



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

**N1332**

2023/08/23

**Source** Requirements (ARA)

**Title** Technical Specification - Avatar Representation and Animation (MPAI-ARA) WD  
for Community Comments

**Target** MPAI Members

This document is a working draft of Technical Specification: Avatar Representation and Animation (MPAI-ARA) published with a request for Community Comments. Comments should be sent to the [MPAI Secretariat](#) by 2023/09/27T23:59 UTC to enable MPAI to consider comments for potential inclusion in the final text of the Technical Specification planned to be approved for publication by the 36<sup>th</sup> General Assembly (2023/09/29).

The draft Standard will be presented online on September 07 at 08 and 15 UTC.



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

## MPAI Technical Specification

### Avatar Representation and Animation MPAI-ARA

**WD for Community Comments**

#### **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

## Avatar Representation and Animation

### V1 (WD for Community Comments)

1	Introduction (Informative).....	5
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	Scope of Use Case .....	9
5.2	Client (Transmission side) .....	11
5.2.1	Functions of Client (Transmission side) .....	11
5.2.2	Reference Architecture of Client (Transmission side) .....	11
5.2.3	Input and output data of Client (Transmission side) .....	12
5.2.4	Functions of Client (Transmission side)'s AI Modules .....	13
5.2.5	I/O Data of Client (Transmission side)'s AI Modules .....	13
5.3	Server .....	13
5.3.1	Functions of Server .....	13
5.3.2	Reference Architecture of Server .....	14
5.3.3	I/O data of Server .....	14
5.3.4	Functions of Server AI Modules .....	15
5.3.5	I/O Data of Server AI Modules .....	15
5.4	Virtual Secretary .....	15
5.4.1	Functions of Virtual Secretary .....	15
5.4.2	Reference Architecture .....	16
5.4.3	I/O Data of Virtual Secretary .....	17
5.5	Client (Receiving side).....	17
5.5.1	Functions of Client (Receiving side).....	17
5.5.2	Reference Architecture of Client (Receiving side) .....	17
5.5.3	I/O Data of Client (Receiving side).....	18
5.5.4	Functions of Client (Receiving side)'s AI Modules .....	18
5.5.5	I/O Data of Client (Receiving side)'s AI Modules .....	19
6	Composite AI Modules .....	19
6.1	Personal Status Extraction (PSE) .....	19
6.1.1	Scope of Composite AIM.....	19
6.1.2	Reference architecture .....	19
6.1.3	I/O Data of Personal Status Extraction .....	20
6.2	Personal Status Display (PSD) .....	20
6.2.1	Scope of Composite AIM.....	20
6.2.2	Reference Architecture .....	21
6.2.3	I/O Data of Personal Status Display .....	21
6.2.4	Functions of AI Modules of Personal Status Display .....	21
6.2.5	I/O Data of AI Modules of Personal Status Display .....	22
6.2.6	JSON Metadata of Personal Status Display .....	22
7	Data Formats .....	22
7.1	Environment .....	22

7.2	Body .....	22
7.2.1	Body Model.....	22
7.2.2	Body Descriptors.....	23
7.2.3	Head Descriptors .....	24
7.3	Face .....	24
7.3.1	Face Model.....	24
7.3.2	Face Descriptors .....	24
7.4	Avatars .....	25
7.4.1	Avatar Model.....	25
7.4.2	Avatar Descriptors.....	25
7.5	Scene Descriptors .....	26
7.5.1	Spatial Attitude.....	26
7.5.2	Audio.....	26
7.5.3	Visual .....	26
7.6	Additional Data Types .....	26
7.6.1	Text.....	26
7.6.2	Language identifier .....	27
7.6.3	Meaning.....	27
7.6.4	Personal Status .....	27
Annex 1 - MPAI Basics .....		28
1	General .....	28
2	Governance of the MPAI Ecosystem .....	28
3	AI Framework .....	29
4	Audio-Visual Scene Description .....	30
	Audio Scene Descriptors .....	30
	Visual Scene Descriptors .....	30
Annex 2 - General MPAI Terminology .....		31
Annex 3 - Notices and Disclaimers Concerning MPAI Standards (Informative).....		34
Annex 4 - AIW and AIM Metadata of ABV-CTX .....		36
1	Metadata for ABV-CTX AIW.....	36
2	Metadata for ARA-CTX AIMs .....	42
	Audio Scene Description.....	42
	Visual Scene Description .....	43
	SpeechRecognition.....	45
	LanguageUnderstanding.....	45
	PersonalStatusExtraction.....	46
	AvatarDescription .....	48
Annex 5 - AIW and AIM Metadata of ABV-SRV.....		50
3	AIW metadata for ABV-SRV .....	50
4	Metadata for ABV-SRV AIMs .....	56
2.1	ParticipantAutentication.....	56
2.2	Translation.....	57
Annex 6 - AIW and AIM Metadata of ARA-VSV .....		59
1	Metadata for VSV AIW .....	59
3.	Mtadata for MMC-VSV .....	65
1	Metadata for MMC-VSV AIW .....	65
2.	AIM metadata for ARA-VSV .....	72
2.1	SpeechRecognition.....	72
2.2	AvatarDescriptorParsing .....	73
2.3	LanguageUnderstanding.....	74

2.4	PersonalStatusExtraction.....	75
2.5	Summarisation.....	77
2.6	DialogueProcessing .....	78
2.7	PersonalStatusDisplay .....	80
	Annex 7 - AIW and AIM Metadata of ABV-CRX .....	82
1	AIW metadata for ABV-CRX.....	82
2	Metadata for ABV-CRX AIMs .....	87
	VisualSceneCreation .....	87
	AudioSceneCreation.....	88
	AVSceneViewer.....	89
	Annex 8 - Metadata of Personal Status Display Composite AIM.....	91
1.	Metadata of PersonalStatusDisplay .....	91
1.1	SpeechSynthesisPS.....	97
1.2	FaceDescription.....	98
1.3	BodyDescription.....	99
1.4	AvatarDescription .....	100
1.5	AvatarSynthesisPS .....	101

## 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. For instance, a transmitting client sends data that a remote receiving client should unambiguously interpret to reproduce a digital human as intended by the transmitting client.

These new usage scenarios require forms of standardisation and Technical Specification: Avatar Representation and Animation (in the following “Standard”) is a first response to the need of a user wishing to enable their transmitting client to send data that a remote client can interpret to render a digital human, having the body movement and the facial expression representing their own movements and expression.

The Standard specified technologies that enable the implementation of the Avatar-Based Videconference (ARA-ABV) Use Case where:

1. Remotely located **transmitting clients** send:
  - 1.1 Avatar Models and Language Preferences (at the beginning of the videoconference).
  - 1.2 Avatar Descriptors, and Speech Objects to a Server (continuously).
2. **A Server:**
  - 2.1 Selects an Environment, i.e., a meeting room (at the beginning).
  - 2.2 Equips the room with objects, i.e., meeting table and chairs (at the beginning).
  - 2.3 Places Avatar Models around the table (at the beginning).
  - 2.4 Distributes Environment, Avatars, and their positions to all receiving clients (at the beginning).
  - 2.5 Translates speech objects from participants according to Language Preferences (continuously).
  - 2.6 Sends Avatar Descriptors and Speech Objects to receiving clients (continuously).
3. **Receiving clients:**

- 3.1 Create Audio and Visual Scene Descriptors.
- 3.2 Render the Audio-Visual Scene corresponding to the Point of View selected by the human participant.

MPAI employs the standard MPAI-ARA technologies in other Use Cases such as Human-Connected Autonomous Vehicle (CAV) Interaction (MMC-HCI) and plans on using them in future versions of the MPAI Metaverse Model (MPAI-MMM) project.

## 2 Scope

Technical Specification: Avatar Representation and Animation (MPAI-ARA) specifies the technologies enabling the implementation of the Avatar-Based Videoconference Use Case specified in Chapter 5 – Avatar-Based Videoconference. Specifically, it enables the Digital Representation of:

- 1 A **Model** of a Digital Human.
- 2 The **Descriptors** of human faces and bodies.
- 3 The **Animation** of a Digital Human Model using the Descriptors captured from a human face and body.

The Avatar-Based Videoconference Use Case requires technologies standardised by 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.

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

1. An *AIW*, shall:
  - a. Perform the AIW function specified in the appropriate Section of Chapter.
  - b. All AIMs, their topology and connections should conform with the AIW Architecture specified in the appropriate Section of Chapter.
  - c. The AIW and AIM input and output data should have the formats specified in the appropriate Subsection of Section.
2. An *AIM*, shall:
  - a. Perform the AIM function specified by the appropriate section of Chapter.
  - b. Receive and produce the data specified in the appropriate Subsection of Section.
  - c. Receive as input and produce as output data having the format specified in Section.
3. A data *Format*, the data shall have the format specified in Section.

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 specification of this Standard.
6. Implementers and Users should consider the notices and disclaimers of Annex 2.

The current version of the Standard has been developed by the Requirements Standing Committee. MPAI may issue new versions of MPAI-ARA extending or replacing the current Standard.

### 3 Terms and Definitions

In this document, the words beginning with a capital letter are defined in *Table 1*; words beginning with a small letter have the normal meaning consistent with the relevant context. If a Term in *Table 1* is preceded by a dash “-”, it means the following:

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

*Table 1 – 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.
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	A Physical or Digital space.
Environment Model	The static audio and visual components of the Environment, e.g., walls, table, and chairs.
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.

<b>Gesture</b>	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.
<b>Grade</b>	The intensity of a Factor.
<b>Human</b>	
- <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.
<b>Meaning</b>	Information extracted from Text such as syntactic and semantic information, and Personal Status.
<b>Modality</b>	One of Text, Speech, Face, or Gesture.
<b>Object</b>	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>De-criptor</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.
<b>Orientation</b>	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° counter-clockwise (right-to-left) with the x axis and its z axis pointing up toward the viewer.
<b>Persona</b>	A manifestation of a human as a rendered Digital Human.
<b>Personal Status</b>	The ensemble of information internal to a person, including Emotion, Cognitive State, and Attitude.
<b>Point of View</b>	The Spatial Attitude of a Digital Human watching the Environment.
<b>Position</b>	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).
<b>Scene</b>	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 as Spatial Attitude.
- <i>Visual</i>	The Visual Objects of an Environment with Object metadata such as as Spatial Attitude.
- <i>Presentation</i>	The rendering of a Scene in a format suitable for human perception.
<b>Text</b>	A sequence of characters drawn from a finite alphabet.
<b>Representation</b>	Data that digitally represent an entity of a Real Environment.

## 4 References

### 4.1 Normative References

This standard normatively references the following documents, both from MPAI and other standards organisations. MPAI standards are publicly available at .

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

### 4.2 Informative References

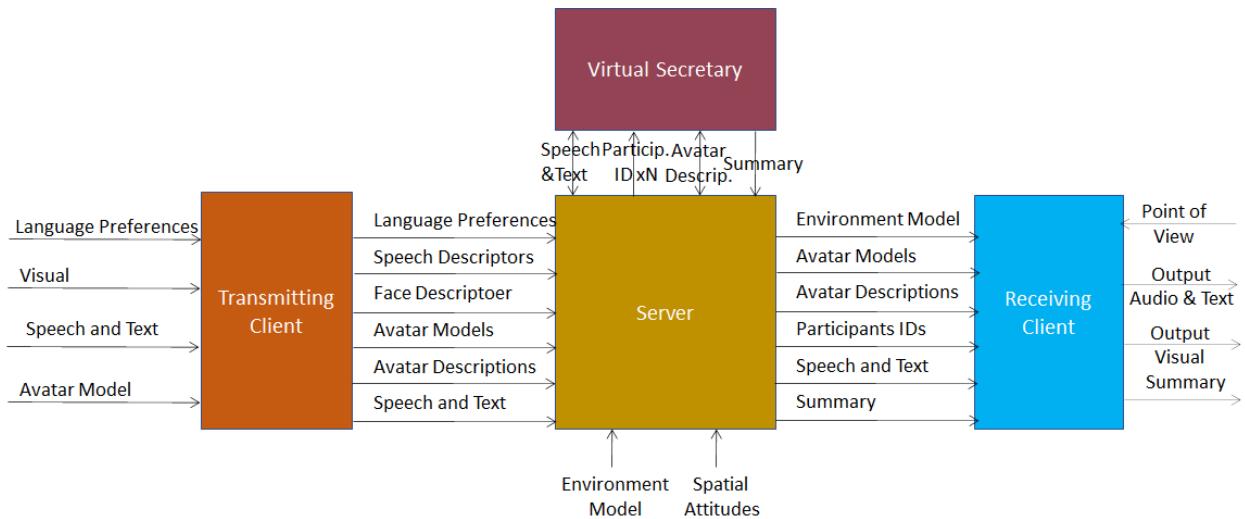
These references are provided for information purposes.

13. MPAI; Published MPAI Standards; <https://mpai.community/standards/resources/>.
14. Ekman, Paul (1999), "Basic Emotions", in Dalgleish, T; Power, M (eds.), Handbook of Cognition and Emotion (PDF), Sussex, UK: John Wiley & Sons.

## 5 Avatar-Based Videoconference

### 5.1 Scope of Use Case

Figure 1 depicts the components of the system supporting the conference of a group of humans participating through avatars having their visual appearance and uttering the participants' real voice meeting in a virtual environment.



*Figure 1 – Avatar-Based Videoconference end-to-end diagram*

This is the workflow of the conference:

1. Geographically separated humans, some of which are co-located in the same room, participate in a conference held in a Virtual Environment where they are represented by avatars whose faces have a visual appearance highly similar with theirs.
2. The members of a co-located group of humans participate in the Virtual Environment as individual avatars.
3. A Virtual Secretary avatar not corresponding to any participant attends the conference.
4. The Virtual Environment is equipped with a table and an appropriate number of chairs.
5. At the beginning of the conference,
  - 5.1. Participants send to the Server:
    - 5.1.1. The Descriptors of their face and speech for authentication.
    - 5.1.2. Their own Avatar Models.
    - 5.1.3. Their language preferences.
  - 5.2. The Server
    - 5.2.1. Selects the Visual Environment Model.
    - 5.2.2. Authenticates participants using their speech and face Descriptors.
    - 5.2.3. Assigns IDs to authenticated participants.
    - 5.2.4. Sets the positions of the participants' and Virtual Secretary's Avatars on the chairs.
    - 5.2.5. Sets the common conference language.
    - 5.2.6. Sends the Environment Model, the Avatar Models, participant IDs to the Server.
6. During the conference:
  - 6.1. Participants send to the Server:
    - 6.1.1. Their Utterances.
    - 6.1.2. The compressed Descriptors of their bodily motion and facial expressions (compressed Avatar Descriptors).
  - 6.2. The Server:
    - 6.2.1. Translates the speech signals to the requested languages based on the language preferences.
    - 6.2.2. Forwards the participants' IDs, translated utterances and compressed Avatar Descriptors to participants' clients and the Virtual Secretary.
  - 6.3. The Virtual Secretary:
    - 6.3.1. Works on the common meeting language
    - 6.3.2. Collects the statements made by participating avatars while monitoring the avatars' Personal Statuses conveyed by their speech, face, and gesture.

- 6.3.3. Makes a summary by combining all recognised texts and Personal Statuses.
  - 6.3.4. Displays the summary in the Environment for avatars to read and edit the Summary directly.
  - 6.3.5. Alternatively, edits the Summary based on Text-and-Speech conversations with avatars using the avatars' Personal Statuses conveyed by Text, Speech, Face and Gesture.
  - 6.3.6. Sends the synthetic Speech and compressed Avatar Descriptors to the Server.
  - 6.4. The Server forwards the Virtual Secretary's synthetic Speech and compressed Avatar Descriptors to the participants' clients.
7. The Receiving Clients:
- 7.1. Decompress the compressed Avatar Descriptors.
  - 7.2. Synthesise the Avatars.
  - 7.3. Render the Visual Scene.
  - 7.4. Render the Audio Scene by spatially adding the participants' utterances to the Spatial Attitude of the respective avatars' mouths.
8. The rendering of the Audio and Visual Scene may be done from a Point of View, possibly different from the position assigned to their Avatars in the Environment, selected by participant who use a device of their choice (HMD or 2D display/earpad).

## **5.2 Client (Transmission side)**

### **5.2.1 Functions of Client (Transmission side)**

The function of a Transmitting Client is to:

- 1. Receive:
  - a. Input Audio from the microphone (array).
  - b. Input Video from the camera (array).
  - c. Participant's Avatar Model.
  - d. Participant's spoken language preferences (e.g., EN-US, IT-CH).
- 2. Send to the Server:
  - a. Speech Descriptors (for Authentication).
  - b. Face Descriptors (for Authentication).
  - c. Participant's spoken language preferences.
  - d. Speech.
  - e. Avatar Model.
  - f. Compressed Avatar Descriptors.

### **5.2.2 Reference Architecture of Client (Transmission side)**

Figure 2 gives the architecture of Transmitting Client AIW. Red text refers to data sent at meeting start.

