



Moving Picture, Audio and Data Coding  
by Artificial Intelligence  
[www.mpai.community](http://www.mpai.community)

# **MPAI Technical Specification**

## **Portable Avatar Format MPAI-PAF**

<b>V1</b>
-----------

### **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 use of this Technical Specification.

Readers are invited to review Annex 3 - Notices and Disclaimers.

# Technical Specification

## Portable Avatar Format (MPAI-PAF) V1

1	Introduction (Informative).....	3
2	Scope .....	6
3	Terms and Definitions .....	7
4	References .....	9
4.1	Normative References .....	9
4.2	Informative References .....	9
5	Avatar-Based Videoconference.....	9
5.1	Introduction .....	9
5.2	Scope of Use Case.....	10
5.3	Transmitting Client .....	11
5.3.1	Functions of Transmitting Client .....	11
5.3.2	Reference Architecture of Transmitting Client.....	12
5.3.3	Input and output data of Transmitting Client .....	13
5.3.4	Functions of Transmitting Client's AI Modules .....	13
5.3.5	I/O Data of Transmitting Client's AI Modules .....	13
5.3.6	JSON Metadata of Transmitting Client.....	14
5.4	Server .....	14
5.4.1	Functions of Server .....	14
5.4.2	Reference Architecture of Server .....	14
5.4.3	I/O Data of Server .....	15
5.4.4	Functions of Server AI Modules .....	15
5.4.5	I/O Data of Server AI Modules .....	15
5.4.6	JSON Metadata of Server.....	16
5.5	Virtual Secretary (informative) .....	16
5.5.1	Functions of Virtual Secretary .....	16
5.5.2	Reference Architecture.....	16
5.5.3	I/O Data of Virtual Secretary .....	17
5.5.4	JSON Metadata of Virtual Secretary.....	18
5.6	Receiving Client.....	18
5.6.1	Functions of Receiving Client.....	18
5.6.2	Reference Architecture of Receiving Client .....	18
5.6.3	I/O Data of Receiving Client.....	19
5.6.4	Functions of Receiving Client's AI Modules.....	19
5.6.5	I/O Data of Receiving Client's AI Modules.....	19
5.6.6	JSON Metadata of Receiving Client .....	20
6	Composite AI Modules .....	20
6.1	Personal Status Extraction (PSE) (Informative).....	20
6.1.1	Scope of Composite AIM.....	20
6.1.2	Reference architecture.....	20
6.1.3	I/O Data of Personal Status Extraction .....	21
6.2	Personal Status Display (PSD).....	21
6.2.1	Scope of Composite AIM.....	21
6.2.2	Reference Architecture.....	22
6.2.3	I/O Data of Personal Status Display.....	22

6.2.4	Functions of AI Modules of Personal Status Display .....	22
6.2.5	I/O Data of AI Modules of Personal Status Display .....	23
6.2.6	JSON Metadata of Personal Status Display .....	23
7	Data Formats .....	23
7.1	Portable Avatar Format .....	24
7.2	Environment .....	27
7.2.1	Audio Environment .....	27
7.2.2	Visual Environment .....	27
7.3	Body .....	27
7.3.1	Body Model .....	27
7.3.2	Body Descriptors .....	28
7.3.3	Head Descriptors .....	29
7.4	Face .....	29
7.4.1	Face Model .....	29
7.4.2	Face Descriptors .....	29
7.5	Avatar .....	30
7.5.1	Avatar Model .....	30
7.6	Objects and Scenes .....	30
7.6.1	Spatial Attitude and Point of View .....	30
7.6.2	Audio Scene Descriptors .....	31
7.6.3	Visual Scene Descriptors .....	31
7.7	Other Data Types .....	31
7.7.1	Text .....	31
7.7.2	Language identifier .....	31
7.7.3	Meaning .....	31
7.7.4	Personal Status .....	31
	Annex 1 - MPAI Basics (Informative) .....	32
1	General .....	32
2	Governance of the MPAI Ecosystem .....	32
3	AI Framework .....	33
4	Audio-Visual Scene Description .....	34
4.1.1	Audio Scene Descriptors .....	34
4.1.2	Visual Scene Descriptors .....	34
	Annex 2 - General MPAI Terminology .....	35
	Annex 3 - Notices and Disclaimers Concerning MPAI Standards (Informative) .....	38
	Annex 4 - Patent Declarations .....	40
	Annex 5 - AIW and AIM Metadata of ABV-CTX .....	41
1	Metadata for ABV-CTX .....	41
2	AIW metadata for ABV-SRV .....	45
3	Metadata for MMC-VSV .....	50
4	AIW metadata for ABV-CRX .....	56
5	Metadata of MMC-PSE .....	60

## 1 Introduction (Informative)

There is a long history of computer-created objects called “digital humans”, i.e., digital objects having a human appearance when rendered. In most cases the underlying assumption of these objects has been that creation, animation, and rendering is done in a closed environment. Such digital humans had little or no need for standards.

In a communication and more so in a metaverse context, there are many cases where a digital human is not constrained within a closed environment thus requiring forms of standardisation. *Technical Specification: Portable Avatar Format (MPAI-PAF)* – in the following also called MPAI-PAF – is a first response to the requirements of new usage contexts. MPAI-PAF specifies a standard for Portable Avatar Format (PAF) enabling a receiving party to render a digital human as intended by the sending party.

MPAI-PAF has been developed by MPAI – Moving Picture, Audio, and Data Coding by Artificial Intelligence [1], the international, unaffiliated, non-profit organisation developing standards for Artificial Intelligence (AI)-based data coding with clear Intellectual Property Rights licensing frameworks in compliance with the rigorous MPAI Process [2] in pursuit of the following policies:

1. Be friendly to the AI context but, to the extent possible, agnostic to the technology – AI or Data Processing – used in an implementation.
2. Be attractive to different industries, end users, and regulators.
3. Address three levels of standardisation: data types, components (called AI Modules), configurations of components (called AI Workflows) all exposing standard interfaces with an aggregation level decided by the implementer.
4. Specify the data exchanged by components with a clear semantic to the extent possible.

As manager of the MPAI Ecosystem specified by Governance of MPAI Ecosystem (MPAI-GME) [3], MPAI ensures that a user can:

1. Operate the reference implementation of the Technical Specification, by providing a Reference Software Specification with annexed software.
2. Test the conformance of an implementation with the Technical Specification, by providing Conformance Testing Specification.
3. Assess the performance of an implementation of a Technical Specification, by providing the Performance Assessment Specification.
4. Get conforming implementations possibly with a performance assessment report from a trusted source through the MPAI Store.

Technical Specification: AI Framework (MPAI-AIF) V2 [4] enables an effective implementation of some of the policies. Figure 1 depicts the Reference Model.

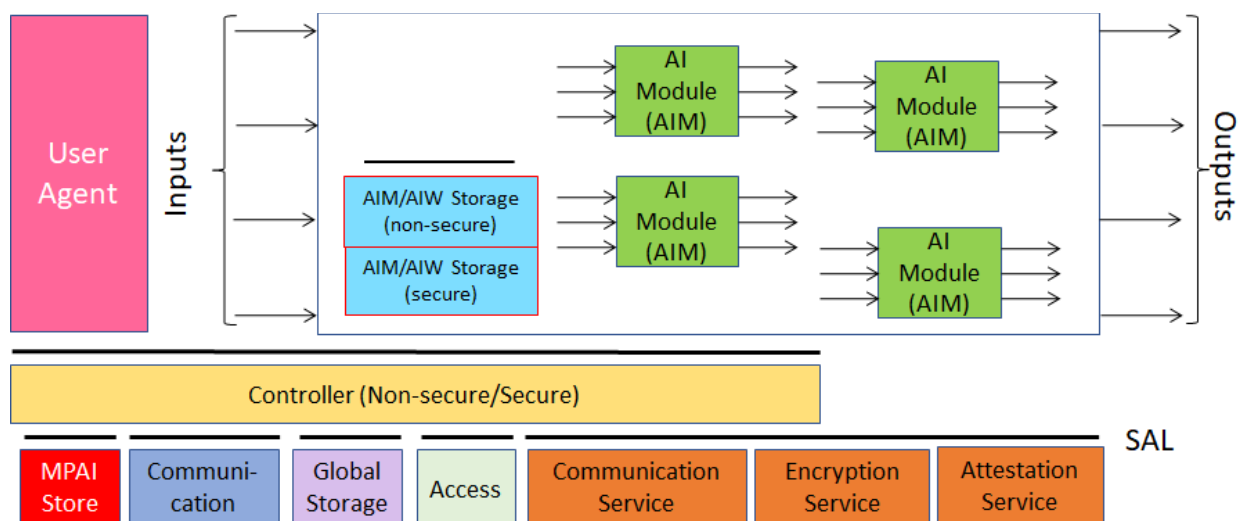


Figure 1 - The AI Framework (MPAI-AIF) V2 Reference Model

MPAI-AIF specifies an environment called AI Framework (AIF) enabling the secure execution of AI Workflows (AIW) that can be constituted by AI Modules (AIM). Thus, users can have machines whose internal operation they understand to some degree, rather than machines that are just “black boxes” resulting from unknown training with unknown data and component developers can provide components with standard interfaces that can have improved performance compared to other implementations.

An AIW and its AIMs may have 3 interoperability levels:

*Level 1* – Implementer-specific and satisfying the MPAI-AIF Standard.

*Level 2* – Specified by an MPAI Application Standard.

*Level 3* – Specified by an MPAI Application Standard and certified by a Performance Assessor.

MPAI offers Users access to the promised benefits of AI with a guarantee of increased transparency, trust and reliability as the Interoperability Level of an Implementation moves from 1 to 3.

AI Modules can execute data processing or Artificial Intelligence algorithms and can be implemented in hardware, software, or hybrid hardware/software.

The MPAI-PAF Technical Specification can be implemented in one of the following modalities:

1. As a specific AIW implementing a Use Case, as specified in this document.
2. As a specific AIM, as specified in this document.
3. As a specific data type, as specified in this document.

However, MPAI does not mandate the choice of modality, which remains the sole decision of the implementer.

The MPAI-PAF Technical Specification will be accompanied by the Reference Software, Conformance Testing, and Performance Assessment Specifications. Conformance Testing specifies methods enabling users to ascertain whether a data type generated by an AIM, an AIM, or an AIW conform with this Technical Specification.

The MPAI-PAF Technical Specification applies the technologies to the Avatar-Based Videoconference (PAF-ABV) Use Case where:

1. **Transmitting clients** send PAFs containing:
  - 1.1 At the beginning: Avatar Models and Language Preferences.
  - 1.2 Continuously: Avatar Descriptors, and Speech and Face Objects to a Server.
2. **Server:**
  - 2.1 At the beginning:
    - 2.1.1 Selects an Environment, i.e., a meeting room and equips it with objects, i.e., meeting table and chairs.
    - 2.1.2 Places Avatar Models around the table.
    - 2.1.3 Distributes a PAF containing Environment, Avatar Models, and their positions to receiving clients.
  - 2.2 Continuously sends to receiving clients:
    - 2.2.1 Translated Speech from participants according to Language Preferences.
    - 2.2.2 Sends PAFs containing Avatar Descriptors and translated Speech.
3. **Receiving clients:**
  - 3.1 At the beginning: receive Environment and PAFs containing Avatar Models and Language Preferences
  - 3.2 Continuously:
    - 3.2.1 Receive PAFs containing Avatar Descriptors and translated Speech.

3.2.2 Create Audio and Visual Scene Descriptors.

3.2.3 Render the Audio-Visual Scene as seen from the human-selected Point of View.

MPAI-PAF utilises MPAI technologies specified by other MPAI standards to specify the Avatar-Based Videoconference Use Case. Similarly, other MPAI standards utilise standard MPAI-PAF technologies in other Use Cases such as Human-Connected Autonomous Vehicle (CAV) Interaction (CAV-HCI).

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

## 2 Scope

Technical Specification: Portable Avatar Format (MPAI-PAF) specifies:

1. The Portable Avatar Format and related Data Formats allowing a sender to enable a receiver to decode and render an Avatar as intended by the sender.
2. The Personal Status Display Composite AI Module allowing the conversion of a Text and a Personal Status to a Portable Avatar.
3. The AI Framework (MPAI-AIF)-conforming AI Workflows and AI Modules [4] composing the Avatar-Based Videoconference Use Case also using Data Types from other MPAI Technical Specifications.

The Use Case normatively defines:

1. The Functions of the AIWs and of the AIMs.
2. The Connections between and among the AIMs
3. The Semantics and the Formats of the input and output data of the AIW and the AIMs, in particular Avatar Descriptors.

The word *normatively* implies that an Implementation claiming Conformance to:

1. An *AIW*, shall:
  - a. Perform the function specified in the appropriate Section of Chapter 5.
  - b. Use AIMs connected with the topology and connections conform with the AIW Architecture specified in Chapter 5.
  - c. Have input and output data with the formats specified in Chapter 7.
2. An *AIM*, shall:
  - a. Perform the AIM function specified by the appropriate section of Chapter 5.
  - b. Be composed of AIMs as specified in Chapter 6.
  - c. Receive and produce the data specified in Chapter 7.
3. A data *Format*, the data shall have the format specified in Chapter 7.

Users of this Technical Specification should note that:

1. This Technical Specification defines Interoperability Levels but does not mandate any.
2. Implementers decide the Interoperability Level their Implementation satisfies.
3. Implementers can use the Reference Software of this Technical Specification to develop their Implementations.
4. The Conformance Testing specification can be used to test the conformity of an Implementation to this Standard.
5. Performance Assessors can assess the level of Performance of an Implementation based on the Performance Assessment associated with the MPAI-PAF Technical Specification.
6. Implementers and Users should consider the notices and disclaimers of Annex 2.

The current version of the Standard has been developed by the Portable Avatar Format (PAF) of the Requirements Standing Committee. MPAAI may issue new versions of MPAAI-PAF that extend or replace the current Technical Specification.

### 3 Terms and Definitions

In this document, the following conventions apply:

1. Words beginning with a capital letter are defined in *Table 1*.
2. Words beginning with a small letter have the normal meaning consistent with the relevant context.

In *Table 1*, a dash “-” preceding a Term means the following:

1. If the font of the Term is normal, the Term in the table without a dash and preceding the one with a dash should be placed before that Term. The notation is used to concentrate in one place all the Terms that are composed of, e.g., the word Audio followed by one of the words Object, Scene, and Scene Descriptors.
2. If the font of the Term 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 Attitude preceded by one of the words Social or Spatial.

