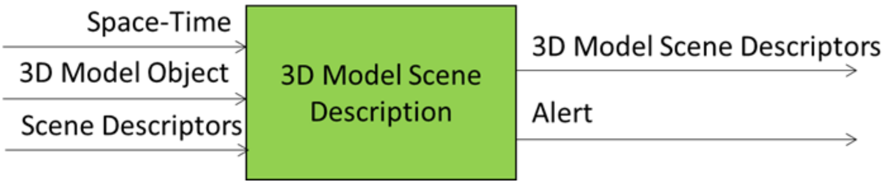


Figure 1 – The 3D Model Scene Description (OSD-3SD) AIM

7.2.3 Input/Output Data

Table 1 specifies the Input and Output Data of the 3D Model Scene Description (OSD-3SD) AIM.

Table 1 – I/O Data of the 3D Model Scene Description (OSD-3SD) AIM

Input	Description
Space-Time	Space-Time of input Objects.
3D Model Objects	Input 3D Model Objects.

Scene Descriptors	Input Scene Descriptors.
Output	**Description**
3D Model Scene Descriptors	The output 3D Model Descriptors.
Alert	Data signalling potential anomalies in Object.

7.2.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/3DModelSceneDescription.json

7.2.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-3SD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-3SD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	3D Model Objects	Shall validate against Object schema. Media-specific Data shall conform with their Qualifiers.
	Scene Descriptors	Shall validate against Scene Descriptors schema.
Produces	3D Model Scene Descriptors	Shall validate against Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.3 Audio Event Description

7.3.1 Functions

Audio Event Description (OSD-AUE) V1.3 produces the Descriptors of an Audio Event from a sequence of its Audio Scene Descriptors:

Receives	*Audio Scene Descriptors*.	A sequence of Audio Scene Descriptors.
Produces	*Audio-Visual Event Descriptors*	The output Audio Event Descriptors

7.3.2 Reference Model

The Audio Event Description (OSD-AUE) AIM Reference Model is depicted in Figure 1.

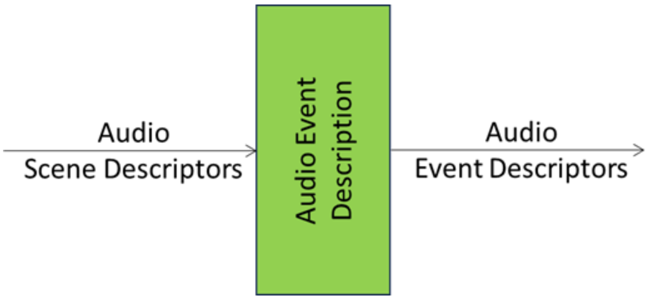


Figure 1 – The Audio Event Description (OSD-AUE) AIM Reference Model

7.3.3 Input/Output Data

Table 1 specifies the Input and Output Data of the AudioVisual Event Description AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Audio-Visual Event Description (OSD-AUE) AIM

Input	**Description**
Audio Scene Descriptors	Sequence of Audio-Visual Scene Descriptors.
Output	**Description**
Audio Event Descriptors	The Audio-Visual Event Descriptors of the Audio-Visual Scene.

7.3.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/AudioEventDescription.json

7.3.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-AUE AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-AVE AIM

Receives	Audio Scene Descriptors	Shall validate against AV Scene Descriptors schema
Produces	Audio Event Descriptors	Shall validate against AV Event Descriptors schema

7.4 Audio Scene Description

7.4.1 Functions

Audio Scene Description (OSD-ASD) V1.3 produces the Descriptors of a Scene composed by Audio Objects and Scenes:

Receives	Space-Time	of the input Objects having the same time base.
	Audio Objects	individual Audio Objects.

	Scene Descriptors	Scene to Objects belong to.
Integrates	Space-Time and 3D Model Object	with Scene Descriptors.
Produces	Audio Scene Descriptors	Output#1 of AIM
	Alert	Output#2 of AIM signalling potential anomalies in Object.

7.4.2 Reference Model

The Reference Architecture is depicted in Figure 1.

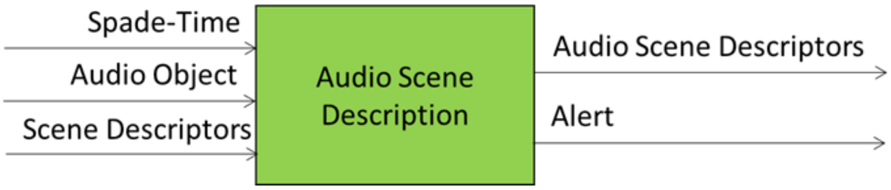


Figure 1 – The Audio Scene Description (OSD-ASD) AIM

7.4.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio Scene Description (OSD-ASD) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Audio Scene Description (OSD-ASD) AIM

Input	**Description**
Space-Time	Space-Time of input Objects.
Audio Objects	Input Objects.
Scene Descriptors	Input Scene Descriptors.
Output	**Description**
Audio Scene Descriptors	The output Audio Scene Descriptors.
Alert	Data signalling potential anomalies in Object.

7.4.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/SceneDescription.json

7.4.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-3SD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-3SD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Audio Objects	Shall validate against Audio Object schema. Media-specific Data shall conform with their Qualifiers.
	Scene Descriptors	Shall validate against Scene Descriptors schema.
Produces	Audio Scene Descriptors	Shall validate against Audio Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.5 Audio-Visual Alignment

7.5.1 Functions

Audio-Visual Alignment (OSD-AVA) V1.3 provides the Descriptors of an Audio-Visual Scene whose Audio and Visual Objects that have the same Position, have compatible Identifiers.

Receives	*Speech Scene Descriptors*	Descriptors of potentially present Speech Scene.
	Audio Scene Descriptors	Descriptors of potentially present Audio Scene.
	Visual Scene Descriptors	Descriptors of Visual Scene.
Aligns	Speech, Audio, and Visual Objects	Sharing the same Spatial Attitude
Produces	*Audio-Visual Scene Descriptors*	Where Speech Objects, Audio Objects and Visual Objects having the same Spatial Attitude have compatible Identifiers.

7.5.2 Reference Model

Figure 1 specifies the Reference Model of the Audio-Visual Alignment (OSD-AVA) AIM.

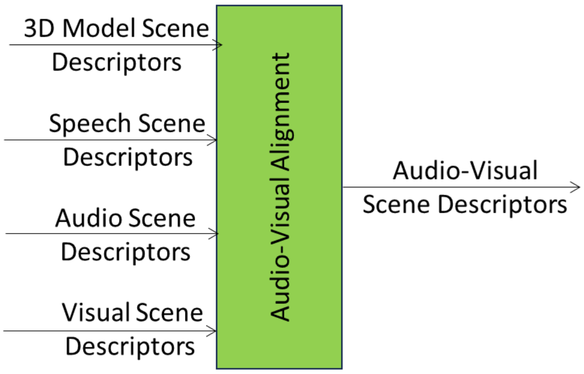


Figure 1 - Reference Model of the Audio-Visual Alignment (OSD-AVA) AIM

7.5.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio-Visual Alignment (OSD-AVA) AIM.

Table 1 – I/O Data of the Audio-Visual Alignment AIM

Input	Description
Speech Scene Descriptors	The IDs and the geometry of the Speech Objects of the Scene.
Audio Scene Descriptors	The IDs and the geometry of the Audio Objects of the Scene.
Visual Scene Descriptors	The IDs and the geometry of the Audio Objects of the Scene.
Output	**Description**
Audio-Visual Scene Descriptors	The IDs and the geometry of the Audio, Visual and Audio-Visual Objects of the Scene.

7.5.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/AudioVisualAlignment.json

7.5.5 Reference Software

7.5.5.1 Disclaimers

1. This OSD-AVA Reference Software Implementation is released with the BSD-3-Clause licence.
2. The purpose of this Reference Software is to show a working Implementation of OSD-AVA, not to provide a ready-to-use product.
3. MPAI disclaims the suitability of this Reference Software for any other purposes and does not guarantee that it is secure.
4. Use of this Reference Software may require acceptance of licences from the respective repositories. Users shall verify that they have the right to use any third-party software required by this Reference Software.

7.5.5.2 Guide to OSD-AVA code

OSD-AVA arranges the output Visual Objects and Speech Objects with corresponding Time information: scene cuts/transitions and speakers' turns. Each Object is bounded by two adjacent times from a list of unique times that are either 1) scene cuts/transitions or 2) starts and ends of speakers' turns.

Use of this Reference Software for the OSD-AVA AI Module is for developers who are familiar with Python, Docker, and RabbitMQ.

OSD-AVA computes segments as unique intervals from scene bounds and from speech segments. Moreover, OSD-AVA outputs visual objects and speech objects.

The OSD-AVA Reference Software is found at the MPAI gitlab site. It contains:

1. src: a folder with the Python code implementing the AIM
2. Dockerfile: a Docker file containing only the libraries required to build the Docker image and run the container
3. requirements.txt: dependencies installed in the Docker image.

7.5.5.3 *Acknowledgements*

This version of the MMC-ASR Reference Software has been developed by the MPAI *AI Framework* Development Committee (AIF-DC).

7.5.6 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-AVA AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-AVA AIM

Receives	Speech Scene Descriptors	Shall validate against Speech Scene Descriptors schema
	Audio Scene Descriptors	Shall validate against Audio Scene Descriptors schema
	Visual Scene Descriptors	Shall validate against Visual Scene Descriptors schema
Produces	Audio-Visual Scene Descriptors	Shall validate against AV Scene Descriptors schema

7.5.7 Performance Assessment

Performance Assessment of an OSD-AVA AIM Implementation shall be performed using a dataset of scenes containing Audio and/or Speech and Visual objects.

The Performance Assessment Report of an OSD-AVA AIM Implementation shall include:

1. The Identifier of the OSD-AVA AIM whose Performance is being Assessed.
2. The Identifier of the scene dataset used which include the identifiers of the aligned objects.
3. The data type of the scenes: analogue, digital, without or with separated objects.
4. The Performance of the OSD-AVA AIM expressed as:
    1. The number of times the OSD-AVA AIM being Assessed for Performance correctly identifies as aligned the objects that the data set declares as aligned divided by the total number of aligned objects (Truly aligned objects).
    2. The number of time the OSD-AVA AIM being Assessed for Performance incorrectly identifies as aligned the object that the dataset declares aligned in the dataset divided by the total number of aligned objects (Falsely aligned objects).
    3. The number of time the OSD-AVA AIM being Assessed for Performance incorrectly identifies as non-aligned object that are declared aligned in the dataset referenced in 2 divided by the total number of aligned objects (Missed aligned objects).

7.6 Audio-Visual Event Description

7.6.1 1 Functions

Audio-Visual Event Description (OSD-AVE) V1.3 produces the Descriptors of of an Audio-Visual Event from a sequence of Audio-Visual Scene Descriptors:

Receives *Audio-Visual Scene Descriptors*. A sequence.
Produces *Audio-Visual Event Descriptors*

7.6.2 Reference Model

The Audio-Visual Event Description (OSD-AVE) AIM Reference Model is depicted in Figure 1.

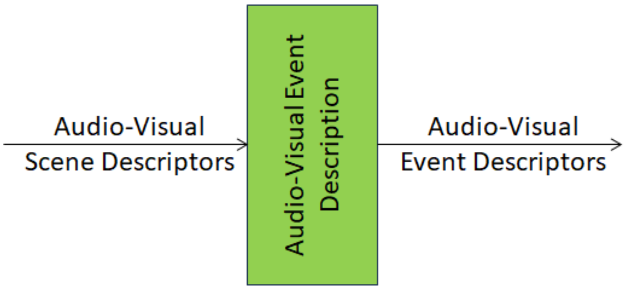

Audio-Visual Scene Descriptors
Audio-Visual Event Description
Audio-Visual Event Descriptors

Figure 1 – The Audio-Visual Event Description (OSD-AVE) AIM Reference Model

7.6.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio-Visual Event Description AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Audio-Visual Event Description (OSD-AVE) AIM

Input	Description
Audio-Visual Scene Descriptors	Sequence of Audio-Visual Scene Descriptors.
Output	**Description**
Audio-Visual Event Descriptors	The Audio-Visual Event Descriptors of the Audio-Visual Scene.

7.6.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/AudioVisualEventDescription.json

7.6.5 Reference Software

1. This OSD-AVE Reference Software Implementation is released with the BSD-3-Clause licence.
2. The purpose of this Reference Software is to show a working Implementation of OSD-AVE, not to provide a ready-to-use product.
3. MPAI disclaims the suitability of the Software for any other purposes and does not guarantee that it is secure.
4. Use of this Reference Software may require acceptance of licences from the respective repositories. Users shall verify that they have the right to use any third-party software required by this Reference Software.

7.6.5.1 Guide to the OSD-AVE code

OSD-AVE arranges the audio-visual scene descriptors from OSD-AVS into Audio-Visual Event Descriptors.

Use of this Reference Software for the OSD-AVE AI Module is for developers who are familiar with Python, Docker, and RabbitMQ.

The OSD-AVE Reference Software is found at the MPAI gitlab site. It contains:

1. src: a folder with the Python code implementing the AIM
2. Dockerfile: a Docker file containing only the libraries required to build the Docker image and run the container
3. requirements.txt: dependencies installed in the Docker image.

7.6.5.2 Acknowledgements

This version of the OSD-AVE Reference Software has been developed by the MPAI *AI Framework* Development Committee (AIF-DC).

7.6.6 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-AVE AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-AVE AIM

Receives	Audio-Visual Scene Descriptors	Shall validate against AV Scene Descriptors schema
Produces	Audio-Visual Event Descriptors	Shall validate against AV Event Descriptors schema

7.7 Audio-Visual Scene Demultiplexing

7.7.1 Functions

Audio-Visual Scene Demultiplexing (OSD-SDX) extracts the component elements from Audio-Visual Scene Descriptors:

Receives	*Audio-Visual Scene Descriptors*
Demultiplexes	*Audio-Visual Scene Descriptors*
Produces	*Speech Scene Geometry*
	Audio Scene Geometry
	Visual Scene Geometry
	Speech Objects
	Audio Objects
	Visual Objects

7.7.2 Reference Model

Figure 1 depicts the Reference Model of the Audio-Visual Scene Demultiplexing (OSD-SDX) AIM.

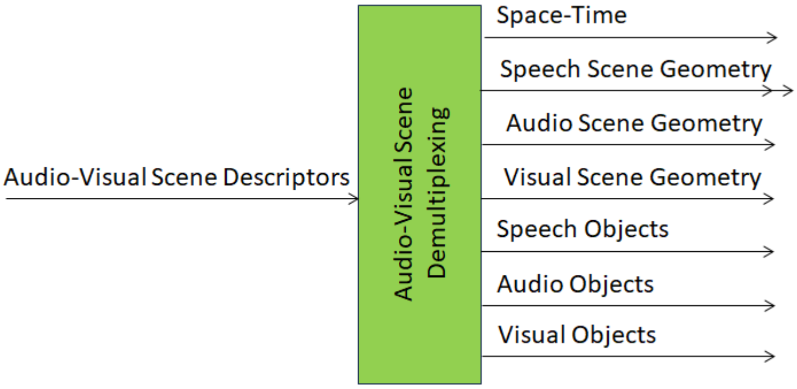


Figure 1 – Audio-Visual Scene Demultiplexing (OSD-SDX) AIM Reference Model

7.7.3 Input/Output Data

Table 1 specifies the Input and Output Data of the of the Audio-Visual Scene Demultiplexing (OSD-SDX) AIM.

Table 1 – I/O Data of the Audio-Visual Scene Demultiplexing (OSD-SDX) AIM

Input	Description
Audio-Visual Scene Descriptors	The Descriptors of the Audio-Visual Scene.
Output	**Description**
Space-Time	Space-Time information of the Audio-Visual Scene
Speech Scene Geometry	The Descriptors of the Speech Scene.
Audio Scene Geometry	The Descriptors of the Audio Scene.
Visual Scene Geometry	The Descriptors of the Visual Scene.
Audio Object	The Audio Objects in the Scene.
Speech Object	The Speech Objects in the Scene.
Visual Object	The Visual Objects in the Scene.

7.7.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/AudioVisualSceneDemultiplexing.json

7.7.5 Conformance Testing

Table 2 provides the Conformance Testing Method for the OSD-SDX AIM as a Basic AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-SDX AIM

Receives	Audio-Visual Scene Descriptors	Shall validate against AV Scene Descriptors schema
Produces	Speech Scene Geometry	Shall validate against Speech Scene Geometry schema
	Audio Scene Geometry	Shall validate against Audio Scene Geometry schema
	Visual Scene Geometry	Shall validate against Visual Scene Geometry schema
	Speech Objects	Shall validate against Speech Objects schema Speech Data shall conform with Qualifier
	Audio Objects	Shall validate against Audio Objects schema Audio Data shall conform with Qualifier
	Visual Objects	Shall validate against Visual Objects schema Visual Data shall conform with Qualifier

7.8 Audio-Visual Scene Description

7.8.1 Functions

Audio-Visual Scene Description (OSD-AVS) produces Audio-Visual Scene Descriptors from Speech, Audio, Visual and Audio-Visual Scene Descriptors:

Receives	*Space-Time*	Of output Audio-Visual Scene Descriptors,
	Speech Objects	
	Audio Objects	
	Visual Objects	
	Audio-Visual Scene Descriptors	Of Scene to be augmented.
Augments	*Audio-Visual Scene Descriptors*	
Produces	*Audio-Visual Scene Descriptors*	

7.8.2 Reference Model

Figure 1 specified the Reference Model of Audio-Visual Scene Description (OSD-AVS) aim.

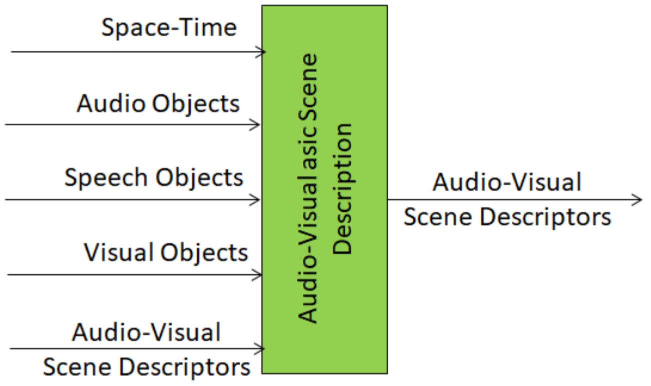

Space-Time
Audio Objects
Speech Objects
Visual Objects
Audio-Visual Scene Descriptors
Audio-Visual asic Scene Description
Audio-Visual Scene Descriptors

Figure 1 – The Audio-Visual Scene Description (OSD-AVS) AIM

7.8.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio-Visual Scene Description (OSD-AVS) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Audio-Visual Scene Description (OSD-AVS) AIM

Input	Description
Space-Time	Space-Time information of output Audio-Visual Scene Descriptors
Speech Object	Speech Object
Audio Objects	Audio Objects.
Visual Objects	Visual Objects.
Audio-Visual Scene Descriptors	The Audio-Visual Descriptors of the Scene part of the target Audio-Visual Scene.
Output	**Description**
Audio-Visual Scene Descriptors	The Audio-Visual Descriptors of the Scene.

7.8.4 SubAIMs

Figure 2 specified the Reference Model of Audio-Visual Scene Description (CAE-ASD) Composite AIM.

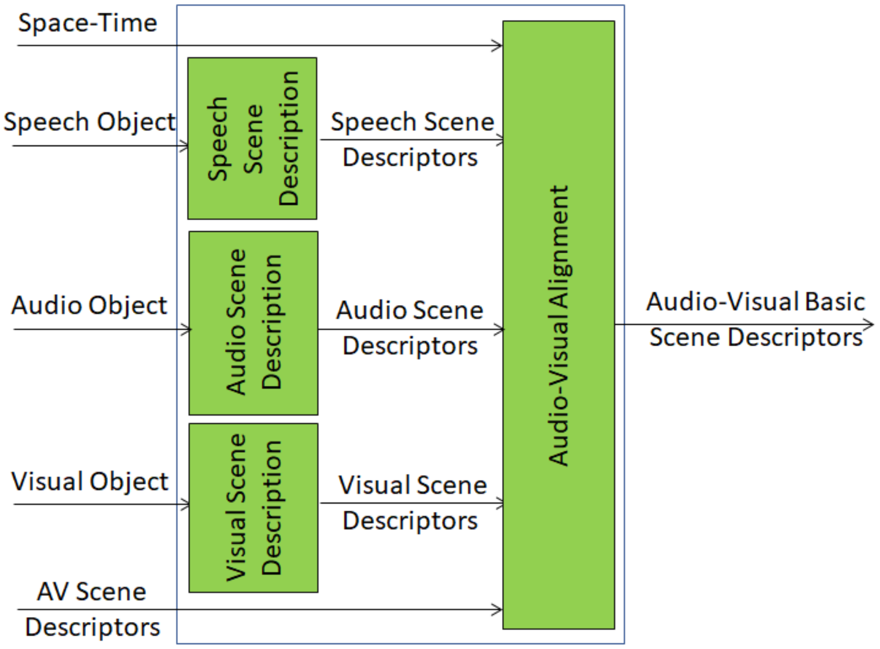


Figure 2 – The Audio-Visual Scene Description (OSD-AVS) Composite AIM

Table 2 provides the links to the specifications of the OSD-AVS AIMs.