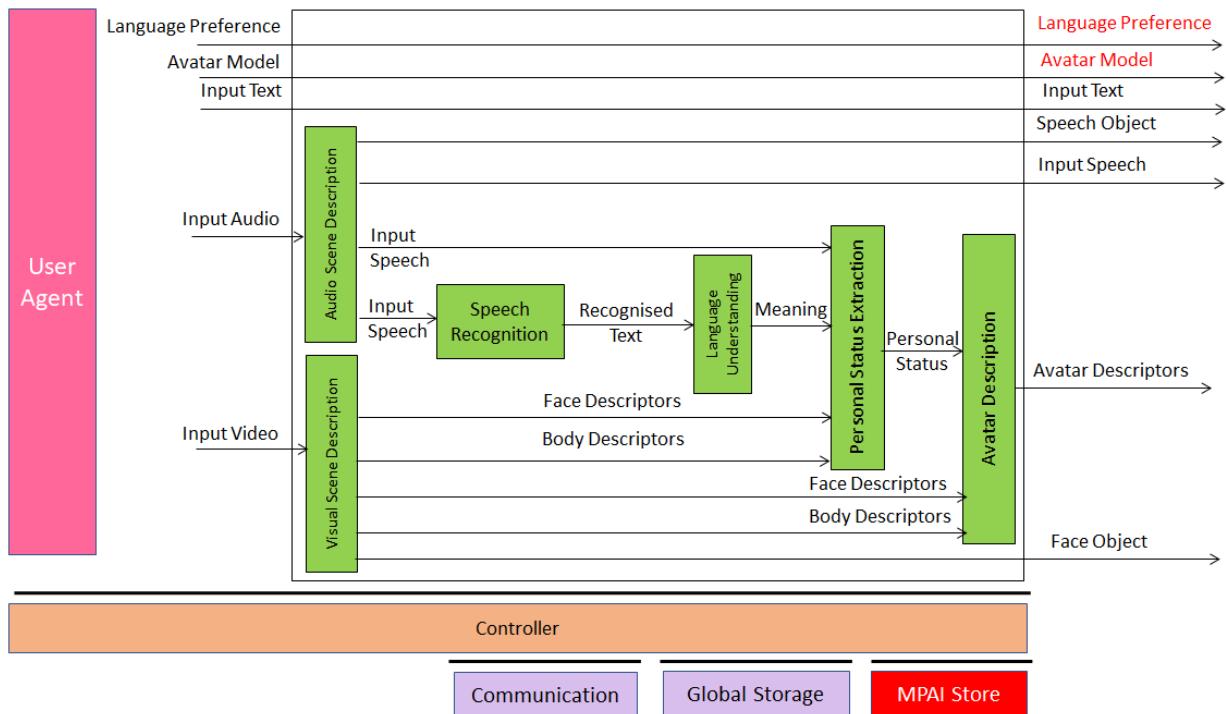


Figure 2 – Reference Model of Avatar Videoconference Transmitting Client

At the start, each participant sends to the Server:

1. Language preferences
2. Avatar model.

During the meeting

1. The following AIMs of the Transmitting Clients produce:
  - 1.1. Audio Scene Description: Audio Scene Descriptors.
  - 1.2. Visual Scene Description: Visual Scene Descriptors.
  - 1.3. Speech Recognition: Recognised Text.
  - 1.4. Face Description: Face Descriptors.
  - 1.5. Body Description: Body Descriptors.
  - 1.6. Personal Status Extraction: Personal Status.
  - 1.7. Language Understanding: Meaning.
  - 1.8. Avatar Description: Avatar Descriptors.
2. The Transmitting Clients send to the Server for distribution to all participants:
  - 2.1. Speech.
  - 2.2. Avatar Descriptors.

### 5.2.3 Input and output data of Client (Transmission side)

Table 2 gives the input and output data of the Transmitting Client AIW:

Table 2 – Input and output data of Client Transmitting AIW

Input	Comments
Text	Chat text used to communicate with Virtual Secretary or other participants
Language Preference	The language participant wishes to speak and hear at the videoconference.
Input Audio	Audio of participant's Speech and Environment Audio.

Input Video	Video of participants' upper part of the body.
Avatar Model	The avatar model selected by the participant.
<b>Output</b>	<b>Comments</b>
Language Preference	As in input.
Participant's Speech	Speech as separated from Environment Audio.
Compressed Avatar Descriptors	Compressed Descriptors produced by Transmitting Client.

#### 5.2.4 Functions of Client (Transmission side)'s AI Modules

Table 4 gives the functions of AI Modules of Transmitting Client AIW.

*Table 3 – AI Modules of Client (Transmission side) AIW*

AIM	Input
<b>Audio Scene Description</b>	Provides audio objects and their scene geometry.
<b>Visual Scene Description</b>	Provides visual objects and their scene geometry.
<b>Speech Recognition</b>	Recognises the speech of a human.
<b>Language Understanding</b>	Extracts the Meaning of the Recognised Text.
<b>Personal Status Extraction</b>	Extracts the Personal Status of Speech, Meaning, and Face and Body Descriptors.
<b>Avatar Description</b>	Provides the Description of the human represented by the Avatar.

#### 5.2.5 I/O Data of Client (Transmission side)'s AI Modules

Table 4 gives the AI Modules of Transmitting Client AIW.

*Table 4 – AI Modules of Client (Transmission side) AIW*

AIM	Input	Output
<b>Audio Scene Description</b>	Input Audio	Audio Scene Descriptors
<b>Visual Scene Description</b>	Input Video	Visual Scene Descriptors
<b>Speech Recognition</b>	Speech Objects	Recognised Text
<b>Language Understanding</b>	Recognised Text	Refined Text Meaning
<b>Personal Status Extraction</b>	Recognised Text Speech Face Object Human Object	Personal Status
<b>Avatar Description</b>	Meaning Personal Status Face Descriptors Gesture Descriptors	. Compressed Avatar Descriptors.

### 5.3 Server

#### 5.3.1 Functions of Server

The Server:

1. At the start:
  - 1.1. Selects an Environment Model.
  - 1.2. Selects the positions of the participants' Avatar Models.
  - 1.3. Authenticates Participants.

- 1.4. Selects the common meeting language.
2. During the videoconference
  - 2.1. Receive participants' text, speech, and compressed Avatar Descriptors.
  - 2.2. Translate participants' speech signals according to their language preferences.
  - 2.3. Send participants' text, speech translated to the common meeting language, and compressed Avatar Descriptors to Virtual Secretary.
  - 2.4. Receive text, speech, and compressed Avatar Descriptors from Virtual Secretary.
  - 2.5. Translate Virtual Secretary's speech signal according to each participant's language preferences.
  - 2.6. Send participants' and Virtual Secretary's text, translated speech, and compressed Avatar Descriptors to participants' clients.

### 5.3.2 Reference Architecture of Server

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

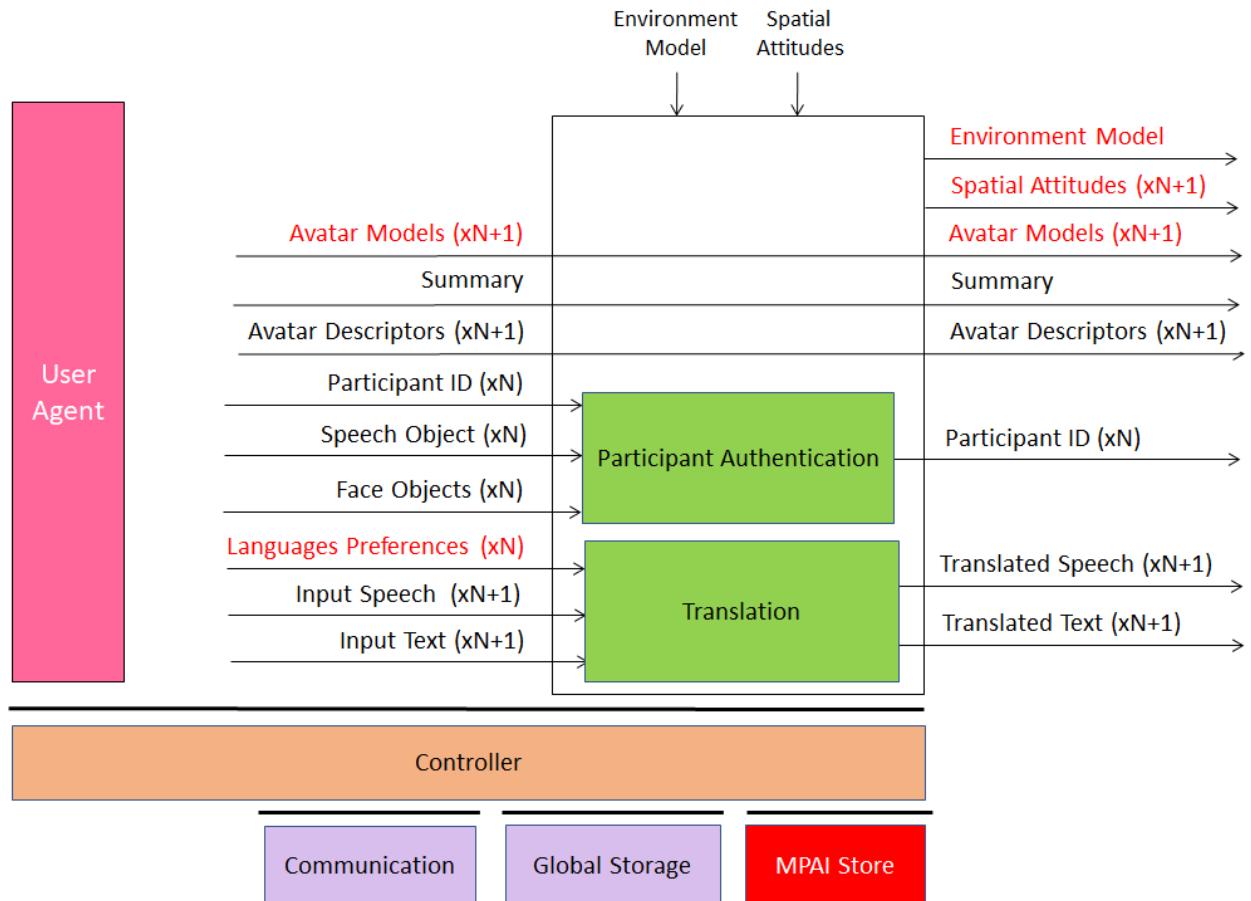


Figure 3 – Reference Model of Avatar-Based Videoconference Server

### 5.3.3 I/O data of Server

Table 5 gives the input and output data of Server AIW.

Table 5 – Input and output data of Server AIW

Input	Comments
Participant Identities (xN)	Assigned by Conference Manager
Speech Object (xN)	Participant's Speech Object for Authentication
Face Object (xN)	Participant's Face Object for Authentication

Selected Languages (xN)	From all participants
Speech (xN+1)	From all participants and Virtual Secretary
Text (xN+1)	From all participants and Virtual Secretary
Avatar Model (xN+1)	From all participants and Virtual Secretary
Avatar Descriptors (xN+1)	From all participants and Virtual Secretary
Summary	From Virtual Secretary
Outputs	Comments
Environment Model	From Server Manager
Avatar Model (xN+1)	From all participants and Virtual Secretary
Avatar Descriptors (xN+1)	Participants + Virtual Secretary Compressed Avatar D.
Participant ID (xN+1)	Participants + Virtual Secretary IDs
Speech (xN+1)	Participants + Virtual Secretary Speech
Text (xN+1)	Participants + Virtual Secretary Speech

### 5.3.4 Functions of Server AI Modules

Figure 4 gives the AI Modules of the Server AIW.

*Table 6 - AI Modules of Server AIW*

AIM	Functions
<b>Participant Authentication</b>	Authenticates Participants using their Speech.
<b>Text and Speech Translation</b>	For all participants 1. Selects an active speech and text streams. 2. Translates the Speech and Text in the Selected Languages. 3. Assigns a translated Speech to the appropriate set of Participants.

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

Figure 4 gives the AI Modules of the Server AIW.

*Table 7 - AI Modules of Server AIW*

AIM	Input	Output
<b>Participant Authentication</b>	Speech Descriptors Face Descriptors	Participant Authentication
<b>Text and Speech Translation</b>	Language Preferences Text Speech	Translates Text Translated Speech

## 5.4 Virtual Secretary

### 5.4.1 Functions of Virtual Secretary

The functions of the Virtual Secretary are to:

1. Listen to the Speech of each avatar.
2. Synthesise Avatars using compressed Avatar Descriptors.
3. Compute 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 (out of verbal conversation, i.e., via chat).
  - 2.2. The Virtual Secretary edits the Summary interpreting Speech, Text, and the avatars' Personal Statuses.

Reference [4] specifies the Personal Status Extraction Composite AIM.

#### 5.4.2 Reference Architecture

Figure 4 depicts the architecture of the Virtual Secretary AIW. 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.

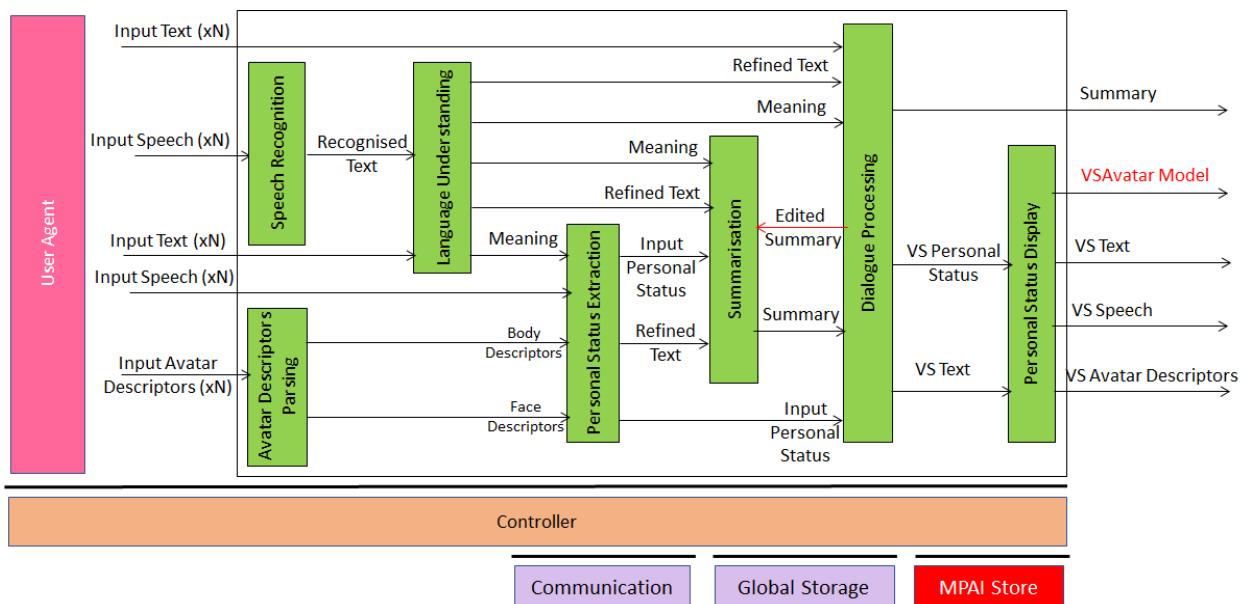


Figure 4 – Reference Model of Virtual Secretary

The Virtual Secretary workflow operates as follows:

1. Speech Recognition extracts Text from an avatar speech.
2. Visual Scene Description provides the N Face Descriptors and N Body Descriptors.
3. Personal Status Extraction extracts Personal Status from Meaning, Speech, Face Descriptors, and Body Descriptors.
4. Language Understanding:
  - 4.1. Receives Personal Status and Recognised Text.
  - 4.2. Creates
    - 4.2.1. Refined Text.
    - 4.2.2. Meaning of the sentence uttered by an avatar.
5. Summarisation
  - 5.1. Receives:
    - 5.1.1. Refined Text.
    - 5.1.2. Personal Status.
    - 5.1.3. Meaning.

- 5.2. Creates Summary expressed by Text in the meeting's common language and graphical symbols.
- 5.3. Receives Edited Summary from Dialogue Processing.
- 6. Dialogue Processing
  - 6.1. Receives
    - 6.1.1. Refined Text.
    - 6.1.2. Text from an avatar via chat.
    - 6.1.3. Meaning.
    - 6.1.4. Summary.
  - 6.2. Creates Edited Summary.
  - 6.3. Sends Edited Summary back to Summarisation.
  - 6.4. Outputs Text and Personal Status.
- 7. Personal Status Display
  - 7.1. Forwards Virtual Secretary's Text.
  - 7.2. Utters a synthesised speech from Outputs Text with appropriate Personal Status.
  - 7.3. Generates Virtual Secretary's avatar visually showing Personal Status represented as compressed Avatar Descriptors.

#### **5.4.3 I/O Data of Virtual Secretary**

Table 6 gives the input and output data of Virtual Secretary Composite AIM.

*Table 8 - I/O data of Virtual Secretary*

<b>Input data</b>	<b>From</b>	<b>Comment</b>
Text (xN)	Server	Remarks on the summary, etc.
Speech (xN)	Server	Utterances by avatars
Input Avatar Descriptors	Server	Separate for Face and Gesture
<b>Output data</b>	<b>To</b>	<b>Comments</b>
Summary	Avatars	Summary of avatars' interventions
VS Avatar Model	Application	
VS Speech	Avatars	Speech to avatars
VS Text	Avatars	Response to chat.
VS Avatar Descriptors	Avatars	Face to avatars

#### **5.5 Client (Receiving side)**

##### **5.5.1 Functions of Client (Receiving side)**

The Function of the Client (Receiving Side) is to:

- 1. Create the Environment using the Environment Model.
- 2. Place and animate the Avatar Models at their Spatial Attitudes.
- 3. Add the relevant Speech to each Avatar.
- 4. Render the Audio-Visual Scene as seen from the participant-selected Point of View.

##### **5.5.2 Reference Architecture of Client (Receiving side)**

The Receiving Client:

- 1. Creates the AV Scene using:
  - 1.1. The Environment Model.
  - 1.2. The Avatar Models and Avatar Descriptors.
  - 1.3. The Speech of each Avatar.