*Table 1 – Terms and Definitions*

Term	Definition
Attitude	
- <i>Social</i>	A Factor of the Personal Status related to the way a human or Avatar intends to position vis-à-vis the Environment or subsets of it, e.g., “Respectful”, “Confrontational”, “Soothing”.
- <i>Spatial</i>	Position and Orientation and their velocities and accelerations of a Human and Physical Object in a Digital Environment.
Audio	Digital representation of an analogue audio signal sampled at a frequency between 8-192 kHz with a number of bits/sample between 8 and 32, and non-linear and linear quantisation.
Authentication	The process of determining whether a device or a human is what it states it is.
Avatar	A rendered Digital Human.
Cognitive State	An element of the internal status reflecting the way a human or avatar understands the Environment, such as “Confused”, “Dubious”, “Convinced”.
Data	Information in digital form.
- Format	The syntax and semantics of a Data Type.
- Type	A particular type of Data.
Descriptor	Coded representation of text, audio, speech, or visual feature.
Device	A piece of equipment used to interact and have Experience in a Digital Environment.
Emotion	The coded representation of the internal state resulting from the interaction of a human or avatar with the Environment or subsets of it, such as “Angry”, “Sad”, “Determined”.
Environment	
- Model	The static audio and visual components of the Environment, e.g., walls,

	table, and chairs.
- <i>Virtual</i>	An Environment in a Digital Space.
- <i>Real</i>	An Environment in a Physical Space.
Experience	The state of a human whose senses are continuously affected for a meaningful period.
Face	A digital representation of a human face.
Factor	One of Emotion, Cognitive State, and Spatial Attitude.
Gesture	A movement of a Digital Human or part of it, such as the head, arm, hand, and finger, often a complement to a vocal utterance.
Grade	The intensity of a Factor.
Human	
- <i>Digital</i>	A Digitised or a Virtual Human.
- <i>Digitised</i>	An Object that has the appearance of a specific human when rendered.
- <i>Virtual</i>	An Object created by a computer that has a human appearance when rendered but is not a Digitised Human.
Meaning	Information extracted from Text such as syntactic and semantic information, and Personal Status.
Modality	One of Text, Speech, Face, or Gesture.
Object	A data structure that can be rendered to cause an Experience.
- <i>Audio</i>	Coded representation of Audio information with its metadata. An Audio Object can include other Audio Objects.
- <i>Audio-Visual</i>	Coded representation of Audio-Visual information with its metadata. An Audio-Visual Object can include other Audio-Visual Objects.
- <i>Descriptor</i>	The Digital Representation of a feature of an Object in a Scene, including its Spatial Attitude.
- <i>Digital</i>	A Digitised or a Virtual Object.
- <i>Digitised</i>	The digital representation of a real object.
- <i>Visual</i>	Coded representation of Visual information with its metadata. A Video Object can include other Video Objects.
- <i>Virtual</i>	An Object not representing an object in a Real Environment.
Orientation	The set of the 3 roll, pitch, yaw angles indicating the rotation around the principal axis (x) of an Object, its y axis having an angle of 90° counterclockwise (right-to-left) with the x axis and its z axis pointing up toward the viewer.
Persona	A manifestation of a human as a rendered Digital Human.
Personal Status	The ensemble of information internal to a person, including Emotion, Cognitive State, and Attitude.
Point of View	The Spatial Attitude of a Digital Human watching the Environment.
Position	The 3 coordinates of a representative point for an object in a Real or Virtual space with respect to a set of coordinate axes (x,y,z).
Scene	A Digital Environment populated by Objects.
- <i>Audio</i>	The Audio Objects of an Environment with Object metadata such as Spatial Attitude.
- <i>Audio-Visual</i>	(AV Scene) The Audio-Visual Objects of an Environment Object metadata such as Spatial Attitude.
- <i>Visual</i>	The Visual Objects of an Environment with Object metadata such as Spatial Attitude.
- <i>Presentation</i>	The rendering of a Scene in a format suitable for human perception.
Text	A sequence of characters drawn from a finite alphabet.



Representation	Data that digitally represents an entity of a Real Environment.
----------------	---

## 4 References

### 4.1 Normative References

Technical Specification: Portable Avatar Format (MPAI-PAF) normatively references the following documents, both from MPAI and other standards organisations. Referenced MPAI standards are publicly available at the URL indicated in the reference.

1. MPAI; The MPAI Statutes; <https://mpai.community/statutes/>.
2. MPAI; The MPAI Patent Policy; <https://mpai.community/about/the-mpai-patent-policy/>.
3. MPAI; Technical Specification: The governance of the MPAI ecosystem (MPAI-GME), V1.1; <https://mpai.community/standards/mpai-gme/>
4. MPAI; Technical Specification; AI Framework (MPAI-AIF) V2; <https://mpai.community/standards/mpai-aif/>
5. MPAI; Technical Specification: Context-based Audio Enhancement (MPAI-CAE) V2; <https://mpai.community/standards/mpai-cae/>
6. MPAI; Technical Specification; Multimodal Conversation (MPAI-MMC) V2; <https://mpai.community/standards/mpai-mmc/>
7. MPAI; Technical Specification; Object and Scene Description (MPAI-OSD) V1; <https://mpai.community/standards/mpai-osd/>
8. Khronos; Graphics Language Transmission Format (glTF); October 2021; <https://registry.khronos.org/glTF/specs/2.0/glTF-2.0.html>
9. ISO/IEC 19774-1:2019 Information technology – Computer graphics, image processing and environmental data representation – Part 1: Humanoid animation (HAnim) architecture; see <https://www.web3d.org/documents/specifications/19774-1/V2.0/index.html>
10. ISO/IEC 19774-2:2019 Information technology – Computer graphics, image processing and environmental data representation – Part 2: Humanoid animation (HAnim) motion data animation; <https://www.web3d.org/documents/specifications/19774/V2.0/MotionDataAnimation/MotionDataAnimation.html>
11. ISO 639; Codes for the Representation of Names of Languages — Part 1: Alpha-2 Code.
12. ISO/IEC 10646; Information technology – Universal Coded Character Set

### 4.2 Informative References

These references are provided for information purposes.

13. Ekman, Paul (1999), "Basic Emotions", in Dalglish, T; Power, M (eds.), Handbook of Cognition and Emotion (PDF), Sussex, UK: John Wiley & Sons.

## 5 Avatar-Based Videoconference

### 5.1 Introduction

The Portable Avatar Format enables an implementation of the Avatar-Based Videoconference Use Case. Table 2 lists all Data Types required by the Portable Avatar Data Type and MPAI Technical Specifications supporting them. Chapter 7 provides the full specification including references of the MPAI-PAF Data Formats.

Table 2 - Data Types of the Portable Avatar Format

Section	Data Type	Technical Specification
7.1	Portable Avatar Format	MPAI-PAF
7.2.2	Visual Environment	MPAI-PAF
7.3.1	Body Model	MPAI-PAF
7.3.2	Body Descriptors	MPAI-PAF
7.4.1	Face Model	MPAI-PAF
7.4.2	Face Descriptors	MPAI-PAF
7.5.1	Avatar Model	MPAI-PAF
7.5.2	Avatar Descriptors	MPAI-PAF
7.6.1	Spatial Attitude	MPAI-PAF
7.6.2	Audio	MPAI-CAE
7.6.3	Visual	MPAI-PAF
7.7.1	Text	MPAI-MMC
7.7.2	Language identifier	MPAI-MMC
7.7.3	Meaning	MPAI-MMC
7.7.4	Personal Status	MPAI-MMC

## 5.2 Scope of Use Case

The MPAI-PAF Avatar-Based Videoconference (PAF-ABV) Use Case enables a form of videoconference held in a Virtual Environment populated by Avatars representing humans showing their visual appearance and uttering their voices. Figure 2 depicts the system composed of four types of subsystems:

1. Transmitting Clients
2. Server
3. Virtual Secretary
4. Receiving Clients.

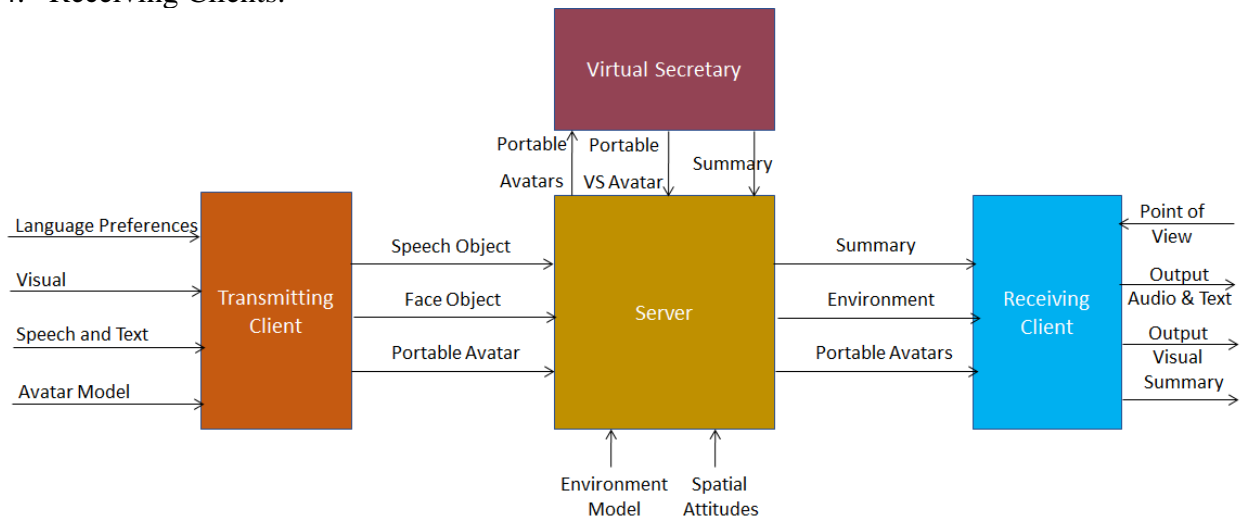


Figure 2 – Avatar-Based Videoconference end-to-end diagram

The components of the MPAI-ABV system:

1. **Participant:** is a human joining an ABV either individually or as a member of a group of humans in the same physical room.
2. **Environment:** is a virtual space equipped with Objects such as a table and an appropriate number of chairs.

3. **Portable Avatar:** represents a human participant represented in the Portable Avatar Format (PAF).
4. **Transmitting Client:**
  - 4.1. At the beginning of the conference,
    - 4.1.1. Receives from Participants and sends to the Server Portable Avatars containing the Avatar Models and Language Preferences.
    - 4.1.2. Sends to the Server Speech Object and Face Object for Authentication.
  - 4.2. Continuously sends to the Server Portable Avatars containing Avatar Descriptors and Speech.
5. **The Server**
  - 5.1. At the beginning:
    - 5.1.1. Selects a Visual Environment Model, e.g., a meeting room.
    - 5.1.2. Equips the room with objects, i.e., meeting table and chairs.
    - 5.1.3. Places Avatar Models around the table with a given Spatial Attitude.
    - 5.1.4. Distributes Environment and Portable Avatars containing Avatars Models, and their Spatial Attitudes to all Receiving Clients.
    - 5.1.5. Authenticates Speech and Face Objects and assigns IDs to Avatars.
    - 5.1.6. Sets the common conference language.
  - 5.2. Continuously:
    - 5.2.1. Translates Speech to Participants according to their Language Preferences.
    - 5.2.2. Sends Portable Avatars containing Avatar Descriptors, Speech, and Spatial Attitude of Participants and Virtual Secretary to all Receiving Clients and Virtual Secretary.
6. **Virtual Secretary** is an Avatar not corresponding to any Participant that continuously:
  - 6.1. Uses the common meeting language.
  - 6.2. Understands Avatars' utterances and extracts their Personal Statuses.
  - 6.3. Drafts a Summary of its understanding of Avatars' Text and Personal Status.
  - 6.4. Displays the Summary either to:
    - 6.4.1. Outside of the Environment for Participants to read and edit directly, or
    - 6.4.2. The Visual Environment for Avatars to comment, e.g., via Text.
  - 6.5. Refines the Summary.
  - 6.6. Sends its Portable Avatar containing its Avatar Descriptors to the Server.
7. **Receiving Client:**
  - 7.1. At the beginning
  - 7.2. receives Visual Environment and Portable Avatars containing Avatar Models with Spatial Attitudes.
  - 7.3. Continuously:
    - 7.3.1. Receives Portable Avatars with Avatar Descriptors and Speech.
    - 7.3.2. Produces Visual and Audio Scene Descriptors.
    - 7.3.3. Renders the Audio-Visual Scene by spatially adding the participants' utterances to the Spatial Attitude of the respective Avatars' mouths. Rendering may be done from a Point of View, possibly different from the position assigned to their Avatars in the Visual Environment, selected by participant who use a device of their choice (HMD or 2D display/earpad).

## 5.3 Transmitting Client

### 5.3.1 Functions of Transmitting Client

The function of a Transmitting Client is to:

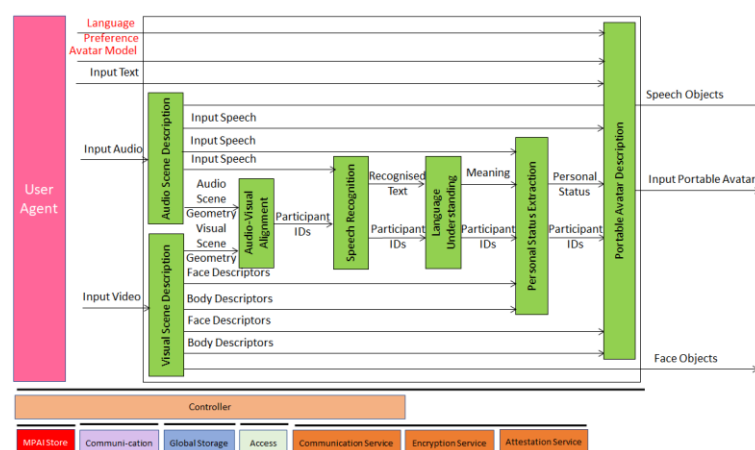
1. Receive from a Participant:
  - 1.1. Input Audio from the microphone.

- 1.2. Input Video from the camera.
- 1.3. Participant's Avatar Model.
- 1.4. Participant's language preferences (e.g., EN-US, IT-CH).
2. Send to the Server:
  - 2.1. Speech Object (for Authentication).
  - 2.2. Face Object (for Authentication).
  - 2.3. Input Portable Avatars containing:
    - 2.3.1. Language preferences (at the start).
    - 2.3.2. Avatar Model (at the start).
    - 2.3.3. Speech.
    - 2.3.4. Avatar Descriptors.

Figure 3 gives the architecture of Transmitting Client AIW. Red text refers to data sent at meeting start.

1. Language preferences
2. Avatar model.
3. Speech Object (for Authentication).
4. Face Object (for Authentication).

AIM	Data
Audio Scene Description	Audio Scene Descriptors.
Visual Scene Description	Visual Scene Descriptors.
Audio-Visual Alignment	Audio, Visual, and Audio-Visual Descriptor Identifiers.
Speech Recognition	Recognised Text.
Face Description	Face Descriptors.
Body Description	Body Descriptors.
Personal Status Extraction	Personal Status.
Language Understanding	Meaning.
Portable Avatar Description	Avatar Descriptors.



### 5.3.3 Input and output data of Transmitting Client

Table 3 gives the input and output data of the Transmitting Client AIW:

*Table 3 – Input and output data of Transmitting Client AIW*

<b>Input</b>	<b>Comments</b>
Input Text	Chat text used by a human to communicate with Virtual Secretary or other participants
Language Preference	The language participant wishes to speak and hear.
Input Audio	Audio of Speech of participants in a meeting room.
Input Video	Video of participants in a meeting room.
Avatar Model	The avatar model selected by the participant.
<b>Output</b>	<b>Comments</b>
Input Portable Avatar	Portable Avatar produced by Transmitting Client.
Speech Object	For authentication by Server.
Face Object	For authentication by Server.

### 5.3.4 Functions of Transmitting Client's AI Modules

Table 5 gives the functions of AI Modules of the Transmitting Client AIW.