Table 2 – AIMs of the Audio-Visual Scene Description (OSD-AVS) Composite AIM

AIMs	Names	JSON
MMC-SSD	Speech Scene Description	X
CAE-ASD	Audio Scene Description	X
OSD-VSD	Visual Scene Description	X
OSD-AVA	Audio-Visual Alignment	X

7.8.5 JSON Metadata

https://schemas.mpai.community/OSDV1.3/AIMs/AudioVisualSceneDescription.json

7.8.6 Reference Software

7.8.6.1 Disclaimers

1. This OSD-AVS Reference Software Implementation is released with the BSD-3-Clause licence.
2. The purpose of this OSD-AVS Reference Software is to show a working Implementation of OSD-AVS, not to provide a ready-to-use product.
3. MPAI disclaims the suitability of the Software for any other purposes and does not guarantee that it is secure.
4. Use of this Reference Software may require acceptance of licences from the respective repositories. Users shall verify that they have the right to use any third-party software required by this Reference Software.

7.8.6.2 *Guide to the OSD-AVS code*

OSD-AVS arranges the aligned visual and speech objects into Audio-Visual Scene Descriptors.

Use of this Reference Software for the OSD-AVS AI Module is for developers who are familiar with Python, Docker, and RabbitMQ.

The OSD-AVS Reference Software is found at the MPAI gitlab site. It contains:

1. src: a folder with the Python code implementing the AIM
2. Dockerfile: a Docker file containing only the libraries required to build the Docker image and run the container
3. requirements.txt: dependencies installed in the Docker image.

7.8.6.3 *Acknowledgements*

This OSD-AVS Reference Software has been developed by the MPAI *AI Framework* Development Committee (AIF-DC).

7.8.7 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-AVS AIM. AIM. Conformance Testing of the individual AIMs of the OSD-AVS Composite AIM are given by the individual AIM Specification.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-AVS AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Speech Objects	Shall validate against Speech Objects schema. Speech Data shall conform with Qualifier.
	Audio Objects	Shall validate against Audio Objects schema. Audio Data shall conform with Qualifier.
	Visual Objects	Shall validate against Visual Objects schema. Visual Data shall conform with Qualifier.
Produces	Audio-Visual Scene Descriptors	Shall validate against AV Scene Descriptors schema.

7.9 Audio-Visual Scene Multiplexing

7.9.1 Functions

Audio-Visual Scene Multiplexing (OSD-SMX) produces Audio-Visual Scene Descriptors by combining Speech, Audio, and Visual Objects and Scene Geometries:

Receives	*Space-Time*
	Speech Objects

	Audio Objects
	Visual Objects
	Speech Scene Geometry
	Audio Scene Geometry
	Visual Scene Geometry
Produces	*Audio-Visual Scene Descriptors*

7.9.2 Reference Model

Figure 1 specifies the Reference Model of the Audio-Visual Scene Multiplexing (OSD-SMX) AIM.

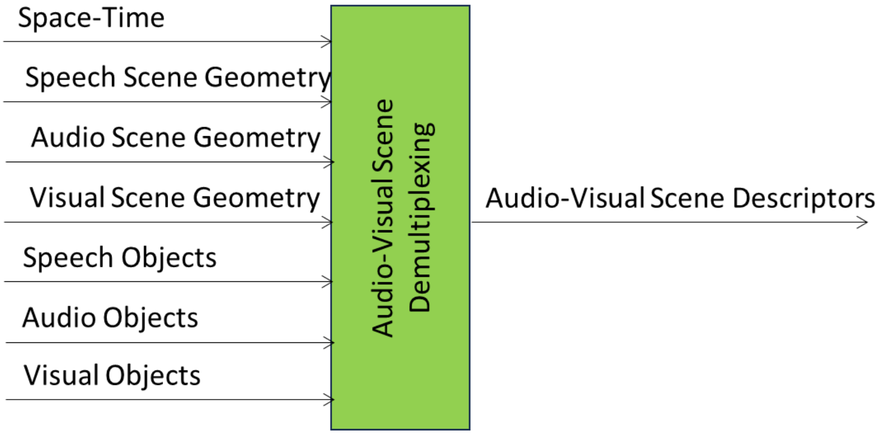


Figure 1 – The Audio-Visual Scene Multiplexing (OSD-SMX) AIM Reference Model

7.9.3 Input/Output Data

Table 13 specifies the Input and Output Data of the Audio-Visual Scene Multiplexing (OSD-SMX).

Table 13 – I/O Data of the Audio-Visual Scene Multiplexing (OSD-SMX) AIM

Input	Description
Space-Time	The Space-Time information.
Speech Object	The Speech Objects of the Scene.
Audio Object	The Audio Objects of the Scene.
Visual Object	The Visual Objects of the Scene.
Speech Scene Geometry	The Geometry of the Audio, Visual, Audio-Visual Objects of the Scene.
Audio Scene Geometry	The Geometry of the Audio, Visual, Audio-Visual Objects of the Scene.
Visual Scene Geometry	The Geometry of the Audio, Visual, Audio-Visual Objects of the Scene.
Output	**Description**

Audio-Visual Scene Descriptors	The combination of the Audio, Visual, and Audio-Visual Objects, and the Geometry of the Objects of the Scene.

7.9.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/AudioVisualSceneMultiplexing.json

7.9.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-SMX AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-SMX AIM

Receives	SpaceTime	Shall validate against Space-Time schema.
	Speech Objects	Shall validate against Speech Objects schema. Speech Data shall conform with Qualifier.
	Audio Objects	Shall validate against Audio Objects schema. Audio Data shall conform with Qualifier.
	Visual Objects	Shall validate against Visual Objects schema. Visual Data shall conform with Qualifier.
	Speech Scene Geometry	Shall validate against Speech Scene Geometry schema.
	Audio Scene Geometry	Shall validate against Audio Scene Geometry schema.
	Visual Scene Geometry	Shall validate against Visual Scene Geometry schema.
Produces	Audio-Visual Scene Descriptors	Shall validate against AV Scene Descriptors schema.

7.10 Basic 3D Model Scene Description

7.10.1 Functions

Basic 3D Model Scene Description (OSD-B3D) V1.3 produces the Descriptors of a Scene composed of 3D Model Objects and optionally an Alert:

Receives	Space-Time	of the input Objects having the same time base.
	3D Model Objects	Individual Objects.
Produces	Basic 3D Model Scene Descriptors	Scene Descriptors produced by AIM.
	Alert	Data concerning potential anomalies in input Object.

7.10.2 Reference Model

The Reference Architecture is depicted in Figure 1.

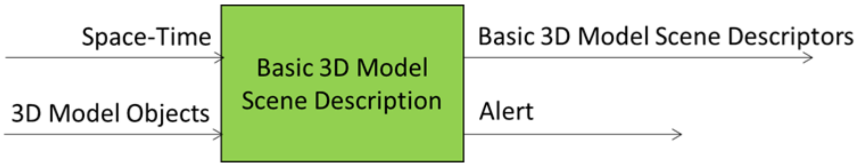

Space-Time
3D Model Objects
Basic 3D Model Scene Description
Basic 3D Model Scene Descriptors
Alert

Figure 1 – The Basic 3D Model Scene Description (OSD-B3D) AIM

7.10.3 Input/Output Data

Table 1 specifies the Input and Output Data of the 3D Model Scene Description (OSD-B3D) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Scene Description (OSD-B3D) AIM

Input	Description
Space-Time	Space-Time of input Objects.
3D Model Objects	Input Objects.
Output	**Description**
Basic 3D Model Scene Descriptors	The output Descriptors.
Alert	Data concerning potential anomalies in input Object.

7.10.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/Basic3DModelSceneDescription.json

7.10.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-B3D AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-B3D AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	3D Model Objects	Shall validate against Object schema. Media-specific Data shall conform with their Qualifiers.
Produces	Basic 3D Model Scene Descriptors	Shall validate against Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.11 Basic Audio Scene Description

7.11.1 Functions

Basic Audio Scene Description (OSD-BAD) V1.3 produces the Descriptors of a Scene composed of Audio Objects and optionally an Alert:

Receives	Space-Time	of the input Objects having the same time base.
	Audio Objects	Individual Objects.
Produces	Basic Audio Scene Descriptors	Scene Descriptors produced by AIM.
	Alert	Data concerning potential anomalies in input Object.

7.11.2 Reference Model

The Reference Architecture is depicted in Figure 1.

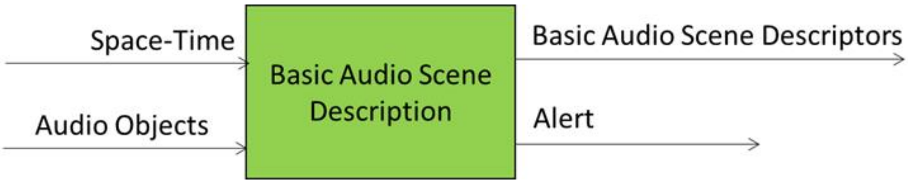


Figure 1 – The Basic Audio Scene Description (OSD-BAD) AIM

7.11.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio Scene Description (OSD-BAD) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Scene Description (OSD-BAD) AIM

Input	Description
Space-Time	Space-Time of input Objects.
Audio Objects	Input Objects.
Output	**Description**
Basic Audio Scene Descriptors	The output Descriptors.
Alert	Data concerning potential anomalies in input Object.

7.11.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/BasicAudioSceneDescription.json

7.11.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-BAD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-BAD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Audio Objects	Shall validate against Object schema. Media-specific Data shall conform with their Qualifiers.
Produces	Basic Audio Scene Descriptors	Shall validate against Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.12 Basic Audio-Visual Scene Description

7.12.1 Functions

Basic Audio-Visual Scene Description (OSD-BMS):

Receives	Space-Time.	of Audio Basic Visual Scene.
	Audio Objects.	In input.
	Speech Objects.	In input.
	Visual Objects.	In input.
	Audio-Visual Objects.	In input.
Processes	All Objects.	
Creates	Audio, Speech, Visual, and Audio-Visual Scene Descriptors	From Objects, if possible
Combines	Scene Descriptors	Of all Objects.
Produces	Audio-Visual Scene Descriptors	

7.12.2 Reference Model

Figure 1 specifies the Reference Model of the Audio-Visual Basic Scene Description (OSD-BMS) AIM.

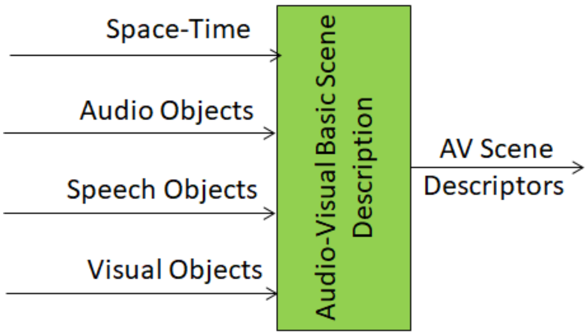


Figure 1 – The Basic Audio-Visual Scene Description (OSD-BMS) AIM

7.12.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Audio-Visual Basic Scene Description (OSD-BMS).

Table 1 – I/O Data of the Basic Audio-Visual Scene Description (OSD-BMS) AIM

Input	Description
SpaceTime	Space-Time information of Objects.
Audio Objects	Input Audio Objects.
Speech Objects	Input Speech Objects.
Visual Objects	Input Visual Objects.
Output	**Description**
Audio-Visual Scene Descriptors	The Audio-Visual Descriptors of the Scene.

7.12.4 SubAIMs

Audio-Visual Basic Scene Description (OSD-AVB) is a Composite AIM whose reference Model is depicted in Figure 2.

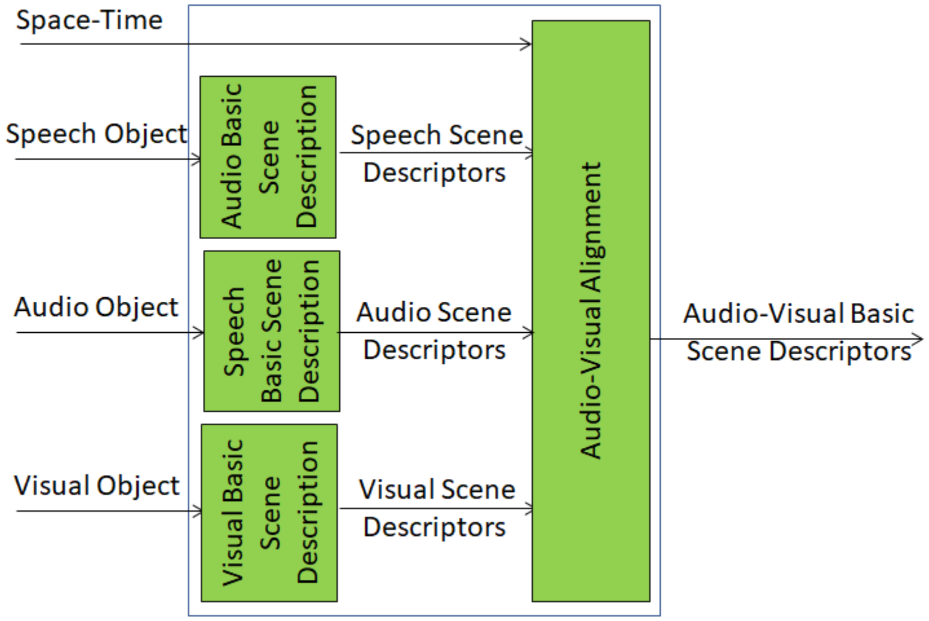


Figure 2 - Reference Model of Basic Audio-Visual Scene Description Composite (OSD-BMS) AIM

Table 2 provides the AI Modules composing the AIM.

Table 2 – AI Modules of the Audio-Visual Basic Scene Description (OSD-AVB) AIM

AIM	Acron.	AIMs	JSON

OSD-AVS		Basic Audio-Visual Basic Scene Description	X
	CAE-ABS	Basic Audio Scene Description	X
	MMC-SBS	Basic Speech Scene Description	X
	OSD-VBS	Basic Visual Scene Description	X
	OSD-AV	Audio-Visual Alignment	X

7.12.5 5 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/BasicAudioVisualSceneDescription.json

7.12.6 Conformance Testing

Table 3 provides the Conformance Testing Method for OSD-BMS AIM. Conformance Testing of the individual AIMs of the OSD-BMS Composite AIM are given by the individual AIM Specification.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 3 – Conformance Testing Method for OSD-BMS AIM

Receives	Space-Time	Shall validate against Space-Time schema
	Speech Objects	Shall validate against Speech Objects schema Speech Data shall conform with Qualifier
	Audio Objects	Shall validate against Audio Objects schema Audio Data shall conform with Qualifier
	Visual Objects	Shall validate against Visual Objects schema Visual Data shall conform with Qualifier
	Audio-Visual Objects	Shall validate against Audio-Visual Objects schema Audio-Visual Data shall conform with Qualifier
Produces	Audio-Visual Scene Descriptors	Shall validate against AV Scene Descriptors schema

7.13 Basic LiDAR Scene Description

7.13.1 Functions

Basic LiDAR Scene Description (OSD-BLD) V1.3 produces the Descriptors of a Scene composed of LiDAR Objects and optionally an Alert:

Receives	Space-Time	of the input Objects having the same time base.
	LiDAR Objects	Individual Objects.
Produces	Basic LiDAR Scene Descriptors	Scene Descriptors produced by AIM.
	Alert	Data concerning potential anomalies in input Object.

7.13.2 Reference Model

The Reference Architecture is depicted in Figure 1.

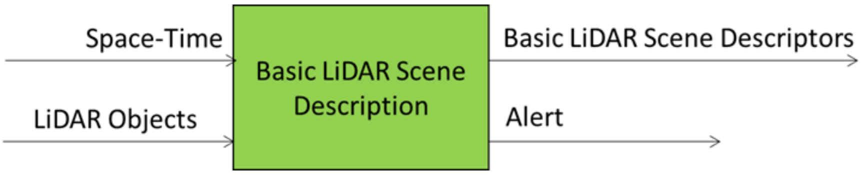


Figure 1 – The Basic LiDAR Scene Description (OSD-BLD) AIM

7.13.3 Input/Output Data

Table 1 specifies the Input and Output Data of the LiDAR Scene Description (OSD-BLD) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Basic LiDAR Scene Description (OSD-BLD) AIM

Input	**Description**
Space-Time	Space-Time of input Objects.
LiDAR Objects	Input Objects.
Output	**Description**
Basic LiDAR Scene Descriptors	The output Descriptors.
Alert	Data concerning potential anomalies in input Object.

7.13.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/BasicLiDARSceneDescription.json

7.13.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-BLD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-BLD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	LiDAR Objects	Shall validate against Object schema. Media-specific Data shall conform with their Qualifiers.
Produces	Basic LiDAR Scene Descriptors	Shall validate against Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.14 Basic Offline Map Scene Description

7.14.1 Functions

Basic Offline Map Scene Description (OSD-BOD) V1.3 produces the Descriptors of a Scene composed of LiDAR Objects and optionally an Alert:

Receives	Space-Time	of the input Objects having the same time base.
	Offline Map Objects	Individual Objects.
Produces	Basic Offline Map Scene Descriptors	Scene Descriptors produced by AIM.
	Alert	Data concerning potential anomalies in input Object.

7.14.2 Reference Model

The Reference Architecture is depicted in Figure 1.

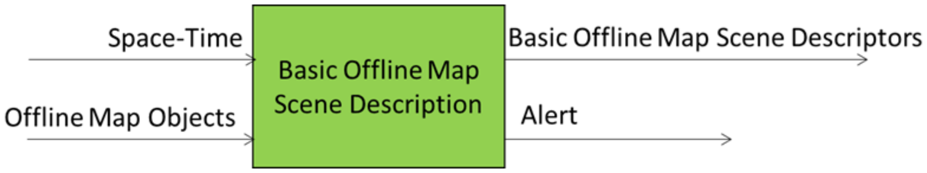

Space-Time
Offline Map Objects
Basic Offline Map Scene Description
Basic Offline Map Scene Descriptors
Alert

Figure 1 – The Basic Offline Map Scene Description (OSD-BOD) AIM

7.14.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Offline Map Scene Description (OSD-BOD) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Basic Offline Map Scene Description (OSD-BOD) AIM

Input	Description
Space-Time	Space-Time of input Objects.
Offline Map Objects	Input Objects.
Output	**Description**
Basic Offline Map Scene Descriptors	The output Descriptors.
Alert	Data concerning potential anomalies in input Object.

7.14.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/BasicOfflineMapSceneDescription.json

7.14.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-BOD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-BOD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Offline Map Objects	Shall validate against Object schema. Media-specific Data shall conform with their Qualifiers.
Produces	Basic Offline Map Scene Descriptors	Shall validate against Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.15 Basic RADAR Scene Description

7.15.1 Functions

Basic RADAR Scene Description (OSD-BRD) V1.3 produces the Descriptors of a Scene composed of RADAR Objects and optionally an Alert:

Receives	Space-Time	of the input Objects having the same time base.
	RADAR Objects	Individual Objects.
Produces	Basic RADAR Scene Descriptors	Scene Descriptors produced by AIM.
	Alert	Data concerning potential anomalies in input Object.

7.15.2 Reference Model

The Reference Architecture is depicted in Figure 1.

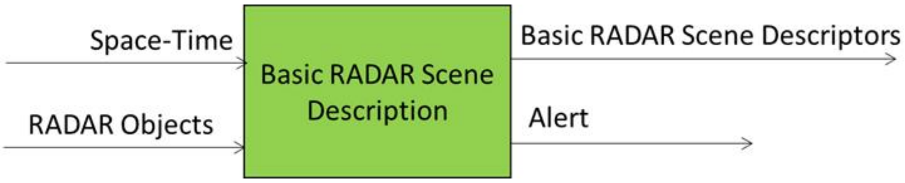


Figure 1 – The Basic RADAR Scene Description (OSD-BRD) AIM

7.15.3 Input/Output Data

Table 1 specifies the Input and Output Data of the RADAR Scene Description (OSD-BRD) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Basic RADAR Scene Description (OSD-BRD) AIM

Input	Description
Space-Time	Space-Time of input Objects.

RADAR Objects	Input Objects.
Output	**Description**
Basic RADAR Scene Descriptors	The output Descriptors.
Alert	Data concerning potential anomalies in input Object.

7.15.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/BasicRADARSceneDescription.json

7.15.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-BRD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-BRD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	RADAR Objects	Shall validate against Object schema. Media-specific Data shall conform with their Qualifiers.
Produces	Basic RADAR Scene Descriptors	Shall validate against Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.16 Basic Scene Description

7.16.1 Functions

Basic Scene Description (OSD-BSC) V1.3 produces the Descriptors of a Scene composed of Objects of different Media Types and optionally and Alert:

Receives	Space-Time	of the input Objects having the same time base.
	Objects	Individual Objects.
Produces	Scene Descriptors	Scene Descriptors produced by AIM.
	Alert	Data concerning potential anomalies in input Object.

7.16.2 Reference Model

The Reference Architecture is depicted in Figure 1.

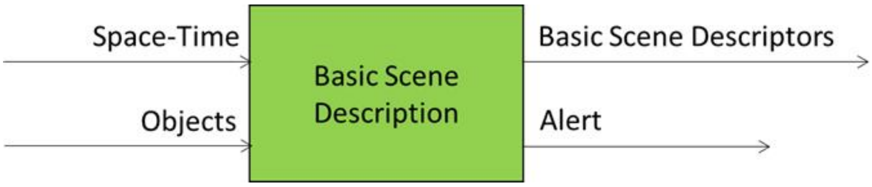


Figure 1 – The Basic Scene Description (OSD-BSC) AIM

7.16.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Basic Scene Description (OSD-BSC) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Scene Description (OSD-BSC) AIM

Input	Description
Space-Time	Space-Time of input Objects.
Objects	Input Objects.
Output	**Description**
Basic Scene Descriptors	The output Basic Scene Descriptors.
Alert	Data concerning potential anomalies in input Object.