2. Presents the Audio-Visual Scene based on the selected Point of View in the Environment.

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

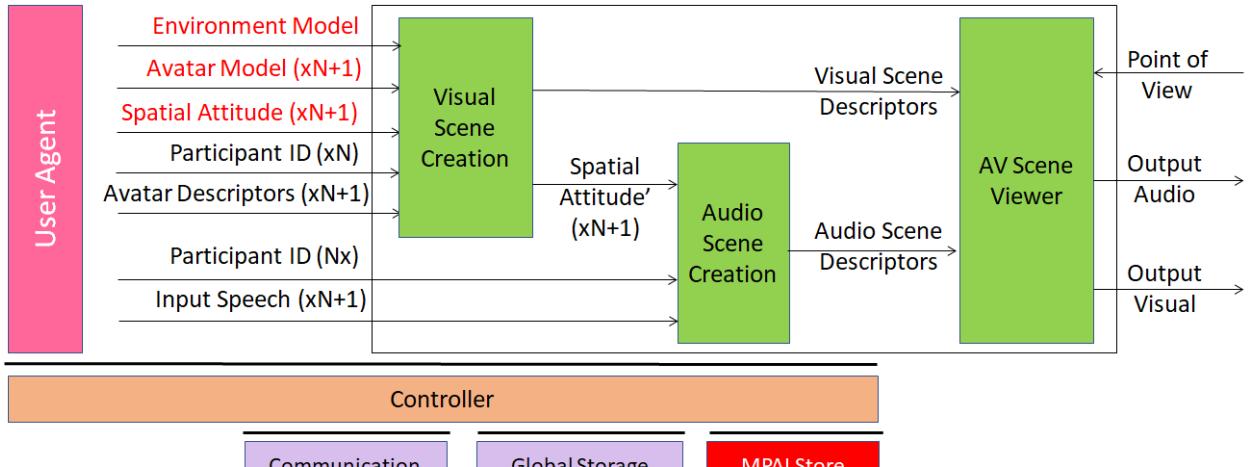


Figure 5 – Reference Model of Avatar-Based Videoconference Client (Receiving Side)

An implementation may decide to display text with the visual image for accessibility purposes.

### 5.5.3 I/O Data of Client (Receiving side)

Table 9 gives the input and output data of Client (Receiving Side) AIW.

Table 9 – Input and output data of Client (Receiving Side) AIW

Input	Comments
Point of View	Participant-selected point to see visual objects and hear audio objects in the Virtual Environment.
Spatial Attitudes (xN+1)	Avatars' Positions and Orientations in Environment.
Participant IDs (xN)	Unique Participants' IDs
Speech (xN+1)	Participant's Speech (e.g., translated).
Environment Model	Environment Model.
Compressed Avatar Descriptors (xN+1)	Descriptors of animated Avatars.
Output	Comments
Output Audio	Presented using loudspeaker (array)/earphones.
Output Visual	Presented using 2D or 3D display.

### 5.5.4 Functions of Client (Receiving side)'s AI Modules

Figure 9 gives the AI Modules of Client (Receiving Side) AIW.

Table 10 - AI Modules of Client (Receiving Side)

AIM	Input
<b>Audio Scene Creation</b>	Creates the Audio Scene
<b>Visual Scene Creation</b>	Creates the Visual Scene
<b>AV Scene Viewer</b>	Renders the AV Scene

### 5.5.5 I/O Data of Client (Receiving side)'s AI Modules

Figure 9 gives the AI Modules of Client (Receiving Side) AIW.

*Table 11 - AI Modules of Client (Receiving Side)*

AIM	Input	Output
<b>Audio Scene Creation</b>	Spatial Attitudes (xN+1) Participants IDs (xN) Input Speech (xN+1)	Audio Scene
<b>Visual Scene Creation</b>	Environment Model Avatar Models (xN+1) Spatial Attitudes (xN+1) Participants IDs (xN) Avatar Descriptors (xN+1)	Visual Scene Spatial Attitudes' (xN+1)
<b>AV Scene Viewer</b>	Audio Scene Descriptors Visual Scene Descriptors Point of View	Output Audio Output Video

## 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)

Reference [4] specifies the Personal Status Extraction Composite AIM. Here only the Scope, Reference Model and Input/Output Data are reported.

#### 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.

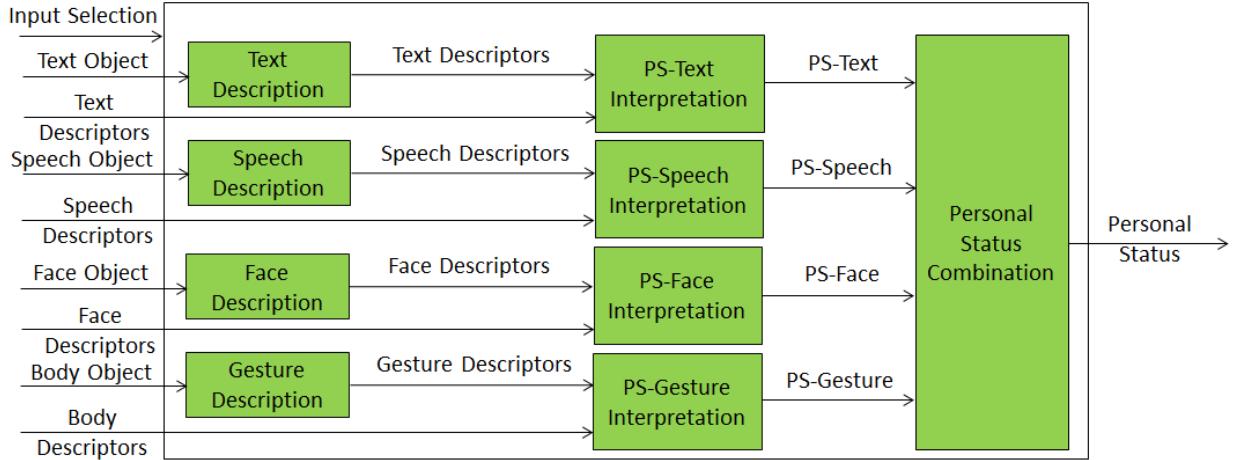
#### 6.1.2 Reference architecture

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 6 depicts the Personal Status estimation process:

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



*Figure 6 – Reference Model of Personal Status Extraction*

Figure 6 represents the possibility that PSE receives some 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 12* gives the input/output data of Personal Status Extraction.

*Table 12 – 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	
Speech	Microphone	Speech of human.
Speech Descriptors	An upstream AIM	
Face Object	Visual Scene Description	The face of the human.
Face Descriptors	An upstream AIM	
Body Object	Visual Scene Description	The upper part of the body.
Body Descriptors	An upstream AIM	
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

A Personal Status Display (PSD) is a Composite AIM receiving Text and Personal Status and generating an avatar producing Text and uttering Speech with the intended Personal Status while the avatar's Face and Gesture show the intended Personal Status. Instead of a ready-to-render avatar, the output can be provided as Avatar Descriptors. The Personal Status driving the avatar can be extracted from a human or can be synthetically generated by a machine as a result of its conversation with a human or another avatar. This Composite AIM is used in the Use Case figures of this document as a replacement for the combination of the AIMs depicted in Figure 7.

## 6.2.2 Reference Architecture

Figure 7 represents the AIMs required to implement Personal Status Display.

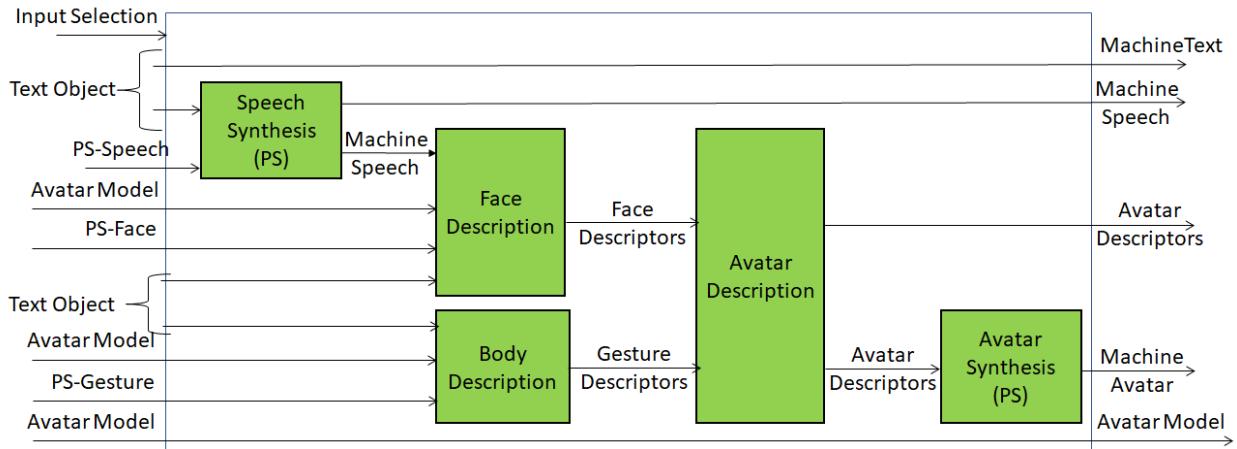


Figure 7 – Reference Model of Personal Status Display

The Personal Status Display operates as follows:

1. Selection determines the type of avatar output – ready-to-render avatar or avatar descriptors.
2. Text is passed as output and synthesised as Speech using the Personal Status provided by PS-Speech)
3. Machine Speech and PS-Face are used to produce the Face Descriptors.
4. PS-Gesture and Text are used for Body Descriptors using the Avatar Model.
5. Avatar Description produces a complete set of Avatar Descriptors (Body and Face).
6. Avatar Synthesis produces a ready-to-render Avatar.

## 6.2.3 I/O Data of Personal Status Display

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

Table 13 – I/O data of Personal Status Display

Input data	From	Comment
Selection	Switch	PSD output
Text Object	Keyboard, Speech Recognition, Machine	
PS-Speech	Personal Status Extractor or Machine	
Avatar Model	From AIM/AIW or embedded	
PS-Face	Personal Status Extractor or Machine	
PS-Gesture	Personal Status Extractor or Machine	
Output data	To	Comments
Machine Text	Human or Avatar (i.e., an AIM)	
Machine Speech	Human or Avatar (i.e., an AIM)	
Compressed Descriptors	AIM/AIW downstream	
Body Object	Presentation Device	Ready-to-render Avatar
Avatar Model	As in input	

## 6.2.4 Functions of AI Modules of Personal Status Display

Table 14 gives functions of the AIMs.

*Table 14 – AI Modules of Personal Status Extraction*

AIM	Functions
<b>Speech Synthesis (PS)</b>	Synthesises Text with Personal Status.
<b>Face Description</b>	Produces the Face Descriptors.
<b>Body Description</b>	Produces the Body Descriptors.
<b>Avatar Description</b>	Produces the Avatar Descriptors.
<b>Descriptor Compression</b>	Compresses the Visual Avatar Descriptors.
<b>Avatar Synthesis</b>	Produces the Avatar.

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

Error! Reference source not found. *Table 15* gives the list of AIMs with their functions.

*Table 15 – 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	Face Descriptors
<b>Gesture Description</b>	Avatar Model Text and Machine PS-Gesture	Body Descriptors
<b>Avatar Description</b>	Face Descriptors Body Descriptors	Avatar Descriptors
<b>Avatar Synthesis</b>	Avatar Descriptors	Avatar

### 6.2.6 JSON Metadata of Personal Status Display

Specified in Section.

## 7 Data Formats

### 7.1 Environment

The Environment represents:

1. A bounded or unbounded space, e.g., a room, a public square surrounded by buildings, etc.
2. Generic objects (e.g., table and chairs).

It is represented according to glTF syntax and transmitted as a file at the beginning of the Avatar-Based Videoconference.

### 7.2 Body

#### 7.2.1 Body Model

MPAI adopts the Humanoid animation (H-Anim) architecture [7]. An implementation of H-Anim 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 by giving access to the joint and end-effector hierarchy of a human figure.

The structure of a humanoid character model 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 H-Anim 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 (integral part of the model).

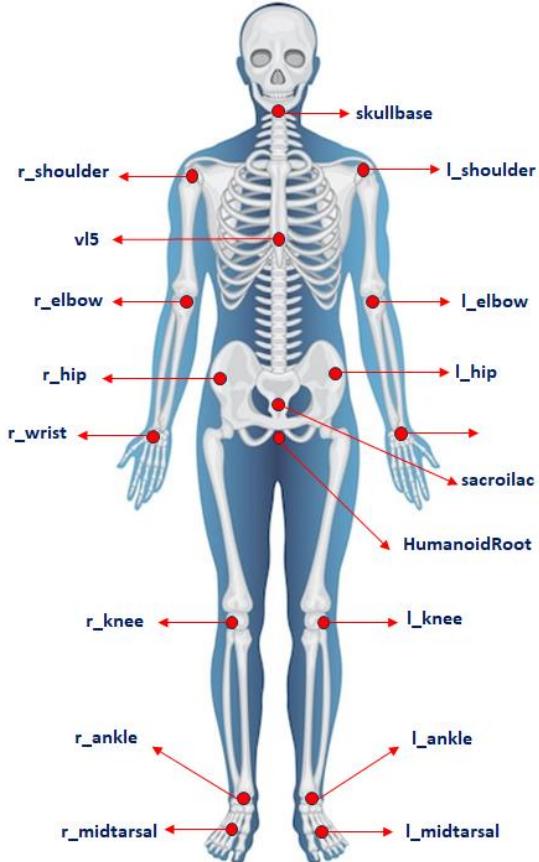


Figure 8 - Some joints of the Body Model

The Body Model is transmitted as a file at the beginning of the Avatar-Based Videoconference in glTF format.

### 7.2.2 Body Descriptors

Body Descriptors are included in the data set describing the root and joints movement in the form of a data sequence representing the delta value of the set of following parameters at the actual time vs the preceding time:

- 1 Position and Orientation of the root with respect to the Position at the preceding time.
- 2 Rotation angle of the y axis in Figure 9.
- 3 Rotation angles of the joints.
- 4 (The rotation of the head is treated as any other joint).

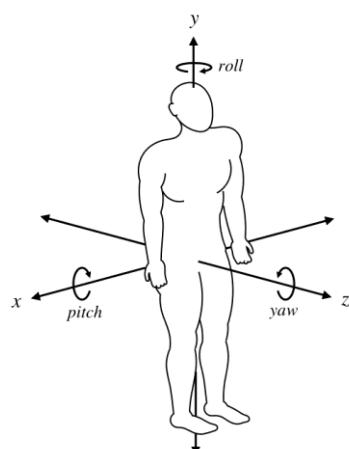


Figure 9 – Pitch, Roll, and Yaw of Body

### 7.2.3 Head Descriptors

The Head is described by:

Roll: head moves toward one of the shoulders.

Pitch: head moves up and down.

Yaw: head rotates left to right (around the vertical axis of the head).

Figure 10 depicts Roll, Pitch, and Yaw of a Head.

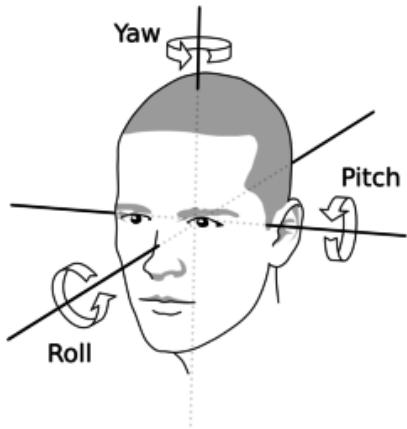


Figure 10 - Roll, Pitch, and Yaw of human head

## 7.3 Face

### 7.3.1 Face Model

The Face Model is represented according to the glTF syntax.

### 7.3.2 Face Descriptors

MPAI adopts as Face Descriptors the Actions Units of the Facial Action Coding System (FACS) initially proposed by [14].

AU	Description	Facial muscle
<u>1</u>	Inner Brow Raiser	<i>Frontalis, pars medialis</i>
<u>2</u>	Outer Brow Raiser	<i>Frontalis, pars lateralis</i>
<u>4</u>	Brow Lowerer	<i>Corrugator supercilii, Depressor supercilii</i>
<u>5</u>	Upper Lid Raiser	<i>Levator palpebrae superioris</i>
<u>6</u>	Cheek Raiser	<i>Orbicularis oculi, pars orbitalis</i>
<u>7</u>	Lid Tightener	<i>Orbicularis oculi, pars palpebralis</i>
<u>9</u>	Nose Wrinkler	<i>Levator labii superioris alaqueae nasi</i>
<u>10</u>	Upper Lip Raiser	<i>Levator labii superioris</i>
11	Nasolabial Deepener	<i>Zygomaticus minor</i>
<u>12</u>	Lip Corner Puller	<i>Zygomaticus major</i>
13	Cheek Puffer	<i>Levator anguli oris (a.k.a. Caninus)</i>
14	Dimpler	<i>Buccinator</i>
<u>15</u>	Lip Corner Depressor	<i>Depressor anguli oris (a.k.a. Triangularis)</i>
16	Lower Lip Depressor	<i>Depressor labii inferioris</i>
<u>17</u>	Chin Raiser	<i>Mentalis</i>

<u>18</u>	Lip Puckerer	<i>Incisivii labii superioris and Incisivii labii inferioris</i>
<u>20</u>	Lip stretcher	<i>Risorius with platysma</i>
22	Lip Funneler	<i>Orbicularis oris</i>
<u>23</u>	Lip Tightener	<i>Orbicularis oris</i>
<u>24</u>	Lip Pressor	<i>Orbicularis oris</i>
<u>25</u>	Lips part**	<i>Depressor labii inferioris or relaxation of Mentalis, or Orbicularis oris</i>
<u>26</u>	Jaw Drop	<i>Masseter, relaxed Temporalis and internal Pterygoid</i>
<u>27</u>	Mouth Stretch	<i>Pterygoids, Digastric</i>
28	Lip Suck	<i>Orbicularis oris</i>
41	Lid droop**	<i>Relaxation of Levator palpebrae superioris</i>
42	Slit	<i>Orbicularis oculi</i>
43	Eyes Closed	<i>Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis</i>
44	Squint	<i>Orbicularis oculi, pars palpebralis</i>
45	Blink	<i>Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis</i>
46	Wink	<i>Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis</i>
61	Eyes turn left	
62	Eyes turn right	
63	Eyes up	
64	Eyes down	

## 7.4 Avatars

### 7.4.1 Avatar Model

The Avatar Model combines the Body and Face Models. It is transmitted as a file at the beginning of the Avatar-Based Videoconference.