*Table 4 – AI Modules of Transmitting Client AIW*

<b>AIM</b>	<b>Input</b>
<b>Audio Scene Description</b>	Provides Audio Objects and Audio Scene Geometry.
<b>Visual Scene Description</b>	Provides Visual Objects and Visual Scene Geometry.
<b>Audio-Visual Alignment</b>	Provides Identifiers to Audio, Visual, and Audio-Visual Objects.
<b>Speech Recognition</b>	Recognises the Speech.
<b>Language Understanding</b>	Extracts the Meaning of the Recognised Text.
<b>Personal Status Extraction</b>	Extracts the Personal Status of Meaning, Speech, Face Descriptors, and Body Descriptors.
<b>Portable Avatar Description</b>	Provides the Portable Avatars of the Participants in a given location.

### 5.3.5 I/O Data of Transmitting Client's AI Modules

Table 5 gives the AI Modules of Transmitting Client AIW.

*Table 5 – AI Modules of Transmitting Client AIW*

<b>AIM</b>	<b>Input</b>	<b>Output</b>
<b>Audio Scene Description</b>	Input Audio	Audio Scene Geometry Input Speech Speech Objects
<b>Visual Scene Description</b>	Input Video	Visual Scene Geometry Face Descriptors Body Descriptors Face Objects
<b>Audio-Visual Alignments</b>	Audio Scene Geometry Visual Scene Geometry	Participant IDs
<b>Speech Recognition</b>	Speech Objects	Recognised Text

		Participant ID
<b>Language Understanding</b>	Recognised Text	Meaning Participant ID
<b>Personal Status Extraction</b>	Meaning Speech Face Object Human Object	Personal Status Participant ID
<b>Portable Avatar Description</b>	Language Preference Avatar Model Input Text Input Speech Personal Status Participant ID	Portable Avatars.

### 5.3.6 JSON Metadata of Transmitting Client

Specified in Annex 5 - AIW Metadata Chapter 1.

## 5.4 Server

### 5.4.1 Functions of Server

The Server:

1. *At the start:*
  - 1.1. Receives Speech Object and Speech Objects of each Participant.
  - 1.2. Authenticates Participants.
  - 1.3. Receives Portable Avatars each containing Language Preference and Avatar Model.
  - 1.4. Selects a Visual Environment.
  - 1.5. Selects the Spatial Attitudes of Avatar Models.
  - 1.6. Selects the common meeting language.
  - 1.7. Distributes all Portable Avatars each containing: Visual Environment, Language Preference, Avatar Model, and Spatial Attitude.
2. *During the videoconference:*
  - 2.1. Receives Participants' and Virtual Secretary's Avatar Descriptors.
  - 2.2. Translates participants' Speech according to their language preferences.
  - 2.3. Sends Portable Avatars containing Avatar ID, Text, Speech translated to the common meeting language, Face Descriptors and Gesture Descriptors to Virtual Secretary.
  - 2.4. Receives Virtual Secretary's Portable Avatar containing Avatar ID, Text, Speech in the common meeting language, Face Descriptors and Gesture Descriptors.
  - 2.5. Translates Virtual Secretary's Speech according to each participant's language preferences.
  - 2.6. Sends Participants' and Virtual Secretary's Portable Avatars containing Avatar ID, Text, Translated Speech, Face Descriptors and Gesture Descriptors to Receiving Clients.

### 5.4.2 Reference Architecture of Server

Figure 5 gives the architecture of Server AIW. Red text refers to data sent at meeting start.

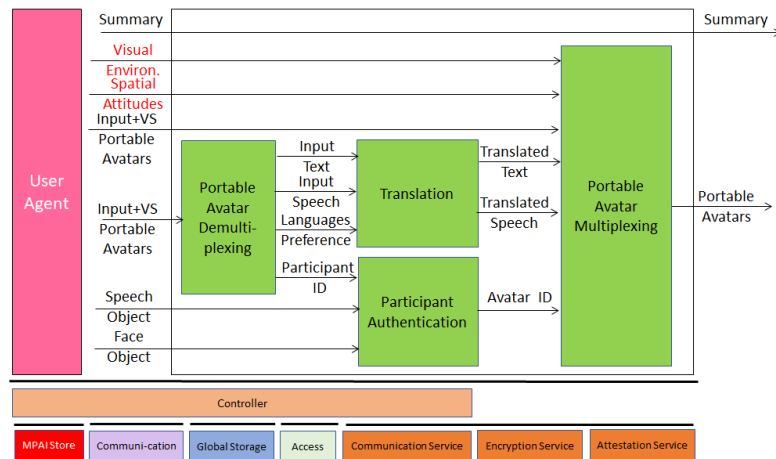


Figure 4 – Reference Model of Avatar-Based Videoconference Server

### 5.4.3 I/O Data of Server

Table 6 gives the input and output data of Server AIW.

Table 6 – Input and output data of Server AIW

Input	Comments
Summary	From Virtual Secretary.
Visual Environment Model	Set by Server.
Spatial Attitudes	Set by Server.
Input+VS Portable Avatars	From Transmitting Clients and Virtual Secretary
Speech Objects	Participants' Speech Object for Authentication.
Face Objects	Participants' Face Object for Authentication.
Outputs	Comments
Summary	As above.
Portable Avatars	As re-multiplexed by Server.

### 5.4.4 Functions of Server AI Modules

Table 7 gives the functions of the AI Modules of the Server AIW.

Table 7 - AI Modules of Server AIW

AIM	Functions
<b>Portable Avatar Demultiplexing</b>	Makes available Input Text, Input Speech, Language Preferences and Participant ID from Input and VS Portable Avatars.
<b>Translation</b>	Translates active Speech and Text of all Participants to the Selected Languages.
<b>Participant Authentication</b>	Authenticates Participants using Speech and Face Objects. Connects Participant ID to Avatar ID.
<b>Portable Avatar Multiplexing</b>	Multiplexes components of Portable Avatars for transmission to Receiving Clients.

### 5.4.5 I/O Data of Server AI Modules

Table 8 gives the Input/Output Data of the AI Modules of the Server AIW.

Table 8 - AI Modules of Server AIW

AIM	Input	Output
<b>Portable Avatar Demultiplexing</b>	Input and VS Portable Avatars	Input Text Input Speech Language Preferences Participant ID
<b>Translation</b>	Language Preferences Text Speech	Translates Text Translated Speech
<b>Participant Authentication</b>	Speech Descriptors Face Descriptors	Avatar ID
<b>Portable Avatar Multiplexing</b>	Visual Environment Spatial Attitudes Avatar Descriptors Translated Text Translated Speech Avatar ID	Portable Avatars

#### 5.4.6 JSON Metadata of Server

Specified in Annex 5 - AIW Metadata Chapter 2.

### 5.5 Virtual Secretary (informative)

Virtual Secretary is normatively specified by *Technical Specification: Multimodal Conversation (MPAI-MMC)* [6]. Here only the Scope, Reference Model, and Input/Output Data are reported for information.

#### 5.5.1 Functions of Virtual Secretary

The functions of the Virtual Secretary are to:

1. Listen to each Avatar's Speech.
2. Demultiplex Avatar ID, and Face and Body Descriptors from Input Portable Avatars.
3. Extract Personal Status.
4. Draft a Summary using text in the meeting common language and graphics symbols representing the Personal Status.

The Summary can be handled in two different ways:

1. Transferred to an external application so that Participants can edit the Summary.
2. Displayed to avatars:
  - 2.1. Avatars make Speech or Text comments (e.g., via chat).
  - 2.2. The Virtual Secretary refines the Summary by interpreting Speech, Text, and the Avatars' Personal Statuses.

#### 5.5.2 Reference Architecture

Figure 5 depicts the architecture of the Virtual Secretary. Data labelled in red refers to data sent only once at meeting start. Summary and Edited Summary are data back and forth from Summarisation to Dialogue Processing to Summarisation. Summary is continuously sent in an updated form to Dialogue Processing which returns it updated by Avatars' comments in the form of Edited Summary.



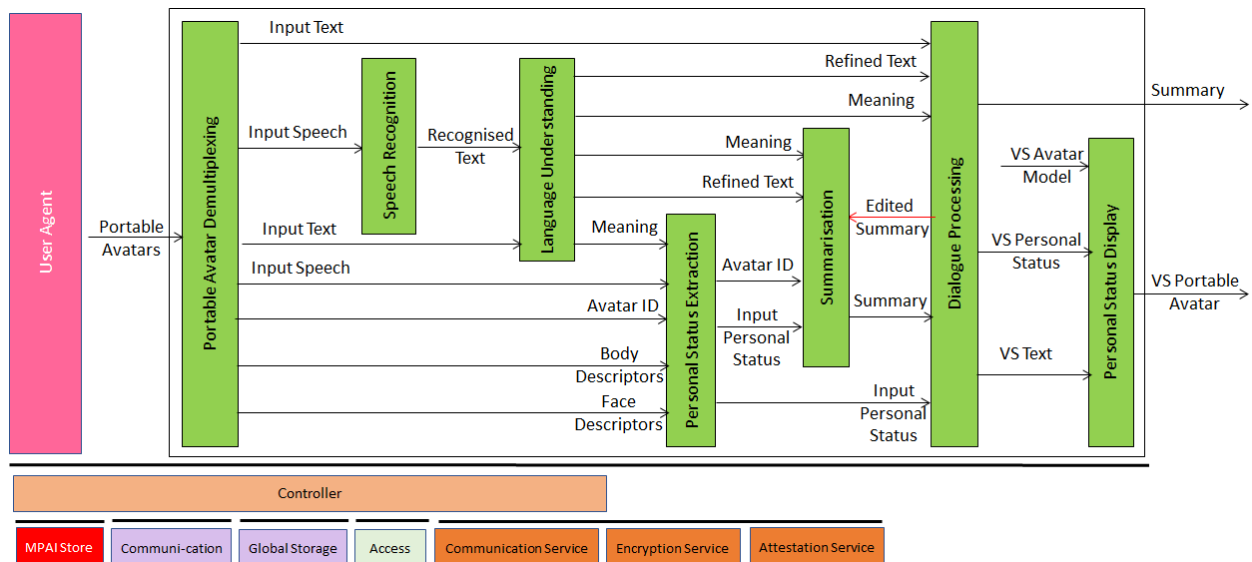


Figure 5 – Reference Model of Virtual Secretary

The Virtual Secretary workflow operates as follows:

1. Input Avatar Demultiplexing produces: Input Text, Input Speech, Avatar ID, Body Descriptors, and Face Descriptors.
2. Speech Recognition produces Recognised Text from Input Speech.
3. Language Understanding produces Refined Text and Meaning from Recognised Text.
4. Personal Status Extraction produces Input Personal Status from Meaning, Speech, Face Descriptors, and Body Descriptors.
5. Summarisation produces Summary, i.e., Text in the meeting's common language and graphical symbols, from Refined Text, Personal Status, Meaning, and from Edited Summary received from Dialogue Processing.
6. Dialogue Processing
  - 6.1. Sends Summary to external application.
  - 6.2. Sends Edited Summary produced from Refined Text (from Speech), Avatar's Text (from chat), Meaning, and Summary back to Summarisation.
  - 6.3. Produces VS Text and VS Personal Status.
7. Personal Status Display produces VS Portable Avatar containing VS Avatar Model, VS Text, VS Speech, and VS Avatar Descriptors.

### 5.5.3 I/O Data of Virtual Secretary

Table 7 gives the Input and Output Data of Virtual Secretary.

Table 9 - I/O data of Virtual Secretary

Input data	From	Comment
Input Portable Avatars	Server	Contain: <ol style="list-style-type: none"> <li>1. Spatial Attitudes</li> <li>2. Avatar Descriptors</li> <li>3. Translated Text</li> <li>4. Translated Speech</li> <li>5. Avatar ID</li> </ol>
Output data	To	Comments
Summary	Server	Summary of Avatars' interventions
VS Avatar Model	Application	The

VS Speech	Avatars	Speech to avatars
VS Text	Avatars	Response to chat.
VS Portable Avatar	Server	Contains: 1. At start: VS Avatar Model 2. Subsequently: 1. VS Speech 2. VS Text 3. VS Avatar Descriptors

### 5.5.4 JSON Metadata of Virtual Secretary

Specified in Annex 5 - AIW Metadata Chapter 3.

## 5.6 Receiving Client

### 5.6.1 Functions of Receiving Client

The Function of the Receiving Client is to:

1. Create the Environment using the Environment Model.
2. Place and animate the Avatar Models with their Spatial Attitudes.
3. Add Speech to Avatar's mouth.
4. Render the Audio-Visual Scene as seen from the Participant-selected Point of View.

### 5.6.2 Reference Architecture of Receiving Client

The Receiving Client:

#### 1. At the start

1.1. Receives the Visual Environment and the Portable Avatars containing:

1.1.1. The Visual Environment.

1.1.2. The Avatar Models

1.1.3. The Spatial Attitudes

1.2. Creates the initial AV Scene.

#### 2. During the Videoconference:

2.1. Receives the Avatar Models containing:

2.1.1. Speech

2.1.2. Body Descriptors

2.1.3. Face Descriptors

2.2. Creates the running AV Scene using each Avatar's:

2.3. The Body and Face Descriptors.

2.4. The Speech.

3. Renders the Audio-Visual Scene based on the selected Point of View.

Figure 6 gives the architecture of Receiving Client AIW. Red text for data received at the start.

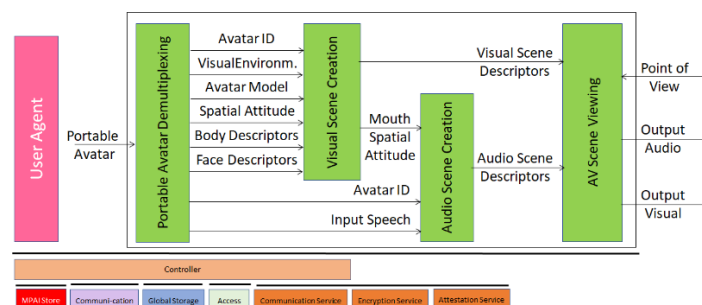


Figure 6 – Reference Model of Avatar-Based Videoconference Receiving Client

Notes:

1. An implementation may decide to display text with a visual image for accessibility purposes.
2. Audio Environment is added for completeness. However, This Standard does not provide a specification for it.

### 5.6.3 I/O Data of Receiving Client

Table 10 gives the input and output data of Receiving Client.

*Table 10 – Input and output data of Receiving Client AIW*

<b>Input</b>	<b>Comments</b>
Point of View	Participant-selected point of view to see visual objects and hear audio objects in the Virtual Environment.
Portable Avatars	Portable Avatars from Server.
<b>Output</b>	<b>Comments</b>
Output Audio	Presented using loudspeaker (array)/earphones.
Output Visual	Presented using 2D or 3D display.

### 5.6.4 Functions of Receiving Client's AI Modules

Table 11 gives the AI Modules of Receiving Client AIW.

*Table 11 – Functions of AI Receiving Clients' AI Modules*

<b>AIM</b>	<b>Input</b>
<b>Portable Avatar Demultiplexing</b>	Extracts Avatar ID, Visual Environment, Avatar Model, Spatial Attitude, Body Descriptors, Face Descriptors and Input Speech from Portable Avatars.
<b>Visual Scene Creation</b>	Creates the Visual Scene providing the Spatial Attitudes of the mouths of all Avatars.
<b>Audio Scene Creation</b>	Creates the Audio Scene.
<b>AV Scene Viewing</b>	Provides a ready-to-rendered AV Scene.

### 5.6.5 I/O Data of Receiving Client's AI Modules

Table 12 gives the AI Modules of Receiving Client AIW.