7.16.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/BasicSceneDescription.json

7.16.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-BSC AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-BSC AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Objects	Shall validate against Object schema. Media-specific Data shall conform with their Qualifiers.
Produces	Basic Scene Descriptors	Shall validate against Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.17 Basic Speech Scene Description

7.17.1 Functions

Basic Scene Description (OSD-BSC) V1.3 produces the Descriptors of a Scene composed of Objects of different Media Types and optionally and Alert:

Receives	Space-Time	of the input Objects having the same time base.
	Objects	Individual Objects.
Produces	Scene Descriptors	Scene Descriptors produced by AIM.
	Alert	Data concerning potential anomalies in input Object.

7.17.2 Reference Model

The Reference Architecture is depicted in Figure 1.

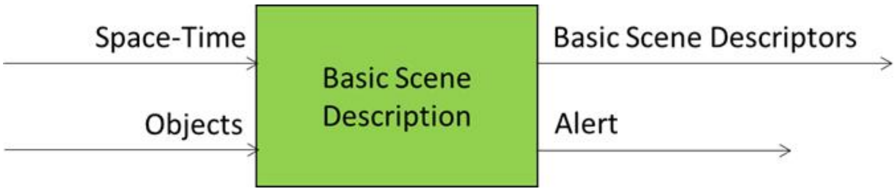


Figure 1 – The Basic Scene Description (OSD-BSC) AIM

7.17.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Basic Scene Description (OSD-BSC) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Scene Description (OSD-BSC) AIM

Input	Description
Space-Time	Space-Time of input Objects.
Objects	Input Objects.
Output	**Description**
Basic Scene Descriptors	The output Basic Scene Descriptors.
Alert	Data concerning potential anomalies in input Object.

7.17.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/BasicSceneDescription.json

7.17.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-BSC AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-BSC AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Objects	Shall validate against Object schema. Media-specific Data shall conform with their Qualifiers.
Produces	Basic Scene Descriptors	Shall validate against Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.18 Basic Ultrasound Scene Description

7.18.1 Functions

Basic Ultrasound Scene Description (OSD-BUD) V1.3 produces the Descriptors of a Scene composed of Ultrasound Objects and optionally an Alert:

Receives	Space-Time	of the input Objects having the same time base.
	Ultrasound Objects	Individual Objects.
Produces	Basic Ultrasound Scene Descriptors	Scene Descriptors produced by AIM.
	Alert	Data concerning potential anomalies in input Object.

7.18.2 Reference Model

The Reference Architecture is depicted in Figure 1.

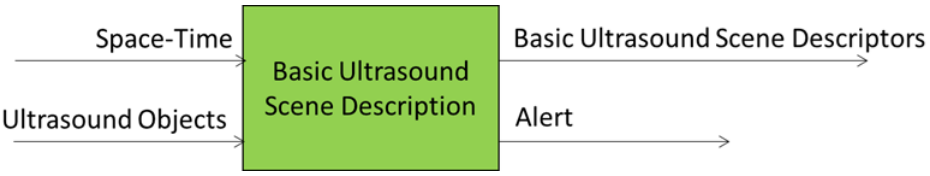

Space-Time
Ultrasound Objects
Basic Ultrasound Scene Description
Basic Ultrasound Scene Descriptors
Alert

Figure 1 – The Basic Ultrasound Scene Description (OSD-BUD) AIM

7.18.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Ultrasound Scene Description (OSD-BUD) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Basic Ultrasound Scene Description (OSD-BUD) AIM

Input	Description
Space-Time	Space-Time of input Objects.
Ultrasound Objects	Input Objects.

Output	Description
Basic Ultrasound Scene Descriptors	The output Descriptors.
Alert	Data concerning potential anomalies in input Object.

7.18.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/BasicUltrasoundSceneDescription.json

7.18.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-BUD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-BUD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Ultrasound Objects	Shall validate against Object schema. Media-specific Data shall conform with their Qualifiers.
Produces	Basic Ultrasound Scene Descriptors	Shall validate against Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.19 Basic Visual Scene Description

7.19.1 Functions

Basic Visual Scene Description (OSD-BVD) V1.3 produces the Descriptors of a Scene composed of Visual Objects and optionally an Alert:

Receives	Space-Time	of the input Objects having the same time base.
	Visual Objects	Individual Objects.
Produces	Basic Visual Scene Descriptors	Scene Descriptors produced by AIM.
	Alert	Data concerning potential anomalies in input Object.

7.19.2 Reference Model

The Reference Architecture is depicted in Figure 1.

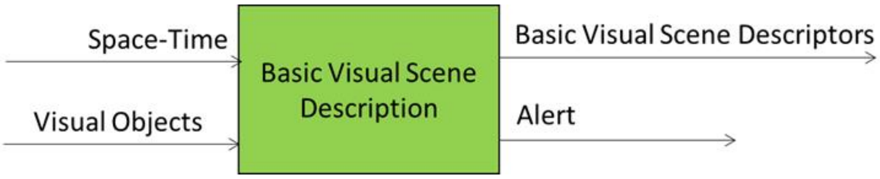

Space-Time
Visual Objects
Basic Visual Scene Description
Basic Visual Scene Descriptors
Alert

Figure 1 – The Basic Visual Scene Description (OSD-BVD) AIM

7.19.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Scene Description (OSD-BVD) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Basic Visual Scene Description (OSD-BVD) AIM

Input	Description
Space-Time	Space-Time of input Objects.
Visual Objects	Input Objects.
Output	**Description**
Basic Visual Scene Descriptors	The output Descriptors.
Alert	Data concerning potential anomalies in input Object.

7.19.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/BasicVisualSceneDescription.json

7.19.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-BVD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-BVD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Visual Objects	Shall validate against Object schema. Media-specific Data shall conform with their Qualifiers.
Produces	Basic Visual Scene Descriptors	Shall validate against Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.20 Direct Visual Identification

7.20.1 Functions

Direct Visual Identification (OSD-DVI) V1.3 produces the identifier of an input Visual Object based on a Taxonomy using the Point of View of a human body pointing to it:

Receives	*Visual Object*	An individual Visual Object or a Visual Object containing individual Visual Objects.
Produces	*Visual Visual ID*	That identifies the Visual Object or a Visual Object contained in the input Visual Object.

7.20.2 Reference Model

Figure 1 depicts the Direct Visual Identification (OSD-DVI) AIM Reference Model.

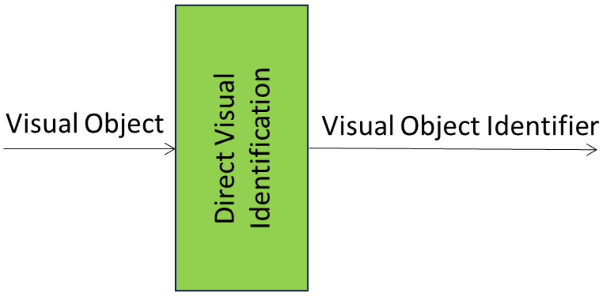


Figure - The Direct Visual Identification (OSD-DVI) AIM Reference Model

7.20.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Direct Visual Identification (OSD-DVI)AIM.

Table 1 – I/O Data of the Direct Visual Identification (OSD-DVI) AIM

Input	Description
Visual Object	The Visual Objects in the Visual Scene that are not Entities.
Output	**Description**
Visual Object Instance Identifier	The Identifier of a specific Visual Object.

7.20.4 SubAIMs

Direct Visual Identification (OSD-DVI) is a Composite AIM depicted in Figure 2.

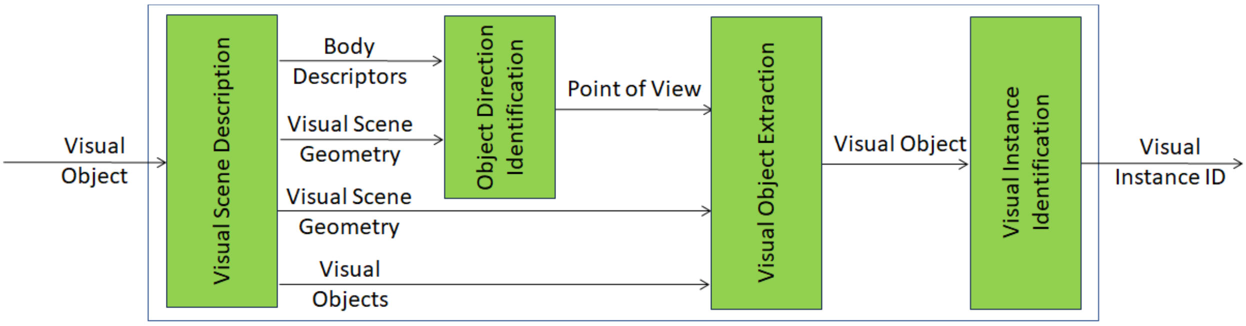


Figure 2 - The Direct Visual Identification (OSD-DVI) Composite AIM

Note that the Direct Visual Direction Identification AIM can parse either an AV Scene Geometry or its Visual Scene Geometry subset.

The AIMs composing the Direct Visual Identification (OSD-DVI) Composite AIM are:

AIM1	AIM2	AIM3	Names	JSON
OSD-DVI			Direct Visual Identification	Link
	OSD-VSD		Visual Scene Description	Link
	OSD-VOI		Visual Object Identification	Link
		OSD-VDI	Visual Direction Identification	Link
		OSD-VOE	Visual Object Extraction	Link
		OSD-VII	Visual Instance Identification	Link

7.20.5 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/DirectVisualIdentification.json

7.20.6 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-DVI AIM. AIM. Conformance Testing of the individual AIMs of the OSD-DVI Composite AIM are given by the individual AIM Specification.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-DVI AIM

Receives Visual Object	Shall validate against Visual Objects schema. Visual Data shall conform with Qualifier.
Produces Visual Object Instance Identifier	Shall validate against Instance ID schema.

7.21 Event Description

7.21.1 Functions

Event Description (OSD-EVD) V1.3 produces the Descriptors of of an Event from a sequence of Scene Descriptors:

Receives	*Scene Descriptors*.	A sequence.
Produces	*Event Descriptors*	

7.21.2 Reference Model

The Event Description (OSD-EVD) AIM Reference Model is depicted in Figure 1.

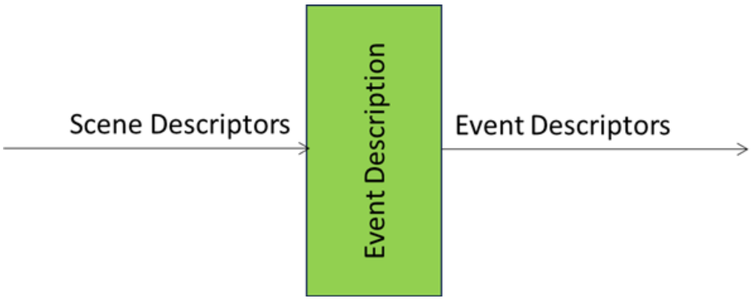


Figure 1 – The Event Description (OSD-EVD) AIM Reference Model

7.21.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Event Description AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Event Description (OSD-EVD) AIM

Input	**Description**
Scene Descriptors	Sequence of Visual Scene Descriptors.
Output	**Description**
Visual Event Descriptors	The Visual Event Descriptors of the Visual Scene.

7.21.4 SubAIMs

No SubAIMs

7.21.5 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/EventDescription.json

7.21.6 Profiles

No Profiles

7.21.7 Reference Software

Under development.

7.21.8 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-EVD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-EVD AIM

Receives	Scene Descriptors	Shall validate against AV Scene Descriptors schema
Produces	Event Descriptors	Shall validate against AV Event Descriptors schema

7.21.9 Performance Assessment

7.22 LiDAR Event Description

7.22.1 Functions

LiDAR Event Description (OSD-LIE) V1.3 produces the Descriptors of a LiDAR Event from a sequence of Visual Scene Descriptors:

Receives *LiDAR Scene Descriptors.* A sequence.
Produces *LiDAR Event Descriptors* a fille.

7.22.2 Reference Model

The LiDAR Event Description (OSD-LIE) AIM Reference Model is depicted in Figure 1.

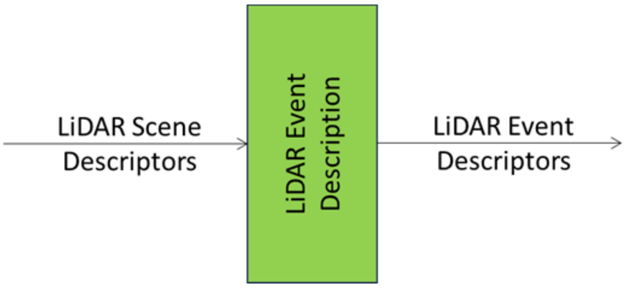


Figure 1 – The LiDAR Event Description (OSD-LIE) AIM Reference Model

7.22.3 Input/Output Data

Table 1 specifies the Input and Output Data of the LiDAR Event Description AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the LiDAR Event Description (OSD-LIE) AIM

Input	**Description**
LiDAR Scene Descriptors	Sequence of LiDAR Scene Descriptors.
Output	**Description**
LiDAR Event Descriptors	The LiDAR Event Descriptors of the LiDAR Scene.

7.22.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/LiDAREventDescription.json

7.22.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-LIE AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-LIE AIM

Receives	LiDAR Scene Descriptors	Shall validate against AV Scene Descriptors schema
Produces	LiDAR Event Descriptors	Shall validate against AV Event Descriptors schema

7.23 LiDAR Scene Description

7.23.1 Functions

LiDAR Scene Description (OSD-LSD) V1.3 produces the Descriptors of a Scene composed by LiDAR Objects and Scenes:

Receives	Space-Time	of the input Objects having the same time base.
	LiDAR Objects	Individual LiDAR Objects.
	Scene Descriptors	Scene the Objects belong to.
Integrates	Space-Time and LiDAR Object	with Scene Descriptors.
Produces	LiDAR Scene Descriptors	Output#1 of AIM
	Alert	Output#2 of AIM signalling potential anomalies in Object.

7.23.2 Reference Model

The Reference Architecture is depicted in Figure 1.

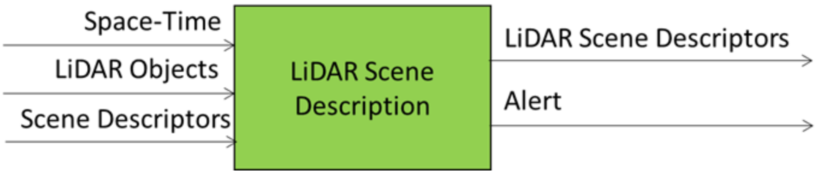


Figure 1 – The LiDAR Scene Description (OSD-LSD) AIM

7.23.3 Input/Output Data

Table 1 specifies the Input and Output Data of the LiDAR Scene Description (OSD-LSD) AIM. .

Table 1 – I/O Data of the LiDAR Scene Description (OSD-LSD) AIM

Input	Description
Space-Time	Space-Time of input Objects.
LiDAR Objects	Input LiDAR Objects.
Scene Descriptors	Input Scene Descriptors.
Output	**Description**
LiDAR Scene Descriptors	The output LiDAR Descriptors.

	Alert	Data signaling potential anomalies in Object.

7.23.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/LiDARSceneDescription.json

7.23.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-LSD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-3SD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	LiDAR Objects	Shall validate against LiDAR Object schema. Media-specific Data shall conform with their Qualifiers.
	Scene Descriptors	Shall validate against Scene Descriptors schema.
Produces	LiDAR Scene Descriptors	Shall validate against LiDAR Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.24 Offline Map Event Description

7.24.1 Functions

Offline Map Event Description (OSD-OME) V1.3 produces the Descriptors of of a Offline Map Event from a sequence of Offline Map Scene Descriptors:

Receives	*Offline Map Scene Descriptors*.	A sequence.
Produces	*Offline Map Event Descriptors*	A File.

7.24.2 Reference Model

The Offline Map Event Description (OSD-OME) AIM Reference Model is depicted in Figure 1.

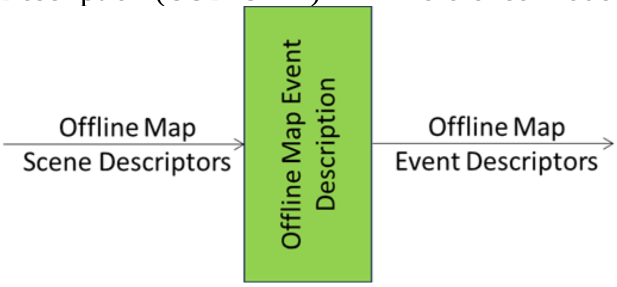

Offline Map Scene Descriptors
Offline Map Event Description
Offline Map Event Descriptors

Figure 1 – The Offline Map Event Description (OSD-OME) AIM Reference Model

7.24.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Offline Map Event Description AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Offline Map Event Description (OSD-OME) AIM

Input	Description
Offline Map Scene Descriptors	Sequence of Offline Map Scene Descriptors.
Output	**Description**
Offline Map Event Descriptors	The Offline Map Event Descriptors of the Offline Map Scene.

7.24.4 SubAIMs

No SubAIMs

7.24.5 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/OfflineMapEventDescription.json

7.24.6 Profiles

No Profiles

7.24.7 Reference Software

Under development.

7.24.8 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-OME AIM.
If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-OME AIM

Receives	Offline Map Scene Descriptors	Shall validate against AV Scene Descriptors schema
Produces	Offline Map Event Descriptors	Shall validate against AV Event Descriptors schema

7.25 Offline Map Scene Description

7.25.1 Functions

Offline Map Scene Description (OSD-OSD) V1.3 produces the Descriptors of a Scene composed by Offline Map Objects and Scenes:

Receives	Space-Time	of the input Objects having the same time base.
	Offline Map Objects	Individual Offline Map Objects.
	Scene Descriptors	Scene the Objects belong to.
Integrates	Space-Time and Offline Map Object	with Scene Descriptors.
Produces	Offline Map Scene Descriptors	Output#1 of AIM

Alert	Output#2 of AIM signalling potential anomalies in Object.

7.25.2 Reference Model

The Reference Architecture is depicted in Figure 1.

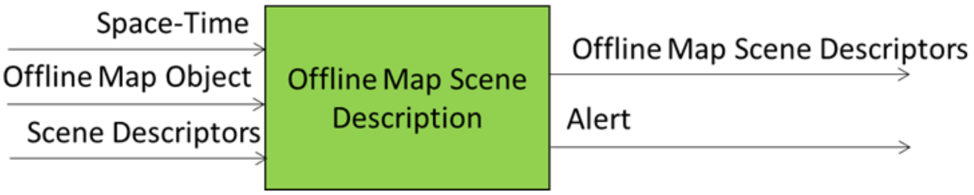


Figure 1 – The Offline Map Scene Description (OSD-OSD) AIM

7.25.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Offline Map Scene Description (OSD-OSD) AIM.

Table 1 – I/O Data of the Offline Map Scene Description (OSD-OSD) AIM

Input	Description
Space-Time	Space-Time of input Objects.
Offline Map Objects	Input Offline Map Objects.
Scene Descriptors	Input Scene Descriptors.
Output	**Description**
Offline Map Scene Descriptors	The output Offline Map Scene Descriptors.
Alert	Data signalling potential anomalies in Object.

7.25.4 5 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/OfflineMapSceneDescription.json

7.25.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-OSD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-OSD AIM

Receives Space-Time	Shall validate against Space-Time schema.

	Offline Map Objects	Shall validate against Offline Map Object schema. Media-specific Data shall conform with their Qualifiers.
	Scene Descriptors	Shall validate against Scene Descriptors schema.
Produces	Offline Map Scene Descriptors	Shall validate against Offline Map Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.26 RADAR Event Description

7.26.1 Functions

RADAR Event Description (OSD-RAE) V1.3 produces the Descriptors of a RADAR Event from a sequence of Visual Scene Descriptors:

Receives *RADAR Scene Descriptors*. A sequence.
Produces *RADAR Event Descriptors* a fille.

7.26.2 Reference Model

The RADAR Event Description (OSD-RAE) AIM Reference Model is depicted in Figure 1.

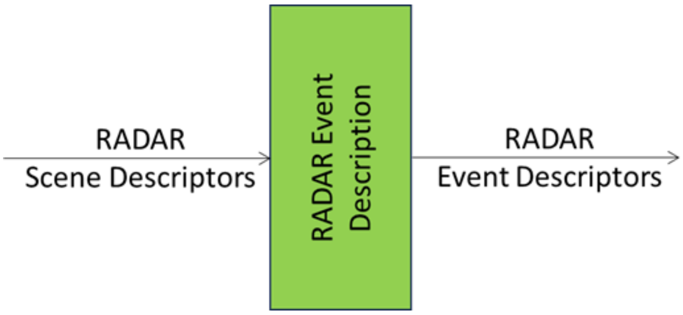


Figure 1 – The RADAR Event Description (OSD-RAE) AIM Reference Model

7.26.3 Input/Output Data

Table 1 specifies the Input and Output Data of the RADAR Event Description AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the RADAR Event Description (OSD-RAE) AIM

Input	Description
RADAR Scene Descriptors	Sequence of RADAR Scene Descriptors.
Output	**Description**
RADAR Event Descriptors	The RADAR Event Descriptors of the RADAR Scene.