### 7.4.2 Avatar Descriptors

Avatar Descriptors is a data stream including:

*Table 16 – Variables composing the Avatar Descriptors*

Variable name	Code
Timestamp type	Absolute/relative
Timestamp value	In seconds
Space type	Absolute/relative
Unit of measure	Metres
Spatial Attitude	
Body Descriptors	

Face Descriptors	
Speech Segment	
Text snippet	

```

"$schema": "http://json-schema.org/draft-07/schema#",
"title": "Personal Status",
"type": "object",
"properties": {
    "Timestamp": {
        "type": "object",
        "properties": {
            "Timestamp type": {
                "type": "string"
            },
            "Timestamp value": {
                "type": "string",
                "oneOf": [
                    { "format" : "date-time" },
                    { "const" : "0" }
                ]
            }
        },
        "required": ["Timestamp value"],
        "if": {
            "properties": { "Timestamp value": { "const": "0" } }
        },
        "then": {
            "properties": { "Timestamp type": { "type": "null" } }
        },
        "else": {
            "required": ["Timestamp type"]
        }
    }
},

```

## 7.5 Scene Descriptors

### 7.5.1 Spatial Attitude

Spatial Attitude of an Object is specified in MPAI-OSD V1 [5]

### 7.5.2 Audio

Audio Scene Descriptors are specified in MPAI-CAE V2 [3]. They describe a sound field containing speech sources with:

1. SpeechID: Speech source ID
2. ChannelID: Channel ID
3. AzimuthDirection: Azimuth direction in degrees.
4. ElevationDirection: Elevation direction in degrees.
5. Distance: Distance in m.
6. DistanceFlag: 0: Valid, 1: NonValid.

### 7.5.3 Visual

A Visual Scene is Described according to glTF [6]. It is produced by the Client (Receiving part). The Spatial Attitude of a Body is defined with respect to a set of Cartesian axes.

## 7.6 Additional Data Types

### 7.6.1 Text

Specified in MPAI-MMC V2 [4].

### **7.6.2 Language identifier**

Specified in MPAI-MMC V2 [4].

### **7.6.3 Meaning**

Specified in MPAI-MMC V2 [4].

### **7.6.4 Personal Status**

Specified in MPAI-MMC V2 [4].

## Annex 1 - MPAI Basics

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

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.

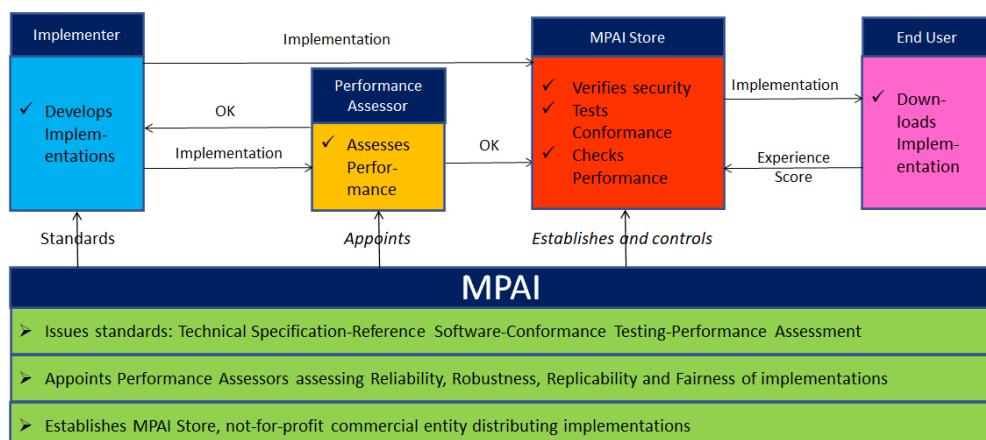
### 2 Governance of the MPAI Ecosystem

The technical foundations of the MPAI Ecosystem are currently provided by the following documents 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*

Technical Specification: Governance of the MPAI Ecosystem Table 17 identifies the following roles in the MPAI Ecosystem:

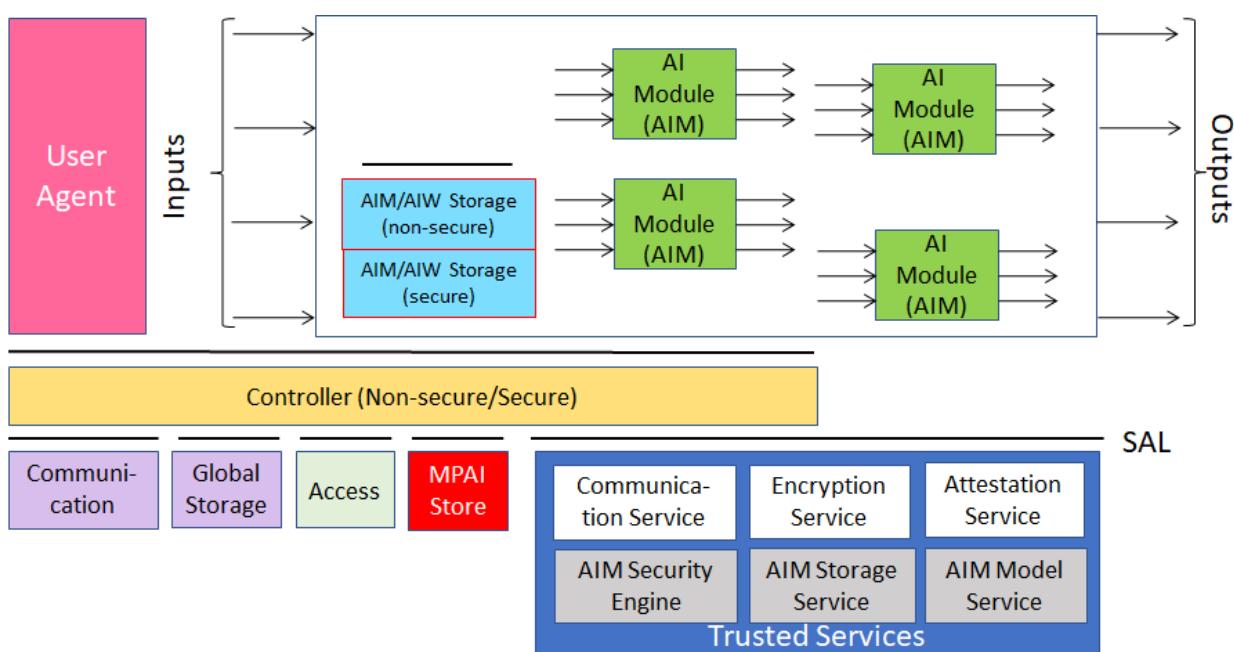
*Table 17 - Roles in the MPAI Ecosystem*

MPAI	Publishes Standards. Establishes the not-for-profit MPAI Store. Appoints Performance Assessors.
Implementers	Submit Implementations to Performance Assessors.
Performance Assessors	Inform Implementation submitters and the MPAI Store if Implementation Performance is acceptable.
Implementers	Submit Implementations to the MPAI Store.
MPAI Store	Assign unique ImplementerIDs (IID) to Implementers 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 [2].



*Figure 12 – The AI Framework (AIF) Reference Model*

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

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

MPAI also defines 3 Interoperability Levels of an AIF that executes an AIW. Table 18 gives the characteristics of an AIW and its AIMs of a given Level:

*Table 18 - MPAI Interoperability Levels*

Level	AIW	AIMs
1	An implementation of a use case	Implementations able to call the MPAI-AIF APIs.
2	An Implementation of an MPAI Use Case	Implementations of the MPAI Use Case
3	An Implementation of an MPAI Use Case certified by a Performance Assessor	Implementations of the MPAI Use Case certified by Performance Assessors

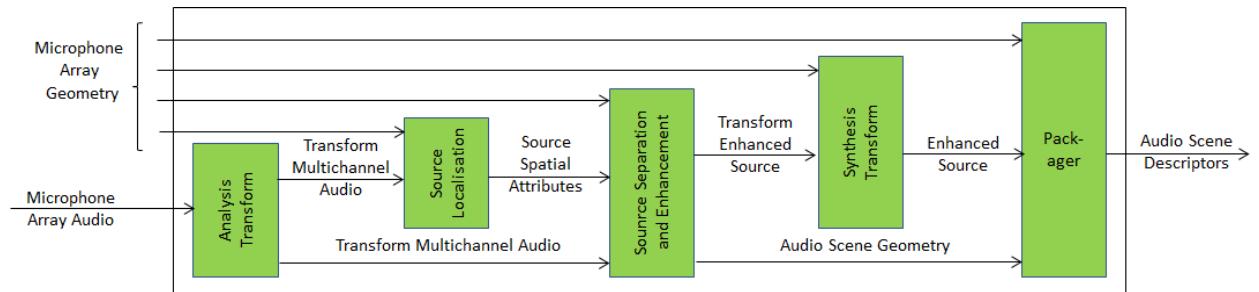
## 4 Audio-Visual Scene Description

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) [3] that includes Audio Scene Descriptors and uses a subset of Graphics Language Transmission Format (glTF) [6] to describe a visual scene.

### Audio Scene Descriptors

Audio Scene Description is a Composite AI Module (AIM) specified by Technical Specification: Context-based Audio Enhancement (MPAI-CAE) [3]. The position of an Audio Object is defined by Azimuth, Elevation, Distance.

The Composite AIM and its composing AIMs are depicted in [3].



*Figure 13 - The Audio Scene Description Composite AIM*

### Visual Scene Descriptors

MPAI uses a subset of Graphics Language Transmission Format (glTF) [6] 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 19*.

*Table 19 – MPAI-wide Terms*

<b>Term</b>	<b>Definition</b>
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 Module (AIM)	A processing element receiving AIM-specific Inputs and producing AIM-specific Outputs according to its Function.
- Composite AIM	An AIM aggregating more than one AIM.
AI Workflow (AIW)	A structured aggregation of AIMs implementing a Use Case receiving AIM-specific inputs and producing AIM-specific inputs according to its Function.
AIF Metadata	The data set describing the capabilities of an AIF set by the AIF Implementer.
AIM Metadata	The data set describing the capabilities of an AIM set by the AIM Implementer.
Application Programming Interface (API)	A software interface that allows two applications to talk to each other
Application Standard	An MPAI Standard specifying AIWs, AIMs, Topologies and Formats suitable for a particular application domain.
Channel	A physical or logical connection between an output Port of an AIM and an input Port of an AIM. The term “connection” is also used as a synonym.
Communication	The infrastructure that implements message passing between AIMs.
Component	One of the 9 AIF elements: Access, AI Module, AI Workflow, Communication, Controller, Internal Storage, Global Storage, MPAI Store, and User Agent.
Conformance	The attribute of an Implementation of being a correct technical Implementation of a Technical Specification.
- Tester	An entity authorised by MPAI 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.
- Semantics	The meaning of Data.

Device	A hardware and/or software entity running at least one instance of an AIF.
Ecosystem	The ensemble of the following actors: MPAI, MPAI Store, Implementers, Conformance Testers, Performance Testers and Users of MPAI-AIF Implementations as needed to enable an Interoperability Level.
Event	An occurrence acted on by an Implementation.
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.
Identifier	A name that uniquely identifies an Implementation.
Implementation	<ul style="list-style-type: none"> <li>1. An embodiment of the MPAI-AIF Technical Specification, or</li> <li>2. An AIW or AIM of a particular Level (1-2-3).</li> </ul>
Interoperability	The ability to functionally replace an AIM/AIW with another AIM/AIW having the same Interoperability Level
Interoperability Level	<p>The attribute of an AIW and its AIMs to be executable in an AIF Implementation and to be:</p> <ol style="list-style-type: none"> <li>1. Implementer-specific and satisfying the MPAI-AIF Standard (<i>Level 1</i>).</li> <li>2. Specified by an MPAI Application Standard (<i>Level 2</i>).</li> <li>3. Specified by an MPAI Application Standard and certified by a Performance Assessor (<i>Level 3</i>).</li> </ol>
Knowledge Base	Structured and/or unstructured information made accessible to AIMs via MPAI-specified interfaces
Message	A sequence of Records.
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.
Performance Assessment Means	Procedures, tools, data sets and/or data set characteristics to Assess the Performance of an Implementation.
Performance Assessor	An entity authorised by MPAI to Assess the Performance of an Implementation in a given Application domain
Port	A physical or logical communication interface of an AIM.
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	Data with a specified structure.
Reference Model	The AIMs and theirs Connections in an AIW.
Reference Software Implementation	The technically correct software implementation of a Technical Specification attached to a Reference Software Specification.
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.
Specification	A collection of normative clauses.
- <i>Technical</i>	(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.
- <i>Reference Software</i>	The normative document specifying the use of the Reference Software Implementation.
- <i>Conformance Testing</i>	The normative document specifying the Means to Test the Conformance of an Implementation.
- <i>Performance Assessment</i>	The normative document specifying the procedures, the tools, the data sets and/or the data set characteristics to Assess the Grade of Performance of an Implementation.
Standard	The ensemble of Technical Specification, Reference Software, Conformance Testing and Performance Assessment of an MPAI application Standard.
Storage	
- Storage	A Component to store data shared by the AIMs.
- Storage	A Component to store data of the individual AIMs.
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.
- 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 - AIW and AIM Metadata of ABV-CTX

### 1 Metadata for ABV-CTX AIW

```
{  
    "$schema": "https://json-schema.org/draft/2020-12/schema",  
    "$id": "https://mpai.community/standards/resources/MPAI-AIF/V2/AIW-AIM-metadata.schema.json",  
    "title": "HCI AIF V2 AIW/AIM metadata",  
    "Identifier": {  
        "ImplementerID": /* String assigned by IIDRA */,  
        "Specification": {  
            "Standard": "MPAI-ARA",  
            "AIW": "ABV-CTX",  
            "AIM": "ABV-CTX",  
            "Version": "1"  
        }  
    },  
    "APIProfile": "Secure",  
    "Description": " This AIW is used to send participant information to the ABV Server",  
    "Types": [  
        {  
            "Name": "LanguageID_t",  
            "Type": "uint16[]"  
        },  
        {  
            "Name": "AvatarModel_t",  
            "Type": "uint8[]"  
        },  
        {  
            "Name": "Text_t",  
            "Type": "{uint8[] | uint16[]}"  
        },  
        {  
            "Name": "Audio_t",  
            "Type": "uint16[]"  
        },  
        {  
            "Name": "ArrayAudio_t",  
            "Type": "Audio_t[]"  
        },  
        {  
            "Name": "Video_t",  
            "Type": "{uint32[] | uint40[]}"  
        },  
        {  
            "Name": "Speech_t",  
            "Type": "{uint8[] | uint18[]}"  
        },  
        {  
            "Name": "AvatarDescriptors_t",  
            "Type": "{uint8[]}"  
        },  
        {  
            "Name": "FaceObject_t",  
            "Type": "{uint32[]}"  
        }  
    ],  
    "Ports": [  
        {  
            "Name": "InputAudio",  
            "Direction": "InputOutput",  
            "RecordType": "ArrayAudio_t",  
            "Technology": "Software",  
            "Protocol": "",  
            "IsRemote": false  
        },  
        {  
            "Name": "AvatarModel",  
            "Direction": "InputOutput",  
            "RecordType": "AvatarModel_t",  
            "Technology": "Software",  
            "Protocol": "",  
            "IsRemote": false  
        }  
    ]  
}
```

```

        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"InputText",
        "Direction":"InputOutput",
        "RecordType":"Text_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"InputAudio",
        "Direction":"InputOutput",
        "RecordType":"AudioArray_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"InputVideo",
        "Direction":"InputOutput",
        "RecordType":"Video_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"SpeechObject",
        "Direction":"InputOutput",
        "RecordType":"Speech_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"InputSpeech",
        "Direction":"InputOutput",
        "RecordType":"Speech_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"AvatarDescriptors",
        "Direction":"OutputInput",
        "RecordType":"AvatarDescriptors_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"FaceObject",
        "Direction":"OutputInput",
        "RecordType":"FaceObject_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    }
],
"SubAIMs": [
    {
        "Name":"VisualSceneDescription",
        "Identifier": {
            "ImplementerID": /* String assigned by IIDRA */,
            "Specification": {
                "Standard": "MPAI-ARA",
                "AIW": "ABV-CTX",
                "AIM": "VisualSceneDescription",
                "Version": "1"
            }
        }
    }
]
}

```

```

    {
      "Name": "AudioSceneDescription",
      "Identifier": {
        "ImplementerID": "/* String assigned by IIDRA */",
        "Specification": {
          "Standard": "MPAI-ARA",
          "AIW": "ABV-CTX",
          "AIM": "Audio SceneDescription",
          "Version": "2"
        }
      }
    },
    {
      "Name": "SpeechRecogniton",
      "Identifier": {
        "ImplementerID": "/* String assigned by IIDRA */",
        "Specification": {
          "Standard": "MPAI-ARA",
          "AIW": "ABV-CTX",
          "AIM": "SpeechRecogniton",
          "Version": "1"
        }
      }
    },
    {
      "Name": " LanguageUnderstanding",
      "Identifier": {
        "ImplementerID": "/* String assigned by IIDRA */",
        "Specification": {
          "Standard": "MPAI-ARA",
          "AIW": "ABV-CTX",
          "AIM": "LanguageUnderstanding",
          "Version": "1"
        }
      }
    },
    {
      "Name": "PersonalStatusExtraction",
      "Identifier": {
        "ImplementerID": "/* String assigned by IIDRA */",
        "Specification": {
          "Standard": "MPAI-ARA",
          "AIW": "ABV-CTX",
          "AIM": "PersonalStatusExtraction",
          "Version": "2"
        }
      }
    },
    {
      "Name": "AvatarDescription",
      "Identifier": {
        "ImplementerID": "/* String assigned by IIDRA */",
        "Specification": {
          "Standard": "MPAI-ARA",
          "AIW": "ABV-CTX",
          "AIM": " AvatarDescription ",
          "Version": "2"
        }
      }
    }
  ],
  "Topology": [
    {
      "Output": {
        "AIMName": "",
        "PortName": "LanguagePreference"
      },
      "Input": {
        "AIMName": "",
        "PortName": "LanguagePreference"
      }
    },
    {
      "Output": {