*Table 12 – I/O Data of Receiving Clients' AI Modules*

<b>AIM</b>	<b>Input</b>	<b>Output</b>
<b>Portable Avatar Demultiplexing</b>	Portable Avatars	Avatar ID Visual Environment Avatar Model Spatial Attitude Body Descriptors Face Descriptors Input Speech
<b>Visual Scene Creation</b>	Avatar ID Visual Environment Avatar Model Spatial Attitude Body Descriptors	Visual Scene Descriptors Mouth Spatial Attitudes

	Face Descriptors	
<b>Audio Scene Creation</b>	Avatar ID Input Speech Mouth Spatial Attitudes	Audio Scene Descriptors
<b>AV Scene Viewing</b>	Audio Scene Descriptors Visual Scene Descriptors Point of View	Output Audio Output Video

### 5.6.6 JSON Metadata of Receiving Client

Specified in Annex 5 - AIW Metadata Chapter 4.

## 6 Composite AI Modules

Some MPAI Use Cases need combinations of AI Modules called Composite AI Modules. This Chapter specifies the Personal Status Display Composite AIM using a format like the one adopted for Uses Cases.

### 6.1 Personal Status Extraction (PSE) (Informative)

The Personal Status Extraction Composite AIM is normatively specified by *Technical Specification: Multimodal Conversation (MPAI-MMC)* [6]. Here only the Scope, Reference Model, and Input/Output Data are reported for information.

#### 6.1.1 Scope of Composite AIM

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

#### 6.1.2 Reference architecture

Personal Status Extraction produces the estimate of the Personal Status of a human or an avatar by analysing each Modality in three steps:

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

Figure 7 depicts the Personal Status estimation process:

1. *Descriptors are extracted* from Text, Speech, Face Object, and Body Object. An AI Module upstream can provide Descriptors, depending on the value of the Input Selections informing PSE whether a Modality or its Descriptors are used.
2. *Descriptors are interpreted* and the specific indicators of the Personal Status in the Text, Speech, Face, and Gesture Modalities are derived.
3. *Personal Status is obtained* by combining the estimates of different Modalities of the Personal Status.

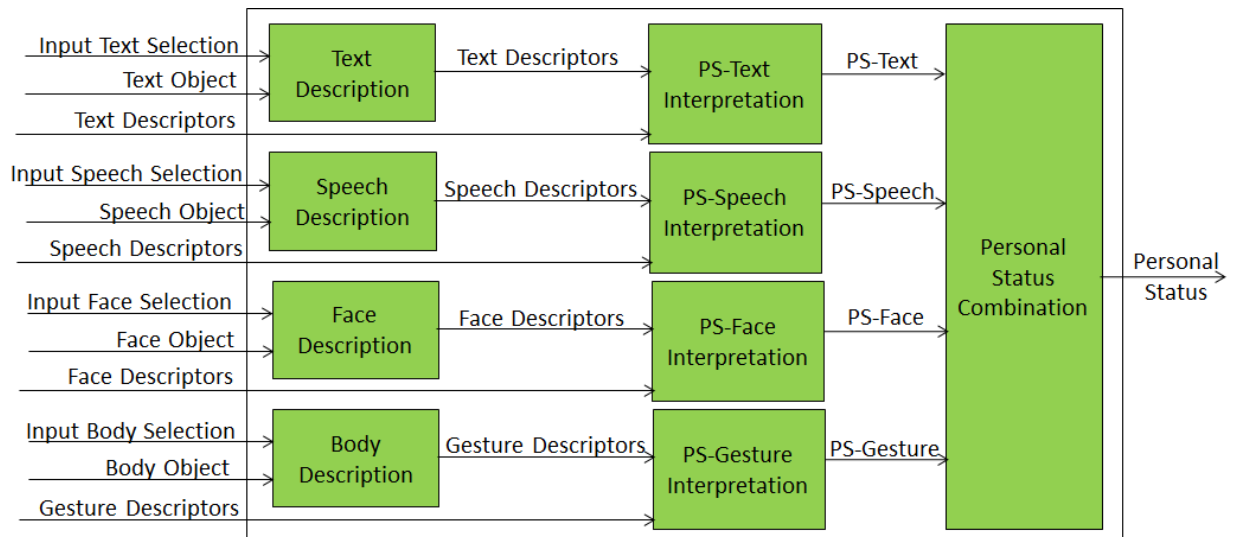


Figure 7 – Reference Model of Personal Status Extraction

Figure 7 represents the possibility that PSE receives some Modality Descriptors as input, thus bypassing the Modality (Text, Speech, etc.) Description AIM.

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

### 6.1.3 I/O Data of Personal Status Extraction

Table 13 gives the input/output data of Personal Status Extraction.

Table 13 – I/O data of Personal Status Extraction

Input data	From	Comment
Selection	An external signal	
Text	Keyboard or Speech Recognition	Text or recognised speech.
Text Descriptors	An upstream AIM	Depending on Selection
Speech	Microphone	Speech of human.
Speech Descriptors	An upstream AIM	Depending on Selection
Face Object	Visual Scene Description	The face of the human.
Face Descriptors	An upstream AIM	Depending on Selection
Body Object	Visual Scene Description	The upper part of the body.
Body Descriptors	An upstream AIM	Depending on Selection
Output data	To	Comments
Personal Status	A downstream AIM	For further processing

## 6.2 Personal Status Display (PSD)

### 6.2.1 Scope of Composite AIM

Personal Status Display (PSD) is a Composite AIM receiving Text and Personal Status and generating the Face and Body Descriptors of an Avatar producing Text and uttering Speech with the intended Personal Status while the avatar's Face and Gesture show the intended Personal Status. The Personal Status driving the avatar can be extracted from a human or can be synthetically generated by a machine as a result of its conversation with a human or another avatar. Several Use

Cases from MPAI-MMC and other MPAI standards use this Composite AIM as a replacement for the combination of the AIMs depicted in Figure 8.

### 6.2.2 Reference Architecture

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

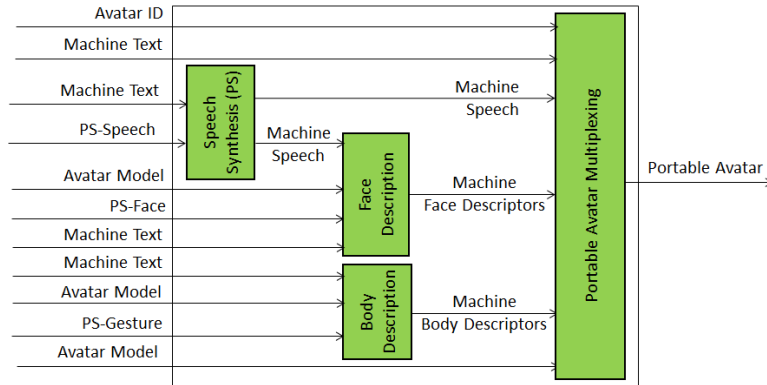


Figure 8 – Reference Model of Personal Status Display

The Personal Status Display operates as follows:

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

### 6.2.3 I/O Data of Personal Status Display

Table 14 gives the Input/Output Data of Personal Status Display.

Table 14 – I/O Data of Personal Status Display

Input data	From	Comment
Avatar ID	Upstream AIM	Portable Avatar's ID
Avatar Model	From upstream AIM or embedded	Part of Portable Avatar
Text Object	Keyboard or upstream AIM	Texts accompanying Portable Avatar
PS-Speech	Personal Status Extraction or Machine	To synthesise Speech
PS-Face	Personal Status Extractor or Machine	To describe Face
PS-Gesture	Personal Status Extractor or Machine	To describe Body
Output data	To	Comments
Portable Avatar	Downstream AIM or renderer	As Portable Avatar

### 6.2.4 Functions of AI Modules of Personal Status Display

Table 15 gives the functions of the AIMs.

Table 15 – Functions of AI Modules of Personal Status Extraction

AIM	Functions
Speech Synthesis (PS)	Synthesises Text with Personal Status.
Face Description	Produces the Machine Face Descriptors with Personal Status.

<b>Body Description</b>	Produces the Machine Body Descriptors with Personal Status.
<b>Portable Avatar Multiplexing</b>	Multiplexes Data into Portable Avatar with Personal Status.

### 6.2.5 I/O Data of AI Modules of Personal Status Display

Table 16 gives the list of AIMs with their functions.

*Table 16 – AI Modules of Personal Status Extraction*

AIM	Receives	Produces
<b>Speech Synthesis (PS)</b>	Text PS-Speech	Machine Speech
<b>Face Description</b>	Avatar Model Machine Speech and PS-Face	Machine Face Descriptors
<b>Gesture Description</b>	Avatar Model Text and Machine PS-Gesture	Machine Body Descriptors
<b>Portable Avatar Multiplexing</b>	Avatar ID Machine Text Machine Speech Machine Body Descriptors Machine Face Descriptors Machine Avatar Avatar Model	Portable Avatar

### 6.2.6 JSON Metadata of Personal Status Display

Specified in Annex 5 - AIW Metadata Chapter 4.2.

## 7 Data Formats

This Technical Specification specifies the Formats of the Data Types listed in Table 17. The first column gives the name of the Data Type, the second the subsection where the Format of the Data is specified and the third the Use Cases making use of it.

*Table 17 – Data Formats*

Note: Entries in bold indicate Data Format categories.

Name of Data Type	Subsection	Use Case
<b>Portable Avatar Format</b>	7.1	PAF-CTX
		PAF-SRV
		PAF-CRX
		CAV-HCI
		MMC-VSV
<b>Environment</b>		PAF-ABV
Audio Environment	7.2.1	PAF-ABV
Visual Environment	7.2.2	PAF-ABV
<b>Body</b>		
Body Model	7.3.1	PAF-ABV
		CAV-HCI

		MMC-VSV
Body Descriptors	7.3.2	PAF-ABV
		CAV-HCI
		MMC-VSV
<b>Face</b>		
Face Model	7.4.1	PAF-ABV
		CAV-HCI
		MMC-VSV
Face Descriptors	7.4.2	PAF-ABV
		CAV-HCI
		MMC-VSV
<b>Avatar</b>	7.5.1	
Avatar Model		PAF-ABV
		CAV-HCI
		MMC-VSV
<b>Object and Scene</b>		
Spatial Attitude	7.6.1	PAF-ABV
		CAV-ESS
		CAV-MAS
		MMC-CAS
Audio Scene Descriptors	7.6.2	PAF-ABV
		CAV-ESS
		CAV-MAS
		MMC-CAS
Visual Scene Descriptors	7.6.3	PAF-ABV
		CAV-ESS
		CAV-MAS
		MMC-CAS
<b>Conversation</b>		
Text	7.7.1	PAF-ABV
		MMC
Language identifier	7.7.2	PAF-ABV
		MMC
Meaning	7.7.3	PAF-ABV
		MMC
Personal Status	7.7.4	PAF-ABV
		MMC

## 7.1 Portable Avatar Format

Portable Avatar Format is a Data Type including:

*Table 18 – Variables composing the Portable Avatar Format*

Note: All elements in Table 18 are optional.

Variable name	Comments
<b>ID</b>	String
Type	0=Relative 1=Absolute



Value	Seconds from - 0000/00/00T00:00 (relative time) - 1970/01/01T00:00 (absolute time)
Visual Environment	Specified by MPAI-PAF
Spatial Attitude	Position and orientation in the global reference system defined by the Visual Environment. If Spatial Attitude and Visual Environment are missing, the Avatar may be placed in any Visual Environment with any Spatial Attitude.
Model	Specified by MPAI-PAF
BodyDescriptors	Specified by MPAI-PAF
FaceDescriptors	Specified by MPAI-PAF
LanguagePreference	Specified by MPAI-MMC V2
SpeechType	String identifies compression
Speech	Byte stream
Text	Specified by MPAI-MMC V2
PersonalStatus	Specified by MPAI-MMC V2

The JSON representation of Table 18 is given below.

```
{
  "$id": "https://schemas.mpai.community/PAF/V1.0/PortableAvatarFormat.json",
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "PortableAvatarFormat",
  "type": "object",
  "properties": {
    "ID": {
      "type": "string"
    },
    "Timestamp": {
      "type": "object",
      "properties": {
        "Type": {
          "type": "string"
        },
        "Value": {
          "type": "string",
          "oneOf": [
            {
              "format": "date-time"
            },
            {
              "const": "0"
            }
          ]
        }
      }
    },
    "required": [
      "Value"
    ],
    "if": {
      "properties": {
        "Value": {
          "const": "0"
        }
      }
    },
    "then": {
      "properties": {
        "Type": {
          "type": "null"
        }
      }
    },
    "else": {
      "required": [
```

```

        "Type"
      ]
    }
  },
  "Placement": {
    "type": "object",
    "properties": {
      "Spatial Attitude": {
        "type": "object",
        "properties": {
          "Position": {
            "type": "array",
            "contains": {
              "type": "number"
            },
            "minContains": 3,
            "maxContains": 3
          },
          "Orientation": {
            "type": "array",
            "contains": {
              "type": "number"
            },
            "minContains": 3,
            "maxContains": 3
          }
        }
      },
      "Visual Environment": {
        "$ref": "https://schemas.mpai.community/PAF/V1.0/data/VisualEnvironment.json"
      }
    }
  },
  "Visual": {
    "type": "object",
    "properties": {
      "Model": {
        "$ref": "https://schemas.mpai.community/PAF/V1.0/data/AvatarModel.json"
      },
      "BodyDescriptors": {
        "$ref": "https://schemas.mpai.community/PAF/V1.0/data/BodyDescriptors.json"
      },
      "FaceDescriptors": {
        "$ref": "https://schemas.mpai.community/PAF/V1.0/data/FaceDescriptors.json"
      }
    }
  },
  "Audio": {
    "type": "object",
    "properties": {
      "LanguagePreference": {
        "type": "string",
        "minLength": 2,
        "maxLength": 2
      },
      "Speech": {
        "type": "object",
        "properties": {
          "Encoding": {
            "enum": [
              "MP3",
              "AAC"
            ]
          },
          "Utterance": {
            "type": "array",
            "contains": {
              "type": "integer"
            }
          }
        }
      }
    }
  }
}

```

```

    },
    "Text": {
      "type": "string"
    },
    "PersonalStatus": {
      "$ref": "https://schemas.mpai.community/MMC/V2.0/data/PersonalStatus.json"
    }
  }
}

```

## 7.2 Environment

### 7.2.1 Audio Environment

Currently, MPAI does not specify the Audio Environment.

### 7.2.2 Visual Environment

The Visual Environment represents:

1. A bounded or unbounded space, e.g., a room, a public square surrounded by buildings, etc.
2. Any object (e.g., tables and chairs) with the exclusion of Avatars.

The Visual Environment defines the global coordinate system, is represented according to the glTF syntax, and is transmitted as a file at the beginning of the Avatar-Based Videoconference. The horizontal plane of the Visual Environment is defined by x,z plane and the y axis points upwards.

## 7.3 Body

### 7.3.1 Body Model

MPAI adopts the Humanoid animation (HAnim) architecture [9] that gives access to the joint and end-effector hierarchy of a human figure. This allows a model-independent animation of a skeleton and related skin vertices associated with joints and geometry/accessories/sensors of individual body segments and sites.

The actual structure of the HAnim architecture depends on the selected element of the Level Of Articulations (LOA) hierarchy: LOA 1, LOA 2, LOA 3, or LOA 4. All joints of an HAnim figure are represented as a tree hierarchy starting with the humanoid\_root joint. For an LOA 1 character, there are 18 joints and 18 segments in the hierarchy.

The bones of the body are described starting from position  $(x_0, y_0, z_0)$  of the root (neck or pelvis).