7.26.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/RADAREventDescription.json

7.26.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-RAE AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-RAE AIM

Receives	RADAR Scene Descriptors	Shall validate against AV Scene Descriptors schema
Produces	RADAR Event Descriptors	Shall validate against AV Event Descriptors schema

7.27 RADAR Scene Description

7.27.1 Functions

RADAR Scene Description (OSD-RSD) V1.3 produces the Descriptors of a Scene composed by RADAR Objects and Scenes:

Receives	Space-Time	of the input Objects having the same time base.
	RADAR Objects	Individual RADAR Objects.
	Scene Descriptors	Scene the Objects belong to.
Integrates	Space-Time and RADAR Object	with Scene Descriptors.
Produces	RADAR Scene Descriptors	Output#1 of AIM
	Alert	Output#2 of AIM signalling potential anomalies in Object.

7.27.2 Reference Model

The Reference Architecture is depicted in Figure 1.

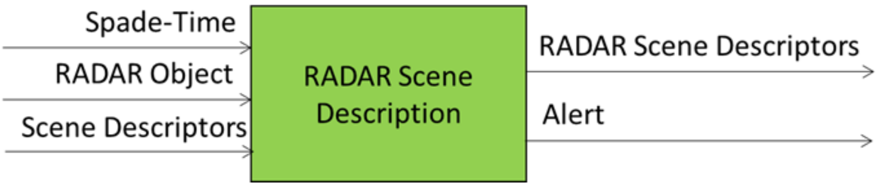


Figure 1 – The RADAR Scene Description (OSD-RSD) AIM

7.27.3 Input/Output Data

Table 1 specifies the Input and Output Data of the RADAR Scene Description (OSD-RSD) AIM. .

Table 1 – I/O Data of the RADAR Scene Description (OSD-RSD) AIM

Input	Description
Space-Time	Space-Time of input Objects.
RADAR Objects	Input RADAR Objects.
Scene Descriptors	Input Scene Descriptors.
Output	**Description**
RADAR Scene Descriptors	The output RADAR Scene Descriptors.
Alert	Data signalling potential anomalies in Object.

7.27.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/RADARSceneDescription.json

7.27.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-RSD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-RSD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	RADAR Objects	Shall validate against RADAR Object schema. Media-specific Data shall conform with their Qualifiers.
	Scene Descriptors	Shall validate against Scene Descriptors schema.
Produces	RADAR Scene Descriptors	Shall validate against RADAR Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.28 Scene Description

7.28.1 Functions

Scene Description (OSD-SCD) V1.3 produces the Descriptors of a Scene composed by Objects and Scenes:

Receives	Space-Time	of the input Objects having the same time base.
	Objects.	individual Objects.
	Scene Descriptors	Scene the input Objects belong to.
Integrates	Space-Time and Object	with Scene Descriptors.
Produces	Scene Descriptors	Output#1 of AIM

Alert	Output#2 of AIM signalling potential anomalies in Object.

7.28.2 Reference Model

The Reference Architecture is depicted in Figure 1.

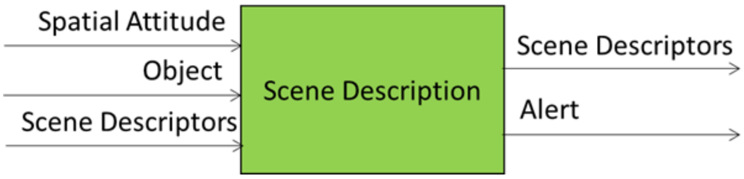


Figure 1 – The Scene Description (OSD-SCD) AIM

7.28.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Scene Description (OSD-VSD) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Scene Description (OSD-SCD) AIM

Input	**Description**
Space-Time	Space-Time of input Objects.
Objects	Input Objects.
Scene Descriptors	Input Scene Descriptors.
Output	**Description**
Scene Descriptors	The output Descriptors.
Alert	Data concerning potential anomalies in input Object.

7.28.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/SceneDescription.json

7.28.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-SCD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-SCD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Objects	Shall validate against Object schema. Media-specific Data shall conform with their Qualifiers.
	Scene Descriptors	Shall validate against Scene Descriptors schema.

Produces	Scene Descriptors	Shall validate against Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.29 Speech Event Description

7.29.1 Functions

Speech Event Description (OSD-SPE) V1.3 produces the Descriptors of a Speech Event from a sequence of Speech Scene Descriptors:

Receives	*Speech Scene Descriptors*.	A sequence.
Produces	*Speech Event Descriptors*	a fille.

7.29.2 Reference Model

The Speech Event Description (OSD-SPE) AIM Reference Model is depicted in Figure 1.

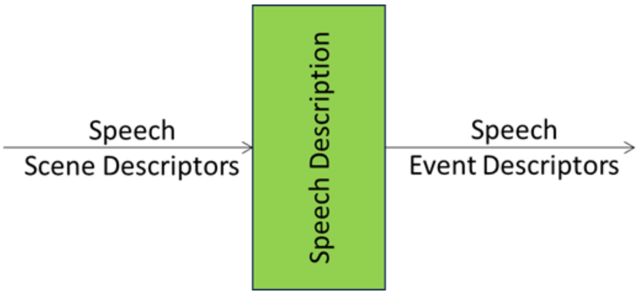


Figure 1 – The Speech Event Description (OSD-SPE) AIM Reference Model

7.29.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Speech Event Description AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Speech Event Description (OSD-SPE) AIM

Input	**Description**
Speech Scene Descriptors	Sequence of Speech Scene Descriptors.
Output	**Description**
Speech Event Descriptors	The Speech Event Descriptors of the Speech Scene.

7.29.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/SpeechEventDescriptors.json

7.29.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-SPE AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-SPE AIM

Receives	Speech Scene Descriptors	Shall validate against AV Scene Descriptors schema
Produces	Speech Event Descriptors	Shall validate against AV Event Descriptors schema

7.30 Speech Scene Description

7.30.1 Functions

Speech Scene Description (OSD-SSD) V1.3 produces the Descriptors of a Scene composed by Speech Objects and Scenes:

Receives	Space-Time	of the input Objects having the same time base.
	Speech Objects	Individual Speech Objects.
	Scene Descriptors	Scene the Objects belong to.
Integrates	Space-Time and Speech Object	with Scene Descriptors.
Produces	Speech Scene Descriptors	Output#1 of AIM
	Alert	Output#2 of AIM signalling potential anomalies in Object.

7.30.2 Reference Model

The Reference Architecture is depicted in Figure 1.

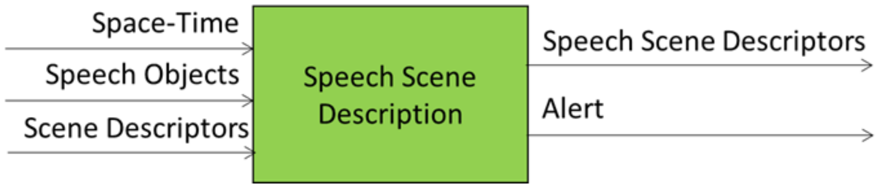


Figure 1 – The Speech Scene Description (OSD-SSD) AIM

7.30.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Speech Scene Description (OSD-SSD) AIM. .

Table 1 – I/O Data of the Speech Scene Description (OSD-SSD) AIM

Input	Description
Space-Time	Space-Time of input Objects.
Speech Objects	Input Speech Objects.

Scene Descriptors	Input Scene Descriptors.
Output	**Description**
Speech Scene Descriptors	The output Speech Scene Descriptors.
Alert	Data signalling potential anomalies in Object.

7.30.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/SpeechSceneDescription.json

7.30.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-SSD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-SSD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Speech Objects	Shall validate against Speech Object schema. Media-specific Data shall conform with their Qualifiers.
	Scene Descriptors	Shall validate against Scene Descriptors schema.
Produces	Speech Scene Descriptors	Shall validate against Speech Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.31 Television Splitting

7.31.1 Functions

Television Splitting (OSD-TVS):

Receives	Audio-Visual Object	As a file.
	Auxiliary Text	A Text Object.
Demultiplexes	Audio-Visual Object	
Produces	Text Object	
	Audio Object	
	Video Object	

7.31.2 Reference Model

The Reference Model is specified in Figure 1.

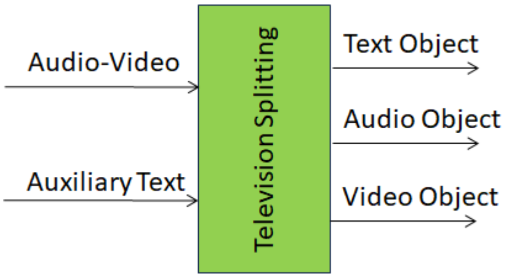


Figure 1 – The Television Splitting (OSD-TVS) AIM Reference Model

7.31.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Television Splitting (OSD-TVS) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Television Splitting (OSD-TVS) AIM

Input	**Description**
Audio-Video	Input audio-video file.
Auxiliary Text	Text provided to facilitate operation.
Output	**Description**
Text	The Text component of the input Audio-Video-Text file.
Audio	The Audio component of the input Audio-Video-Text file.
Video	The Video component of the input Audio-Video-Text file.

7.31.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/TelevisionSplitting.json

7.31.5 Reference Software

7.31.5.1 Disclaimers

1. This OSD-TVS Reference Software Implementation is released with the BSD-3-Clause licence.
2. The purpose of this Reference Software is to show a working Implementation of OSD-TVS, not to provide a ready-to-use product.
3. MPAI disclaims the suitability of the Software for any other purposes and does not guarantee that it is secure.
4. Users shall verify that they have the right to use any third-party software required by this Reference Software.

7.31.5.2 Guide to the code

Use of this AI Modules is for developers who are familiar with Python, Docker, RabbitMQ, and downloading models from HuggingFace.

OSD-TVS extracts a WAV (uncompressed audio) from an input video using FFMPEG. Therefore, any input Audio-Visual file that can be demultiplexed by FFMPEG can be used as input. The current OSD-TVS implementation does not support Auxiliary Text.

The OSD-TVS Reference Software is found at the MPAI gitlab site. It contains:

1. src: a folder with the Python code implementing the AIM
2. Dockerfile: a Docker file containing only the libraries required to build the Docker image and run the container
3. requirements.txt: dependencies installed in the Docker image

Library: https://www.ffmpeg.org/

7.31.5.3 Acknowledgements

This version of the OSD-TVS Reference Software has been developed by the MPAI *AI Framework* Development Committee (AIF-DC).

7.31.6 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-TVS AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-TVS AIM

Receives	Audio-Video file	Shall validate against Audio-Visual Object schema. Audio-Visual Data shall conform with Qualifier.
	Auxiliary Text Object	Shall validate against Text Object schema. Text Data shall conform with Qualifier.
Produces	Text File	Shall validate against Text Object schema. Text Data shall conform with Qualifier.
	Audio File	Shall validate against Audio Object schema. Audio Data shall conform with Qualifier.
	Video File	Shall validate against Visual Object schema. Visual Data shall conform with Qualifier.

7.32 Ultrasound Event Description

7.32.1 Functions

Ultrasound Scene Description (OSD-USD) V1.3 produces the Descriptors of a Scene composed by Ultrasound Objects and Scenes:

Receives	Space-Time	of the input Objects having the same time base.
	Ultrasound Objects	Individual Ultrasound Objects.

	Scene Descriptors	Scene the Objects belong to.
Integrates	Space-Time and Ultrasound Object	with Scene Descriptors.
Produces	Ultrasound Scene Descriptors	Output#1 of AIM
	Alert	Output#2 of AIM signalling potential anomalies in Object.

7.32.2 Reference Model

The Reference Architecture is depicted in Figure 1.

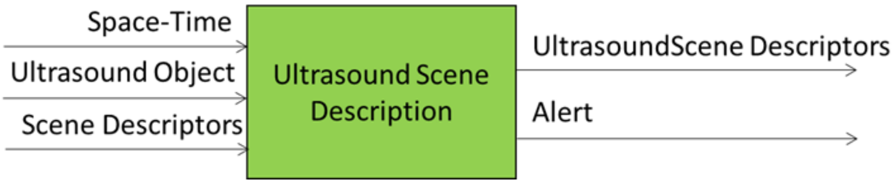


Figure 1 – The Ultrasound Scene Description (OSD-USD) AIM

7.32.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Ultrasound Scene Description (OSD-USD) AIM.

Table 1 – I/O Data of the Ultrasound Scene Description (OSD-USD) AIM

Input	Description
Space-Time	Space-Time of input Objects.
Ultrasound Objects	Input Ultrasound Objects.
Scene Descriptors	Input Scene Descriptors.
Output	**Description**
Ultrasound Scene Descriptors	The output Ultrasound Scene Descriptors.
Alert	Data signalling potential anomalies in Object.

7.32.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/UltrasoundSceneDescription.json

7.32.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-USD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-USD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Ultrasound Objects	Shall validate against Ultrasound Object schema. Media-specific Data shall conform with their Qualifiers.
	Scene Descriptors	Shall validate against Scene Descriptors schema.
Produces	Ultrasound Scene Descriptors	Shall validate against Ultrasound Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.33 Ultrasound Scene Description

7.33.1 Functions

Ultrasound Scene Description (OSD-USD) V1.3 produces the Descriptors of a Scene composed by Ultrasound Objects and Scenes:

Receives	Space-Time	of the input Objects having the same time base.
	Ultrasound Objects	Individual Ultrasound Objects.
	Scene Descriptors	Scene the Objects belong to.
Integrates	Space-Time and Ultrasound Object	with Scene Descriptors.
Produces	Ultrasound Scene Descriptors	Output#1 of AIM
	Alert	Output#2 of AIM signalling potential anomalies in Object.

7.33.2 Reference Model

The Reference Architecture is depicted in Figure 1.

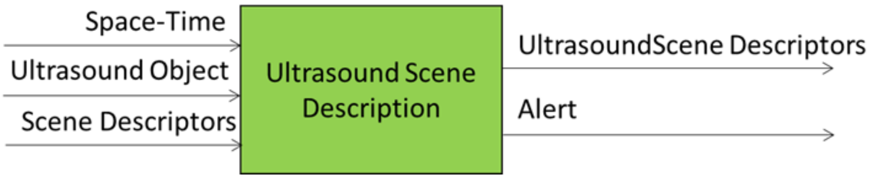


Figure 1 – The Ultrasound Scene Description (OSD-USD) AIM

7.33.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Ultrasound Scene Description (OSD-USD) AIM.

Table 1 – I/O Data of the Ultrasound Scene Description (OSD-USD) AIM

Input	Description
Space-Time	Space-Time of input Objects.
Ultrasound Objects	Input Ultrasound Objects.

Scene Descriptors	Input Scene Descriptors.
Output	**Description**
Ultrasound Scene Descriptors	The output Ultrasound Scene Descriptors.
Alert	Data signalling potential anomalies in Object.

7.33.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/UltrasoundSceneDescription.json

7.33.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-USD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-USD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Ultrasound Objects	Shall validate against Ultrasound Object schema. Media-specific Data shall conform with their Qualifiers.
	Scene Descriptors	Shall validate against Scene Descriptors schema.
Produces	Ultrasound Scene Descriptors	Shall validate against Ultrasound Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

7.34 Visual Direction Identification

7.34.1 Functions

Visual Direction Identification (OSD-VDI) identifies the Point of view of the index finger or a human body:

Receives	Visual Scene Geometry	The Geometry of the Visual Scene
	Body Descriptors	The Descriptors of a Body.
Produces	Point of View	The direction of a line traversing a point of the forefinger of the Entity.

7.34.2 Reference Model

Figure 1 depicts the Reference Model of the Visual Direction Identification (OSD-VOI) AIM.

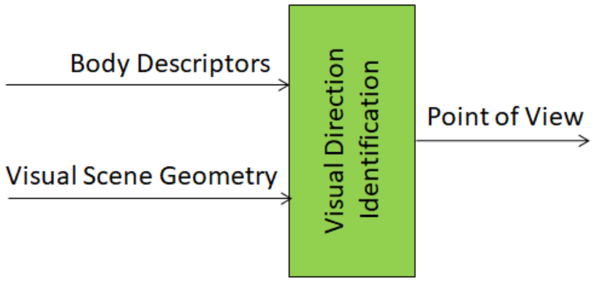


Figure 1 – The Visual Direction Identification (OSD-VOI) AIM Reference Model

7.34.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Direction Identification (OSD-VOI) AIM.

Table 1 – I/O Data of the Visual Direction Identification (OSD-VOI) AIM

Input	**Description**
Body Descriptors	The Descriptors of the Body Objects in the Visual Scene.
Visual Scene Geometry	The digital representation of the spatial arrangement of the Visual Objects of the Scene.
Output	**Description**
Point ov View	The direction of the line traversing the forefinger of the target Entity.

7.34.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/VisualDirectionIdentification.json

7.34.5 Reference Software

7.34.6 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-VDI AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-VDI AIM

Receives	Visual Scene Geometry	Shall validate against Visual Scene Geometry schema
	Body Descriptors	Shall validate against Body Descriptors XML schema
Produces	Point of View	Shall validate against Point of View schema

7.35 Visual Event Description

7.35.1 Functions

Visual Event Description (OSD-VIE) V1.3 produces the Descriptors of of a Visual Event from a sequence of Visual Scene Descriptors:

Receives *Visual Scene Descriptors*. A sequence.
Produces *Visual Event Descriptors*

7.35.2 Reference Model

The Visual Event Description (OSD-VIE) AIM Reference Model is depicted in Figure 1.

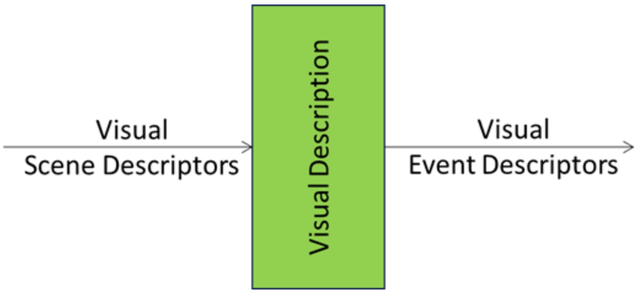


Figure 1 – The Visual Event Description (OSD-VIE) AIM Reference Model

7.35.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Event Description AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Visual Event Description (OSD-VIE) AIM

Input	Description
Visual Scene Descriptors	Sequence of Visual Scene Descriptors.
Output	**Description**
Visual Event Descriptors	The Visual Event Descriptors of the Visual Scene.

7.35.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/VisualEventDescription.json

7.35.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-VIE AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-VIE AIM

Receives	Visual Scene Descriptors	Shall validate against AV Scene Descriptors schema
Produces	Visual Event Descriptors	Shall validate against AV Event Descriptors schema

7.36 Visual Instance Identification

7.36.1 Functions

Visual Instance Identification (OSD-VII) V1.3 identifies a Visual Object based on a Taxonomy:

Receives	*Visual Object*	To be identified.
Produces	An *Instance ID*	Identifying an element of a set of Visual Objects belonging to a level in a taxonomy.

7.36.2 Reference Model

Figure 1 specifies the Reference Model of the Visual Instance Identification (OSD-VII) AIM.

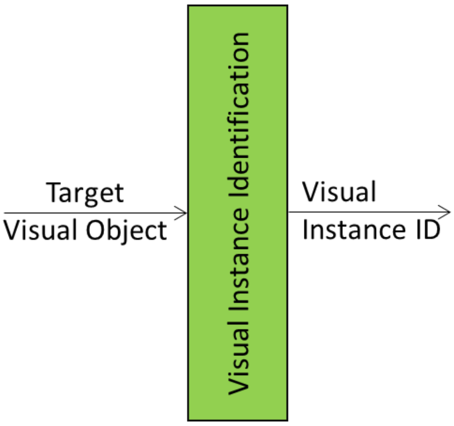


Figure 1 – The Visual Instance Identification (OSD-VII) AIM Reference Model

7.36.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Instance Identification (OSD-VII) AIM.

Table 1 – I/O Data of Visual Instance Identification (OSD-VII) AIM

Input	**Description**
Target Visual Object	The Visual Object crossed by the line traversing the forefinger of the Entity.
Output	**Description**
Visual Instance Identifier	The Identifier of the specific Visual Object belonging to a level in the taxonomy.

7.36.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/VisualInstanceIdentification.json

7.36.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-VII AIM.

Note that a schema may contain references to other schemas. In this case, validation of data for the primary schema implies that any data that refers to a secondary schema shall also validate.

Table 2 – Conformance Testing Method for OSD-VII AIM

Receives	Visual Object	Shall validate against Visual Object schema. Visual Data shall conform with Qualifier.
Produces	Instance ID	Shall validate against Instance ID schema.

Table 3 provides an example of MMC-AQM AIM conformance testing.

Table 3 – An example MMC-AQM AIM conformance testing

Input Data	**Data Type**	**Input Conformance Testing Data**
Input Image	JPEG	All input Text files to be drawn from Image files.
Output Data	**Data Type**	**Data Format**
Object Instance ID	Identifier	All Identifiers of Visual Objects shall be represented according to Instance Identifier

7.36.6 Performance Assessment

Performance Assessment of an OSD-VII AIM Implementation shall be performed using a dataset of object of a category of objects of an identified Taxonomy.

The Performance Assessment Report of an OSD-VII AIM Implementation shall include:

1. The Identifier of the OSD-VII AIM.
2. The Identifier of the object dataset.
3. The data type of object: analogue, digital, 2D, 3D etc.
4. The Performance of the OSD-VII AIM expressed as the Accuracy of the Identifiers provided by the OSD-VII AIM computed on all objects of the dataset referenced in 2.