```

```

        "AIMName":"",
        "PortName":AvatarModel"""
    },
    "Input":{

        "AIMName":"",
        "PortName":AvatarModel"
    }
},
{
    "Output":{

        "AIMName":"",
        "PortName":InputText"
    },
    "Input":{

        "AIMName":"",
        "PortName":InputText"
    }
},
{
    "Output":{

        "AIMName":"",
        "PortName":InputAudio"
    },
    "Input":{

        "AIMName":AudioSceneDescription",
        "PortName":InputAudio"
    }
},
{
    "Output":{

        "AIMName":"",
        "PortName":InputVideo"
    },
    "Input":{

        "AIMName":VisualSceneDescription",
        "PortName":InputVideo"
    }
},
{
    "Output":{

        "AIMName": AudioSceneDescription",
        "PortName":InputSpeech2"
    },
    "Input":{

        "AIMName":PersonalStatusExtraction",
        "PortName":InputSpeech2"
    }
},
{
    "Output":{

        "AIMName": AudioSceneDescription ",
        "PortName":InputSpeech3"
    },
    "Input":{

        "AIMName":SpeechRecognition",
        "PortName":InputSpeech3"
    }
},
{
    "Output":{

        "AIMName": SpeechRecognition",
        "PortName":RecognisedText"
    },
    "Input":{

        "AIMName":SpeechRecognition",
        "PortName":RecognisedText"
    }
},
{
    "Output":{

        "AIMName":AudioSceneDescription",
        "PortName":InputSpeech1"
    },
    "Input":{


```

```

        "AIMName":"PersonalStatusExtraction",
        "PortName":"InputSpeech1"
    }
},
{
    "Output":{
        "AIMName":"LanguageUnderstanding",
        "PortName":"Meaning"
    },
    "Input":{
        "AIMName":"PersonalStatusExtraction",
        "PortName":"Meaning"
    }
},
{
    "Output":{
        "AIMName":"VisualSceneDescription",
        "PortName":"FaceDescriptors1"
    },
    "Input":{
        "AIMName":"PersonalStatusExtraction",
        "PortName": "FaceDescriptors1"
    }
},
{
    "Output":{
        "AIMName":"VisualSceneDescription",
        "PortName": "BodyDescriptors1"
    },
    "Input":{
        "AIMName":"PersonalStatusExtraction",
        "PortName": "BodyDescriptors1"
    }
},
{
    "Output":{
        "AIMName":"PersonalStatusExtraction",
        "PortName": "PersonalStatus"
    },
    "Input":{
        "AIMName": "AvatarDescription",
        "PortName": "PersonalStatus"
    }
},
{
    "Output":{
        "AIMName": "VisualSceneDescription",
        "PortName": "FaceDescriptors2"
    },
    "Input":{
        "AIMName": "AvatarDescription",
        "PortName": "FaceDescriptors2"
    }
},
{
    "Output":{
        "AIMName": "VisualSceneDescription",
        "PortName": "BodyDescriptors2"
    },
    "Input":{
        "AIMName": "AvatarDescription",
        "PortName": "BodyDescriptors2"
    }
},
{
    "Output": {
        "AIMName": "",
        "PortName": "LanguagePreference"
    },
    "Input": {
        "AIMName": "",
        "PortName": "LanguagePreference"
    }
},
{
    "Output": {

```

```

        "AIMName":"",
        "PortName":"AvatarModel"
    },
    "Input":{

        "AIMName":"",
        "PortName":"AvatarModel"
    }
},
{
    "Output":{

        "AIMName":"",
        "PortName":"InputText"
    },
    "Input":{

        "AIMName":"",
        "PortName":"InputText"
    }
},
{
    "Output":{

        "AIMName":"AudioSceneDescription",
        "PortName":"SpeechObject"
    },
    "Input":{

        "AIMName":"",
        "PortName":"SpeechObject"
    }
},
{
    "Output":{

        "AIMName":"AudioSceneDescription",
        "PortName":"InputSpeech1"
    },
    "Input":{

        "AIMName":"",
        "PortName":"InputSpeech1"
    }
},
{
    "Output":{

        "AIMName":"AvatarDescription",
        "PortName":"AvatarDescriptors"
    },
    "Input":{

        "AIMName":"",
        "PortName":"AvatarDescriptors"
    },
{
    "Output":{

        "AIMName":"VisualSceneDescription",
        "PortName":"FaceObject"
    },
    "Input":{

        "AIMName":"",
        "PortName":"FaceObject"
    }
}
],
"Implementations": [
{
    "BinaryName":"ctx.exe",
    "Architecture":"x64",
    "OperatingSystem":"Windows",
    "Version":"v0.1",
    "Source":"MPAISource",
    "Destination":""
}
],
"ResourcePolicies": [
{
    "Name":"Memory",
    "Minimum":"50000",
    "Maximum":"100000",
    "Request":"75000"
}
]
```

```

},
{
  "Name": "CPUNumber",
  "Minimum": "1",
  "Maximum": "2",
  "Request": "1"
},
{
  "Name": "CPU:Class",
  "Minimum": "Low",
  "Maximum": "High",
  "Request": "Medium"
},
{
  "Name": "GPU:CUDA:FrameBuffer",
  "Minimum": "11GB_GDDR5X",
  "Maximum": "8GB_GDDR6X",
  "Request": "11GB_GDDR6"
},
{
  "Name": "GPU:CUDA:MemorySpeed",
  "Minimum": "1.60GHz",
  "Maximum": "1.77GHz",
  "Request": "1.71GHz"
},
{
  "Name": "GPU:CUDA:Class",
  "Minimum": "SM61",
  "Maximum": "SM86",
  "Request": "SM75"
},
{
  "Name": "GPU:Number",
  "Minimum": "1",
  "Maximum": "1",
  "Request": "1"
}
],
"Documentation": [
  {
    "Type": "tutorial",
    "URI": "https://mpai.community/standards/mpai-ara/"
  }
]
}

```

## 2 Metadata for ARA-CTX AIMs

### Audio Scene Description

```

{
  "Identifier": {
    "ImplementerID": "/* String assigned by IIDRA */",
    "Specification": {
      "Name": "MPAI-ARA",
      "AIM": "ARA-CTX",
      "AIM": "AudioSceneDescription",
      "Version": "1"
    },
    "Description": "This AIM implements the audio scene description function for VSV-CTX.",
    "Types": [
      {
        "Name": "Audio_t",
        "Type": "uint16[]"
      },
      {
        "Name": "ArrayAudio_t",
        "Type": "Audio_t[]"
      },
      {
        "Name": "Speech_t",
        "Type": "{uint8[] | uint16[]}"
      }
    ],
  }
}
```

```

"Ports": [
    {
        "Name": "ArrayAudio",
        "Direction": "InputOutput",
        "RecordType": "ArrayAudio_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "SpeechObject",
        "Direction": "OutputInput",
        "RecordType": "Speech_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "InputSpeech1",
        "Direction": "OutputInput",
        "RecordType": "Speech_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "InputSpeech2",
        "Direction": "OutputInput",
        "RecordType": "Speech_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    }
],
"SubAIMs": [
],
"Topology": [
],
"Implementations": [
],
"Documentation": [
    {
        "Type": "Tutorial",
        "URI": "https://mpai.community/standards/mpai-ara/"
    }
]
}
}

```

## Visual Scene Description

```

{
    "Identifier": {
        "ImplementerID": /* String assigned by IIDRA */,
        "Specification": {
            "Name": "MPAI-ARA",
            "AIW": "ABVCTX",
            "AIM": "VisualSceneDescription",
            "Version": "1"
        },
        "Description": "This AIM implements the visual scene description function for ABV-CTX.",
        "Types": [
            {
                "Name": "Video_t",
                "Type": "uint32[]"
            },
            {
                "Name": "FaceDescriptors_t",
                "Type": "uint8[]"
            },
            {
                "Name": "Image_t",
                "Type": "uint8[]"
            }
        ]
    }
}

```

```

{
  "Name":"BodyDescriptors_t",
  "Type":"{uint8[]}"
},
{
  "Name":"FaceObject_t",
  "Type":"{uint32[]}"
},
],
"Ports":[
{
  "Name":"InputVideo",
  "Direction":"InputOutput",
  "RecordType":"Video_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"FaceDescriptors1",
  "Direction":"OutputInput",
  "RecordType":"FaceDescriptors_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"BodyDescriptors1",
  "Direction":"OutputInput",
  "RecordType":"BodyDescriptors_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"FaceDescriptors2",
  "Direction":"OutputInput",
  "RecordType":"FaceDescriptors_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"BodyDescriptors2",
  "Direction":"OutputInput",
  "RecordType":"BodyDescriptors_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"FaceObject",
  "Direction":"OutputInput",
  "RecordType":"FaceObject_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
}
],
"SubAIMs":[
],
"Topology":[
],
"Implementations":[
],
"Documentation": [
  {
    "Type":"Tutorial",
    "URI":"https://mpai.community/standards/mpai-ara/"
  }
]
]
```

```
    }
}
```

## SpeechRecognition

```
{
  "Identifier": {
    "ImplementerID": /* String assigned by IIDRA */,
    "Specification": {
      "Name": "MPAI-ARA",
      "AIW": "ABV-CTX",
      "AIM": "SpeechRecognition",
      "Version": "1"
    },
    "Description": "This AIM implements the speech recognition function for ARA-VSV: it converts the user's speech to text.",
    "Types": [
      {
        "Name": "Speech_t",
        "Type": "{uint8[] | uint16[]}"
      },
      {
        "Name": "Text_t",
        "Type": "{uint8[] | uint16[]}"
      }
    ],
    "Ports": [
      {
        "Name": "InputSpeechs",
        "Direction": "InputOutput",
        "RecordType": "Speech_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
      },
      {
        "Name": "RecognisedText",
        "Direction": "OutputInput",
        "RecordType": "Text_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
      }
    ],
    "SubAIMs": [
    ],
    "Topology": [
    ],
    "Implementations": [
    ],
    "Documentation": [
      {
        "Type": "Tutorial",
        "URI": "https://mpai.community/standards/mpai-ara/"
      }
    ]
  }
}
```

## LanguageUnderstanding

```
{
  "Identifier": {
    "ImplementerID": /* String assigned by IIDRA */,
    "Specification": {
      "Name": "MPAI-ARA",
      "AIW": "ABV-CTX",
      "AIM": "LanguageUnderstanding",
      "Version": "1"
    },

```

```

    "Description":"This AIM extracts Meaning from Recognised Text.",
    "Types": [
        {
            "Name":"Text_t",
            "Type":"{uint8[] | uint16[]}"
        },
        {
            "Name":"Tagging_t",
            "Type":"{string<256 set; string<256 result}"
        },
        {
            "Name":"Meaning_t",
            "Type":"{Tagging_t POS_tagging; Tagging_t NE_tagging; Tagging_t dependency_tagging; Tagging_t SRL_tagging}"
        }
    ],
    "Ports": [
        {
            "Name":"RecognisedText",
            "Direction":"InputOutput",
            "RecordType":"Text_t",
            "Technology":"Software",
            "Protocol":"",
            "IsRemote":false
        },
        {
            "Name":"Meaning",
            "Direction":"OutputInput",
            "RecordType":"Meaning_t",
            "Technology":"Software",
            "Protocol":"",
            "IsRemote":false
        }
    ],
    "SubAIMs": [
    ],
    "Topology": [
    ],
    "Implementations": [
    ],
    "Documentation": [
        {
            "Type":"Tutorial",
            "URI":"https://mpai.community/standards/mpai-ara/"
        }
    ]
}

```

## PersonalStatusExtraction

```

{
    "Identifier": {
        "ImplementerID": "/* String assigned by IIDRA */",
        "Specification": {
            "Name": "MPAI-ARA",
            "AIW": "ABV-CTX",
            "AIM": "PersonalStatusExtraction",
            "Version": "1"
        },
        "Description": "This AIM extracts the combined Personal Status from Meaning, Speech, Face, and Gesture."
    },
    "Types": [
        {
            "Name": "Speech_t",
            "Type": "{uint16[]}"
        },
        {
            "Name": "FaceDescriptors_t",
            "Type": "uint8[]"
        }
    ]
}

```

```

{
    "Name": "BodyDescriptors_t",
    "Type": "uint8[]"
},
{
    "Name": "Tagging_t",
    "Type": "{string<256 set; string<256 result}"
},
{
    "Name": "Meaning_t",
    "Type": "{Tagging_t POS_tagging; Tagging_t NE_tagging; Tagging_t dependency_tagging; Tagging_t SRL_tagging}"
},
{
    "Name": "PersonalStatus_t",
    "Type": "uint8[]"
}
],
"Ports": [
{
    "Name": "InputSpeech2",
    "Direction": "InputOutput",
    "RecordType": "Speech_t",
    "Technology": "Software",
    "Protocol": "",
    "IsRemote": false
},
{
    "Name": "Meaning",
    "Direction": "InputOutput",
    "RecordType": "Meaning_t",
    "Technology": "Software",
    "Protocol": "",
    "IsRemote": false
},
{
    "Name": "FaceDescriptors",
    "Direction": "InputOutput",
    "RecordType": "FaceDescriptors_t",
    "Technology": "Software",
    "Protocol": "",
    "IsRemote": false
},
{
    "Name": "BodyDescriptors",
    "Direction": "InputOutput",
    "RecordType": "BodyDescriptors_t",
    "Technology": "Software",
    "Protocol": "",
    "IsRemote": false
},
{
    "Name": "PersonalStatus",
    "Direction": "OutputInput",
    "RecordType": "PersonalStatus_t",
    "Technology": "Software",
    "Protocol": "",
    "IsRemote": false
}
],
"SubAIMs": [
],
"Topology": [
],
"Implementations": [
],
"Documentation": [
{
    "Type": "Tutorial",
    "URI": "https://mpai.community/standards/mpai-ara/"
}
]
```

```
        ]
    }
}
```

## AvatarDescription

```
{
  "Identifier": {
    "ImplementerID": "/* String assigned by IIDRA */",
    "Specification": {
      "Name": "MPAI-ARA",
      "AIW": "ABV-CTX",
      "AIM": "PersonalStatusDisplay",
      "Version": "1"
    },
    "Description": "This AIM outputs the Avatar Descriptors.",
    "Types": [
      {
        "Name": "PersonalStatus_t",
        "Type": "{uint8[]}"
      },
      {
        "Name": "FaceDescriptors_t",
        "Type": "{uint8[]}"
      },
      {
        "Name": "BodyDescriptors_t",
        "Type": "{uint8[]}"
      },
      {
        "Name": "AvatarDescriptors_t",
        "Type": "uint8[]"
      }
    ],
    "Ports": [
      {
        "Name": "PersonalStatus",
        "Direction": "InputOutput",
        "RecordType": "PersonalStatus_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
      },
      {
        "Name": "FaceDescriptors",
        "Direction": "InputOutput",
        "RecordType": "FaceDescriptors_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
      },
      {
        "Name": "BodyDescriptors",
        "Direction": "InputOutput",
        "RecordType": "BodyDescriptors_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
      },
      {
        "Name": "AvatarDescriptors",
        "Direction": "OutputInput",
        "RecordType": "AvatarDescriptors_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
      }
    ],
    "SubAIMS": [
    ],
    "Topology": [
    ],
  ]
```

```
"Implementations": [  
    ],  
    "Documentation": [  
        {  
            "Type": "Tutorial",  
            "URI": "https://mpai.community/standards/mpai-mmC/"  
        }  
    ]  
}
```