The orientation of a bone attached to the root is defined by  $(\alpha, \beta, \gamma)$  where  $\alpha$  is the angle of the bone with the x axis, and so on. The joint of a bone attached to the preceding bone has a position  $(x_1, y_1, z_1)$  determined by the angles  $(\alpha_1, \beta_1, \gamma_1)$  and the length of the bone.

The Body Model contains:

1. Pose composed by:
  - 1.1. The position of the root.
  - 1.2. The angles of the bones with the  $(x, y, z)$  coordinate axes.
  - 1.3. The orientation of the body defined by 3 angles.
2. The standard bone lengths.
3. Lengths of the bones of the specific model.
4. Surface-related
  - 4.1. Surface
  - 4.2. Texture
  - 4.3. Material
  - 4.4. Cloth (an integral part of the model).

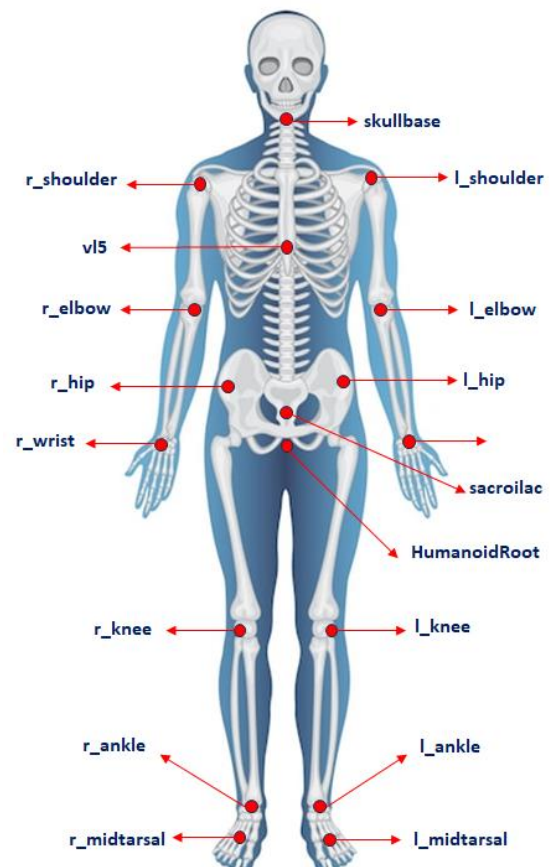


Figure 9 - Some joints of the Body Model

The Body Model is represented in the glTF format and transmitted in the initial Portable Avatar at the beginning of the Avatar-Based Videoconference.

The Spatial Attitude of a Body is defined with respect to the global coordinate system defined by the Visual Environment.

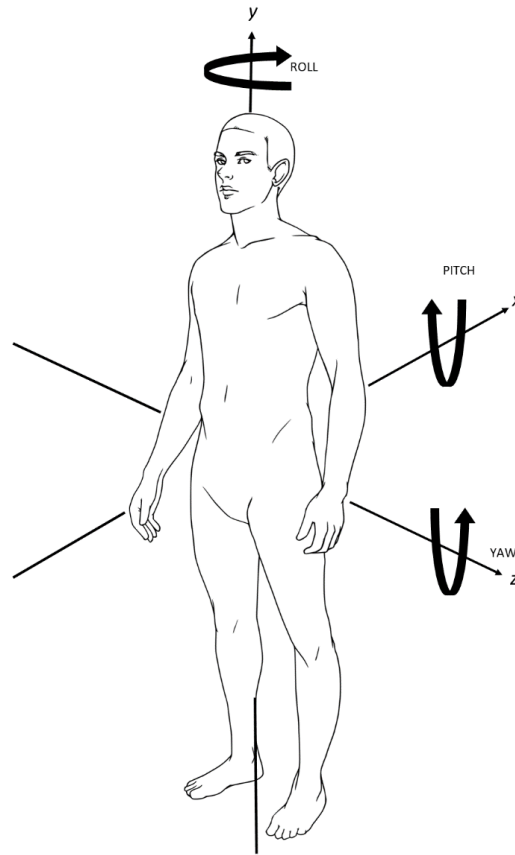
### 7.3.2 Body Descriptors

Body Descriptors are included in the data set describing the root and joints as:

- 1 Position and Orientation of the root.
- 2 Rotation angles of the joints.

The rotation of the head is treated as any other joint.

Figure 10 depicts Roll (rotation around y), Pitch (rotation around y, and Yaw (rotation around z) of a Body.



*Figure 10 – Pitch, Roll, and Yaw of Body*

### 7.3.3 Head Descriptors

The Head Descriptors are the angles of:

1. Roll: head has moved toward one of the shoulders.
2. Pitch: head has moved up and down.
3. Yaw: head has rotated left to right (around the vertical axis of the head).

These are useful in case when a body-less head is animated.

## 7.4 Face

### 7.4.1 Face Model

The Face Model is represented according to the glTF syntax.

### 7.4.2 Face Descriptors

MPAI adopts as Face Descriptors the Actions Units of the Facial Action Coding System (FACS). FACS was originally developed by Carl-Herman Hjortsjö, adopted by Paul Ekman and Wallace V. Friesen (1978) and updated by Ekman, Friesen, and Joseph C. Hager (2002) [13].

AU	Description	Facial muscle generating the Action
1	Inner Brow Raiser	Frontalis, pars medialis
2	Outer Brow Raiser	Frontalis, pars lateralis
4	Brow Lowerer	Corrugator supercilii, Depressor supercilii
5	Upper Lid Raiser	Levator palpebrae superioris

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

## 7.5 Avatar

### 7.5.1 Avatar Model

The Avatar Model combines the Body and Face Models. It is a component of the Portable Avatar Model.

## 7.6 Objects and Scenes

### 7.6.1 Spatial Attitude and Point of View

*Spatial Attitude* is a real vector containing Position and Orientation, and their Velocities and Accelerations, in the following order:

Table 19 – Components of Spatial Attitude

x	y	z	xVel	tVel	zVel	xAcc	yAcc	zAcc	$\phi$	$\theta$	$\psi$	$\phi_{Vel}$	$\theta_{Vel}$	$\psi_{Vel}$	$\phi_{Acc}$	$\theta_{Acc}$	$\psi_{Acc}$
---	---	---	------	------	------	------	------	------	--------	----------	--------	--------------	----------------	--------------	--------------	----------------	--------------

$x, y, z$  are Cartesian coordinates and  $\phi, \theta$ , and  $\psi$  the Euler angles.

The vector including only Position and Orientation is called Point of View.

*Table 20 – Components of Point of View*

x	y	z	$\phi$	$\theta$	$\psi$
---	---	---	--------	----------	--------

### **7.6.2 Audio Scene Descriptors**

Audio Scene Descriptors are specified in MPAI-CAE V2 [5].

### **7.6.3 Visual Scene Descriptors**

A Visual Scene is Described according to glTF [8]. It is produced by Client (Receiving part).

## **7.7 Other Data Types**

### **7.7.1 Text**

Specified in MPAI-MMC V2 [6]. It is a component of the Portable Avatar Model.

### **7.7.2 Language identifier**

Specified in MPAI-MMC V2 [6]. It is a component of the Portable Avatar Model.

### **7.7.3 Meaning**

Specified in MPAI-MMC V2 [6].

### **7.7.4 Personal Status**

Specified in MPAI-MMC V2 [6]. It is a component of the Portable Avatar Model.

## Annex 1 - MPAI Basics (Informative)

### 1 General

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

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

As a rule, MPAI standards include four documents: Technical Specification, Reference Software Specifications, Conformance Testing Specifications, and Performance Assessment Specifications. The last – and new in standardisation – type of Specification includes standard operating procedures that enable users of MPAI Implementations to make informed decision about their applicability based on the notion of Performance, defined as a set of attributes characterising a reliable and trustworthy implementation.

### 2 Governance of the MPAI Ecosystem

The technical foundations of the MPAI Ecosystem are currently provided by *Technical Specification: Governance of the MPAI Ecosystem (MPAI-GME)* [9] developed and maintained by MPAI:

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

An MPAI Standard is a collection of a variable number of the 5 document types.

Figure 11 depicts the MPAI ecosystem operation for conforming MPAI implementations.

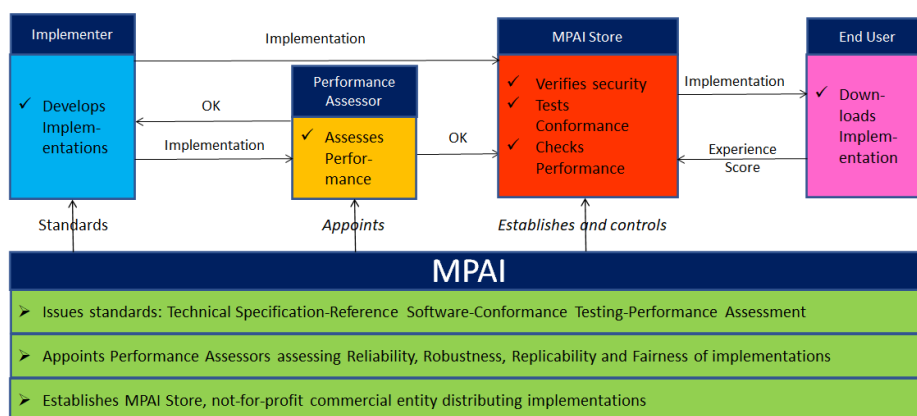


Figure 11 – The MPAI ecosystem operation

Table 21 identifies the following roles in the MPAI Ecosystem:



Table 21 - Roles in the MPAI Ecosystem

MPAI	Publishes Standards. Establishes the not-for-profit MPAI Store. Appoints Performance Assessors.
Implementers	Submit Implementations to Performance Assessors and the MPAI Store.
Performance Assessors	Inform Implementation submitters and the MPAI Store if Implementation Performance is acceptable.
MPAI Store	Assign unique ImplementerIDs (IID) to Implementers of MPAI Technical Specification in its capacity as ImplementerID Registration Authority (IIDRA) <sup>1</sup> . Verifies security and Tests Implementation Conformance.
Users	Download Implementations and report their experience to MPAI.

### 3 AI Framework

In general, MPAI Application Standards are defined as aggregations – called AI Workflows (AIW) – of processing elements – called AI Modules (AIM) – executed in an AI Framework (AIF). MPAI defines Interoperability as the ability to replace an AIW or an AIM Implementation with a functionally equivalent Implementation.

Figure 12 depicts the MPAI-AIF Reference Model. Implementations of MPAI Application Standards and user-defined MPAI-AIF Conforming applications operate in an AI Framework [4].

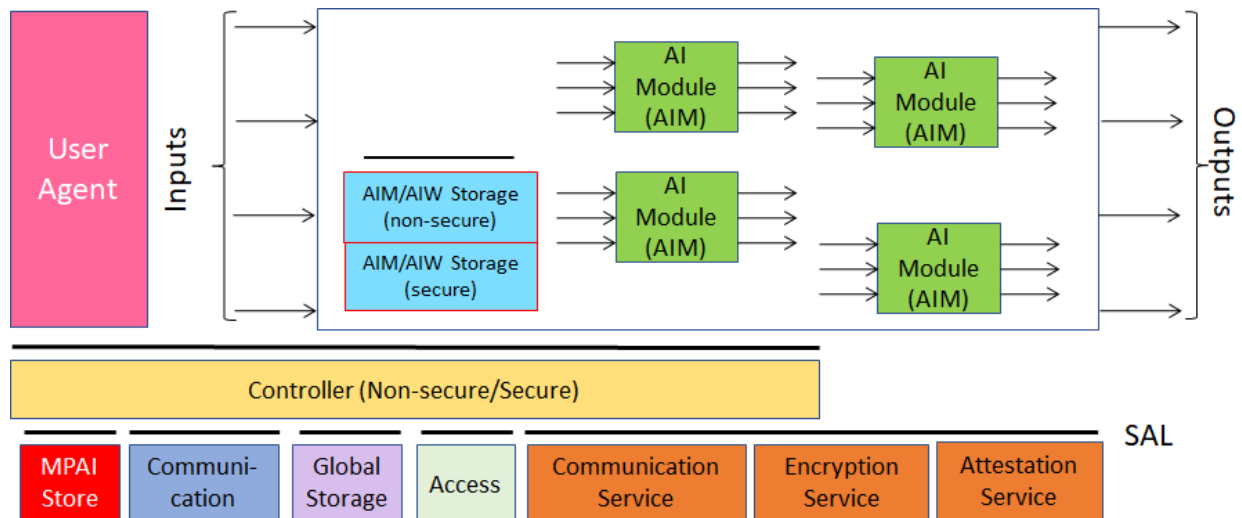


Figure 12 – The AI Framework (AIF) Reference Model

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

An AIW is defined by its Function and input/output Data and by the topology of its AIMs. Likewise, an AIM is defined by its Function and input/output Data. MPAI standard are silent on

<sup>1</sup> At the time of publication of this Technical Report, the MPAI Store was assigned as the IIDRA.

the technology used to implement the AIM which may be based on AI or data processing, and implemented in software, hardware or hybrid software and hardware technologies.

AIW and its AIMs may have 3 interoperability levels:

*Level 1* – Proprietary and satisfying the MPAI-AIF Standard.

*Level 2* – Specified by an MPAI Application Standard.

*Level 3* – Specified by an MPAI Application Standard and certified by a Performance Assessor.

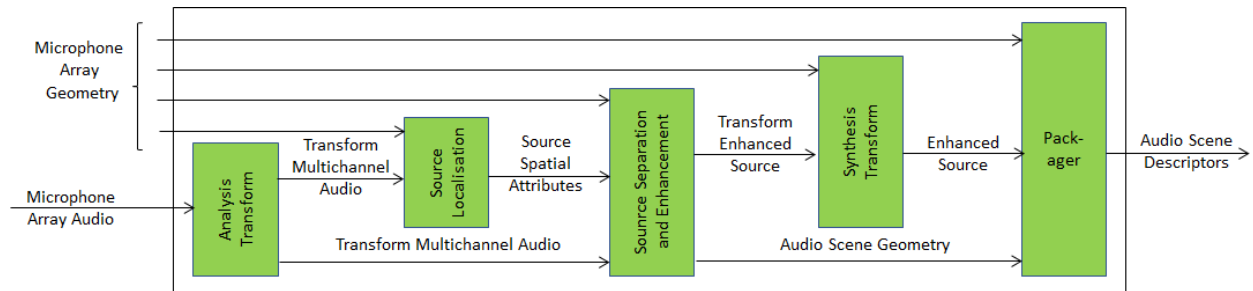
## 4 Audio-Visual Scene Description

The ability to describe (i.e., digitally represent) an audio-visual scene is a key requirement of several MPAI Technical Specifications and Use Cases. MPAI has developed Technical Specification: Context-based Audio Enhancement (MPAI-CAE) [5] that includes Audio Scene Descriptors and uses a subset of Graphics Language Transmission Format (glTF) [8] to describe a visual scene.

### 4.1.1 Audio Scene Descriptors

Audio Scene Description is a Composite AI Module (AIM) specified by Technical Specification: Context-based Audio Enhancement (MPAI-CAE) [5]. The position of an Audio Object is defined by Azimuth, Elevation, Distance.

The Composite AIM and its composing AIMs are depicted in [5].



*Figure 13 - The Audio Scene Description Composite AIM*

### 4.1.2 Visual Scene Descriptors

MPAI uses Graphics Language Transmission Format (glTF) [8] to describe a visual scene.