7.37 Visual Object Extraction

7.37.1 Functions

Visual Object Extraction (OSD-VOE) V1.3 extracts from a Visual Scene Geometry the Visual Object crossed by the line of a Point of View:

Receives	*Visual Scene Geometry*	Spatial description of object arrangement.
	Visual Objects	To be extracted for identification.
	Point of View	Crossed by line.
Extracts	*Visual Object*	Crossed by line from Point of View.

7.37.2 Reference Model

Figure 1 depicts the Reference Model of the Visual Object Extraction (OSD-VOE) AIM.

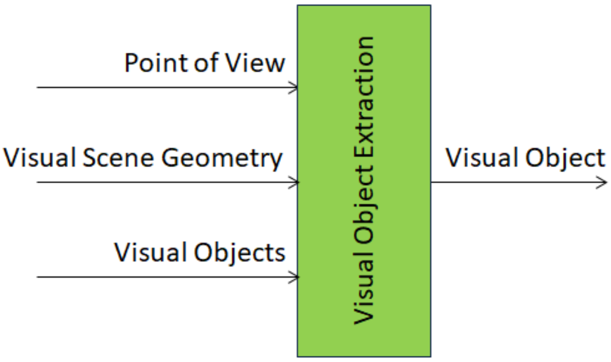


Figure 1 – The Visual Object Extraction (OSD-VOE) AIM Reference Model

7.37.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Object Extraction (OSD-VOE) AIM.

Table 1 – I/O Data of the Visual Object Extraction (OSD-VOE) AIM

Input	Description
Point of View	The direction of the line traversing the forefinger of the Entity.
Visual Scene Geometry	The digital representation of the spatial arrangement of the Visual Objects of the Scene.
Visual Objects	The Visual Objects of the Visual Scene Geometry.
Output	**Description**
Target Visual Object ID	The Visual Object crossed by the line traversing the forefinger of the Entity.

7.37.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/VisualObjectExtraction.json

7.37.5 Conformance Testing

Table 2 provides the Conformance Testing Method for MMC-VOE AIM.

Note that a schema may contain references to other schemas. In this case, validation of data for the primary schema implies that any data that refers to a secondary schema shall also validate.

Table 2 – Conformance Testing Method for MMC-VOE AIM

Receives	Visual Scene Geometry	Shall validate against Visual Scene Geometry schema.
	Visual Objects	Shall validate against Visual Object schema. Visual Data shall conform with Qualifier.
	Point of View	Shall validate against Point of View schema.
Extracts	Visual Object	Shall validate against Visual Object schema. Visual Data shall conform with Qualifier.

7.38 Visual Object Identification

7.38.1 Functions

Visual Object Identification (OSD-VOI) V1.3 identifies a Visual Object included in a Visual Scene Geometry by providing the Point of View:

Receives	*Visual Scene Geometry*	The arrangement of the objects in the Scene, a subset of Visual Scene Descriptors.
	Visual Objects	The Objects in the Scene.
	Body Descriptors	Descriptors of the Body indicating the object.
Produces	*Visual Instance ID*	Identifying a Visual Object in the Scene that belongs to some level in a taxonomy.

7.38.2 Reference Model

Figure 1 specifies the Reference Model of Visual Object Identification (OSD-VOI) AIM.

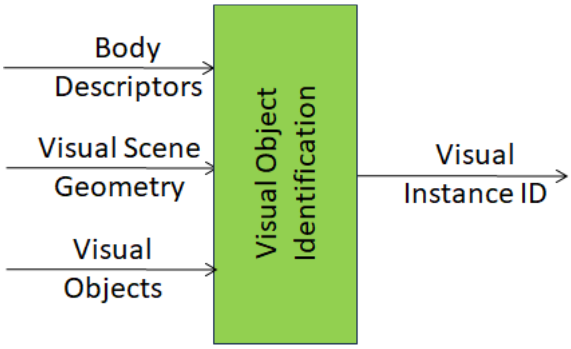


Figure - The Visual Object Identification (OSD-VOI) AIM Reference Model

7.38.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Object Identification (OSD-VOI) AIM.

Table 1 – I/O Data of the Visual Object Identification (OSD-VOI) AIM

Input	Description
Body Descriptors	The Descriptors of the Body Objects of Entities in the Visual Scene.
Visual Scene Geometry	The digital representation of the spatial arrangement of the Visual Objects of the Scene.
Visual Object	The Visual Objects in the Visual Scene that are not Entities.
Output	**Description**
Visual Instance Identifier	The Identifier of the specific Visual Object belonging to a level in the taxonomy.

7.38.4 SubAIMs

Visual Object Identification (OSD-VOI) is a Composite AIM specified by Figure 2.

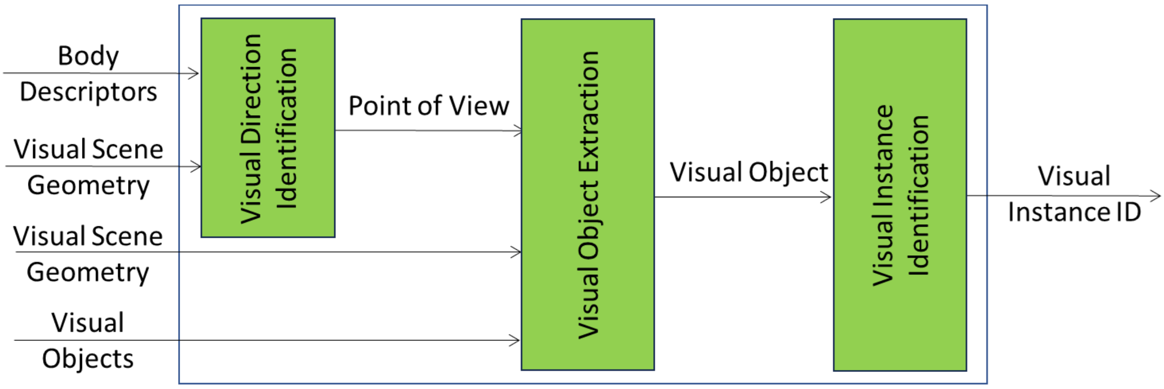


Figure 2 - The Visual Object Identification (OSD-VOI) Composite AIM

Note that the Visual Direction Identification AIM can parse either an AV Scene Geometry or its Visual Scene Geometry subset.

The AIMs composing the Visual Object Identification (OSD-VOI) Composite AIM are:

AIM	AIMs	Names	JSON
OSD-VOI		Visual Object Identification	Link
	OSD-VDI	Visual Direction Identification	Link
	OSD-VOE	Visual Object Extraction	Link
	OSD-VII	Visual Instance Identification	Link

7.38.5 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/VisualObjectIdentification.json

7.38.6 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-VOI AIM. Conformance Testing of the individual AIMs of the OSD-VOI Composite AIM are given by the individual AIM Specification.

Note that a schema may contain references to other schemas. In this case, validation of data for the primary schema implies that any data that refers to a secondary schema shall also validate.

Table 2 – Conformance Testing Method for OSD-VOI AIM

Receives	Visual Scene Geometry	Shall validate against Visual Scene Geometry schema.
	Visual Objects	Shall validate against Visual Objects schema. Visual Data shall conform with Qualifier.
	Body Descriptors	Shall validate against Body Descriptors XML schema.
Produces	Visual Instance ID	Shall validate against Instance ID schema.

7.39 Visual Object Identification by Body

7.39.1 Functions

Visual Object Identification by Body (OSD-VOB) V1.3:

Receives	Visual Object known to be a Body Object
	A generic Visual Object to be identified
Processes	Body Object to identify an index finger.
Finds	The Point of View defined by the direction of the index finger.
Produces	The ID of the Visual Object crossed by the line corresponding to the Point of View.

7.39.2 Reference Model

The Reference Model of the Visual Object Identification by Body (OSD-VOB) AIM is specified in Figure 1.

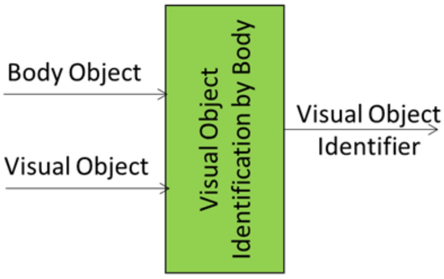


Figure 1 - The Visual Object Identification by Body (OSD-VOB) AIM Reference Model

7.39.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Object Identification by Body (OSD-VOB) AIM. Links are to the Data Type specifications.

Table 1 – I/O Data of the Visual Scene Description AIM

Input	Description
Visual Object	A Body Object

Visual Object	Generic Visual Object to be identified
Output	**Description**
Instance Identifier	Identifier of generic visual Objects

7.39.4 SubAIMs

Visual Object Identification by Body (OSD-VOB) is a Composite AIM specified by Figure 2.

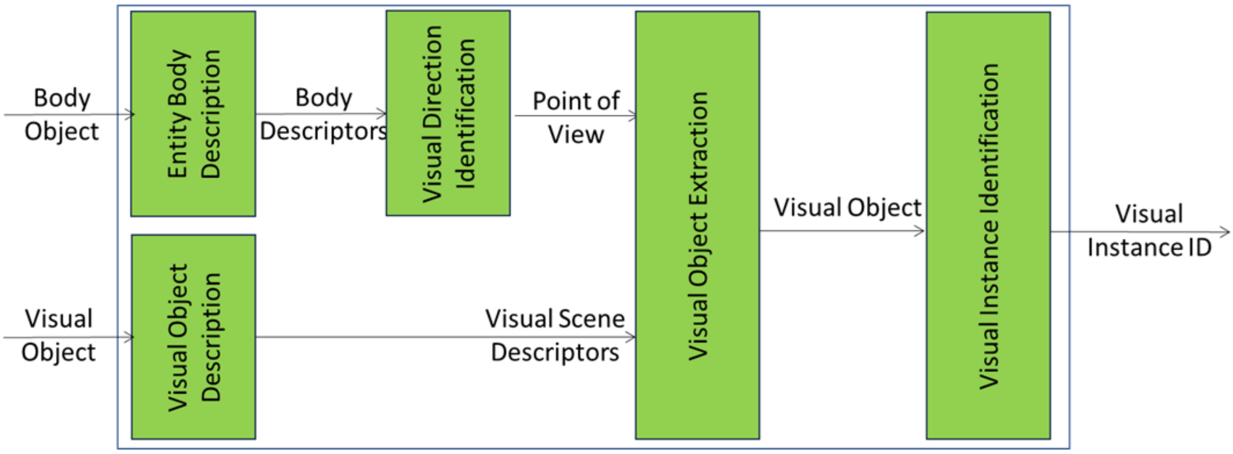


Figure 1 - The Visual Object Identification by Body (OSD-VOB) Composite AIM Reference Model

The AIMs composing the Visual Object Identification by Body (OSD-VOB) Composite AIM are:

Table2 – AIMs of the Visual Object Identification by Body (OSD-VOB) Composite AIM

AIM	**AIMs**	**Names**	**JSON**
OSD-VOB		Visual Object Identification by Body	Link
	OSD-EBD	Entity Body Description	Link
	OSD-VOD	Visual Scene Description	Link
	OSD-VDI	Visual Direction Identification	Link
	OSD-VOE	Visual Object Extraction	Link
	OSD-VII	Visual Instance Identification	Link

7.39.5 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/VisualObjectIdentificationByBody.json

7.39.6 Conformance Testing

Table 3 provides the Conformance Testing Method for OSD-VOB AIM. Conformance Testing of the individual AIMs of the OSD-VOB Composite AIM are given by the individual AIM Specification.

Note that a schema may contain references to other schemas. In this case, validation of data for the primary schema implies that any data that refers to a secondary schema shall also validate.

Table 3 – Conformance Testing Method for OSD-VOI AIM

Receives Visual Object (Body)	Shall validate against Visual Object schema. Visual Data shall conform with Qualifier.
Visual Object	Shall validate against Visual Objects schema. Visual Data shall conform with Qualifier.
Produces Visual Instance ID	Shall validate against Instance ID schema.

7.40 Visual Scene Description

7.40.1 Functions

Visual Scene Description (OSD-VSD) V1.3 produces the Descriptors of a Scene composed by Visual Objects and Scenes:

Receives	Space-Time	of the input Objects having the same time base.
	Visual Objects	Individual Visual Objects.
	Scene Descriptors	Scene the Objects belong to.
Integrates	Space-Time and Visual Object	with Scene Descriptors.
Produces	Visual Scene Descriptors	Output#1 of AIM
	Alert	Output#2 of AIM signalling potential anomalies in Object.

7.40.2 Reference Model

The Reference Architecture is depicted in Figure 1.

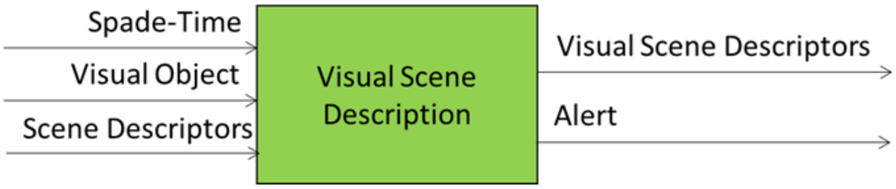


Figure 1 – The Visual Scene Description (OSD-VSD) AIM

7.40.3 Input/Output Data

Table 1 specifies the Input and Output Data of the Visual Scene Description (OSD-VSD) AIM.

Table 1 – I/O Data of the Visual Scene Description (OSD-VSD) AIM

Input	Description
Space-Time	Space-Time of input Objects.
Visual Objects	Input Visual Objects.
Scene Descriptors	Input Scene Descriptors.

Output	Description
Visual Scene Descriptors	The output Visual Scene Descriptors.
Alert	Data signalling potential anomalies in Object.

7.40.4 JSON Metadata

https://schemas.mpai.community/OSD/V1.3/AIMs/VisualSceneDescription.json

7.40.5 Conformance Testing

Table 2 provides the Conformance Testing Method for OSD-VSD AIM.

If a schema contains references to other schemas, conformance of data for the primary schema implies that any data referencing a secondary schema shall also validate against the relevant schema, if present and conform with the Qualifier, if present.

Table 2 – Conformance Testing Method for OSD-VSD AIM

Receives	Space-Time	Shall validate against Space-Time schema.
	Visual Objects	Shall validate against Visual Object schema. Media-specific Data shall conform with their Qualifiers.
	Scene Descriptors	Shall validate against Scene Descriptors schema.
Produces	Visual Scene Descriptors	Shall validate against Visual Scene Descriptors schema.
	Alert	Shall validate against Alert schema.

8 Data Types

Table 1 provides the Data Types specified by ***Technical Specification: Object Scene Description (MPAI-OSD) V1.3***.

Table 1 - Data Types specified by MPAI-OSD V1.3

OSD-3EV	3D Model Event Descriptors	X	OSD-BBX	Bounding Box	X
OSD-3DO	3D Model Object	X	OSD-CRD	Coordinates	X
OSD-3SD	3D Model Scene Descriptors	X	OSD-DSG	Device Scene Geometry	X
OSD-3SG	3D Model Scene Geometry	X	OSD-EVT	Event Descriptors	X
OSD-ANN	Annotation	X	OSD-IID	Instance Identifier	X
OSD-AEV	Audio Event Descriptors	X	OSD-LEV	LiDAR Event Descriptors	X
OSD-AUO	Audio Object	X	OSD-LIO	LiDAR Object	X

OSD-ASD	Audio Scene Descriptors	X	OSD-LSD	LiDAR Scene Descriptors	X
OSD-ASG	Audio Scene Geometry	X	OSD-LSG	LiDAR Scene Geometry	X
OSD-MEV	Audio-Visual Event Descriptors	X	OSD-LOC	Location	X
OSD-AVO	Audio-Visual Object	X	OSD-OBJ	Object	X
OSD-MSD	Audio-Visual Scene Descriptors	X	OSD-OEV	Offline Map Event Descriptors	X
OSD-MSG	Audio-Visual Scene Geometry	X	OSD-OMO	Offline Map Object	X
OSD-B3O	Basic 3D Model Object	X	OSD-OSD	Offline Map Scene Descriptors	X
OSD-B3S	Basic 3D Model Scene Descriptors	X	OSD-OSG	Offline Map Scene Geometry	X
OSD-B3G	Basic 3D Model Scene Geometry	X	OSD-OOR	Orientation	X
OSD-BAO	Basic Audio Object	X	OSD-PAT	Path	X
OSD-BAS	Basic Audio Scene Descriptors	X	OSD-PCE	Perceptible Entity	X
OSD-BAG	Basic Audio Scene Geometry	X	OSD-OPV	Point of View	X
OSD-BMO	Basic Audio-Visual Object	X	OSD-OPS	Position	X
OSD-BMS	Basic Audio-Visual Scene Descriptors	X	OSD-REV	RADAR Event Descriptors	X
OSD-BMG	Basic Audio-Visual Scene Geometry	X	OSD-RAO	RADAR Object	X
OSD-BLO	Basic LiDAR Object	X	OSD-RSD	RADAR Scene Descriptors	X
OSD-BLS	Basic LiDAR Scene Descriptors	X	OSD-RSG	RADAR Scene Geometry	X
OSD-BLG	Basic LiDAR Scene Geometry	X	OSD-RPP	Right Parallelepiped	X
OSD-BLC	Basic Location	X	OSD-SCD	Scene Descriptors	X
OSD-BOB	Basic Object	X	OSD-SCG	Scene Geometry	X
OSD-BOO	Basic Offline Map Object	X	OSD-SEL	Selector	X
OSD-BOS	Basic Offline Map Scene Descriptors	X	OSD-SPT	Space-Time	X

OSD-BOG	Basic Offline Map Scene Geometry	X	OSD-OSA	Spatial Attitude	X
OSD-BRO	Basic RADAR Object	X	OSD-SEV	Speech Event Descriptors	X
OSD-BRS	Basic RADAR Scene Descriptors	X	OSD-SPO	Speech Object	X
OSD-BRG	Basic RADAR Scene Geometry	X	OSD-SSD	Speech Scene Descriptors	X
OSD-BSD	Basic Scene Descriptors	X	OSD-SSG	Speech Scene Geometry	X
OSD-BSG	Basic Scene Geometry	X	OSD-TOB	Text Object	X
OSD-BSO	Basic Speech Object	X	OSD-TIM	Time	X
OSD-BSS	Basic Speech Scene Descriptors	X	OSD-TRJ	Trajectory	X
OSD-BSG	Basic Speech Scene Geometry	X	OSD-UEV	Ultrasound Event Descriptors	X
OSD-BTO	Basic Text Object	X	OSD-USO	Ultrasound Object	X
OSD-BUO	Basic Ultrasound Object	X	OSD-USD	Ultrasound Scene Descriptors	X
OSD-BUS	Basic Ultrasound Scene Descriptors	X	OSD-USG	Ultrasound Scene Geometry	X
OSD-BUG	Basic Ultrasound Scene Geometry	X	OSD-VEV	Visual Event Descriptors	X
OSD-BVO	Basic Visual Object	X	OSD-VIO	Visual Object	X
OSD-BVS	Basic Visual Scene Descriptors	X	OSD-VSD	Visual Scene Descriptors	X
OSD-BVG	Basic Visual Scene Geometry	X	OSD-VSG	Visual Scene Geometry	X

Note that:

1. All previously specified MPAI-OSD Data Types that are specified by V1.2 are superseded. Use of earlier versions of Data Types is permitted if their version is explicitly signalled.
2. MPAI-OSD AIWs and AIMs utilise Data Types specified by other MPAI Technical Specifications.
3. MPAI-OSD V1.3 specifies Basic Object and Object; Basic Scene Descriptors and Scene Descriptors; and Basic Scene Geometry and Scene Geometry, but not their media specific versions because they cab be obtained by referencing their Qualifiers.

8.1 3D Model Event Descriptors

8.1.1 Definition

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

8.1.2 Functional Requirements

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

8.1.3 Syntax

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

8.1.4 Semantics

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

8.1.5 Conformance Testing

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

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

8.2 3D Model Object

8.2.1 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.

8.2.2 Functional Requirements

A 3D Model Object may include:

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

Note that.

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

8.2.3 Syntax

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

8.2.4 Semantics

Label	Size	Description
Header	N1 Bytes	3D Model Object Header
– Standard-3D ModelObject	9 Bytes	The characters “OSD-3DO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
3DModelObjectID	N5 Bytes	Identifier of the 3D Model Object.
3DModelObjectSpaceTime	N6 Bytes	Space-Time of 3D Model Object.
Basic3DModelObjectCount	N7 Bytes	Set of Parent 3D Model Objects.
Basic3DModelObjects[]	N8 Bytes	Set of Basic 3D Model Objects.
- SpaceTime	N9 Bytes	Space Time of a Basic 3D Model Object in the 3D Model Object.
- Basic3DModelObject	N10 Bytes	A Basic 3D Model Object in the 3D Model Object.
3DModelObjectCount	N11 Bytes	Number of 3D Model Objects.
3DModelObjects[]	N12 Bytes	Set of 3D Model Objects.
- SpaceTime	N13 Bytes	Space Time of an 3D Model Object in the 3D Model Object.
- 3DModelObject	N14 Bytes	A 3D Model Object in the 3D Model Object
Annotations[]	N15 Bytes	Set of 3D Model Object Annotation.
– Annotation	N16 Bytes	An Annotation.
– AnnotationSpaceTime	N17 Bytes	Where Annotation is attached and when it will be active.

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

8.2.5 Conformance Testing

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

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

8.3 3D Model Scene Descriptors

8.3.1 Definition

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

8.3.2 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.