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

### 3 AIW metadata for ABV-SRV

```
{  
    "$schema": "https://json-schema.org/draft/2020-12/schema",  
    "$id": "https://mpai.community/standards/resources/MPAI-AIF/V2/AIW-AIM-metadata.schema.json",  
    "title": "HCI AIF V2 AIW/AIM metadata",  
    "Identifier": {  
        "ImplementerID": /* String assigned by IIDRA */,  
        "Specification": {  
            "Standard": "MPAI-ARA",  
            "AIW": "ABV-SRV",  
            "AIM": "ABV-SRV",  
            "Version": "1"  
        },  
        "APIProfile": "Secure",  
        "Description": "At the start, this AIF selects and distributes the Environment, receives, places and distributes Avatar Models, and continuously receives and distributes Speech and Avatar Descriptors.,  
        "Types": [  
            {  
                "Name": "EnvironmentModel_t",  
                "Type": "uint8[]"  
            },  
            {  
                "Name": "SpatialAttitude_t",  
                "Type": "float32[18]"  
            },  
            {  
                "Name": "AvatarModel_t",  
                "Type": "uint8[]"  
            },  
            {  
                "Name": "Summary_t",  
                "Type": "uint8[]"  
            },  
            {  
                "Name": "AvatarDescriptor_t",  
                "Type": "uint8[]"  
            },  
            {  
                "Name": "ParticipantID_t",  
                "Type": "uint8[]"  
            },  
            {  
                "Name": "Speech_t",  
                "Type": "uint16[]"  
            },  
            {  
                "Name": "FaceObject_t",  
                "Type": "uint32[]"  
            },  
            {  
                "Name": "LanguagePreference_t",  
                "Type": "uint16[]"  
            },  
            {  
                "Name": "Text_t",  
                "Type": "{uint8[] | uint16[]}"  
            }]  
        "Ports": [  
            {  
                "Name": "EnvironmentModel",  
                "Direction": "InputOutput",  
                "RecordType": "EnvironmentModel_t",  
                "Technology": "Software",  
                "Protocol": "",  
                "IsRemote": false  
            }]}  
}
```

```
},
{
  "Name":"SpatialAttitude",
  "Direction":"InputOutput",
  "RecordType":"SpatialAttitude_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"AvatarModel",
  "Direction":"OutputInput",
  "RecordType":"AvatarModel_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"Summary",
  "Direction":"OutputInput",
  "RecordType":"Summary_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"AvatarDescriptor",
  "Direction":"OutputInput",
  "RecordType":"AvatarDescriptor_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"ParticipantID",
  "Direction":"OutputInput",
  "RecordType":"ParticipantID_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"SpeechObject",
  "Direction":"OutputInput",
  "RecordType":"Speech_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"FaceObject",
  "Direction":"OutputInput",
  "RecordType":"FaceObject_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"LanguagePreference",
  "Direction":"OutputInput",
  "RecordType":"LanguagePreference_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"InputSpeech",
  "Direction":"OutputInput",
  "RecordType":"Speech_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"OutputSpeech",
  "Direction":"OutputInput",
  "RecordType":"Speech_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
}
```

```

        "Name":"InputText",
        "Direction":"OutputInput",
        "RecordType":"Text_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"EnvironmentModel",
        "Direction":"OutputInput",
        "RecordType":"EnvironmentModel_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"SpatialAttitude",
        "Direction":"OutputInput",
        "RecordType":"SpatialAttitude_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"AvatarModel",
        "Direction":"OutputInput",
        "RecordType":"AvatarModel_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"Summary",
        "Direction":"OutputInput",
        "RecordType":"Summary_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"AvatarDescriptor",
        "Direction":"OutputInput",
        "RecordType":"_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"ParticipantID",
        "Direction":"OutputInput",
        "RecordType":"_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"TranslatedSpeech",
        "Direction":"OutputInput",
        "RecordType":"Speech_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"TranslatedText",
        "Direction":"OutputInput",
        "RecordType":"Text_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    }
],
"SubAIMs": [
    {

```

```

        "Name":"ParticipantAuthentication",
        "Identifier":{
            "ImplementerID":/* String assigned by IIDRA */,
            "Specification":{
                "Standard":"MPAI-ARA",
                "AIW":"ABV-SRV",
                "AIM":"ParticipantAuthentication",
                "Version":"1"
            }
        }
    },
    {
        "Name":"Translation",
        "Identifier":{
            "ImplementerID":/* String assigned by IIDRA */,
            "Specification":{
                "Standard":"MPAI-ARA",
                "AIW":"ABV-SRV",
                "AIM":"Translation",
                "Version":"1"
            }
        }
    }
],
"Topology": [
    {
        "Output": {
            "AIMName":"",
            "PortName":"EnvironmentModel"
        },
        "Input": {
            "AIMName":"",
            "PortName":"EnvironmentModel"
        }
    },
    {
        "Output": {
            "AIMName":"",
            "PortName":"SpatialAttitude"
        },
        "Input": {
            "AIMName":"",
            "PortName":"SpatialAttitude"
        }
    },
    {
        "Output": {
            "AIMName":"",
            "PortName":"AvatarModel"
        },
        "Input": {
            "AIMName":"",
            "PortName":"AvatarModel"
        }
    },
    {
        "Output": {
            "AIMName":"",
            "PortName":"Summary"
        },
        "Input": {
            "AIMName":"",
            "PortName":"Summary"
        }
    },
    {
        "Output": {
            "AIMName":"",
            "PortName":"AvatarDescriptors"
        },
        "Input": {
            "AIMName":"",
            "PortName":"AvatarDescriptor"
        }
    }
]

```

```
},
{
  "Output": {
    "AIMName": "",
    "PortName": "ParticipantID1"
  },
  "Input": {
    "AIMName": "ParticipantAuthentication",
    "PortName": "ParticipantID1"
  }
},
{
  "Output": {
    "AIMName": "",
    "PortName": "SpeechObject"
  },
  "Input": {
    "AIMName": "ParticipantAutentication",
    "PortName": "SpeechObject"
  }
},
{
  "Output": {
    "AIMName": "",
    "PortName": "FaceObject"
  },
  "Input": {
    "AIMName": "ParticipantAutentication",
    "PortName": "FaceObject"
  }
},
{
  "Output": {
    "AIMName": "",
    "PortName": "LanguagePreference"
  },
  "Input": {
    "AIMName": "Translation",
    "PortName": "LanguagePreference"
  }
},
{
  "Output": {
    "AIMName": "",
    "PortName": "InputSpeech"
  },
  "Input": {
    "AIMName": "Translation",
    "PortName": "InputSpeech"
  }
},
{
  "Output": {
    "AIMName": "",
    "PortName": "InputText"
  },
  "Input": {
    "AIMName": "Translation",
    "PortName": "InputText"
  }
},
{
  "Output": {
    "AIMName": "ParticipantAuthentication",
    "PortName": "ParticipantID2"
  },
  "Input": {
    "AIMName": "",
    "PortName": "ParticipantID2"
  }
},
{
  "Output": {
    "AIMName": "Translation",
```

```

        "PortName":"Speech"
    },
    "Input":{

        "AIMName":"",
        "PortName":"Speech"
    }
},
{
    "Output":{

        "AIMName":"Translation",
        "PortName":"Text"
    },
    "Input":{

        "AIMName":"",
        "PortName":"Text"
    }
}
],
"Implementations":[
{
    "BinaryName":"arasrv.exe",
    "Architecture":"x64",
    "OperatingSystem":"Windows",
    "Version":"v0.1",
    "Source":"MPAIStrore",
    "Destination":""
}
],
"ResourcePolicies":[
{
    "Name":"Memory",
    "Minimum":"50000",
    "Maximum":"100000",
    "Request":"75000"
},
{
    "Name":"CPUNumber",
    "Minimum":"1",
    "Maximum":"2",
    "Request":"1"
},
{
    "Name":"CPU:Class",
    "Minimum":"Low",
    "Maximum":"High",
    "Request":"Medium"
},
{
    "Name":"GPU:CUDA:FrameBuffer",
    "Minimum":"11GB_GDDR5X",
    "Maximum":"8GB_GDDR6X",
    "Request":"11GB_GDDR6"
},
{
    "Name":"GPU:CUDA:MemorySpeed",
    "Minimum":"1.60GHz",
    "Maximum":"1.77GHz",
    "Request":"1.71GHz"
},
{
    "Name":"GPU:CUDA:Class",
    "Minimum":"SM61",
    "Maximum":"SM86",
    "Request":"SM75"
},
{
    "Name":"GPU:Number",
    "Minimum":"1",
    "Maximum":"1",
    "Request":"1"
}
],
"Documentation":[
{

```

```

        "Type":"tutorial",
        "URI":"https://mpai.community/standards/mpai-ara/"
    }
]
}

{
  "Identifier":{

    "ImplementerID":/* String assigned by IIDRA */,
    "Specification":{

      "Name":"MPAI-ARA",
      "AIW":"ABV-SRV",
      "AIM":"ParticipantAuthentication",
      "Version":"1"
    },
    "Description":"This AIM identifies participants via speech and face.",
    "Types":[
      {
        "Name":"ParticipantID_t",
        "Type":"uint8[]"
      },
      {
        "Name":"Speech_t",
        "Type":"uint16[]"
      },
      {
        "Name":"FaceObject_t",
        "Type":"uint32[]"
      }
    ],
    "Ports":[
      {
        "Name":"ParticipantID1",
        "Direction":"InputOutput",
        "RecordType":"ParticipantID_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
      },
      {
        "Name":"InputSpeech",
        "Direction":"InputOutput",
        "RecordType":"Speech_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
      },
      {
        "Name":"FaceObject",
        "Direction":"OutputInput",
        "RecordType":"FaceObject_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
      },
      {
        "Name":"ParticipantID2",
        "Direction":"OutputInput",
        "RecordType":"ParticipantID_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
      }
    ],
    "SubAIMs":[
    ],
    "Topology":[
    ]
  }
}

```

## 4 Metadata for ABV-SRV AIMs

### 2.1 ParticipantAutentication

```

{
  "Identifier":{

    "ImplementerID":/* String assigned by IIDRA */,
    "Specification":{

      "Name":"MPAI-ARA",
      "AIW":"ABV-SRV",
      "AIM":"ParticipantAuthentication",
      "Version":"1"
    },
    "Description":"This AIM identifies participants via speech and face.",
    "Types":[
      {
        "Name":"ParticipantID_t",
        "Type":"uint8[]"
      },
      {
        "Name":"Speech_t",
        "Type":"uint16[]"
      },
      {
        "Name":"FaceObject_t",
        "Type":"uint32[]"
      }
    ],
    "Ports":[
      {
        "Name":"ParticipantID1",
        "Direction":"InputOutput",
        "RecordType":"ParticipantID_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
      },
      {
        "Name":"InputSpeech",
        "Direction":"InputOutput",
        "RecordType":"Speech_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
      },
      {
        "Name":"FaceObject",
        "Direction":"OutputInput",
        "RecordType":"FaceObject_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
      },
      {
        "Name":"ParticipantID2",
        "Direction":"OutputInput",
        "RecordType":"ParticipantID_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
      }
    ],
    "SubAIMs":[
    ],
    "Topology":[
    ]
  }
}

```

```

        ],
        "Implementations": [
            ],
            "Documentation": [
                {
                    "Type": "Tutorial",
                    "URI": "https://mpai.community/standards/mpai-ara/"
                }
            ]
        }
    }
}

```

## 2.2 Translation

```

{
    "Identifier": {
        "ImplementerID": /* String assigned by IIDRA */,
        "Specification": {
            "Name": "MPAI-ARA",
            "AIW": "SRV",
            "AIM": "Translation",
            "Version": "1"
        },
        "Description": "This AIM translates an input speech or text in a language into speech or text in another language.",
        "Types": [
            {
                "Name": "LanguagePreference_t",
                "Type": "uint8[]"
            },
            {
                "Name": "Speech_t",
                "Type": "uint16[]"
            },
            {
                "Name": "Text_t",
                "Type": "{uint8[] | uint16[]}"
            }
        ],
        "Ports": [
            {
                "Name": "LanguagePreference",
                "Direction": "InputOutput",
                "RecordType": " LanguagePreference_t",
                "Technology": "Software",
                "Protocol": "",
                "IsRemote": false
            },
            {
                "Name": "InputSpeech",
                "Direction": "InputOutput",
                "RecordType": "Speech_t",
                "Technology": "Software",
                "Protocol": "",
                "IsRemote": false
            },
            {
                "Name": "InputText",
                "Direction": "OutputInput",
                "RecordType": "Text_t",
                "Technology": "Software",
                "Protocol": "",
                "IsRemote": false
            },
            {
                "Name": "TranslatedSpeech",
                "Direction": "OutputInput",
                "RecordType": "Speech_t",
                "Technology": "Software",
                "Protocol": "",
                "IsRemote": false
            }
        ]
    }
}

```

```
        "Name":"TranslatedText",
        "Direction":"InputOutput",
        "RecordType":"Text_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
],
"SubAIMs":[
],
"Topology":[
],
"Implementations":[
],
"Documentation":[
{
    "Type":"Tutorial",
    "URI":"https://mpai.community/standards/mpai-ara/"
}
]
}
}
```

## Annex 6 - AIW and AIM Metadata of ARA-VSV

### 1 Metadata for VSV AIW

```
{  
    "$schema": "https://json-schema.org/draft/2020-12/schema",  
    "$id": "https://mpai.community/standards/resources/MPAI-AIF/V2/AIW-AIM-  
metadata.schema.json",  
    "title": "VSV AIF V2 AIW/AIM metadata",  
    "Identifier": {  
        "ImplementerID": "/* String assigned by IIDRA */",  
        "Specification": {  
            "Standard": "MPAI-MMC",  
            "AIW": "MMC-VSV",  
            "AIM": "MMC-VSV ",  
            "Version": "2"  
        }  
    },  
    "APIProfile": "Secure",  
    "Description": " This AIF is used to produce the visual and vocal appearance of the Virtual Secretary and the Summary of the Avatar-Based Videoconference",  
    "Types": [  
        {  
            "Name": "Text_t",  
            "Type": "{uint8[] | uint16[]}"  
        },  
        {  
            "Name": "Speech_t",  
            "Type": "uint16[]"  
        },  
        {  
            "Name": "AvatarDescriptors_t",  
            "Type": "uint8[]"  
        },  
        {  
            "Name": "Summary_t",  
            "Type": "uint8[]"  
        },  
        {  
            "Name": "AvatarModel_t",  
            "Type": "uint8[]"  
        },  
        {  
            "Name": "AvatarDescriptors_t",  
            "Type": "uint8[]"  
        }  
    ],  
    "Ports": [  
        {  
            "Name": "InputText1",  
            "Direction": "InputOutput",  
            "RecordType": "Text_t",  
            "Technology": "Software",  
            "Protocol": "",  
            "IsRemote": false  
        },  
        {  
            "Name": "InputSpeech1",  
            "Direction": "InputOutput",  
            "RecordType": "Speech_t",  
            "Technology": "Software",  
            "Protocol": "",  
            "IsRemote": false  
        },  
        {  
            "Name": "InputText2",  
            "Direction": "InputOutput",  
            "RecordType": "Text_t",  
            "Technology": "Software",  
            "Protocol": ""  
        }  
    ]}
```

```

        "IsRemote":false
    },
    {
        "Name":"InputSpeech2",
        "Direction":"InputOutput",
        "RecordType":"Speech_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"AvatarDescriptors",
        "Direction":"InputOutput",
        "RecordType":"AvatarDescriptors_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"Summary",
        "Direction":"OutputInput",
        "RecordType":"Summary_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"VSAvatarModel",
        "Direction":"OutputInput",
        "RecordType":"AvatarModel_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"VSText",
        "Direction":"OutputInput",
        "RecordType":"Speech_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"VSSpeech",
        "Direction":"OutputInput",
        "RecordType":"Speech_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"VSAvatarDescriptors",
        "Direction":"OutputInput",
        "RecordType":"AvatarDescriptors_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    }
],
"SubAIMs":[
    {
        "Name":"SpeechRecognition",
        "Identifier":{
            "ImplementerID":/* String assigned by IIDRA */,
            "Specification":{
                "Standard":"MPAI-MMC",
                "AIW":"MMC-VSV",
                "AIM":"SpeechRecognition",
                "Version":"1"
            }
        }
    },
    {
        "Name":"AvatarDescriptorsParsing",
        "Identifier":{


```

```

        "ImplementerID":/* String assigned by IIDRA */,
        "Specification":{
            "Standard":"MPAI-MMC ",
            "AIW":"MMC-VSV",
            "AIM":"AvatarDescriptorsParsing",
            "Version":"2"
        }
    }
},
{
    "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": [
{
    "Output": {
        "AIMName": ""
    }
}
]

```

```

        "PortName":"InputSpeech1"
    },
    "Input":{

        "AIMName":"SpeechRecognition",
        "PortName":"InputSpeech1"
    }
},
{
    "Output":{

        "AIMName":"",
        "PortName":"InputAvatarDescriptors"
    },
    "Input":{

        "AIMName":"AvatarDescriptorsParsing",
        "PortName":"InputAvatarDescriptors"
    }
},
{
    "Output":{

        "AIMName":"SpeechRecognition",
        "PortName":"RecognisedText"
    },
    "Input":{

        "AIMName":"LanguageUnderstanding",
        "PortName":"RecognisedText"
    }
},
{
    "Output":{

        "AIMName":"LanguageUnderstanding",
        "PortName":"Meaning2"
    },
    "Input":{

        "AIMName":"PersonalStatusExtraction",
        "PortName":"Meaning2"
    }
},
{
    "Output":{

        "AIMName":"LanguageUnderstanding",
        "PortName":"Meaning2"
    },
    "Input":{

        "AIMName":"PersonalStatusExtraction",
        "PortName":"Meaning2"
    }
},
{
    "Output":{

        "AIMName":"",
        "PortName":"InputSpeech2"
    },
    "Input":{

        "AIMName":"PersonalStatusExtraction",
        "PortName":"InputSpeech2"
    }
},
{
    "Output":{

        "AIMName":"AvatarDescriptorsParsing",
        "PortName":"BodyDescriptors"
    },
    "Input":{

        "AIMName":"PersonalStatusExtraction",
        "PortName":"BodyDescriptors"
    }
},
{
    "Output":{

        "AIMName":"AvatarDescriptorParsing",
        "PortName":"FaceDescriptors"
    },
    "Input":{

        "AIMName":"PersonalStatusExtraction",
        "PortName":"PersonalStatusExtraction"
    }
}