## Annex 2 - General MPAI Terminology

The Terms used in this standard whose first letter is capital and are not already included in *Table 1* are defined in Table 22. To concentrate in one place all the Terms that are composed of a common name followed by other words (e.g., the word Data followed by one of the words Format, Type, or Semantics), the definition given to a Terms preceded by a dash “-” applies to a Term composed by that Term without the dash preceded by the Term that precedes it in the column without a dash.

Table 22 – MPAI-wide Terms

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

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

Reliability	The attribute of an Implementation that performs as specified by the Application Standard, profile, and version the Implementation refers to, e.g., within the application scope, stated limitations, and for the period of time specified by the Implementer.
Replicability	The attribute of an Implementation whose Performance, as Assessed by a Performance Assessor, can be replicated, within an agreed level, by another Performance Assessor.
Robustness	The attribute of an Implementation that copes with data outside of the stated application scope with an estimated degree of confidence.
Scope	The domain of applicability of an MPAI Application Standard
Service Provider	An entrepreneur who offers an Implementation as a service (e.g., a recommendation service) to Users.
Standard	A set of Technical Specification, Reference Software, Conformance Testing, Performance Assessment, and Technical Report of an MPAI application Standard.
Technical Specification	(Framework) the normative specification of the AIF. (Application) the normative specification of the set of AIWs belonging to an application domain along with the AIMs required to Implement the AIWs that includes: <ol style="list-style-type: none"> <li>1. The formats of the Input/Output data of the AIWs implementing the AIWs.</li> <li>2. The Connections of the AIMs of the AIW.</li> <li>3. The formats of the Input/Output data of the AIMs belonging to the AIW.</li> </ol>
Testing Laboratory	A laboratory accredited to Assess the Grade of Performance of Implementations.
Time Base	The protocol specifying how Components can access timing information
Topology	The set of AIM Connections of an AIW.
Use Case	A particular instance of the Application domain target of an Application Standard.
User	A user of an Implementation.
User Agent	The Component interfacing the user with an AIF through the Controller
Version	A revision or extension of a Standard or of one of its elements.
Zero Trust	A cybersecurity model primarily focused on data and service protection that assumes no implicit trust.

### **Annex 3 - Notices and Disclaimers Concerning MPAI Standards (Informative)**

The notices and legal disclaimers given below shall be borne in mind when [downloading](#) and using approved MPAI Standards.

In the following, “Standard” means the collection of four MPAI-approved and [published](#) documents: “Technical Specification”, “Reference Software” and “Conformance Testing” and, where applicable, “Performance Testing”.

#### **Life cycle of MPAI Standards**

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

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

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

#### **Coverage and Applicability of MPAI Standards**

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

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

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

TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

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

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

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

#### Copyright

MPAI draft and approved standards, whether they are in the form of documents or as web pages or otherwise, are copyrighted by MPAI under Swiss and international copyright laws. MPAI Standards are made available and may be used for a wide variety of public and private uses, e.g., implementation, use and reference, in laws and regulations and standardisation. By making these documents available for these and other uses, however, MPAI does not waive any rights in copyright to its Standards. For inquiries regarding the copyright of MPAI standards, please contact the [MPAI Secretariat](#).

The Reference Software of an MPAI Standard is released with the [MPAI Modified Berkeley Software Distribution licence](#). However, implementers should be aware that the Reference Software of an MPAI Standard may reference some third party software that may have a different licence.

## **Annex 4 - Patent Declarations**

The following table will include the entities declaring to agree to licence their standard essential patents reading on the Technical Specification: Portable Avatar Format (PAF):

<b>Entity</b>	<b>Name</b>	<b>Email address</b>



## Annex 5 - AIW and AIM Metadata of ABV-CTX

### 1 Metadata for ABV-CTX

#### 1.1 AIW Metadata

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://schemas.mpai.community/AIF/2.0/AIW-AIM-metadata.schema.json",
  "title": "AIW/AIM metadata for CAS AIW",
  "Identifier": {
    "ImplementerID": "/* String assigned by IIDRA */",
    "Specification": {
      "Standard": "MPAI-PAF",
      "AIW": "ABV-CRX",
      "AIM": "ABV-CRX",
      "Version": "1"
    }
  },
  "APIProfile": "Secure",
  "Description": "This AIW composes and renders the Avatar-Based Videoconference scene.",
  "Types": [
    {
      "Name": "PortableAvatar_t",
      "Type": "uint8[]"
    },
    {
      "Name": "PointOfView_t",
      "Type": "float32[6]"
    },
    {
      "Name": "OutputAudio_t",
      "Type": "uint16[]"
    },
    {
      "Name": "OutputVisual_t",
      "Type": "uint8[]"
    }
  ],
  "Ports": [
    {
      "Name": "PortableAvatar",
      "Direction": "InputOutput",
      "RecordType": "AvatarModel_t",
      "Technology": "Software",
      "Protocol": "",
      "IsRemote": false
    },
    {
      "Name": "PointOfView",
      "Direction": "InputOutput",
      "RecordType": "PointOfView_t",
      "Technology": "Software",
      "Protocol": "",
      "IsRemote": false
    },
    {
      "Name": "OutputAudio",
      "Direction": "OutputInput",
      "RecordType": "OutputAudio_t",
      "Technology": "Software",
      "Protocol": "",
      "IsRemote": false
    },
    {
      "Name": "OutputVisual",
      "Direction": "OutputInput",
      "RecordType": "OutputVisual_t",
      "Technology": "Software",
      "Protocol": "",
      "IsRemote": false
    }
  ]
}
```

```

    }
  ],
  "SubAIMs":[
    {
      "Name":"PortableAvatarDemultiplexing",
      "Identifier":{
        "ImplementerID":"/ * String assigned by IIDRA */",
        "Specification":{
          "Standard":"MPAI-PAF",
          "AIW":"PAF-CRX",
          "AIM":"PortableAvatarDemultiplexing",
          "Version":"1"
        }
      }
    },
    {
      "Name":"VisualSceneCreation",
      "Identifier":{
        "ImplementerID":"/ * String assigned by IIDRA */",
        "Specification":{
          "Standard":"MPAI-PAF",
          "AIW":"PAF-CRX",
          "AIM":"VisualSceneCreation",
          "Version":"1"
        }
      }
    },
    {
      "Name":"AudioSceneCreation",
      "Identifier":{
        "ImplementerID":"/ * String assigned by IIDRA */",
        "Specification":{
          "Standard":"MPAI-PAF",
          "AIW":"PAF-CRX",
          "AIM":"AudioSceneCreation",
          "Version":"1"
        }
      }
    },
    {
      "Name":"AVSceneViewing",
      "Identifier":{
        "ImplementerID":"/ * String assigned by IIDRA */",
        "Specification":{
          "Standard":"MPAI-PAF",
          "AIW":"PAF-CRX",
          "AIM":"AVSceneViewing",
          "Version":"1"
        }
      }
    }
  ],
  "Topology":[
    {
      "_comment": "Input to first AIM column"
    },
    {
      "Output":{
        "AIMName": "",
        "PortName": "PortableAvatar"
      },
      "Input":{
        "AIMName": "PortableAvatarDemultiplexing",
        "PortName": "PortableAvatar"
      }
    },
    {
      "_comment": "Input to second AIM column"
    },
    {
      "Output":{
        "AIMName": "PortableAvatarDemultiplexing",
        "PortName": "AvatarID"
      },
    },
  ]

```

```

        "Input":{
            "AIMName":"VisualSceneCreation",
            "PortName":"AvatarID"
        }
    },
    {
        "Output":{
            "AIMName":"PortableAvatarDemultiplexing",
            "PortName":"VisualEnvironment"
        },
        "Input":{
            "AIMName":"VisualSceneCreation",
            "PortName":"VisualEnvironment"
        }
    },
    {
        "Output":{
            "AIMName":"PortableAvatarDemultiplexing",
            "PortName":"AvatarModel"
        },
        "Input":{
            "AIMName":"VisualSceneCreation",
            "PortName":"AvatarModel"
        }
    },
    {
        "Output":{
            "AIMName":"PortableAvatarDemultiplexing",
            "PortName":"SpatialAttitude"
        },
        "Input":{
            "AIMName":"VisualSceneCreation",
            "PortName":"SpatialAttitude"
        }
    },
    {
        "Output":{
            "AIMName":"PortableAvatarDemultiplexing",
            "PortName":"BodyDescriptors"
        },
        "Input":{
            "AIMName":"VisualSceneCreation",
            "PortName":"BodyDescriptors"
        }
    },
    {
        "Output":{
            "AIMName":"PortableAvatarDemultiplexing",
            "PortName":"FaceDescriptors"
        },
        "Input":{
            "AIMName":"VisualSceneCreation",
            "PortName":"FaceDescriptors"
        }
    },
    {
        {
            "_comment": "Input to third AIM column"
        },
        {
            "Output":{
                "AIMName":"VisualSceneCreation",
                "PortName":"MouthSpatialAttitude"
            },
            "Input":{
                "AIMName":"AudioSceneCreation",
                "PortName":"MouthSpatialAttitude"
            }
        }
    },
    {
        {
            "Output":{
                "AIMName":"PortableAvatarDemultiplexing",
                "PortName":"AvatarID"
            },
            "Input":{

```

```

        "AIMName": "AudioSceneCreation",
        "PortName": "AvatarID"
    },
    {
        "Output": {
            "AIMName": "PortableAvatarDemultiplexing",
            "PortName": "InputSpeech"
        },
        "Input": {
            "AIMName": "AudioSceneCreation",
            "PortName": "InputSpeech"
        }
    },
    {
        "_comment": "Input to fourth AIM column"
    },
    {
        "Output": {
            "AIMName": "VisualSceneCreation",
            "PortName": "VisualSceneDescriptors"
        },
        "Input": {
            "AIMName": "AVSceneViewing",
            "PortName": "VisualSceneDescriptors"
        }
    },
    {
        "Output": {
            "AIMName": "AudioSceneCreation",
            "PortName": "AudioSceneDescriptors"
        },
        "Input": {
            "AIMName": "AVSceneViewing",
            "PortName": "AudioSceneDescriptors"
        }
    },
    {
        "_comment": "Input to output"
    },
    {
        "Output": {
            "AIMName": "",
            "PortName": "PointofView"
        },
        "Input": {
            "AIMName": "AVSceneViewing",
            "PortName": "PointofView"
        }
    },
    {
        "Output": {
            "AIMName": "AVSceneViewing",
            "PortName": "OutputAudio"
        },
        "Input": {
            "AIMName": "",
            "PortName": "OutputAudio"
        }
    },
    {
        "Output": {
            "AIMName": "AVSceneViewing",
            "PortName": "OutputVisual"
        },
        "Input": {
            "AIMName": "",
            "PortName": "OutputVisual"
        }
    }
],
"Implementations": [
],

```

```

    "ResourcePolicies":[
    ],
    "Documentation":[
    {
        "Type":"tutorial",
        "URI":"https://mpai.community/standards/MPAI-PAF/"
    }
    ]
}

```

## 1.2 AIM Metadata

The AIM Metadata are available from

```

AudioSceneDescription": https://schemas.mpai.community/CAE/V2.0/ASD/AudioSceneDescription.json
VisualSceneDescription" :https://schemas.mpai.community/PAF/V2.0/CTX/VisualSceneDescription.json
AudioVisualAlignment": https://schemas.mpai.community/PAF/V1.0/CTX/AudioVisualAlignment.json
SpeechRecogniton": https://schemas.mpai.community/MMC/V2.0/CPS/SpeechRecogniton.json
LanguageUnderstanding": https://schemas.mpai.community/MMC/V2.0/CPS/LanguageUnderstanding.json
PersonalStatusExtraction": https://schemas.mpai.community/
https://schemas.mpai.community/MMC/V2.0/PSE/PersonalStatusExtraction.json
PortableAvatarDescription":
https://schemas.mpai.community/PAF/V2.0/CTX/PortableAvatarDescription.json

```

## 2 AIW metadata for ABV-SRV

### 2.1 AIW Metadata

```

{
    "$schema":"https://json-schema.org/draft/2020-12/schema",
    "$id":"https://schemas.mpai.community/AIF/2.0/AIW-AIM-metadata.schema.json",
    "title":" AIW/AIM metadata ABV AIW ",
    "Identifier":{
        "ImplementerID":"/ * String assigned by IIDRA */",
        "Specification":{
            "Standard":"MPAI-PAF",
            "AIW":"ABV-SRV",
            "AIM":"ABV-SRV",
            "Version":"1"
        }
    },
    "APIProfile":"Secure",
    "Description":"At the start, this AIW receives Input Portable Avatars, selects the
VisualEnvironment, places Avatar Models, and distributes the Portable Avatars. Then it
continuously receives Input Portable Avatars translates Text and Speech and distributes
Portable Avatars.",
    "Types":[
        {
            "Name": "Summary_t",
            "Type": "uint8[]"
        },
        {
            "Name": "VisualEnvironment_t",
            "Type": "uint8[]"
        },
        {
            "Name":"SpatialAttitude_t",
            "Type":"{float32[3] Position; Float32[3] Orientation"
        },
        {
            "Name": "PortableAvatar_t",
            "Type": "uint8[]"
        },
        {
            "Name": "ParticipantID_t",
            "Type": "uint8[]"
        },
        {
            "Name": "Speech_t",

```

```

        "Type": "uint16[]"
    },
    {
        "Name": "FaceObject_t",
        "Type": "uint32[]"
    }
],
"Ports": [
    {
        "Name": "Summary1",
        "Direction": "InputOutput",
        "RecordType": "Summary_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "VisualEnvironmentInput",
        "Direction": "InputOutput",
        "RecordType": "VisualEnvironment_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "SpatialAttitude",
        "Direction": "InputOutput",
        "RecordType": "SpatialAttitude_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "PortableAvatar1",
        "Direction": "InputOutput",
        "RecordType": "PortableAvatar_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "PortableAvatar2",
        "Direction": "InputOutput",
        "RecordType": "PortableAvatar_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "SpeechObject",
        "Direction": "InputOutput",
        "RecordType": "Speech_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "FaceObject",
        "Direction": "InputOutput",
        "RecordType": "FaceObject_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "Summary2",
        "Direction": "OutputInput",
        "RecordType": "Summary_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "PortableAvatar3",

```

```

        "Direction": "OutputInput",
        "RecordType": "PortableAvatar_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    }
],
"SubAIMs": [
    {
        "Name": "PortableAvatarDemultiplexing",
        "Identifier": {
            "ImplementerID": "/* String assigned by IIDRA */",
            "Specification": {
                "Standard": "MPAI-PAF",
                "AIW": "ABV-SRV",
                "AIM": "PortableAvatarDemultiplexing",
                "Version": "1"
            }
        }
    },
    {
        "Name": "Translation",
        "Identifier": {
            "ImplementerID": "/* String assigned by IIDRA */",
            "Specification": {
                "Standard": "MPAI-PAF",
                "AIW": "ABV-SRV",
                "AIM": "Translation",
                "Version": "1"
            }
        }
    },
    {
        "Name": "ParticipantAuthentication",
        "Identifier": {
            "ImplementerID": "/* String assigned by IIDRA */",
            "Specification": {
                "Standard": "MPAI-PAF",
                "AIW": "ABV-SRV",
                "AIM": "ParticipantAuthentication",
                "Version": "1"
            }
        }
    },
    {
        "Name": "PortableAvatarMultiplexing",
        "Identifier": {
            "ImplementerID": "/* String assigned by IIDRA */",
            "Specification": {
                "Standard": "MPAI-PAF",
                "AIW": "ABV-SRV",
                "AIM": "PortableAvatarMultiplexing",
                "Version": "1"
            }
        }
    }
],
"Topology": [
    {
        "_comment": "Input to first AIM column"
    },
    {
        "Output": {
            "AIMName": "",
            "PortName": "PortableAvatar"
        },
        "Input": {
            "AIMName": "PortableAvatarDemultiplexing",
            "PortName": "PortableAvatar"
        }
    },
    {
        "_comment": "Input to second AIM column"
    },

```