8.3.3 Syntax

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

8.3.4 Semantics

Label	Size	Description
Header	N1 Bytes	3D Model Scene Descriptors Header
- Standard-3DModelSceneDescriptors	9 Bytes	The characters "OSD-3DD-V"
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneDescriptorsID	N5 Bytes	Identifier of Scene Descriptors.
SceneDescriptorsSpaceTime	N6 Bytes	Space and Time of Scene Descriptors.

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

8.3.5 Conformance Testing

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

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

8.4 3D Model Scene Geometry

8.4.1 Definition

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

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

8.4.2 Functional Requirements

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

8.4.3 Syntax

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

8.4.4 Semantics

Label	Size	Description
Header	N1 Bytes	3D Model Scene Geometry Header
- Standard-3DModelSceneGeometry	9 Bytes	The characters "OSD-3SG-V"

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

8.4.5 Conformance Testing

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

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

8.5 Annotation

8.5.1 Definition

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

8.5.2 Functional Requirements

Elements of an Annotation are:

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

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

8.5.3 Syntax

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

8.5.4 Semantics

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

8.5.5 Conformance Testing

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

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

8.6 Audio Event Descriptors

8.6.1 Definition

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

8.6.2 Functional Requirements

Audio Event Descriptors contains Audio Scene Descriptors for a Time.

8.6.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/AudiolEventDescriptors.json

8.6.4 Semantics

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

8.6.5 Conformance Testing

A Data instance Conforms with MPAI-OSD Audio Event Descriptors V1.3 (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.

8.7 Audio Object

8.7.1 Definition

A Data Type including a collection of Basic Audio Objects.

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

8.7.2 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.

8.7.3 Syntax

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

8.7.4 Semantics

Label	Size	Description
Header	N1 Bytes	Audio Object Header
– Standard-AudioObject	9 Bytes	The characters “OSD-AUO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
AudioObjectID	N5 Bytes	Identifier of the Audio Object.
AudioObjectSpaceTime	N6 Bytes	Space-Time of Audio Object.
BasicAudioObjectCount	N7 Bytes	Set of Parent Audio Objects.
BasicAudioObjects[]	N8 Bytes	Set of Basic Audio Objects.
- SpaceTime	N9 Bytes	Space Time of a Basic Audio Object in the Audio Object.
- BasicAudioObject	N10 Bytes	A Basic Audio Object in the Audio Object.
AudioObjectCount	N11 Bytes	Number of Audio Objects.
AudioObjects[]	N12 Bytes	Set of Audio Objects.
- SpaceTime	N13 Bytes	Space Time of an Audio Object in the Audio Object.
- AudioObject	N14 Bytes	An Audio Object in the Audio Object
Annotations[]	N14 Bytes	Set of Audio Object Annotation.
– Annotation	N15 Bytes	An Annotation.
– AnnotationSpaceTime	N15 Bytes	Where Annotation is attached and when it will be active.

– Rights	N16 Bytes Actions that may be performed on the Annotation
Rights	N17 Bytes Actions that may be performed on the Object.
DescrMetadata	N17 Bytes Descriptive Metadata

8.7.5 Conformance Testing

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

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

8.8 Audio Scene Descriptors

8.8.1 Definition

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

8.8.2 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.

8.8.3 Syntax

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

8.8.4 Semantics

Label	Size	Description
Header	N1 Bytes	Audio Scene Descriptors Header
- Standard-AudioSceneDescriptors	9 Bytes	The characters "OSD-ASD-V"
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneDescriptorsID	N5 Bytes	Identifier of Scene Descriptors.

SceneDescriptorsSpaceTime	N6 Bytes	Space and Time of Scene Descriptors.
ObjectCount	N7 Bytes	Number of Objects in Scene.
Objects[]	N8 Bytes	Set of Objects.
- Object or ObjectID	N9 Bytes	Object in the Scene of its ID.
- ObjectSpaceTime	N10 Bytes	Space Time of Object.
SubSceneCount	N11 Bytes	Number of Sub-Scenes in Scene.
SubScenes[]	N12 Bytes	Set of Sub-Scenes in the Scene.
- SubScene or SubSceneID	N13 Bytes	Sub-Scene in the Scene or its ID.
- SubSceneSpaceTime	N14 Bytes	Space Time of Sub-Scene.
DescrMetadata	N15 Bytes	Descriptive Metadata

8.8.5 Conformance Testing

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

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

8.9 Audio Scene Geometry

8.9.1 Definition

An Data Type including the arrangement of the Audio Objects in a scene with their Audio Qualifiers.

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

8.9.2 Functional Requirements

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

8.9.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/AudioSceneGeometry.json

8.9.4 Semantics

Label	Size	Description
Header	N1 Bytes	Audio Scene Geometry Header

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

8.9.5 Conformance Testing

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

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

8.10 Audio-Visual Event Descriptors

8.10.1 Definition

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

8.10.2 Functional Requirements

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

8.10.3 Syntax

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

8.10.4 Semantics

Label	Size	Description

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

8.10.5 Conformance Testing

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

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

8.11 Audio-Visual Object

8.11.1 Definition

Data whose rendering has both Audio and Visual perceptibility attributes.

8.11.2 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.

8.11.3 Syntax

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

8.11.4 Semantics

Label	Size	Description

Header	N1 Bytes	Audio-Visual Object Header
- Standard-AudioVisualObject	9 Bytes	The characters “OSD-AVO-V”
- Version	N2 Byte	Major version – 1 or 2 Bytes
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 Bytes
MInstanceID	N4 Bytes	Identifier of M-Instance.
AudioVisualObjectID	N5 Bytes	Identifier of Audio-Visual Object.
AudioVisualObjectSpaceTime	N6 Bytes	Space-Time of Audio-Visual Object
AudioVisualQualifier	N7 Bytes	Qualifier of the Audio-Visual Object
3DModelObjectData	N8 Bytes	3D Model Object Data
- 3DModelObjectID and/or 3DModelObject	N9 Bytes	3D Model Object ID and/or Object
- 3DModelObjectSpaceTime	N10 Bytes	Space-Time of Speech Object
SpeechObjectData	N11 Bytes	Speech Object Data
- SpeechObjectID and/or Speech Object	N12 Bytes	Speech Object ID and/or Object
- SpeechObjectSpaceTime	N13 Bytes	Space-Time of Speech Object
AudioObjectData	N14 Bytes	Audio Object Data
- AudioObjectID and/or Audio Object	N15 Bytes	Audio Object ID and/or Object
- AudioObjectSpaceTime	N16 Bytes	Space-Time of Audio Object
VisualObjectData	N17 Bytes	Visual Object Data
- VisualObjectID and/or Visual Object	N18 Bytes	Visual Object ID and/or Object
- VisualObjectSpaceTime	N19 Bytes	Space-Time of Visual Object
Annotations[]	N20 Bytes	Set of Audio Object Annotation.
– Annotation	N21 Bytes	An Annotation.
– AnnotationSpaceTime	N22 Bytes	Where Annotation is attached and when it will be active.
– Rights	N23 Bytes	Actions that may be performed on the Annotation
Rights	N124Bytes	Actions that may be performed on the Object.
DescrMetadata	N25 Bytes	Descriptive Metadata

8.11.5 Conformance Testing

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

1. The Data validates against the Audio-Visual Object’s JSON Schema.

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

8.12 Audio-Visual Scene Descriptors

8.12.1 Definition

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

8.12.2 Functional Requirements

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

8.12.3 Syntax

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

8.12.4 Semantics

Label	**Size**	**Description**
Header	N1 Bytes	Audio-Visual Scene Descriptors Header
- Standard-AVSceneDescriptors	9 Bytes	The characters "OSD-AVS-V"
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
AVBasicSceneDescriptorsID	N5 Bytes	Identifier of the AV Object.
ObjectCount	N6 Bytes	Number of Objects in Scene
AVSceneSpaceTime	N7 Bytes	Data about Space and Time
SpeechObjects[]	N8 Bytes	Set of Speech Objects
- SpeechObject	N9 Bytes	Speech Object
- SpeechObjectSpaceTime	N10 Bytes	Space-Time of Speech Object
AudioObjects[]	N11 Bytes	Set of Audio Objects
- AudioObject	N12 Bytes	ID of Audio Object
- AudioObjectSpaceTime	N13 Bytes	Space-Time of Audio Object
VisualObjects[]	N14 Bytes	Set of Visual Objects
- VisualObjectID	N15 Bytes	ID of Visual Object
- VisualObjectSpaceTime	N16 Bytes	Space-Time of Visual Object
AudioVisualObjects[]	N17 Bytes	Set of Audio-Visual Objects
- AudioVisualObjectID	N18 Bytes	ID of Audio-Visual Object
- AudioObjectSpaceTime	N19 Bytes	Space-Time of Audio-Visual Object
SubSceneCount	N20 Bytes	Number of Sub-Scenes in Scene
SpeechSubScenes[]	N21 Bytes	Set of Speech Objects

- SpeechSubScene	N22 Bytes Speech SubScene
- SpeechSubSceneSpaceTime	N23 Bytes Space-Time of Speech SubScene
AudioSubScenes[]	N24 Bytes Set of Audio SubScenes
- AudioSubScene	N25 Bytes ID of Audio SubScene
- AudioSubSceneSpaceTime	N26 Bytes Space-Time of Audio SubScene
VisualSubScenes[]	N27 Bytes Set of Visual SubScenes
- VisualSubSceneID	N28 Bytes ID of Visual SubScene
- VisualSubSceneSpaceTime	N29 Bytes Space-Time of Visual SubScene
AudioVisualSubScenes[]	N30 Bytes Set of Audio-Visual SubScenes
- AudioVisualSubSceneID	N31 Bytes ID of Audio-Visual SubScene
- AudioSubSceneSpaceTime	N31 Bytes Space-Time of Audio-Visual SubScene
DescrMetadata	N33 Bytes Descriptive Metadata

8.12.5 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.

8.13 Audio-Visual Scene Geometry

8.13.1 Definition

An 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

8.13.2 Functional Requirements

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

8.13.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/AudioVisualSceneGeometry.json

8.13.4 Semantics

Label	Size	Description
Header	N1 Bytes	Audio-Visual Scene Geometry Header
- Standard-AudioVisualSceneGeometry	9 Bytes	The characters "OSD-AVG-V"

- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneGeometryID	N5 Bytes	Identifier of Scene Geometry.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SubSceneCount	N7 Bytes	Number of Sub-Scenes in Scene.
SceneGeometrySpaceTime	N8 Bytes	Space and Time of Scene Geometry.
SceneObjects[]	N9 Bytes	Set of Data related to Objects.
- SceneObjectQualifiers	N10 Bytes	Qualifiers of Object.
- SceneObjectSpaceTime	N11 Bytes	Space Time of Object.
SceneSubScenes[]	N9 Bytes	Set of Sub-Scenes.
- SceneSubSceneSpaceTime	N11 Bytes	Space Time of Sub-Scene.
DescrMetadata	N16 Bytes	Descriptive Metadata

8.13.5 Conformance Testing

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

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

8.14 Basic 3D Model Object

8.14.1 Definition

A Data Type including

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

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

8.14.2 Functional Requirements

A Basic Object may include:

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

8.14.3 Syntax

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

8.14.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic 3D Model Object Header
– Standard-Basic3DModelObject	9 Bytes	The characters “OSD-B3O-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
BasicObjectID	N5 Bytes	Identifier of the Basic Object.
BasicObjectSpaceTime	N6 Bytes	Space-Time info of the Basic Object.
Qualifier	N7 Bytes	Qualifier of Basic Data.
BasicObjectAnnotations[]	N8 Bytes	Annotations of Basic Object.
– Annotation	N9 Bytes	ID of Annotation
– AnnotationSpaceTime	N10 Bytes	Where/when Annotation is attached.
Rights	N11 Bytes	Rights to perform Actions of the Basic Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.14.5 Conformance Testing

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

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

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

8.15 Basic 3D Model Scene Descriptors

8.15.1 Definition

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

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

8.15.2 Functional Requirements

Basic Scene Descriptors include

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

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

8.15.3 Syntax

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

8.15.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic 3D Model Scene Descriptors Header
- Standard-Basic3DModelSceneDescriptors	9 Bytes	The characters "OSD-B3S-V"
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneDescriptorsID	N5 Bytes	Identifier of Scene Descriptors.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SceneDescriptorsSpaceTime	N7 Bytes	Space and Time of Scene Descriptors.
SceneObjects[]	N8 Bytes	Set of Objects.
- SceneObject	N9 Bytes	An Object.
- SceneObjectSpaceTime	N10 Bytes	Space Time of Object.

Rights	N11 Bytes	Rights that may be exercised on the Scene.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.15.5 Conformance Testing

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

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

8.16 Basic 3D Model Scene Geometry

8.16.1 Definition

A Data Type including the arrangement of the 3D Model Objects in a scene.

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

8.16.2 Functional Requirements

Basic Scene Geometry includes the Qualifiers and the Space-Time of the Objects.

8.16.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/Basic3DModelSceneGeometry.json

8.16.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic 3D Model Scene Geometry Header
- Standard-Basic3DModelSceneGeometry	9 Bytes	The characters "OSD-B3G-V"
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneGeometryID	N5 Bytes	Identifier of Scene Geometry.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SceneGeometrySpaceTime	N8 Bytes	Space and Time of Scene Geometry.
SceneObjects[]	N9 Bytes	Set of Data related to Objects.

- SceneObjectQualifiers	N10 Bytes Qualifiers of Object.
- SceneObjectSpaceTime	N11 Bytes Space Time of Object.
DescrMetadata	N12 Bytes Descriptive Metadata

8.16.5 Conformance Testing

A Data instance Conforms with Basic 3D Model Scene Geometry (OSD-B3G) V1.3 if:

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

8.17 Basic Audio Object

8.17.1 Definition

A Data Type including

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

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

8.17.2 Functional Requirements

A Basic Object may include:

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

8.17.3 Syntax

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

8.17.4 Semantics

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

8.17.5 Conformance Testing

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

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

8.18 Basic Audio Scene Descriptors

8.18.1 Definition

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

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

8.18.2 Functional Requirements

Basic Scene Descriptors include

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

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

8.18.3 Syntax

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

8.18.4 Semantics

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

8.18.5 Conformance Testing

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

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

8.19 Basic Audio Scene Geometry

8.19.1 Definition

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

In the following, Data, Objects, Qualifiers, and Scenes should be read as Audio Data, Audio Objects, Audio Qualifiers, and Audio Scenes

8.19.2 Functional Requirements

Basic Scene Geometry includes the Qualifiers and the Space-Time of the Objects.

8.19.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicAudioSceneGeometry.json

8.19.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Audio Scene Geometry Header
- Standard-BasicAudioSceneGeometry	9 Bytes	The characters “OSD-BAG-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneGeometryID	N5 Bytes	Identifier of Scene Geometry.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SceneGeometrySpaceTime	N8 Bytes	Space and Time of Scene Geometry.
SceneObjects[]	N9 Bytes	Set of Data related to Objects.
- SceneObjectQualifiers	N10 Bytes	Qualifiers of Object.
- SceneObjectSpaceTime	N11 Bytes	Space Time of Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.19.5 Conformance Testing

A Data instance Conforms with Basic Audio Scene Geometry (OSD-BAG) V1.3 if:

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

8.20 Basic Audio-Visual Object

8.20.1 Definition

A Data Type including

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

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

8.20.2 Functional Requirements

A Basic Object may include:

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

8.20.3 Syntax

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

8.20.4 Semantics

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

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

8.20.5 Conformance Testing

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

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

8.21 Basic Audio-Visual Scene Descriptors

8.21.1 Definition

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

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

8.21.2 Functional Requirements

Basic Scene Descriptors include

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

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

8.21.3 Syntax

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

8.21.4 Semantics

Label	Size	Description

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

8.21.5 Conformance Testing

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

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

8.22 Basic Audio-Visual Scene Geometry

8.22.1 Definition

A Data Type including the arrangement of the 3D Model Objects in a scene.

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

8.22.2 Functional Requirements

Basic Scene Geometry includes the Qualifiers and the Space-Time of the Objects.

8.22.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicAudioVisualSceneGeometry.json

8.22.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Audio-Visual Scene Geometry Header
- Standard-BasicAudioVisualSceneGeometry	9 Bytes	The characters “OSD-BMG-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneGeometryID	N5 Bytes	Identifier of Scene Geometry.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SceneGeometrySpaceTime	N8 Bytes	Space and Time of Scene Geometry.
SceneObjects[]	N9 Bytes	Set of Data related to Objects.
- SceneObjectQualifiers	N10 Bytes	Qualifiers of Object.
- SceneObjectSpaceTime	N11 Bytes	Space Time of Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.22.5 Conformance Testing

A Data instance Conforms with Basic Audio-Visual Scene Geometry (OSD-BMG) V1.3 if:

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

8.23 Basic LiDAR Object

8.23.1 Definition

A Data Type including

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

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

8.23.2 Functional Requirements

A Basic Object may include:

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

8.23.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicLiDARObject.json

8.23.4 Semantics

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

8.23.5 Conformance Testing

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

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

8.24 Basic LiDAR Scene Descriptors

8.24.1 Definition

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

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

8.24.2 Functional Requirements

Basic Scene Descriptors include

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

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

8.24.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicLiDARSceneDescriptors.json

8.24.4 Semantics

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

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

8.24.5 Conformance Testing

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

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

8.25 Basic LiDAR Scene Geometry

8.25.1 Definition

A Data Type including the arrangement of the LiDAR Objects in a scene.

In the following, Data, Objects, Qualifiers, and Scenes should be read as LiDAR Data, LiDAR Objects, LiDAR Qualifiers, and LiDAR Scenes.

8.25.2 Functional Requirements

Basic Scene Geometry includes the Qualifiers and the Space-Time of the Objects.

8.25.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicLiDARSceneGeometry.json

8.25.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic LiDAR Scene Geometry Header
- Standard-BasicLiDARSceneGeometry	9 Bytes	The characters "OSD-BLG-V"
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneGeometryID	N5 Bytes	Identifier of Scene Geometry.
ObjectCount	N6 Bytes	Number of Objects in Scene.

SceneGeometrySpaceTime	N8 Bytes	Space and Time of Scene Geometry.
SceneObjects[]	N9 Bytes	Set of Data related to Objects.
- SceneObjectQualifiers	N10 Bytes	Qualifiers of Object.
- SceneObjectSpaceTime	N11 Bytes	Space Time of Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.25.5 Conformance Testing

A Data instance Conforms with Basic Audio Scene Geometry (OSD-BLG) V1.3 if:

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

8.26 Basic Location

8.26.1 Definition

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

8.26.2 Functional Requirements

A Basic Location

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

8.26.3 Syntax

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

8.26.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Location Header
– Standard-BasicLocation	9 Bytes	The characters "OSD-BLC-V"
– Version	N2 Bytes	Major version
– Dot-separator	1 Byte	The character "."
– Subversion	N3 Bytes	Minor version
MInstanceID	N4 Bytes	Identifier of M-Instance.
BasicLocationID	N6 Bytes	Identifier of Basic Location.

BasicLocation[]	N7 Bytes	Set of Right Parallelepipeds defining Basic-M-Location.
– LocationQualifier	N8 Bytes	Spatial unit.
– SpaceTime	N9 Bytes	Space-Time of spatial unit.
DescrMetadata	N10 Bytes	Descriptive Metadata.

8.26.5 Conformance Testing

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

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

8.27 Basic Object

8.27.1 Definition

A Data Type including:

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

8.27.2 Functional Requirements

A Basic Object may include:

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

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

Object name	Acronym	JSON
Basic 3DModel Object	OSD-B3O	Link

Basic Audio Object	OSD-BAO	Link
Basic Audio-Visual Object	OSD-BMO	Link
Basic LiDAR Object	OSD-BLO	Link
Basic Offline Map Object	OSD-BOO	Link
BasicRADAR Object	OSD-BRO	Link
Basic Speech Object	OSD-BSO	Link
Basic Text Object	OSD-BTO	Link
Basic Ultrasound Object	OSD-BUO	Link
Basic Visual Object	OSD-BVO	Link

8.27.3 Syntax

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

8.27.4 Semantics

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

8.27.5 Conformance Testing

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

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

8.28 Basic Offline Map Object

8.28.1 Definition

A Data Type including

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

In the following, Data, Qualifier, and Object should be read as Basic Offline Map Data, Basic Offline Map Qualifiers, and Basic Offline Map Object, respectively.

8.28.2 Functional Requirements

A Basic Object may include:

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

8.28.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicOfflineMapObject.json

8.28.4 Semantics

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

8.28.5 Conformance Testing

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

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

8.29 Basic Offline Map Scene Descriptors

8.29.1 Definition

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

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

8.29.2 Functional Requirements

Basic Scene Descriptors include

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

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

8.29.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicOfflineMapSceneDescriptors.json

8.29.4 Semantics

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

8.29.5 Conformance Testing

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

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

8.30 Basic Offline Map Scene Geometry

8.30.1 Definition

A Data Type including the arrangement of the Offline Map Objects in a scene.

In the following, Data, Objects, Qualifiers, and Scenes should be read as Offline Map Data, Offline Map Objects, Offline Map Qualifiers, and Offline Map Scenes.

8.30.2 Functional Requirements

Basic Scene Geometry includes the Qualifiers and the Space-Time of the Objects.