```

```

        "PortName":"FaceDescriptors"
    }
},
{
    "Output": {
        "AIMName": "LanguageUnderstanding",
        "PortName": "Meaning1"
    },
    "Input": {
        "AIMName": "Summarisation",
        "PortName": "Meaning1"
    }
},
{
    "Output": {
        "AIMName": "LanguageUnderstanding",
        "PortName": "RefinedText2"
    },
    "Input": {
        "AIMName": "Summarisation",
        "PortName": "RefinedText2"
    }
},
{
    "Output": {
        "AIMName": "PersonalStatusExtraction",
        "PortName": "InputPersonalStatus1"
    },
    "Input": {
        "AIMName": "Summarisation",
        "PortName": "InputPersonalStatus1"
    }
},
{
    "Output": {
        "AIMName": "",
        "PortName": "InputText1"
    },
    "Input": {
        "AIMName": "DialogueProcessing",
        "PortName": "InputText1"
    }
},
{
    "Output": {
        "AIMName": "LanguageProcessing",
        "PortName": "RefinedText1"
    },
    "Input": {
        "AIMName": "DialogueProcessing",
        "PortName": "RefinedText1"
    }
},
{
    "Output": {
        "AIMName": "LanguageProcessing",
        "PortName": "Meaning1"
    },
    "Input": {
        "AIMName": "DialogueProcessing",
        "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": "DialogueProcessing",
        "PortName": "InputPersonalStatus2"
    }
},
{
    "Output": {
        "AIMName": "DialogueProcessing",
        "PortName": "Summary2"
    },
    "Input": {
        "AIMName": "",
        "PortName": "Summary2"
    }
},
{
    "Output": {
        "AIMName": "DialogueProcessing",
        "PortName": "VSPersonalStatus"
    },
    "Input": {
        "AIMName": "PersonalStatusDisplay",
        "PortName": "VSPersonalStatus"
    }
},
{
    "Output": {
        "AIMName": "DialogueProcessing",
        "PortName": "VSText"
    },
    "Input": {
        "AIMName": "PersonalStatusDisplay",
        "PortName": "VSText"
    }
},
{
    "Output": {
        "AIMName": "PersonalStatusDisplay",
        "PortName": "VSText"
    },
    "Input": {
        "AIMName": "",
        "PortName": "VSText"
    }
},
{
    "Output": {
        "AIMName": "PersonalStatusDisplay",
        "PortName": "VSSpeech"
    },
    "Input": {
        "AIMName": "",
        "PortName": "VSSpeech"
    }
},
{
    "Output": {
        "AIMName": "PersonalStatusDisplay",
        "PortName": "VSAvatarDescriptors"
    },
    "Input": {
        "AIMName": "",
        "PortName": "VSAvatarDescriptors"
    }
}
]

```

```

        }
    },
    "Implementations": [
        {
            "BinaryName": "vsv.exe",
            "Architecture": "x64",
            "OperatingSystem": "Windows",
            "Version": "v0.1",
            "Source": "MPAIStore",
            "Destination": ""
        }
    ],
    "ResourcePolicies": [
        {
            "Name": "Memory",
            "Minimum": "50000",
            "Maximum": "100000",
            "Request": "75000"
        },
        {
            "Name": "CPUNumber",
            "Minimum": "1",
            "Maximum": "2",
            "Request": "1"
        },
        {
            "Name": "CPU:Class",
            "Minimum": "Low",
            "Maximum": "High",
            "Request": "Medium"
        },
        {
            "Name": "GPU:CUDA:FrameBuffer",
            "Minimum": "11GB_GDDR5X",
            "Maximum": "8GB_GDDR6X",
            "Request": "11GB_GDDR6"
        },
        {
            "Name": "GPU:CUDA:MemorySpeed",
            "Minimum": "1.60GHz",
            "Maximum": "1.77GHz",
            "Request": "1.71GHz"
        },
        {
            "Name": "GPU:CUDA:Class",
            "Minimum": "SM61",
            "Maximum": "SM86",
            "Request": "SM75"
        },
        {
            "Name": "GPU:Number",
            "Minimum": "1",
            "Maximum": "1",
            "Request": "1"
        }
    ],
    "Documentation": [
        {
            "Type": "tutorial",
            "URI": "https://mpai.community/standards/mpai-mmci/"
        }
    ]
}

```

### 3. Metadata for MMC-VSV

## 1 Metadata for MMC-VSV AIW

```
{
    "$schema": "https://json-schema.org/draft/2020-12/schema",
    "$id": "https://mpai.community/standards/resources/MPAI-AIF/V2/AIW-AIM-
metadata.schema.json",

```

```

"title":"VSV AIF V2 AIW/AIM metadata",
"Identifier": {
    "ImplementerID": /* String assigned by IIDRA */,
    "Specification": {
        "Standard": "MPAI-MMC",
        "AIW": "MMC-VSV",
        "AIM": "MMC-VSV",
        "Version": "2"
    }
},
"APIProfile": "Secure",
"Description": "This AIF is used to produce the visual and vocal appearance of the Virtual Secretary and the Summary of the Avatar-Based Videoconference",
"Types": [
    {
        "Name": "Text_t",
        "Type": "{uint8[] | uint16[]}"
    },
    {
        "Name": "Speech_t",
        "Type": "uint16[]"
    },
    {
        "Name": "AvatarDescriptors_t",
        "Type": "uint8[]"
    },
    {
        "Name": "Summary_t",
        "Type": "uint8[]"
    },
    {
        "Name": "AvatarModel_t",
        "Type": "uint8[]"
    },
    {
        "Name": "AvatarDescriptors_t",
        "Type": "uint8[]"
    }
],
"Ports": [
    {
        "Name": "InputText1",
        "Direction": "InputOutput",
        "RecordType": "Text_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "InputSpeech1",
        "Direction": "InputOutput",
        "RecordType": "Speech_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "InputText2",
        "Direction": "InputOutput",
        "RecordType": "Text_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "InputSpeech2",
        "Direction": "InputOutput",
        "RecordType": "Speech_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "AvatarDescriptors",
        "Direction": "InputOutput",
        "RecordType": "AvatarDescriptors_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    }
]

```

```

        "Direction":"InputOutput",
        "RecordType":"AvatarDescriptors_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"Summary",
        "Direction":"OutputInput",
        "RecordType":"Summary_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"VSAvatarModel",
        "Direction":"OutputInput",
        "RecordType":"AvatarModel_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"VSText",
        "Direction":"OutputInput",
        "RecordType":"Speech_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"VSSpeech",
        "Direction":"OutputInput",
        "RecordType":"Speech_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"VSAvatarDescriptors",
        "Direction":"OutputInput",
        "RecordType":"AvatarDescriptors_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    }
],
"SubAIMs":[
    {
        "Name":"SpeechRecognition",
        "Identifier":{
            "ImplementerID":/* String assigned by IIDRA */,
            "Specification":{
                "Standard":"MPAI-MMC",
                "AIW":"MMC-VSV",
                "AIM":"SpeechRecognition",
                "Version":"1"
            }
        }
    },
    {
        "Name":"AvatarDescriptorsParsing",
        "Identifier":{
            "ImplementerID":/* String assigned by IIDRA */,
            "Specification":{
                "Standard":"MPAI-MMC ",
                "AIW":"MMC-VSV",
                "AIM":"AvatarDescriptorsParsing",
                "Version":"2"
            }
        }
    },
    {
        "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":[
{
    "Output":{
        "AIMName":"",
        "PortName":"InputSpeech1"
    },
    "Input":{
        "AIMName":"SpeechRecognition",
        "PortName":"InputSpeech1"
    }
},
{
    "Output":{
        "AIMName":"",
        "PortName":"InputAvatarDescriptors"
    },

```

```

    "Input": {
        "AIMName": "AvatarDescriptorsParsing",
        "PortName": "InputAvatarDescriptors"
    },
},
{
    "Output": {
        "AIMName": "SpeechRecognition",
        "PortName": "RecognisedText"
    },
    "Input": {
        "AIMName": "LanguageUnderstanding",
        "PortName": "RecognisedText"
    },
},
{
    "Output": {
        "AIMName": "LanguageUnderstanding",
        "PortName": "Meaning2"
    },
    "Input": {
        "AIMName": "PersonalStatusExtraction",
        "PortName": "Meaning2"
    },
},
{
    "Output": {
        "AIMName": "LanguageUnderstanding",
        "PortName": "Meaning2"
    },
    "Input": {
        "AIMName": "PersonalStatusExtraction",
        "PortName": "Meaning2"
    },
},
{
    "Output": {
        "AIMName": "",
        "PortName": "InputSpeech2"
    },
    "Input": {
        "AIMName": "PersonalStatusExtraction",
        "PortName": "InputSpeech2"
    },
},
{
    "Output": {
        "AIMName": "AvatarDescriptorsParsing",
        "PortName": "BodyDescriptors"
    },
    "Input": {
        "AIMName": "PersonalStatusExtraction",
        "PortName": "BodyDescriptors"
    },
},
{
    "Output": {
        "AIMName": "AvatarDescriptorParsing",
        "PortName": "FaceDescriptors"
    },
    "Input": {
        "AIMName": "PersonalStatusExtraction",
        "PortName": "FaceDescriptors"
    },
},
{
    "Output": {
        "AIMName": "LanguageUnderstanding",
        "PortName": "Meaning1"
    },
    "Input": {
        "AIMName": "Summarisation",
        "PortName": "Meaning1"
    }
},

```

```

    {
      "Output": {
        "AIMName": "LanguageUnderstanding",
        "PortName": "RefinedText2"
      },
      "Input": {
        "AIMName": "Summarisation",
        "PortName": "RefinedText2"
      }
    },
    {
      "Output": {
        "AIMName": "PersonalStatusExtraction",
        "PortName": "InputPersonalStatus1"
      },
      "Input": {
        "AIMName": "Summarisation",
        "PortName": "InputPersonalStatus1"
      }
    },
    {
      "Output": {
        "AIMName": "",
        "PortName": "InputText1"
      },
      "Input": {
        "AIMName": "DialogueProcessing",
        "PortName": "InputText1"
      }
    },
    {
      "Output": {
        "AIMName": "LanguageProcessing",
        "PortName": "RefinedText1"
      },
      "Input": {
        "AIMName": "DialogueProcessing",
        "PortName": "RefinedText1"
      }
    },
    {
      "Output": {
        "AIMName": "LanguagePeocessing",
        "PortName": "Meaning1"
      },
      "Input": {
        "AIMName": "DialogueProcessing",
        "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":"DialogueProcessing",
        "PortName":"InputPersonalStatus2"
    },
},
{
    "Output":{

        "AIMName":"DialogueProcessing",
        "PortName":"Summary2"
    },
    "Input":{

        "AIMName":"",
        "PortName":"Summary2"
    }
},
{
    "Output":{

        "AIMName":"DialogueProcessing",
        "PortName":"VSPersonalStatus"
    },
    "Input":{

        "AIMName":"PersonalStatusDisplay",
        "PortName":"VSPersonalStatus"
    }
},
{
    "Output":{

        "AIMName":"DialogueProcessing",
        "PortName":"VSText"
    },
    "Input":{

        "AIMName":"PersonalStatusDisplay",
        "PortName":"VSText"
    }
},
{
    "Output":{

        "AIMName":"PersonalStatusDisplay",
        "PortName":"VSText"
    },
    "Input":{

        "AIMName":"",
        "PortName":"VSText"
    }
},
{
    "Output":{

        "AIMName":"PersonalStatusDisplay",
        "PortName":"VSSpeech"
    },
    "Input":{

        "AIMName":"",
        "PortName":"VSSpeech"
    }
},
{
    "Output":{

        "AIMName":"PersonalStatusDisplay",
        "PortName":"VSAvatarDescriptors"
    },
    "Input":{

        "AIMName":"",
        "PortName":"VSAvatarDescriptors"
    }
}
],
"Implementations": [
{
    "BinaryName":"vsv.exe",
    "Architecture":"x64",
    "OperatingSystem":"Windows",
    "Version":"v0.1",
    "Source":"MPAIStore",
    "Destination":""
}
]
```

```

],
"ResourcePolicies": [
    {
        "Name": "Memory",
        "Minimum": "50000",
        "Maximum": "100000",
        "Request": "75000"
    },
    {
        "Name": "CPUNumber",
        "Minimum": "1",
        "Maximum": "2",
        "Request": "1"
    },
    {
        "Name": "CPU:Class",
        "Minimum": "Low",
        "Maximum": "High",
        "Request": "Medium"
    },
    {
        "Name": "GPU:CUDA:FrameBuffer",
        "Minimum": "11GB_GDDR5X",
        "Maximum": "8GB_GDDR6X",
        "Request": "11GB_GDDR6"
    },
    {
        "Name": "GPU:CUDA:MemorySpeed",
        "Minimum": "1.60GHz",
        "Maximum": "1.77GHz",
        "Request": "1.71GHz"
    },
    {
        "Name": "GPU:CUDA:Class",
        "Minimum": "SM61",
        "Maximum": "SM86",
        "Request": "SM75"
    },
    {
        "Name": "GPU:Number",
        "Minimum": "1",
        "Maximum": "1",
        "Request": "1"
    }
],
"Documentation": [
    {
        "Type": "tutorial",
        "URI": "https://mpai.community/standards/mpai-mmci/"
    }
]
}

```

## 2. AIM metadata for ARA-VSV

### 2.1 SpeechRecognition

```

{
    "Identifier": {
        "ImplementerID": "/* String assigned by IIDRA */",
        "Specification": {
            "Name": "MPAI-MMCI",
            "AIW": "MMC-VSV",
            "AIM": "SpeechRecognition",
            "Version": "1"
        },
        "Description": "This AIM implements the speech recognition function for ARA-VSV: it converts the user's speech to text.",
        "Types": [
            {
                "Name": "Speech_t",
                "Type": "uint16[]"
            },

```

```

        {
            "Name":"Text_t",
            "Type":"{uint8[] | uint16[]}"
        },
    ],
    "Ports": [
        {
            "Name":"InputSpeech1",
            "Direction":"InputOutput",
            "RecordType":"Speech_t",
            "Technology":"Software",
            "Protocol":"",
            "IsRemote":false
        },
        {
            "Name":"RecognisedText",
            "Direction":"OutputInput",
            "RecordType":"Text_t",
            "Technology":"Software",
            "Protocol":"",
            "IsRemote":false
        }
    ],
    "SubAIMs": [
    ],
    "Topology": [
    ],
    "Implementations": [
    ],
    "Documentation": [
        {
            "Type":"Tutorial",
            "URI":"https://mpai.community/standards/mpai-mmci/"
        }
    ]
}

```

## 2.2 AvatarDescriptorParsing

```

{
    "Identifier": {
        "ImplementerID": /* String assigned by IIDRA */,
        "Specification": {
            "Name": "MPAI-MMC",
            "AIW": "MMC-VSV",
            "AIM": "AvatarDescriptorParsing",
            "Version": "2"
        },
        "Description": "This AIM implements the speech recognition function for ARA-VSV: it converts the user's speech to text."
    },
    "Types": [
        {
            "Name": "AvatarDescriptors_t",
            "Type": "uint8[]"
        },
        {
            "Name": "BodyDescriptors_t",
            "Type": "{uint8[]}"
        },
        {
            "Name": "FaceDescriptors_t",
            "Type": "{uint8[]}"
        }
    ],
    "Ports": [
        {
            "Name": "InputAvatarDescriptors",
            "Direction": "InputOutput",
            "RecordType": "AvatarDescriptors_t",
            "Technology": "Software",
            "Protocol": ""
        }
    ]
}

```

```
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"BodyDescriptors",
        "Direction":"OutputInput",
        "RecordType":"BodyDescriptors_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"FaceDescriptors",
        "Direction":"OutputInput",
        "RecordType":"FaceDescriptors_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    }
],
"SubAIMs":[
],
"Topology":[
],
"Implementations":[
],
"Documentation":[
    {
        "Type":"Tutorial",
        "URI":"https://mpai.community/standards/mpai-ara/"
    }
]
}
}
```

### 2.3 Language Understanding

```

{
  "Identifier": {
    "ImplementerID": "/* String assigned by IIDRA */",
    "Specification": {
      "Name": "MPAI-MMC",
      "AIW": "MMC-VSV",
      "AIM": "LanguageUnderstanding",
      "Version": "1"
    },
    "Description": "This AIM extracts Meaning from Recognised Text supplemented by the ID of the Physical Object and improves Recognised Text.",
    "Types": [
      {
        "Name": "Text_t",
        "Type": "{uint8[] | uint16[]} "
      },
      {
        "Name": "Tagging_t",
        "Type": "{string<256 set; string<256 result}"
      },
      {
        "Name": "Meaning_t",
        "Type": "{Tagging_t POS_tagging; Tagging_t NE_tagging; Tagging_t dependency_tagging; Tagging_t SRL_tagging}"
      }
    ],
    "Ports": [
      {
        "Name": "RecognisedText",
        "Direction": "InputOutput",
        "RecordType": "Text_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
      }
    ]
  }
}

```

```

},
{
  "Name":"InputText",
  "Direction":"InputOutput",
  "RecordType":"Text_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"RefinedText1",
  "Direction":"OutputInput",
  "RecordType":"Text_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"Meaning1",
  "Direction":"OutputInput",
  "RecordType":"Meaning_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"RefinedText2",
  "Direction":"OutputInput",
  "RecordType":"Text_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
}
],
"SubAIMs":[
],
"Topology":[
],
"Implementations":[
],
"Documentation":[
  {
    "Type":"Tutorial",
    "URI":"https://mpai.community/standards/mpai-mmci/"
  }
]
}
}

```

## 2.4 PersonalStatusExtraction

```

{
  "Identifier":{
    "ImplementerID":/* String assigned by IIDRA */,
    "Specification":{
      "Name":"MPAI-MMC",
      "AIW":"MMC-VSV",
      "AIM":"PersonalStatusExtraction",
      "Version":"2"
    },
    "Description":"This AIM extracts the combined Personal Status from Text, Speech, Face, and Gesture.",
    "Types":[
      {
        "Name":"Speech_t",
        "Type":"{uint16[]}"
      },
      {
        "Name":"BodyDescriptors_t",
        "Type":"uint8[]"
      },
      {
        "Name":"Image_t",
        "Type":"{uint8[]}"
      }
    ]
  }
}