```

{
  "Output":{
    "AIMName":"PortableAvatarDemultiplexing",
    "PortName":"InputText"
  },
  "Input":{
    "AIMName":"Translation",
    "PortName":"InputText"
  }
},
{
  "Output":{
    "AIMName":"PortableAvatarDemultiplexing",
    "PortName":"InputSpeech"
  },
  "Input":{
    "AIMName":"Translation",
    "PortName":"InputSpeech"
  }
},
{
  "Output":{
    "AIMName":"PortableAvatarDemultiplexing",
    "PortName":"LanguagePreference"
  },
  "Input":{
    "AIMName":"Translation",
    "PortName":"LanguagePreference"
  }
},
{
  "Output":{
    "AIMName":"PortableAvatarDemultiplexing",
    "PortName":"ParticipantID"
  },
  "Input":{
    "AIMName":"ParticipantAuthentication",
    "PortName":"ParticipantID"
  }
},
{
  "Output":{
    "AIMName":"",
    "PortName":"SpeechObject"
  },
  "Input":{
    "AIMName":"ParticipantAuthentication",
    "PortName":"SpeechObject"
  }
},
{
  "Output":{
    "AIMName":"",
    "PortName":"FaceObject"
  },
  "Input":{
    "AIMName":"ParticipantAuthentication",
    "PortName":"FaceObject"
  }
},
{
  "_comment":"Input to third AIM column"
},
{
  "Output":{
    "AIMName":"",
    "PortName":"VisualEnvironment"
  },
  "Input":{
    "AIMName":"PortableAvatarMultiplexing",
    "PortName":"VisualEnvironment"
  }
},
{

```



```

    "Output":{
      "AIMName": "",
      "PortName": "SpatialAttitudes"
    },
    "Input":{
      "AIMName": "PortableAvatarMultiplexing",
      "PortName": "SpatialAttitudes"
    }
  },
  {
    "Output":{
      "AIMName": "",
      "PortName": "PortableAvatar1"
    },
    "Input":{
      "AIMName": "PortableAvatarMultiplexing",
      "PortName": "PortableAvatar1"
    }
  },
  {
    "Output":{
      "AIMName": "Translation",
      "PortName": "TranslatedText"
    },
    "Input":{
      "AIMName": "PortableAvatarMultiplexing",
      "PortName": "TranslatedText"
    }
  },
  {
    "Output":{
      "AIMName": "Translation",
      "PortName": "TranslatedSpeech"
    },
    "Input":{
      "AIMName": "PortableAvatarMultiplexing",
      "PortName": "TranslatedSpeech"
    }
  },
  {
    "Output":{
      "AIMName": "ParticipantAuthentication",
      "PortName": "AvatarID"
    },
    "Input":{
      "AIMName": "PortableAvatarMultiplexing",
      "PortName": "AvatarID"
    }
  },
  {
    "_comment": "Input to output"
  },
  {
    "Output":{
      "AIMName": "",
      "PortName": "Summary2"
    },
    "Input":{
      "AIMName": "",
      "PortName": "Summary2"
    }
  },
  {
    "Output":{
      "AIMName": "PortableAvatarMultiplexing",
      "PortName": "PortableAvatar"
    },
    "Input":{
      "AIMName": "",
      "PortName": "PortableAvatar"
    }
  }
],
"Implementations": [

```

```

    ],
    "ResourcePolicies":[
    ],
    "Documentation":[
      {
        "Type":"tutorial",
        "URI":"https://mpai.community/standards/MPAI-PAF/"
      }
    ]
  }
}

```

## 2.2 AIM Metadata

The AIM Metadata are available from

PortableAvatarDemultiplexing:  
<https://schemas.mpai.community/PAF/V2.0/SRV/PortableAvatarDemultiplexing.json>  
 SpeechAndTextTranslation:  
<https://schemas.mpai.community/MMC/V2.0/STT/SpeechAndTextTranslation.json>  
 ParticipantAuthentication:  
<https://schemas.mpai.community/PAF/V2.0/SRV/ParticipantAuthentication.json>  
 PortableAvatarMultiplexing:  
<https://schemas.mpai.community/PAF/V2.0/SRV/PortableAvatarMultiplexing.json>

## 3 Metadata for MMC-VSV

### 3.1 AIW Metadata

```

{
  "$schema":"https://json-schema.org/draft/2020-12/schema",
  "$id":"https://schemas.mpai.community/AIF/2.0/AIW-AIM-metadata.schema.json",
  "title":"AIW/AIM metadata for VSV",
  "Identifier":{
    "ImplementerID":"/* String assigned by IIDRA */",
    "Specification":{
      "Standard":"MPAI-MMC",
      "AIW":"MMC-VSV",
      "AIM":"MMC-VSV ",
      "Version":"2"
    }
  },
  "APIProfile":"Secure",
  "Description":" This AIM produces the Portable of the Virtual Secretary and the Summary
in the Avatar-Based Videoconference",
  "Types":[
    {
      "Name":"PortableAvatar_t",
      "Type":"uint8[]"
    },
    {
      "Name":"Summary_t",
      "Type":"uint8[]"
    }
  ],
  "Ports":[
    {
      "Name":"PortableAvatar",
      "Direction":"InputOutput",
      "RecordType":"PortableAvatar_t",
      "Technology":"Software",
      "Protocol":"",
      "IsRemote":false
    },
    {
      "Name":"Summary",
      "Direction":"OutputInput",
      "RecordType":"Summary_t",
      "Technology":"Software",
      "Protocol":"",

```

```

        "IsRemote":false
    },
    {
        "Name":"VSPortableAvatar",
        "Direction":"OutputInput",
        "RecordType":"PortableAvatar_t",
        "Technology":"Software",
        "Protocol": "",
        "IsRemote":false
    }
],
"SubAIMs":[
    {
        "Name":"PortableAvatarDeMultiplexing",
        "Identifier":{
            "ImplementerID":"/ * String assigned by IIDRA * /",
            "Specification":{
                "Standard":"MPAI-MMC ",
                "AIW":"MMC-VSV",
                "AIM":"PortableAvatarDemultiplexing",
                "Version":"2"
            }
        }
    },
    {
        "Name":"SpeechRecognition",
        "Identifier":{
            "ImplementerID":"/ * String assigned by IIDRA * /",
            "Specification":{
                "Standard":"MPAI-MMC",
                "AIW":"MMC-VSV",
                "AIM":"SpeechRecognition",
                "Version":"1"
            }
        }
    },
    {
        "Name":" LanguageUnderstanding",
        "Identifier":{
            "ImplementerID":"/ * String assigned by IIDRA * /",
            "Specification":{
                "Standard":"MPAI-MMC",
                "AIW":"MMC-VSV",
                "AIM":"LanguageUnderstanding",
                "Version":"2"
            }
        }
    },
    {
        "Name":"PersonalStatusExtraction",
        "Identifier":{
            "ImplementerID":"/ * String assigned by IIDRA * /",
            "Specification":{
                "Standard":"MPAI-MMC ",
                "AIW":"MMC-VSV",
                "AIM":"PersonalStatusExtraction",
                "Version":"2"
            }
        }
    },
    {
        "Name":"Summarisation",
        "Identifier":{
            "ImplementerID":"/ * String assigned by IIDRA * /",
            "Specification":{
                "Standard":"MPAI-MMC",
                "AIW":"MMC-VSV",
                "AIM":"Summarisation",
                "Version":"2"
            }
        }
    },
    {
        "Name":"DialogueProcessing",

```

```

        "Identifier":{
            "ImplementerID": "/* String assigned by IIDRA */",
            "Specification":{
                "Standard": "MPAI-MMC",
                "AIW": "MMC-VSV",
                "AIM": "DialogueProcessing",
                "Version": "2"
            }
        },
    {
        "Name": "PersonalStatusDisplay",
        "Identifier":{
            "ImplementerID": "/* String assigned by IIDRA */",
            "Specification":{
                "Standard": "MPAI-MMC",
                "AIW": "MMC-VSV",
                "AIM": "PersonalStatusDisplay",
                "Version": "2"
            }
        }
    }
],
"Topology": [
    {
        "_comment": "Input to first AIM column"
    },
    {
        "Output":{
            "AIMName": "",
            "PortName": "PortableAvatar"
        },
        "Input":{
            "AIMName": "PortableAvatarDemultiplexing",
            "PortName": "PortableAvatar"
        }
    },
    {
        "_comment": "Input to second AIM column"
    },
    {
        "Output":{
            "AIMName": "PortableAvatarDemultiplexing",
            "PortName": "InputSpeech1"
        },
        "Input":{
            "AIMName": "SpeechRecognition",
            "PortName": "InputSpeech1"
        }
    },
    {
        "_comment": "Input to third AIM column"
    },
    {
        "Output":{
            "AIMName": "SpeechRecognition",
            "PortName": "RecognisedText"
        },
        "Input":{
            "AIMName": "LanguageUnderstanding",
            "PortName": "RecognisedText"
        }
    },
    {
        "Output":{
            "AIMName": "PortableAvatarDemultiplexing",
            "PortName": "InputText2"
        },
        "Input":{
            "AIMName": "LanguageUnderstanding",
            "PortName": "InputText2"
        }
    }
],
{

```

```

    "_comment": "Input to fourth AIM column"
  },
  {
    "Output":{
      "AIMName":"LanguageUnderstanding",
      "PortName":"Meaning3"
    },
    "Input":{
      "AIMName":"PersonalStatusExtraction",
      "PortName":"Meaning3"
    }
  },
  {
    "Output":{
      "AIMName":"PortableAvatarDemultiplexing",
      "PortName":"InputSpeech2"
    },
    "Input":{
      "AIMName":"PersonalStatusExtraction",
      "PortName":"InputSpeech2"
    }
  },
  {
    "Output":{
      "AIMName":"PortableAvatarDemultiplexing",
      "PortName":"AvatarID1"
    },
    "Input":{
      "AIMName":"PersonalStatusExtraction",
      "PortName":"AvatarID1"
    }
  },
  {
    "Output":{
      "AIMName":"PortableAvatarDemultiplexing",
      "PortName":"BodyDescriptors"
    },
    "Input":{
      "AIMName":"PersonalStatusExtraction",
      "PortName":"BodyDescriptors"
    }
  },
  {
    "Output":{
      "AIMName":"PortableAvatarDemultiplexing",
      "PortName":"FaceDescriptors"
    },
    "Input":{
      "AIMName":"PersonalStatusExtraction",
      "PortName":"FaceDescriptors"
    }
  },
  {
    "_comment": "Input to fifth AIM column"
  },
  {
    "Output":{
      "AIMName":"LanguageUnderstanding",
      "PortName":"Meaning2"
    },
    "Input":{
      "AIMName":"Summarisation",
      "PortName":"Meaning2"
    }
  },
  {
    "Output":{
      "AIMName":"LanguageUnderstanding",
      "PortName":"RefinedText2"
    },
    "Input":{
      "AIMName":"Summarisation",
      "PortName":"RefinedText2"
    }
  },
  {
    "Output":{

```

```

        "AIMName": "LanguageUnderstanding",
        "PortName": "AvatarID2"
    },
    "Input": {
        "AIMName": "Summarisation",
        "PortName": "AvatarID2"
    }
},
{
    "Output": {
        "AIMName": "PersonalStatusExtraction",
        "PortName": "InputPersonalStatus1"
    },
    "Input": {
        "AIMName": "Summarisation",
        "PortName": "InputPersonalStatus1"
    }
},
{
    "_comment": "Input to sixth AIM column"
},
{
    "Output": {
        "AIMName": "LanguageUnderstanding",
        "PortName": "InputText1"
    },
    "Input": {
        "AIMName": "DialogueProcessing",
        "PortName": "InputText1"
    }
},
{
    "Output": {
        "AIMName": "LanguageUnderstanding",
        "PortName": "RefinedText1"
    },
    "Input": {
        "AIMName": "DialogueProcessing",
        "PortName": "RefinedText1"
    }
},
{
    "Output": {
        "AIMName": "LanguageUnderstanding",
        "PortName": "Meaning1"
    },
    "Input": {
        "AIMName": "Summarisation",
        "PortName": "Meaning1"
    }
},
{
    "Output": {
        "AIMName": "DialogueProcessing",
        "PortName": "EditedSummary"
    },
    "Input": {
        "AIMName": "Summarisation",
        "PortName": "EditedSummary"
    }
},
{
    "Output": {
        "AIMName": "Summarisation",
        "PortName": "Summary1"
    },
    "Input": {
        "AIMName": "DialogueProcessing",
        "PortName": "Summary1"
    }
},
{
    "Output": {
        "AIMName": "PersonalStatusExtraction",
        "PortName": "InputPersonalStatus2"
    },

```

```

        "Input":{
            "AIMName":"Summarisation",
            "PortName":"InputPersonalStatus2"
        }
    },
    {
        "_comment": "Input to seventh AIM column"
    },
    {
        "Output":{
            "AIMName": "",
            "PortName": "VSAvatarModel"
        },
        "Input":{
            "AIMName":"PersonalStatusDisplay",
            "PortName":"VSAvatarModel"
        }
    },
    {
        "Output":{
            "AIMName":"DialogueProcessing",
            "PortName":"VSPersonalStatus"
        },
        "Input":{
            "AIMName":"PersonalStatusDisplay",
            "PortName":"VSPersonalStatus"
        }
    },
    {
        "Output":{
            "AIMName":"DialogueProcessing",
            "PortName":"VSText"
        },
        "Input":{
            "AIMName":"PersonalStatusDisplay",
            "PortName":"VSText"
        }
    },
    {
        "_comment": "Input to output"
    },
    {
        "Output":{
            "AIMName":"DialogueProcessing",
            "PortName":"Summary2"
        },
        "Input":{
            "AIMName": "",
            "PortName":"Summary2"
        }
    },
    {
        "Output":{
            "AIMName":"PersonalStatusDisplay",
            "PortName":"VSPortableAvatar"
        },
        "Input":{
            "AIMName": "",
            "PortName":"VSPortableAvatar"
        }
    }
],
"Implementations":[

],
"ResourcePolicies":[

],
"Documentation":[
    {
        "Type":"tutorial",
        "URI":"https://mpai.community/standards/mpai-mmc/"
    }
]

```