8.30.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicOfflineMapSceneGeometry.json

8.30.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Offline Scene Geometry Header
- Standard-BasicOfflineMapSceneGeometry	9 Bytes	The characters "OSD-BLG-V"
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneGeometryID	N5 Bytes	Identifier of Scene Geometry.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SceneGeometrySpaceTime	N8 Bytes	Space and Time of Scene Geometry.
SceneObjects[]	N9 Bytes	Set of Data related to Objects.
- SceneObjectQualifiers	N10 Bytes	Qualifiers of Object.
- SceneObjectSpaceTime	N11 Bytes	Space Time of Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.30.5 Conformance Testing

A Data instance Conforms with Basic Offline Scene Geometry (OSD-BOG) V1.3 if:

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

8.31 Basic RADAR Object

8.31.1 Definition

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

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

8.31.2 Functional Requirements

A Basic Object may include:

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

8.31.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicRADARObject.json

8.31.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic RADAR Object Header
– Standard-BasicRADARObject	9 Bytes	The characters “OSD-BRO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
BasicObjectID	N5 Bytes	Identifier of the Basic Object.
BasicObjectSpaceTime	N6 Bytes	Space-Time info of the Basic Object.
Qualifier	N7 Bytes	Qualifier of Basic Data.

BasicObjectAnnotations[]	N8 Bytes	Annotations of Basic Object.
– Annotation	N9 Bytes	ID of Annotation
– AnnotationSpaceTime	N10 Bytes	Where/when Annotation is attached.
Rights	N11 Bytes	Rights to perform Actions of the Basic Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.31.5 Conformance Testing

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

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

8.32 Basic RADAR Scene Descriptors

8.32.1 Definition

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

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

8.32.2 Functional Requirements

Basic Scene Descriptors include

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

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

8.32.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicRADARSceneDescriptors.json

8.32.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic RADAR Scene Descriptors Header
- Standard-BasicRADARSceneDescriptors	9 Bytes	The characters "OSD-BRS-V"

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

8.32.5 Conformance Testing

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

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

8.33 Basic RADAR Scene Geometry

8.33.1 Definition

A Data Type including the arrangement of the RADAR Objects in a scene.

In the following, Data, Objects, Qualifiers, and Scenes should be read as RADAR Data, RADAR Objects, RADAR Qualifiers, and RADAR Scenes.

8.33.2 Functional Requirements

Basic Scene Geometry includes the Qualifiers and the Space-Time of the Objects.

8.33.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicRADARSceneGeometry.json

8.33.4 Semantics

Label	Size	Description

Header	N1 Bytes	Basic RADAR Scene Geometry Header
- Standard-BasicRADARSceneGeometry	9 Bytes	The characters “OSD-BRG-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneGeometryID	N5 Bytes	Identifier of Scene Geometry.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SceneGeometrySpaceTime	N8 Bytes	Space and Time of Scene Geometry.
SceneObjects[]	N9 Bytes	Set of Data related to Objects.
- SceneObjectQualifiers	N10 Bytes	Qualifiers of Object.
- SceneObjectSpaceTime	N11 Bytes	Space Time of Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.33.5 Conformance Testing

A Data instance Conforms with Basic RADAR Scene Geometry (OSD-BRG) V1.3 if:

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

8.34 Basic Scene Descriptors

8.34.1 Definition

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

8.34.2 Functional Requirements

Basic Scene Descriptors include:

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

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

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

8.34.3 Syntax

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

8.34.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Scene Descriptors Header
- Standard-BasicSceneDescriptors	9 Bytes	The characters “OSD-BSD-”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
BasicSceneDescriptorsID	N5 Bytes	Identifier of the Basic Scene.
SpaceTime	N6 Bytes	Data about Space-Time of Basic Scene.
ObjectCount	N7 Bytes	Number of Objects in Scene.
Objects[]	N16 Bytes	Set of Objects.
- ObjectID and/or Object	N17 Bytes	ObjectID or Object.
- ObjectSpaceTime	N18 Bytes	Space-Time of Object.
DescrMetadata	N32 Bytes	Descriptive Metadata.

8.34.5 Conformance Testing

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

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

8.35 Basic Scene Geometry

8.35.1 *Definition*

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

8.35.2 *Functional Requirements*

Basic Scene Geometry includes:

1. The ID of a Virtual Space where it is/will be located.
2. The ID of the Basic Scene Geometry.
3. The Basic Scene Space-Time info.
4. The number of Objects in the Scene.
5. The Object ID and Space-Time of each Object:
6. The Rights that may be exercised on the Object.

The following Table gives the Media Type-dependent Scene Geometries.

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

8.35.3 *Syntax*

https://schemas.mpai.community/OSD/V1.3/data/BasicSceneGeometry.json

8.35.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Scene Geometry Header
- Standard-BasicSceneGeometry	9 Bytes	The characters “OSD-BSG-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
BasicSceneGeometryID	N5 Bytes	Identifier of the Basic Scene Geometry.

ObjectCount	N6 Bytes	Number of Objects in Basic Scene
BasicSceneSpaceTime	N7 Bytes	Space and Time info of Basic Scene Geometry
BasicSceneGeometry[]	N8 Bytes	Set of Basic Scene Geometry Data
- ObjectID	N9 Bytes	Object ID.
- ObjectSpaceTime	N10 Bytes	Object Space-Time - may supersede Object's own Space-Time
DescrMetadata	N11 Bytes	Descriptive Metadata

8.35.5 Conformance Testing

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

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

8.36 Basic Speech Object

8.36.1 Definition

A Data Type including:

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

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

8.36.2 Functional Requirements

A Basic Object may include:

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

    1. Space-Time information of the Annotation.
    2. Rights to perform Actions on the Annotation.

8.36.3 Syntax

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

8.36.4 Semantics

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

8.36.5 Conformance Testing

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

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

8.37 Basic Speech Scene Descriptors

8.37.1 Definition

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

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

8.37.2 Functional Requirements

Basic Scene Descriptors include

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

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

8.37.3 Syntax

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

8.37.4 Semantics

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

8.37.5 Conformance Testing

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

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

3. Conform with their Data Qualifiers.

8.38 Basic Speech Scene Geometry

8.38.1 Definition

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

In the following, Data, Objects, Qualifiers, and Scenes should be read as Speech Data, Speech Objects, Speech Qualifiers, and Speech Scenes.

8.38.2 Functional Requirements

Basic Scene Geometry includes the Qualifiers and the Space-Time of the Objects.

8.38.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicSpeechSceneGeometry.json

8.38.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Speech Scene Geometry Header
- Standard-BasicSpeechSceneGeometry	9 Bytes	The characters “OSD-BSG-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneGeometryID	N5 Bytes	Identifier of Scene Geometry.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SceneGeometrySpaceTime	N8 Bytes	Space and Time of Scene Geometry.
SceneObjects[]	N9 Bytes	Set of Data related to Objects.
- SceneObjectQualifiers	N10 Bytes	Qualifiers of Object.
- SceneObjectSpaceTime	N11 Bytes	Space Time of Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.38.5 Conformance Testing

A Data instance Conforms with Basic Speech Scene Geometry (OSD-BSG) V1.3 if:

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

    3. Conform with their Data Qualifiers.

8.39 Basic Text Object

8.39.1 Definition

A Data Type including

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

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

8.39.2 Functional Requirements

A Basic Object may include:

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

8.39.3 Syntax

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

8.39.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Text Object Header
– Standard-BasicTextObject	9 Bytes	The characters “OSD-BTO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters

MInstanceID	N4 Bytes	Identifier of M-Instance.
BasicObjectID	N5 Bytes	Identifier of the Basic Object.
BasicObjectSpaceTime	N6 Bytes	Space-Time info of the Basic Object.
Qualifier	N7 Bytes	Qualifier of Basic Data.
BasicObjectAnnotations[]	N8 Bytes	Annotations of Basic Object.
– Annotation	N9 Bytes	ID of Annotation
– AnnotationSpaceTime	N10 Bytes	Where/when Annotation is attached.
Rights	N11 Bytes	Rights to perform Actions of the Basic Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.39.5 Conformance Testing

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

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

8.40 Basic Ultrasound Object

8.40.1 Definition

A Data Type including

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

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

8.40.2 Functional Requirements

A Basic Object may include:

1. The ID of a Virtual Space (M-Instance) where it is or is intended to be located.
2. The ID of the Basic Object.
3. The ID(s) of Parent Object(s) supporting two cases:
    1. The Parent Object has spawned two (or more) Objects.
    2. Two (or more) Parent Objects have merged into one.
4. The Space-Time information of Parent Objects in an M-Instance.
5. The ID(s) of Child Object(s).

6. The Space-Time information of Child Objects in an M-Instance.
7. The Space-Time information of the Basic Object in an M-Instance.
8. The Qualifier of the specific Data Type.
9. The Rights that can be exercised on the Basic Object.
10. The set of Annotations including, for each Annotation:
    1. Space-Time information of the Annotation.
    2. Rights to perform Actions on the Annotation.

8.40.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicUltrasoundObject.json

8.40.4 Semantics

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

8.40.5 Conformance Testing

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

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

8.41 Basic Ultrasound Scene Descriptors

8.41.1 Definition

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

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

8.41.2 Functional Requirements

Basic Scene Descriptors include

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

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

8.41.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicUltrasoundSceneDescriptors.json

8.41.4 Semantics

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

8.41.5 Conformance Testing

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

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

8.42 Basic Ultrasound Scene Geometry

8.42.1 Definition

A Data Type including the arrangement of the Ultrasound Objects in a scene.

In the following, Data, Objects, Qualifiers, and Scenes should be read as Ultrasound Data, Ultrasound Objects, Ultrasound Qualifiers, and Ultrasound Scenes.

8.42.2 Functional Requirements

Basic Scene Geometry includes the Qualifiers and the Space-Time of the Objects.

8.42.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicUltrasoundSceneGeometry.json

8.42.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Ultrasound Scene Geometry Header
- Standard-BasicUltrasoundSceneGeometry	9 Bytes	The characters "OSD-BUG-V"
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneGeometryID	N5 Bytes	Identifier of Scene Geometry.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SceneGeometrySpaceTime	N8 Bytes	Space and Time of Scene Geometry.
SceneObjects[]	N9 Bytes	Set of Data related to Objects.
- SceneObjectQualifiers	N10 Bytes	Qualifiers of Object.
- SceneObjectSpaceTime	N11 Bytes	Space Time of Object.

DescrMetadata	N12 Bytes	Descriptive Metadata

8.42.5 Conformance Testing

A Data instance Conforms with Basic Ultrasound Scene Geometry (OSD-BUG) V1.3 if:

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

8.43 Basic Visual Object

8.43.1 Definition

A Data Type including:

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

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

8.43.2 Functional Requirements

A Basic Object may include:

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

8.43.3 Syntax

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

8.43.4 Semantics

Label	Size	Description

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

8.43.5 Conformance Testing

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

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

8.44 Basic Visual Scene Descriptors

8.44.1 Definition

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

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

8.44.2 Functional Requirements

Basic Scene Descriptors include

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

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

8.44.3 Syntax

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

8.44.4 Semantics

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

8.44.5 Conformance Testing

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

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

8.45 Basic Visual Scene Geometry

8.45.1 Definition

A Data Type including the arrangement of the Basic Visual Scene Geometry Objects in a scene.

In the following, Data, Objects, Qualifiers, and Scenes should be read as Visual Data, Visual Objects, Visual Qualifiers, and Visual Scenes.

8.45.2 Functional Requirements

Basic Scene Geometry includes the Qualifiers and the Space-Time of the Objects.

8.45.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicVisualSceneGeometry.json

8.45.4 Semantics

Label	Size	Description
Header	N1 Bytes	Basic Visual Scene Geometry Header
- Standard-BasicVisualSceneGeometry	9 Bytes	The characters "OSD-BVG-V"
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
SceneGeometryID	N5 Bytes	Identifier of Scene Geometry.
ObjectCount	N6 Bytes	Number of Objects in Scene.
SceneGeometrySpaceTime	N8 Bytes	Space and Time of Scene Geometry.
SceneObjects[]	N9 Bytes	Set of Data related to Objects.
- SceneObjectQualifiers	N10 Bytes	Qualifiers of Object.
- SceneObjectSpaceTime	N11 Bytes	Space Time of Object.
DescrMetadata	N12 Bytes	Descriptive Metadata

8.45.5 Conformance Testing

A Data instance Conforms with Basic Visual Scene Geometry (OSD-BVG) V1.3 if:

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

8.46 Bounding Box

8.46.1 Definition

A Data Type representing a rectangle (2D Bounding Box) or right parallelepiped (3D Bounding Box) containing a 2D or 3D Visual Object, respectively.

8.46.2 Functional Requirements

The rectangle or right parallelepiped is defined, respectively, by

1. Rectangle (2D): 3 vertices not on a straight line.
2. Right Parallelepiped (3D): 4 vertices not on a plane.

The Visual Object (Content) may fit exactly in the rectangle/parallelepiped and have the same axes of the rectangle/parallelepiped.

Content may be absent.

8.46.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BoundingBox.json

8.46.4 Semantics

Label	Size	Description
Header	N1 Bytes	Bounding Box Header
- Standard-BoundingBox	9 Bytes	The characters “OSD-BBX-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance
BoundingBoxID	N5 Bytes	Identifier of BoundingBox.
Dimensions	2 Bytes	One of 2D, 3D
VisualDataQualifier	N6 Bytes	Qualifier of Visual Data in the BoundingBox.
DescrMetadata	N7 Bytes	Descriptive Metadata

8.46.5 Conformance Testing

A Data instance Conforms with Bounding Box (OSD-BBX) V1.3 if:

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

8.47 Coordinates

8.47.1 Definition

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

8.47.2 Functional Requirements

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

The coordinate systems supported so far are:

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

8.47.3 Syntax

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

8.47.4 Semantics

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

8.47.5 Conformance Testing

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

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

8.48 Device Scene Geometry

8.48.1 Definition

A Data Type including the Point of View of the Devices capturing media information, possibly grouped according to the characteristics of Devices.

8.48.2 Functional Requirements

A Device Scene Geometry may include:

1. The ID of a Virtual Space (M-Instance) the Device Geometry may refer to.
2. Groups of Devices sharing the same characteristics including:
    1. Characteristic Name.
    2. Data about the individual Devices of the group, including:
        1. Device ID.
        2. Point of View.

8.48.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/DeviceSceneGeometry.json

8.48.4 Semantics

Label	Size	Description
Header	N1 Bytes	Device Scene Geometry Header
– Standard-DeviceSceneGeometry	9 Bytes	The characters “OSD-DSG-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
CaptureTime	N5 Bytes	Duration of media information capture.
DeviceEnsembles[]	N6 Bytes	Identifier of the Audio Object.
- CharacteristicName	N7 Bytes	A Name in an Enumeration.
- DeviceEnsemble[]	N8 Bytes	Data of the Devices in the group.
- DeviceID	N9 Bytes	Device ID.
- PointOf View	N10 Bytes	Point of View of the Device.
DescrMetadata	N11 Bytes	Descriptive Metadata

8.48.5 Conformance Testing

A Data instance Conforms with Data Scene Geometry (OSD-DSG) V1.2 if:

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

8.49 Event Descriptors

8.49.1 Definition

An Item represented by a series of Scene Descriptors for a certain duration (from a start Time to an end Time).

8.49.2 Functional Requirements

Event Descriptors contains Scene Descriptors between two Times.

8.49.3 Syntax

https://schemas.mpai.community/OSD/V1.2/data/AudioVisualEventDescriptors.json

8.49.4 Semantics

Label	Size	Description
Header	N1 Bytes	Event Descriptors Header
- Standard - EventDescriptors	9 Bytes	The characters “OSD-EVD-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
EventDescriptorsID	N5 Bytes	Identifier of Event Descriptors.
EventDescriptorsSpaceTime	N6 Bytes	Data about start and end Space-Time.
EventDescriptors[]	N7 Bytes	Collection of Scene Descriptors between Start Time and End Time
- SceneDescriptors	N8 Bytes	Set of Scene Descriptors of IDs.
DescrMetadata	N9 Bytes	Descriptive Metadata

8.49.5 Conformance Testing

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

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

8.50 Instance Identifier

8.50.1 Definition

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

8.50.2 Functional Requirements

Instance Identifier includes:

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

8.50.3 Syntax

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

8.50.4 Semantics

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

8.50.5 Conformance Testing

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

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

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

8.51 LiDAR Event Descriptors

8.51.1 Definition

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

8.51.2 Functional Requirements

LiDAR Event Descriptors contains LiDAR Scene Descriptors for a Time.

8.51.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/LiDAREventDescriptors.json

8.51.4 Semantics

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

8.51.5 Conformance Testing

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

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

8.52 LiDAR Object

8.52.1 Definition

A Data Type including a collection of Basic LiDAR Objects.

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

8.52.2 Functional Requirements

A LiDAR Object may include:

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

Note that.

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

8.52.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/LiDARObject.json

8.52.4 Semantics

Label	**Size**	**Description**
Header	N1 Bytes	LiDAR Object Header
– Standard-LiDARObject	9 Bytes	The characters “OSD-LIO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
LiDARObjectID	N5 Bytes	Identifier of the LiDAR Object.
LiDARObjectSpaceTime	N6 Bytes	Space-Time of LiDAR Object.
BasicLiDARObjectCount	N7 Bytes	Set of Parent LiDAR Objects.
BasicLiDARObjects[]	N8 Bytes	Set of Basic LiDAR Objects.

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

8.52.5 Conformance Testing

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

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

8.53 LiDAR Scene Descriptors

8.53.1 Definition

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

8.53.2 Functional Requirements

LiDAR Scene Descriptors include

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

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

8.53.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/LiDARSceneDescriptors.json

8.53.4 Semantics

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

8.53.5 Conformance Testing

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

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

8.54 LiDAR Scene Geometry

8.54.1 Definition

An Data Type including the arrangement of the LiDAR Objects in a scene with their LiDAR Qualifiers.

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

8.54.2 Functional Requirements

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

8.54.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/LiDARSceneGeometry.json

8.54.4 Semantics

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

8.54.5 Conformance Testing

A Data instance Conforms with LiDAR Scene Geometry (OSD-LSG) V1.3 if:

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

8.55 Location

8.55.1 Definition

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

8.55.2 Functional Requirements

A Location

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

8.55.3 Syntax

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

8.55.4 Semantics

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

8.55.5 Conformance Testing

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

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

8.56 Object

8.56.1 Definition

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

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

8.56.2 Functional Requirements

An Object may include:

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

Note that.

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

The following table gives the Media Type-specific Objects.

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

8.56.3 Syntax

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

8.56.4 Semantics

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

8.56.5 Conformance Testing

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

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

8.57 Offline Map Event Descriptors

8.57.1 Definition

An Item including a series of Offline Map Event Descriptors Scene Descriptors for a certain duration.

8.57.2 Functional Requirements

Offline Map Event Descriptors contains Offline Map Scene Descriptors for a Time.

8.57.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/OfflineMapEventDescriptors.json

8.57.4 Semantics

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

8.57.5 Conformance Testing

A Data instance Conforms with MPAI-OSD Offline Map Event Descriptors V1.3 (OSD-OME) if:

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

8.58 Offline Map Object

8.58.1 Definition

A Data Type including a collection of Basic Offline Map Objects.

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

8.58.2 Functional Requirements

An Offline Map Object may include:

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

Note that.

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

8.58.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/OfflineMapObject.json.json

8.58.4 *Semantics*

Label	**Size**	**Description**
Header	N1 Bytes	Offline Map Object Header
– Standard-AudioObject	9 Bytes	The characters “OSD-OMO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
OfflineMapObjectID	N5 Bytes	Identifier of the Offline Map Object.
OfflineMapObjectSpaceTime	N6 Bytes	Space-Time of Offline Map Object.
BasicOfflineMapObjectCount	N7 Bytes	Set of Parent Offline Map Objects.
BasicOfflineMapObjects[]	N8 Bytes	Set of Basic Offline Map Objects.

- SpaceTime	N9 Bytes	Space Time of a Basic Offline Map Object in the Offline Map Object.
- Basic OfflineMapObject	N10 Bytes	A Basic Offline Map Object in the Offline Map Object.
OfflineMapObjectCount	N11 Bytes	Number of Offline Map Objects.
OfflineMapObjects[]	N12 Bytes	Set of Offline Map Objects.
- SpaceTime	N13 Bytes	Space Time of an Offline Map Object in the Offline Map Object.
- AudioObject	N14 Bytes	An Offline Map Object in the Offline Map Object
Annotations[]	N15 Bytes	Set of Offline Map Object Annotation.
– Annotation	N16 Bytes	An Annotation.
– AnnotationSpaceTime	N17 Bytes	Where Annotation is attached and when it will be active.
– Rights	N18 Bytes	Actions that may be performed on the Annotation
Rights	N19 Bytes	Actions that may be performed on the Object.
DescrMetadata	N20 Bytes	Descriptive Metadata

8.58.5 Conformance Testing

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

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

8.59 Offline Map Scene Descriptors

8.59.1 Definition

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

8.59.2 Functional Requirements

Offline Map Scene Descriptors include

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

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

8.59.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/OfflineMapSceneDescriptors.json

8.59.4 Semantics

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

8.59.5 Conformance Testing

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

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

8.60 Offline Map Scene Geometry

8.60.1 Definition

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

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

8.60.2 Functional Requirements

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

8.60.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/OfflineMapSceneGeometry.json

8.60.4 Semantics

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

8.60.5 Conformance Testing

A Data instance Conforms with Offline Map Scene Geometry (OSD-OSG) V1.3 if:

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

8.61 Orientation

8.61.1 Definition

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

8.61.2 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.