```
}
```

## 3.2 AIM Metadata

The AIM Metadata are available from

PortableAvatarDemultiplexing:  
<https://schemas.mpai.community/PAF/V2.0/SRV/PortableAvatarDemultiplexing.json>  
SpeechRecognition: <https://schemas.mpai.community/MMC/V2.0/CPS/SpeechRecognition.json>  
LanguageUnderstanding: <https://schemas.mpai.community/MMC/V2.0/CPS/LanguageUnderstanding.json>  
PersonalStatusExtraction:  
<https://schemas.mpai.community/MMC/V2.0/PSE/PersonalStatusExtraction.json>  
Summarisation: <https://schemas.mpai.community/MMC/V2.0/VSV/Summarisation.json>  
DialogueProcessing: <https://schemas.mpai.community/MMC/V2.0/VSV/DialogueProcessing.json>  
PersonalStatusDisplay: <https://schemas.mpai.community/PAF/V2.0/PSD/PersonalStatusDisplay.json>

## 4 AIW metadata for ABV-CRX

### 4.1 AIW Metadata

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://schemas.mpai.community/AIF/2.0/AIW-AIM-metadata.schema.json",
  "title": "AIW/AIM metadata for CAS AIW",
  "Identifier": {
    "ImplementerID": "/* String assigned by IIDRA */",
    "Specification": {
      "Standard": "MPAI-PAF",
      "AIW": "ABV-CRX",
      "AIM": "ABV-CRX",
      "Version": "1"
    }
  },
  "APIProfile": "Secure",
  "Description": "This AIW composes and renders the Avatar-Based Videoconference scene.",
  "Types": [
    {
      "Name": "PortableAvatar_t",
      "Type": "uint8[]"
    },
    {
      "Name": "PointOfView_t",
      "Type": "float32[6]"
    },
    {
      "Name": "OutputAudio_t",
      "Type": "uint16[]"
    },
    {
      "Name": "OutputVisual_t",
      "Type": "uint8[]"
    }
  ],
  "Ports": [
    {
      "Name": "PortableAvatar",
      "Direction": "InputOutput",
      "RecordType": "AvatarModel_t",
      "Technology": "Software",
      "Protocol": "",
      "IsRemote": false
    },
    {
      "Name": "PointOfView",
      "Direction": "InputOutput",
      "RecordType": "PointOfView_t",
      "Technology": "Software",
      "Protocol": "",
      "IsRemote": false
    },
    {
      "Name": "OutputAudio",

```



```

        "Direction": "OutputInput",
        "RecordType": "OutputAudio_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "OutputVisual",
        "Direction": "OutputInput",
        "RecordType": "OutputVisual_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    }
],
"SubAIMs": [
    {
        "Name": "PortableAvatarDemultiplexing",
        "Identifier": {
            "ImplementerID": "/* String assigned by IIDRA */",
            "Specification": {
                "Standard": "MPAI-PAF",
                "AIW": "PAF-CRX",
                "AIM": "PortableAvatarDemultiplexing",
                "Version": "1"
            }
        }
    },
    {
        "Name": "VisualSceneCreation",
        "Identifier": {
            "ImplementerID": "/* String assigned by IIDRA */",
            "Specification": {
                "Standard": "MPAI-PAF",
                "AIW": "PAF-CRX",
                "AIM": "VisualSceneCreation",
                "Version": "1"
            }
        }
    },
    {
        "Name": "AudioSceneCreation",
        "Identifier": {
            "ImplementerID": "/* String assigned by IIDRA */",
            "Specification": {
                "Standard": "MPAI-PAF",
                "AIW": "PAF-CRX",
                "AIM": "AudioSceneCreation",
                "Version": "1"
            }
        }
    },
    {
        "Name": "AVSceneViewing",
        "Identifier": {
            "ImplementerID": "/* String assigned by IIDRA */",
            "Specification": {
                "Standard": "MPAI-PAF",
                "AIW": "PAF-CRX",
                "AIM": "AVSceneViewing",
                "Version": "1"
            }
        }
    }
],
"Topology": [
    {
        "_comment": "Input to first AIM column"
    },
    {
        "Output": {
            "AIMName": "",
            "PortName": "PortableAvatar"
        }
    },

```

```

    "Input":{
      "AIMName":"PortableAvatarDemultiplexing",
      "PortName":"PortableAvatar"
    }
  },
  {
    "_comment": "Input to second AIM column"
  },
  {
    "Output":{
      "AIMName":"PortableAvatarDemultiplexing",
      "PortName":"AvatarID"
    },
    "Input":{
      "AIMName":"VisualSceneCreation",
      "PortName":"AvatarID"
    }
  },
  {
    {
      "Output":{
        "AIMName":"PortableAvatarDemultiplexing",
        "PortName":"VisualEnvironment"
      },
      "Input":{
        "AIMName":"VisualSceneCreation",
        "PortName":"VisualEnvironment"
      }
    }
  },
  {
    {
      "Output":{
        "AIMName":"PortableAvatarDemultiplexing",
        "PortName":"AvatarModel"
      },
      "Input":{
        "AIMName":"VisualSceneCreation",
        "PortName":"AvatarModel"
      }
    }
  },
  {
    {
      "Output":{
        "AIMName":"PortableAvatarDemultiplexing",
        "PortName":"SpatialAttitude"
      },
      "Input":{
        "AIMName":"VisualSceneCreation",
        "PortName":"SpatialAttitude"
      }
    }
  },
  {
    {
      "Output":{
        "AIMName":"PortableAvatarDemultiplexing",
        "PortName":"BodyDescriptors"
      },
      "Input":{
        "AIMName":"VisualSceneCreation",
        "PortName":"BodyDescriptors"
      }
    }
  },
  {
    {
      "Output":{
        "AIMName":"PortableAvatarDemultiplexing",
        "PortName":"FaceDescriptors"
      },
      "Input":{
        "AIMName":"VisualSceneCreation",
        "PortName":"FaceDescriptors"
      }
    }
  },
  {
    "_comment": "Input to third AIM column"
  },
  {
    "Output":{
      "AIMName":"VisualSceneCreation",

```

```

        "PortName": "MouthSpatialAttitude"
    },
    "Input": {
        "AIMName": "AudioSceneCreation",
        "PortName": "MouthSpatialAttitude"
    }
},
{
    "Output": {
        "AIMName": "PortableAvatarDemultiplexing",
        "PortName": "AvatarID"
    },
    "Input": {
        "AIMName": "AudioSceneCreation",
        "PortName": "AvatarID"
    }
},
{
    "Output": {
        "AIMName": "PortableAvatarDemultiplexing",
        "PortName": "InputSpeech"
    },
    "Input": {
        "AIMName": "AudioSceneCreation",
        "PortName": "InputSpeech"
    }
},
{
    "_comment": "Input to fourth AIM column"
},
{
    "Output": {
        "AIMName": "VisualSceneCreation",
        "PortName": "VisualSceneDescriptors"
    },
    "Input": {
        "AIMName": "AVSceneViewing",
        "PortName": "VisualSceneDescriptors"
    }
},
{
    "Output": {
        "AIMName": "AudioSceneCreation",
        "PortName": "AudioSceneDescriptors"
    },
    "Input": {
        "AIMName": "AVSceneViewing",
        "PortName": "AudioSceneDescriptors"
    }
},
{
    "_comment": "Input to output"
},
{
    "Output": {
        "AIMName": "",
        "PortName": "PointofView"
    },
    "Input": {
        "AIMName": "AVSceneViewing",
        "PortName": "PointofView"
    }
},
{
    "Output": {
        "AIMName": "AVSceneViewing",
        "PortName": "OutputAudio"
    },
    "Input": {
        "AIMName": "",
        "PortName": "OutputAudio"
    }
},
{

```

```

        "Output":{
            "AIMName":"AVSceneViewing",
            "PortName":"OutputVisual"
        },
        "Input":{
            "AIMName": "",
            "PortName":"OutputVisual"
        }
    },
    "Implementations":[
    ],
    "ResourcePolicies":[
    ],
    "Documentation":[
        {
            "Type":"tutorial",
            "URI":"https://mpai.community/standards/MPAI-PAF/"
        }
    ]
}

```

## 4.2 AIM Metadata

The AIM Metadata are available from

PortableAvatarDemultiplexing: <https://schemas.mpai.community/PAF/V2.0/SRV/PortableAvatarDemultiplexing.json>  
 VisualSceneCreation: <https://schemas.mpai.community/PAF/V2.0/CRX/VisualSceneCreation.json>  
 AudioSceneCreation: <https://schemas.mpai.community/PAF/V2.0/CRX/AudioSceneCreation.json>  
 AVSceneViewing: <https://schemas.mpai.community/PAF/V2.0/CRX/AVSceneViewing.json>

## 5 Metadata of MMC-PSE

### 5.1 AIW Metadata

```

{
    "Identifier":{
        "ImplementerID": "/* String assigned by IIDRA */",
        "Specification":{
            "Name":"MPAI-PAF",
            "AIW": "",
            "AIM":"PersonalStatusDisplay",
            "Version":"1"
        },
        "Description":"This AIM implements Personal Status Display function.",
        "Types":[
            {
                "Name":"AvatarID_t",
                "Type":"uint8[]"
            },
            {
                "Name":"Text_t",
                "Type":"{uint8[] | uint16[]}"
            },
            {
                "Name":"PSSpeech_t",
                "Type":"uint8[]"
            },
            {
                "Name":"AvatarModel_t",
                "Type":"uint8[]"
            },
            {
                "Name":"PSFace_t",
                "Type":"uint8[]"
            },
            {
                "Name":"PSGesture_t",

```

```

        "Type": "uint8[]"
    },
    {
        "Name": "PortableAvatar_t",
        "Type": "uint8[]"
    }
],
"Ports": [
    {
        "Name": "AvatarID",
        "Direction": "InputOutput",
        "RecordType": "AvatarID_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "MachineText1",
        "Direction": "InputOutput",
        "RecordType": "Text_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "MachineText2",
        "Direction": "InputOutput",
        "RecordType": "Text_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "PSSpeech",
        "Direction": "InputOutput",
        "RecordType": "PSSpeech_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "AvatarModel1",
        "Direction": "InputOutput",
        "RecordType": "AvatarModel_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "PSFace",
        "Direction": "InputOutput",
        "RecordType": "PSFace_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "MachineText3",
        "Direction": "InputOutput",
        "RecordType": "Text_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "MachineText4",
        "Direction": "InputOutput",
        "RecordType": "Text_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "AvatarModel2",

```

```

        "Direction": "InputOutput",
        "RecordType": "AvatarModel_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "PSGesture",
        "Direction": "InputOutput",
        "RecordType": "PSGesture_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "AvatarModel3",
        "Direction": "InputOutput",
        "RecordType": "AvatarModel_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "PortableAvatar",
        "Direction": "OutputInput",
        "RecordType": "PortableAvatar_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    }
],
"SubAIMs": [
    {
        "Name": "SpeechSynthesisPS",
        "Identifier": {
            "ImplementerID": "/* String assigned by IIDRA */",
            "Specification": {
                "Standard": "MPAI-MMC",
                "AIW": "",
                "AIM": "SpeechSynthesisPS",
                "Version": "2"
            }
        }
    },
    {
        "Name": "FaceDescription",
        "Identifier": {
            "ImplementerID": "/* String assigned by IIDRA */",
            "Specification": {
                "Standard": "MPAI-PAF",
                "AIW": "",
                "AIM": "FaceDescription",
                "Version": "1"
            }
        }
    },
    {
        "Name": "BodyDescription",
        "Identifier": {
            "ImplementerID": "/* String assigned by IIDRA */",
            "Specification": {
                "Standard": "MPAI-PAF ",
                "AIW": "",
                "AIM": "BodyDescription",
                "Version": "1"
            }
        }
    },
    {
        "Name": "PortableAvatarMultiplexing",
        "Identifier": {
            "ImplementerID": "/* String assigned by IIDRA */",
            "Specification": {
                "Standard": "MPAI-PAF ",

```

```

        "AIW": "",
        "AIM": "PortableAvatar",
        "Version": "1"
    }
}
],
"Topology": [
{
    "_comment": "Input to first AIM column"
},
{
    "Output": {
        "AIMName": "",
        "PortName": " MachineText2"
    },
    "Input": {
        "AIMName": "SpeechSynthesisPS",
        "PortName": " MachineText2"
    }
},
{
    "Output": {
        "AIMName": "",
        "PortName": "PSSpeech"
    },
    "Input": {
        "AIMName": "SpeechSynthesisPS",
        "PortName": "PSSpeech"
    }
},
{
    "_comment": "Input to second AIM column"
},
{
    "Output": {
        "AIMName": "SpeechSynthesisPS",
        "PortName": "MachineSpeech1"
    },
    "Input": {
        "AIMName": "FaceDescription",
        "PortName": "MachineSpeech1"
    }
},
{
    "Output": {
        "AIMName": "",
        "PortName": "AvatarModel1"
    },
    "Input": {
        "AIMName": "FaceDescription",
        "PortName": "AvatarModel1"
    }
},
{
    "Output": {
        "AIMName": "",
        "PortName": "PSFace"
    },
    "Input": {
        "AIMName": "FaceDescription",
        "PortName": "PSFace"
    }
},
{
    "Output": {
        "AIMName": "",
        "PortName": " MachineText3"
    },
    "Input": {
        "AIMName": "FaceDescription",
        "PortName": "MachineText3"
    }
}
],

```

```

{
  "Output":{
    "AIMName": "",
    "PortName": "MachineText4"
  },
  "Input":{
    "AIMName": "BodyDescription",
    "PortName": "MachineText4"
  }
},
{
  "Output":{
    "AIMName": "",
    "PortName": "AvatarModel2"
  },
  "Input":{
    "AIMName": "BodyDescription",
    "PortName": "AvatarModel2"
  }
},
{
  "Output":{
    "AIMName": "",
    "PortName": "PSGesture"
  },
  "Input":{
    "AIMName": "BodyDescription",
    "PortName": "PSGesture"
  }
},
{
  "_comment": "Input to third AIM column"
},
{
  "Output":{
    "AIMName": "",
    "PortName": "AvatarID"
  },
  "Input":{
    "AIMName": "PortableAvatarMultiplexing",
    "PortName": "AvatarID"
  }
},
{
  "Output":{
    "AIMName": "",
    "PortName": "MachineText1"
  },
  "Input":{
    "AIMName": "PortableAvatarMultiplexing",
    "PortName": "MachineText1"
  }
},
{
  "Output":{
    "AIMName": "SpeechSynthesisPS",
    "PortName": "MachineSpeech2"
  },
  "Input":{
    "AIMName": "",
    "PortName": "MachineSpeech2"
  }
},
{
  "Output":{
    "AIMName": "FaceDescription",
    "PortName": "MachineFaceDescriptors2"
  },
  "Input":{
    "AIMName": "PortableAvatarMultiplexing",
    "PortName": "MachineFaceDescriptors2"
  }
},
{

```



```

    "Output":{
      "AIMName":"MachineBodyDescription",
      "PortName":"MachineGestureDescriptors2"
    },
    "Input":{
      "AIMName":"PortableAvatarMultiplexing",
      "PortName":"MachineBodyDescriptors2"
    }
  },
  {
    "Output":{
      "AIMName": "",
      "PortName":"AvatarModel3"
    },
    "Input":{
      "AIMName":"PortableAvatarMultiplexing",
      "PortName":"AvatarModel3"
    }
  },
  {
    "Output":{
      "AIMName":"PortableAvatarMultiplexing",
      "PortName":"PortableAvatar"
    },
    "Input":{
      "AIMName": "",
      "PortName":"PortableAvatar"
    }
  }
],
"Implementations":[
],
"Documentation":[
  {
    "Type":"Tutorial",
    "URI":"https://mpai.community/standards/MPAI-PAF/"
  }
]
}

```

## 5.2 AIM Metadata

The AIM Metadata are available from

SpeechSynthesis: <https://schemas.mpai.community/PAF/V1.0/PSD/SpeechSynthesis.json>  
 FaceDescription: <https://schemas.mpai.community/PAF/V2.0/PSD/FaceDescription.json>  
 BodyDescription: <https://schemas.mpai.community/PAF/V2.0/PSD/BodyDescription.json>  
 PortableAvatarMultiplexing:  
<https://schemas.mpai.community/PAF/V2.0/SRV/PortableAvatarMultiplexing.json>