8.61.3 Syntax

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

8.61.4 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	Size	Description
Header	N1 Bytes	Orientation Header
- Standard-Orientation	9 Bytes	The characters “OSD-OOR-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstance	N4 Bytes	ID id Virtual space Orientation refers tu
ObjectOrientationID	N5 Bytes	Identifier of Object Orientation.
General	N6 Bytes	Set of general data
- ObjectType	N8 Bytes	One of Digital Human, Generic.
- MediaType	N9 Bytes	One of Speech, Audio, Visual, Audio-Visual, Haptic, Smell, RADAR, LiDAR, Ultrasound.
Orientation		
- Orient (α,β,γ)	N10 Bytes	Array (in degrees)
- OrientAccuracy	N11 Bytes	Array of (α,β,γ) Accuracy
Velocity of Orientation		
- OrientVelocity (α,β,γ)	N12 Bytes	Array (in degrees/s)
- OrientVelocityAccuracy	N13 Bytes	Array of (α,β,γ) Velocity Accuracy
Acceleration of Orientation		
- OrientAccel (α,β,γ)	N14 Bytes	Array (in degrees/s2)

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

8.61.5 Conformance Testing

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

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

8.62 Path

8.62.1 Definition

Path is a sequence of Points of View.

8.62.2 Functional Requirements

Path is defined against an M-Instance or an OfflineMap.

8.62.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/Path.json

8.62.4 Semantics

Label	Size	Description
Header	N1 Bytes	Path Header
- Standard	9 Bytes	The characters "OSD-PAT-V"
- Version	N2 Bytes	Major version – 1 or 2 Bytes
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 Bytes
PathID	N4 Bytes	Identifier of Path.
MInstanceID	N5 Bytes	ID of the Virtual Space where Path is defined.
OfflineMapID	N6 Bytes	ID of the referenced OfflineMap.
PathData[]	N7 Bytes	Path Dataset.
- PointOfView	N8 Bytes	Planned Individual Point of View in the Path.
DescrMetadata	N9 Bytes	Descriptive Metadata

8.62.5 Conformance Testing

A Data instance Conforms with Path MPAI-OSD V1.3 (OSD-PAT) if the Data

1. Have the specified type
2. Validate against the Path's JSON Schema.

8.63 Perceptible Entity

8.63.1 Definition

Perceptible Entity is one of

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

8.63.2 Functional Requirements

A Perceptible Entity

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

8.63.3 Syntax

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

8.63.4 Semantics

Label	Size	Description
Header	N1 Bytes	Perceptible Entity Header
- Standard-PerceptibleEntity	9 Bytes	The characters "OSD-PCE-V"
- Version	N2 Bytes	Major version – 1 or 2 Bytes
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 Bytes
PerceptibleEntityID	N4 Bytes	Identifier of Perceptible Entity.
PerceptibleEntity	N5 Bytes	Anyone of the following Objects, Scenes, or Events.
- Object	N6 Bytes	Intended Object
- Scene Descriptors	N7 Bytes	Intended Scene
- Event Descriptors	N8 Bytes	Intended Event
- RightsID or Rights	N9 Bytes	Individual Rights ID
DescrMetadata	N10 Bytes	Descriptive Metadata

8.63.5 Conformance Testing

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

1. The Data validates against the Perceptible Entity's JSON Schema.
2. All Data in the Perceptible Entity's JSON Schema

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

8.64 Point of View

8.64.1 Definition

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

8.64.2 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.

8.64.3 Syntax

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

8.64.4 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	Size	Description
Header	N1 Bytes	Point of View Header
- Standard-Point of View	9 Bytes	The characters “OSD-OPV-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstance	N4 Bytes	ID id Virtual space Orientation refers tu
PointOfViewID	N5 Bytes	Identifier of Object Point of View.
General	N6 Bytes	Set of general data.
- CoordType	N7 Bytes	One of Cartesian, Spherical, Geodesic, Toroidal.
- ObjectType	N8 Bytes	One of Digital Human, Generic.
- MediaType	N9 Bytes	One of Speech, Audio, Visual, Audio-Visual, Haptic, Smell, RADAR, LiDAR, Ultrasound.
PositionAndOrientation		

- CartPosition (X,Y,Z)	N10 Bytes	Array (in metres)
- CartPositionAccuracy (X,Y,Z)	N11 Bytes	Array Of CartPositionAccuracy
- SpherPosition (r,φ,θ)	N12 Bytes	Array (in metres and degrees)
- SpherPositionAccuracy (r,φ,θ)	N13 Bytes	Array of - SpherPositionAccuracy
- Orient (α,β,γ)	N14 Bytes	Array (in degrees)
- OrientAccuracy (α,β,γ)	N15 Bytes	Array of OrientAccuracy
DescrMetadata	N16 Bytes	Descriptive Metadata

8.64.5 Conformance Testing

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

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

8.65 Position

8.65.1 Definition

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

8.65.2 Functional Requirements

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

8.65.3 Syntax

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

8.65.4 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	Size	Description
Header	N1 Bytes	Position Header
- Standard-Position	9 Bytes	The characters “OSD-OPS-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	ID of Virtual Space Position refers to.
ObjectPositionID	N5 Bytes	Identifier of Object Position.
General	N6 Bytes	Set of general data
- CoordinateType	N7 Bytes	One of Cartesian, Spherical, Geodesic, Toroidal.
- ObjectType	N8 Bytes	One of Digital Human, Generic.
- MediaType	N9 Bytes	One of Speech, Audio, Visual, 3D Model, Audio-Visual, Haptic, Smell, RADAR, LiDAR, Ultrasound.
Position		
- CartPosition (X,Y,Z)	N10 Bytes	Array (in metres)
- CartPositionAccuracy (X,Y,Z)	N11 Bytes	Array of CartPositionAccuracy
- SpherPosition (r,φ,θ)	N12 Bytes	Array (in metres and degrees)
- SpherPositionAccuracy (r,φ,θ)	N13 Bytes	Array of SpherPositionAccuracys
Velocity of Position		
- CartVelocity (X,Y,Z)	N14 Bytes	Array (in metres)
- CartVelocityAccuracy (X,Y,Z)	N15 Bytes	Array of - CartVelocityAccuracys (X,Y,Z)
- SpherVelocity (r,φ,θ)	N16 Bytes	Array (in metres and degrees)
- SpherVelocityAccuracy (r,φ,θ)	N17 Bytes	Array of SpherVelocityAccuracys
Acceleration of Position		
- CartAccel (X,Y,Z)	N18 Bytes	Array (in metres)
- CartAccelAccuracy (X,Y,Z)	N19 Bytes	Array of CartAccelAccuracys
- SpherAccel (r,φ,θ)	N20 Bytes	Array (in metres and degrees)
- SpherAccel (r,φ,θ)	N21 Bytes	Array (in metres and degrees)
DescrMetadata	N22 Bytes	Descriptive Metadata

8.65.5 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.

8.66 RADAR Event Descriptors

8.66.1 Definition

An Item including a RADAR Event Descriptors Scene Descriptors for a certain duration.

8.66.2 Functional Requirements

RADAR Event Descriptors contains RADAR Scene Descriptors for a Time.

8.66.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/RADAREventDescriptors.json

8.66.4 Semantics

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

8.66.5 Conformance Testing

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

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

8.67 RADAR Object

8.67.1 Definition

A Data Type including a collection of Basic RADAR Objects.

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

8.67.2 Functional Requirements

A RADAR Object may include:

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

Note that.

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

8.67.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/RADARObject.json

8.67.4 Semantics

Label	**Size**	**Description**
Header	N1 Bytes	RADAR Object Header
– Standard-RADARObject	9 Bytes	The characters “OSD-AUO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
RADARObjectID	N5 Bytes	Identifier of the RADAR Object.
RADARObjectSpaceTime	N6 Bytes	Space-Time of RADAR Object.
BasicRADARObjectCount	N7 Bytes	Set of Parent RADAR Objects.
BasicRADARObjects[]	N8 Bytes	Set of Basic RADAR Objects.
- SpaceTime	N9 Bytes	Space Time of a Basic RADAR Object in the RADAR Object.
- BasicRADARObject	N10 Bytes	A Basic RADAR Object in the RADAR Object.
RADARObjectCount	N11 Bytes	Number of RADAR Objects.
RADARObjects[]	N12 Bytes	Set of RADAR Objects.

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

8.67.5 Conformance Testing

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

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

8.68 RADAR Scene Descriptors

8.68.1 Definition

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

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

8.68.2 Functional Requirements

A Basic Object may include:

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

8.68.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/BasicRADARObject.json

8.68.4 Semantics

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

8.68.5 Conformance Testing

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

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

8.69 RADAR Scene Geometry

8.69.1 Definition

An Data Type including the arrangement of the RADAR Objects in a scene with their RADAR Qualifiers.

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

8.69.2 Functional Requirements

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

8.69.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/RADARSceneGeometry.json

8.69.4 Semantics

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

8.69.5 Conformance Testing

A Data instance Conforms with RADAR Scene Geometry (OSD-RSG) V1.3 if:

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

8.70 Right Parallelepiped

8.70.1 Definition

A Data Type representing a rectangle (2D or right parallelepiped containing a 2D or 3D Visual Object, respectively.

8.70.2 Functional Requirements

The rectangle or right parallelepiped is defined, respectively, by

1. Rectangle (2D): 3 vertices not on a straight line.
2. Right Parallelepiped (3D): 4 vertices not on a plane.

8.70.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/RightParallelepiped.json

8.70.4 Semantics

Label	Size	Description
Header	N1 Bytes	Right Parallelepiped Header
- Standard-RightParallelepiped	9 Bytes	The characters “OSD-RPP-V”
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character “.”
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance
RightParallelepipedID	N5 Bytes	Identifier of Right Parallelepiped.
Dimensions	1 Byte	One of 2D, 3D
RightParallelepipedData	N6 Bytes	Data about Right Parallelepiped.
- Precision	1 Byte	Simple or double precision
- CartPositions	12/24 Bytes	The Positions of the selected vertices of the rectangle/parallelepiped.
DescrMetadata	N6 Bytes	Descriptive Metadata

8.70.5 Conformance Testing

A Data instance Conforms with MPAI-OSD V1.2 Right Parallelepiped (OSD-RPP) if:

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

8.71 Scene Descriptors

8.71.1 Definition

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

8.71.2 Functional Requirements

Scene Descriptors include

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

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

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

8.71.3 Syntax

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

8.71.4 Semantics

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

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

8.71.5 Conformance Testing

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

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

8.72 Scene Geometry

8.72.1 Definition

A Data Type representing the arrangement of the Objects of a Scene that may include Objects and sub-scenes in a hierarchical fashion.

8.72.2 Functional Requirements

Scene Geometry may reference Objects and Scenes in a hierarchical fashion.

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

The following Table gives the Media Type-dependent Scene Geometries.

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

8.72.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/SceneGeometry.json

8.72.4 4 Semantics

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

8.72.5 Conformance Testing

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

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

8.73 Selector

8.73.1 Definition

A Data Type used to indicate specific operating values of an AIW or AIM.

8.73.2 Functional Requirements

Selector informs an AIW/AIM that a communicating Entity uses/requests to use:

1. Specific media – Text, Speech, Visual, or Gesture – as input or output.
2. Specific Language – as input or output.
3. Media or their Descriptors.
4. View an Avatar or a Scene

8.73.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/Selector.json

8.73.4 Semantics

Label	Size	Description
Header	N1 Bytes	Selector Header
- Standard-Selector	9 Bytes	The characters “OSD-SEL-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
InputMedia	N4 Bytes	One or more of Text, Speech, Visual, or Gesture.
OutputMedia	N5 Bytes	One or more of Text, Speech, Visual, or Gesture.
InputLanguage	N6 Bytes	One of a list of languages.
OutputLanguage	N6 Bytes	One of a list of languages.
MediaOrDescriptors	N7 Bytes	One of Text, Speech, Face, Body for MMC-TST
SpeechDescriptors	N8 Bytes	One of No, Yes for MMC-PSE
View	N9 Bytes	One of Avatar or Scene
DescrMetadata	N9 Bytes	Descriptive Metadata

8.73.5 Conformance Testing

A Data instance Conforms with Selector (OSD-SEL) V1.2 if:

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

8.74 Space-Time

8.74.1 Definition

Data Type representing the Spatial Attitude and Time information.

8.74.2 Functional Requirements

Space-Time includes Spatial Attitude and Time.

8.74.3 Syntax

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

8.74.4 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

8.74.5 Conformance Testing

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

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

8.75 Spatial Attitude

8.75.1 Definition

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

8.75.2 Functional Requirements

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

8.75.3 Syntax

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

8.75.4 Semantics

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

Table 1 – Semantics of the Spatial Attitude

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

8.75.5 Conformance Testing

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

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

8.76 Speech Event Descriptors

8.76.1 Definition

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

8.76.2 Functional Requirements

Speech Event Descriptors contains Speech Scene Descriptors for a Time.

8.76.3 Syntax

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

8.76.4 Semantics

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

8.76.5 Conformance Testing

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

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

8.77 Speech Object

8.77.1 *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.

8.77.2 *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.

8.77.3 *Syntax*

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

8.77.4 *Semantics*

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

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

8.77.5 Conformance Testing

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

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

8.78 Speech Scene Descriptors

8.78.1 Definition

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

8.78.2 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.

8.78.3 Syntax

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

8.78.4 Semantics

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

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

8.78.5 Conformance Testing

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

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

8.79 Speech Scene Geometry

8.79.1 Definition

An 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

8.79.2 Functional Requirements

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

8.79.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/SpeechSceneGeometry.json

8.79.4 Semantics

Label	Size	Description
Header	N1 Bytes	Speech Scene Geometry Header
- Standard-SpeechSceneGeometry	9 Bytes	The characters "OSD-SSG-V"

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

8.79.5 Conformance Testing

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

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

8.80 Text Object

8.80.1 *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.

8.80.2 *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.

8.80.3 *Syntax*

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

8.80.4 Semantics

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

8.80.5 Conformance Testing

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

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

8.81 Time

8.81.1 Definition

The start time and the end time of a duration.

8.81.2 Functional Requirements

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

8.81.3 Syntax

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

8.81.4 Semantics

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

8.81.5 Conformance Testing

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

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

8.82 Trajectory

8.82.1 Definition

The sequence of start and end Spatial Attitudes *SA* (*SA_1*, *SA_2*, …, *SA_i*) and corresponding Times *t (*t_1, t_2, t_j*)* expected and actual of a series of segments.

8.82.2 Functional Requirements

A Trajectory is composed of Segments. Each Segment is described by the expected and actual start and end Spatial Attitudes and Times.

8.82.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/Trajectory.json

8.82.4 Semantics

Label	Size	Description
Header	N1 Bytes	Trajectory Header
- Standard-Trajectory	9 Bytes	The characters "CAV-TRJ-V"
- Version	N2 Bytes	Major version – 1 or 2 Bytes
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 Bytes
TrajectoryID	N4 Bytes	Identifier of Trajectory.
TrajectoryData[]	N5Bytes	Data in the Trajectory
- SpaceTime	N6 Bytes	Expected and/or actual Spatial Attitude and Time of a Trajectory segment.
DescrMetadata	N7 Bytes	Descriptive Metadata

8.82.5 Conformance Testing

A Data instance Conforms with Trajectory (OSD-TRJ) V1.3 if:

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

8.83 Ultrasound Event Descriptors

8.83.1 Definition

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

8.83.2 Functional Requirements

Ultrasound Event Descriptors contains Ultrasound Scene Descriptors for a Time.

8.83.3 Syntax

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

8.83.4 Semantics

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

8.83.5 Conformance Testing

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

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

8.84 Ultrasound Object

8.84.1 Definition

A Data Type including a collection of Basic Ultrasound Objects.

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

8.84.2 Functional Requirements

A Ultrasound Object may include:

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

Note that.

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

8.84.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/UltrasoundObject.json

8.84.4 Semantics

Label	Size	Description
Header	N1 Bytes	Ultrasound Object Header
– Standard-UltrasoundObject	9 Bytes	The characters "OSD-USO-V"
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character "."
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
UltrasoundObjectID	N5 Bytes	Identifier of the Ultrasound Object.
UltrasoundObjectSpaceTime	N6 Bytes	Space-Time of Ultrasound Object.
BasicUltrasoundObjectCount	N7 Bytes	Set of Parent Ultrasound Objects.
BasicUltrasoundObjects[]	N8 Bytes	Set of Basic Ultrasound Objects.
- SpaceTime	N9 Bytes	Space Time of a Basic Ultrasound Object in the Ultrasound Object.
- BasicUltrasoundObject	N10 Bytes	A Basic Ultrasound Object in the Ultrasound Object.
UltrasoundObjectCount	N11 Bytes	Number of Ultrasound Objects.
UltrasoundObjects[]	N12 Bytes	Set of Ultrasound Objects.

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

8.84.5 Conformance Testing

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

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

8.85 Ultrasound Scene Descriptors

8.85.1 Definition

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

8.85.2 Functional Requirements

Ultrasound Scene Descriptors include

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

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

8.85.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/UltrasoundSceneDescriptors.json

8.85.4 Semantics

Label	Size	Description
Header	N1 Bytes	Ultrasound Scene Descriptors Header
- Standard-UltrasoundSceneDescriptors	9 Bytes	The characters "OSD-USD-V"

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

8.85.5 Conformance Testing

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

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

8.86 Ultrasound Scene Geometry

8.86.1 Definition

An Data Type including the arrangement of the Ultrasound Objects in a scene with their Ultrasound Qualifiers.

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

8.86.2 Functional Requirements

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

8.86.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/UltrasoundSceneGeometry.json

8.86.4 Semantics

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

8.86.5 Conformance Testing

A Data instance Conforms with Ultrasound Scene Geometry (OSD-USG) V1.3 if:

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

8.87 Visual Event Descriptors

8.87.1 Definition

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

8.87.2 Functional Requirements

Visual Event Descriptors contains Visual Scene Descriptors for a Time.

8.87.3 Syntax

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

8.87.4 Semantics

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

8.87.5 Conformance Testing

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

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

8.88 Visual Object

8.88.1 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.

8.88.2 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.

8.88.3 Syntax

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

8.88.4 *Semantics*

Label	Size	Description
Header	N1 Bytes	Visual Object Header
– Standard-VisualObject	9 Bytes	The characters “OSD-VIO-V”
– Version	N2 Bytes	Major version – 1 or 2 characters
– Dot-separator	1 Byte	The character “.”
– Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
VisualObjectID	N5 Bytes	Identifier of the Visual Object.
VisualObjectSpaceTime	N6 Bytes	Space-Time of Visual Object.
BasicVisualObjectCount	N7 Bytes	Set of Parent Visual Objects.
BasicVisualObjects[]	N8 Bytes	Set of Basic Visual Objects.
- SpaceTime	N9 Bytes	Space Time of a Basic Visual Object in the Visual Object.
- BasicVisualObject	N10 Bytes	A Basic Visual Object in the Visual Object.
VisualObjectCount	N11 Bytes	Number of Visual Objects.
VisualObjects[]	N12 Bytes	Set of Visual Objects.
- SpaceTime	N13 Bytes	Space Time of a Visual Object in the Visual Object.
- VisualObject	N14 Bytes	A Visual Object in the Visual Object
Annotations[]	N15 Bytes	Set of Visual Object Annotation.
– Annotation	N16 Bytes	An Annotation.
– AnnotationSpaceTime	N17 Bytes	Where Annotation is attached and when it will be active.
– Rights	N18 Bytes	Actions that may be performed on the Annotation

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

8.88.5 Conformance Testing

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

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

8.89 Visual Scene Descriptors

8.89.1 Definition

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

8.89.2 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.

8.89.3 Syntax

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

8.89.4 Semantics

Label	Size	Description
Header	N1 Bytes	Visual Scene Descriptors Header
- Standard-VisualSceneDescriptors	9 Bytes	The characters "OSD-VSD-V"
- Version	N2 Bytes	Major version – 1 or 2 characters
- Dot-separator	1 Byte	The character "."
- Subversion	N3 Bytes	Minor version – 1 or 2 characters
MInstanceID	N4 Bytes	Identifier of M-Instance.
VisualSceneDescriptorsID	N5 Bytes	Identifier of Visual Scene Descriptors.
ObjectCount	N6 Bytes	Number of Visual Objects in Scene.

SubSceneCount	N7 Bytes	Number of Visual Scenes in Scene.
VisualSceneDescriptorsSpaceTime	N8 Bytes	Space and Time of Visual Scene Descriptors.
VisualSceneObjects[]	N9 Bytes	Set of Visual Objects.
- VisualSceneObject	N10 Bytes	Visual Object.
- VisualSceneObjectSpaceTime	N11 Bytes	Space Time of Visual Object.
VisualSceneSubScenes[]	N12 Bytes	Set of Visual Sub-Scenes.
- VisualSceneSubScene	N13 Bytes	Visual Sub-Scene.
- VisualSceneSubSceneSpaceTime	N14 Bytes	Space Time of Visual Sub-Scene.
DescrMetadata	N15 Bytes	Descriptive Metadata

8.89.5 Conformance Testing

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

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

8.90 Visual Scene Geometry

8.90.1 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

8.90.2 Functional Requirements

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

8.90.3 Syntax

https://schemas.mpai.community/OSD/V1.3/data/VisualSceneGeometry.json

8.90.4 Semantics

Label	Size	Description
Header	N1 Bytes	Visual Scene Geometry Header
- Standard-VisualSceneGeometry	9 Bytes	The characters "OSD-VSG-V"

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

8.90.5 Conformance Testing

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

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