```

```

{
    "Name":"FaceDescriptors_t",
    "Type":"uint8[]"
},
{
    "Name":"Tagging_t",
    "Type":"{string<256 set; string<256 result}"
},
{
    "Name":"Meaning_t",
    "Type":"{Tagging_t POS_tagging; Tagging_t NE_tagging; Tagging_t dependency_tag-
ging; Tagging_t SRL_tagging}"
},
{
    "Name":"PersonalStatus_t",
    "Type":"uint8[]"
}
],
"Ports": [
{
    "Name":"Meaning3",
    "Direction":"InputOutput",
    "RecordType":"Meaning_t",
    "Technology":"Software",
    "Protocol":"",
    "IsRemote":false
},
{
    "Name":"InputSpeech2",
    "Direction":"InputOutput",
    "RecordType":"Speech_t",
    "Technology":"Software",
    "Protocol":"",
    "IsRemote":false
},
{
    "Name":"BodyDescriptors",
    "Direction":"InputOutput",
    "RecordType":"BodyDescriptors_t",
    "Technology":"Software",
    "Protocol":"",
    "IsRemote":false
},
{
    "Name":"FaceDescriptors",
    "Direction":"InputOutput",
    "RecordType":"FaceDescriptors_t",
    "Technology":"Software",
    "Protocol":"",
    "IsRemote":false
},
{
    "Name":"InputPersonalStatus1",
    "Direction":"OutputInput",
    "RecordType":"PersonalStatus_t",
    "Technology":"Software",
    "Protocol":"",
    "IsRemote":false
},
{
    "Name":"InputPersonalStatus2",
    "Direction":"OutputInput",
    "RecordType":"PersonalStatus_t",
    "Technology":"Software",
    "Protocol":"",
    "IsRemote":false
}
],
"SubAIMs": [
],
"Topology": [
]
,
```

```

        "Implementations": [
            ],
            "Documentation": [
                {
                    "Type": "Tutorial",
                    "URI": "https://mpai.community/standards/mpai-mmci/"
                }
            ]
        }
    }
}

```

## 2.5 Summarisation

```

{
    "Identifier": {
        "ImplementerID": /* String assigned by IIDRA */,
        "Specification": {
            "Name": "MPAI-MMCI",
            "AIW": "MMC-VSV",
            "AIM": "Summarisation",
            "Version": "2"
        },
        "Description": "This AIM produces the Summary of the Videoconference.",
        "Types": [
            {
                "Name": "Meaning_t",
                "{uint8[]}"
            },
            {
                "Name": "Text_t",
                "{uint8[] | uint16[]}"
            },
            {
                "Name": "PersonalStatus_t",
                "Type": "uint16[]"
            },
            {
                "Name": "Summary_t",
                "Type": "uint8[]"
            }
        ],
        "Ports": [
            {
                "Name": "Meaning2",
                "Direction": "InputOutput",
                "RecordType": "Meaning_t",
                "Technology": "Software",
                "Protocol": "",
                "IsRemote": false
            },
            {
                "Name": "RefinedText",
                "Direction": "InputOutput",
                "RecordType": "Text_t",
                "Technology": "Software",
                "Protocol": "",
                "IsRemote": false
            },
            {
                "Name": "InputPersonalStatus1",
                "Direction": "InputOutput",
                "RecordType": "PersonalStatus_t",
                "Technology": "Software",
                "Protocol": "",
                "IsRemote": false
            },
            {
                "Name": "EditedSummary",
                "Direction": "InputOutput",
                "RecordType": "Summary_t",
                "Technology": "Software",
                "Protocol": "",
                "IsRemote": false
            }
        ]
    }
}

```

```

        },
        {
            "Name":"Summary1",
            "Direction":"OutputInput",
            "RecordType":"Summary_t",
            "Technology":"Software",
            "Protocol":"",
            "IsRemote":false
        }
    ],
    "SubAIMs":[
        ],
        "Topology":[
            ],
            "Implementations":[
                ],
                "Documentation":[
                    {
                        "Type":"Tutorial",
                        "URI":"https://mpai.community/standards/mpai-mmci/"
                    }
                ]
            ]
        }
    }
}

```

## 2.6 DialogueProcessing

```

{
    "Identifier":{
        "ImplementerID":/* String assigned by IIDRA */,
        "Specification":{
            "Name":"MPAI-MMC",
            "AIW":"MMC-VSV",
            "AIM":"DialogueProcessing",
            "Version":"1"
        },
        "Description":"This AIM produces the Machine's Text and Personal Status from the human's Text and Personal Status.",
        "Types":[
            {
                "Name":"Text_t",
                "{uint8[] | uint16[]}"
            },
            {
                "Name":"Meaning_t",
                "{uint8[]}"
            },
            {
                "Name":"PersonalStatus_t",
                "Type":"uint8[]"
            },
            {
                "Name":"Summary_t",
                "{uint8[]}"
            },
        ],
        "Ports":[
            {
                "Name":"InputText1",
                "Direction":"InputOutput",
                "RecordType":"Text_t",
                "Technology":"Software",
                "Protocol":"",
                "IsRemote":false
            },
            {
                "Name":"RefinedText",
                "Direction":"InputOutput",
                "RecordType":"Text_t",
                "Technology":"Software",
                "Protocol":"",
                "IsRemote":false
            }
        ]
    }
}

```

```

        "IsRemote":false
    },
{
    "Name":"Meaning1",
    "Direction":"InputOutput",
    "RecordType":"Meaning_t",
    "Technology":"Software",
    "Protocol":"",
    "IsRemote":false
},
{
    "Name":"EditedSummary",
    "Direction":"OutputInput",
    "RecordType":"Summary_t",
    "Technology":"Software",
    "Protocol":"",
    "IsRemote":false
},
{
    "Name":"Summary1",
    "Direction":"InputOutput",
    "RecordType":"Summary_t",
    "Technology":"Software",
    "Protocol":"",
    "IsRemote":false
},
{
    "Name":"InputPersonalStatus1",
    "Direction":"InputOutput",
    "RecordType":"PersonalStatus_t",
    "Technology":"Software",
    "Protocol":"",
    "IsRemote":false
},
{
    "Name":"Summary2",
    "Direction":"OutputInput",
    "RecordType":"Summary_t",
    "Technology":"Software",
    "Protocol":"",
    "IsRemote":false
},
{
    "Name":"VSPersonalStatus",
    "Direction":"InputOutput",
    "RecordType":"PersonalStatus_t",
    "Technology":"Software",
    "Protocol":"",
    "IsRemote":false
},
{
    "Name":"VText",
    "Direction":"InputOutput",
    "RecordType":"Text_t",
    "Technology":"Software",
    "Protocol":"",
    "IsRemote":false
},
],
"SubAIMS":[
],
"Topology":[
],
"Implementations":[
],
"Documentation": [
    {
        "Type":"Tutorial",
        "URI":"https://mpai.community/standards/mpai-mmcl/"
    }
]
]
```

```
    }
}
```

## 2.7 PersonalStatusDisplay

```
{
  "Identifier": {
    "ImplementerID": /* String assigned by IIDRA */,
    "Specification": {
      "Name": "MPAI-MMC",
      "AIW": "MMC-VSV",
      "AIM": "PersonalStatusDisplay",
      "Version": "2"
    },
    "Description": "This AIM outputs the Avatar Model and renders a speaking avatar from text and Personal Status.",
    "Types": [
      {
        "Name": "AvatarModel_t",
        "Type": "{uint8[]}"
      },
      {
        "Name": "Text_t",
        "Type": "{uint8[] | uint16[]}"
      },
      {
        "Name": "Speech_t",
        "Type": "uint16[]"
      },
      {
        "Name": "Avatar_t",
        "Type": "uint8[]"
      }
    ],
    "Ports": [
      {
        "Name": "VSPersonalStatus",
        "Direction": "InputOutput",
        "RecordType": "PersonalStatus_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
      },
      {
        "Name": "VSText1",
        "Direction": "InputOutput",
        "RecordType": "Text_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
      },
      {
        "Name": "AvatarModel",
        "Direction": "OutputInput",
        "RecordType": "3DGraphics_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
      },
      {
        "Name": "VSText2",
        "Direction": "OutputInput",
        "RecordType": "Text_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
      },
      {
        "Name": "VSSpeech",
        "Direction": "OutputInput",
        "RecordType": "Speech_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
      }
    ]
  }
}
```

```
        }
    {
        "Name":"VSAvatarDescriptors",
        "Direction":"OutputInput",
        "RecordType":" AvatarDescriptors_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    }
],
"SubAIMs":[
],
"Topology":[
],
"Implementations":[
],
"Documentation":[
{
    "Type":"Tutorial",
    "URI":"https://mpai.community/standards/mpai-mmci/"
}
]
}
}
```

## Annex 7 - AIW and AIM Metadata of ABV-CRX

### 1 AIW metadata for ABV-CRX

```
{  
    "$schema": "https://json-schema.org/draft/2020-12/schema",  
    "$id": "https://mpai.community/standards/resources/MPAI-AIF/V2/AIW-AIM-metadata.schema.json",  
    "title": "CAS AIF V2 AIW/AIM metadata",  
    "Identifier": {  
        "ImplementerID": /* String assigned by IIDRA */,  
        "Specification": {  
            "Standard": "MPAI-ARA",  
            "AIW": "ABV-CRX",  
            "AIM": "ABV-CRX",  
            "Version": "1"  
        }  
    },  
    "APIProfile": "Secure",  
    "Description": " This AIW composes and renders the Avatar-Based Videoconference scene.",  
    "Types": [  
        {  
            "Name": "EnvironmentModel_t",  
            "Type": "uint8[]"  
        },  
        {  
            "Name": "AvatarModel_t",  
            "Type": "uint8[]"  
        },  
        {  
            "Name": "SpatialAttitude_t",  
            "Type": "float32[6]"  
        },  
        {  
            "Name": "ParticipantID_t",  
            "Type": "uint8[]"  
        },  
        {  
            "Name": "AvatarDescriptor_t",  
            "Type": "uint8[]"  
        },  
        {  
            "Name": "Speech_t",  
            "Type": "uint16[]"  
        },  
        {  
            "Name": "PointOfView_t",  
            "Type": "float32[6]"  
        },  
        {  
            "Name": "PointOfView_t",  
            "Type": "float32[6]"  
        },  
        {  
            "Name": "OutputAudio_t",  
            "Type": "uint16[]"  
        },  
        {  
            "Name": "OutputVisual_t",  
            "Type": "uint8[]"  
        }  
    ],  
    "Ports": [  
        {  
            "Name": "EnvironmentModel",  
            "Direction": "InputOutput",  
            "RecordType": "EnvironmentModel_t",  
            "Technology": "Software",  
            "Protocol": "",  
            "IsRemote": false  
        },  
        {  
        }
```

```
        "Name":"AvatarModel",
        "Direction":"InputOutput",
        "RecordType":"AvatarModel_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"SpatialAttitude1",
        "Direction":"InputOutput",
        "RecordType":"SpatialAttitude_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"ParticipantID1",
        "Direction":"InputOutput",
        "RecordType":"ParticipantID_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"AvatarDescriptor",
        "Direction":"InputOutput",
        "RecordType":"AvatarDescriptor_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"SpatialAttitude2",
        "Direction":"InputOutput",
        "RecordType":"SpatialAttitude_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"ParticipantID2",
        "Direction":"InputOutput",
        "RecordType":"ParticipantID_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"InputSpeech",
        "Direction":"InputOutput",
        "RecordType":"Speech_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"PointOfView",
        "Direction":"OutputInput",
        "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
    }
```

```

        "RecordType":"OutputVisual_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    }
],
"SubAIMs":[
{
    "Name":"VisualSceneCreation",
    "Identifier":{
        "ImplementerID":/* String assigned by IIDRA */,
        "Specification":{
            "Standard":"MPAI-ARA",
            "AIW":"ARA-CRX",
            "AIM":"VisualSceneCreation",
            "Version":"1"
        }
    }
},
{
    "Name":"AudioSceneCreation",
    "Identifier":{
        "ImplementerID":/* String assigned by IIDRA */,
        "Specification":{
            "Standard":"MPAI-ARA",
            "AIW":"ARA-CRX",
            "AIM":"AudioSceneCreation",
            "Version":"1"
        }
    }
},
{
    "Name":"AVSceneViewer",
    "Identifier":{
        "ImplementerID":/* String assigned by IIDRA */,
        "Specification":{
            "Standard":"MPAI-ARA",
            "AIW":"ARA-CRX",
            "AIM":"AVSceneViewer",
            "Version":"1"
        }
    }
}
],
"Topology": [
    {
        "Output": {
            "AIMName": "",
            "PortName": "EnvironmentModel"
        },
        "Input": {
            "AIMName": "VisualSceneCreation",
            "PortName": "EnvironmentModel"
        }
    },
    {
        "Output": {
            "AIMName": "",
            "PortName": "AvatarModel"
        },
        "Input": {
            "AIMName": "VisualSceneCreation",
            "PortName": "AvatarModel"
        }
    },
    {
        "Output": {
            "AIMName": "",
            "PortName": "SpatialAttitude1"
        },
        "Input": {
            "AIMName": "VisualSceneCreation",
            "PortName": "SpatialAttitude1"
        }
    }
]

```

```

},
{
  "Output": {
    "AIMName": "",
    "PortName": "ParticipantID1"
  },
  "Input": {
    "AIMName": "VisualSceneCreation",
    "PortName": "ParticipantID1"
  }
},
{
  "Output": {
    "AIMName": "",
    "PortName": "AvatarDescriptor"
  },
  "Input": {
    "AIMName": "VisualSceneCreation",
    "PortName": "AvatarDescriptor"
  }
},
{
  "Output": {
    "AIMName": "",
    "PortName": "SpatialAttitude2"
  },
  "Input": {
    "AIMName": "AudioSceneCreation",
    "PortName": "SpatialAttitude2"
  }
},
{
  "Output": {
    "AIMName": "",
    "PortName": "ParticipantID2"
  },
  "Input": {
    "AIMName": "AudioSceneCreation",
    "PortName": "ParticipantID2"
  }
},
{
  "Output": {
    "AIMName": "",
    "PortName": "InputSpeech"
  },
  "Input": {
    "AIMName": "AudioSceneCreation",
    "PortName": "Speech"
  }
},
{
  "Output": {
    "AIMName": "AVSceneViewer",
    "PortName": "PointOfView"
  },
  "Input": {
    "AIMName": "",
    "PortName": "PointOfView"
  }
},
{
  "Output": {
    "AIMName": "AVSceneViewer",
    "PortName": "OutputAudio"
  },
  "Input": {
    "AIMName": "",
    "PortName": "OutputAudio"
  }
},
{
  "Output": {
    "AIMName": "AVSceneViewer",
    "PortName": ""
  }
}

```

```

        "PortName":"OutputVisual"
    },
    "Input":{
        "AIMName":"",
        "PortName":"OutputVisual"
    }
}
],
"Implementations": [
{
    "BinaryName": "aracrx.exe",
    "Architecture": "x64",
    "OperatingSystem": "Windows",
    "Version": "v0.1",
    "Source": "MPAISource",
    "Destination": ""
}
],
"ResourcePolicies": [
{
    "Name": "Memory",
    "Minimum": "50000",
    "Maximum": "100000",
    "Request": "75000"
},
{
    "Name": "CPU:Number",
    "Minimum": "1",
    "Maximum": "2",
    "Request": "1"
},
{
    "Name": "CPU:Class",
    "Minimum": "Low",
    "Maximum": "High",
    "Request": "Medium"
},
{
    "Name": "GPU:CUDA:FrameBuffer",
    "Minimum": "11GB_GDDR5X",
    "Maximum": "8GB_GDDR6X",
    "Request": "11GB_GDDR6"
},
{
    "Name": "GPU:CUDA:MemorySpeed",
    "Minimum": "1.60GHz",
    "Maximum": "1.77GHz",
    "Request": "1.71GHz"
},
{
    "Name": "GPU:CUDA:Class",
    "Minimum": "SM61",
    "Maximum": "SM86",
    "Request": "SM75"
},
{
    "Name": "GPU:Number",
    "Minimum": "1",
    "Maximum": "1",
    "Request": "1"
}
],
"Documentation": [
{
    "Type": "tutorial",
    "URI": "https://mpai.community/standards/mpai-ara/"
}
]
}
]
```

## 2 Metadata for ABV-CRX AIMs

### VisualSceneCreation

```
{  
    "Identifier": {  
        "ImplementerID": "/* String assigned by IIDRA */",  
        "Specification": {  
            "Name": "MPAI-ARA",  
            "AIW": "ABV-CRX",  
            "AIM": "VisualSceneCreation",  
            "Version": "1"  
        },  
        "Description": "This AIM composes the Visual Scene.",  
        "Types": [  
            {  
                "Name": "EnvironmentModel_t",  
                "Type": "uint8[]"  
            },  
            {  
                "Name": "AvatarModel_t",  
                "Type": "uint8[]"  
            },  
            {  
                "Name": "SpatialAttitude_t",  
                "Type": "float32[6]"  
            },  
            {  
                "Name": "ParticipantID_t",  
                "Type": "uint8[]"  
            },  
            {  
                "Name": "AvatarDescriptor_t",  
                "Type": "uint8[]"  
            },  
            {  
                "Name": "VisualSceneDescriptor_t",  
                "Type": "uint8[]"  
            }  
        ],  
        "Ports": [  
            {  
                "Name": "EnvironmentModel",  
                "Direction": "InputOutput",  
                "RecordType": "EnvironmentModel_t",  
                "Technology": "Software",  
                "Protocol": "",  
                "IsRemote": false  
            },  
            {  
                "Name": "AvatarModel",  
                "Direction": "InputOutput",  
                "RecordType": "AvatarModel_t",  
                "Technology": "Software",  
                "Protocol": "",  
                "IsRemote": false  
            },  
            {  
                "Name": "SpatialAttitude1",  
                "Direction": "OutputInput",  
                "RecordType": "SpatialAttitude_t",  
                "Technology": "Software",  
                "Protocol": "",  
                "IsRemote": false  
            },  
            {  
                "Name": "ParticipantID1",  
                "Direction": "OutputInput",  
                "RecordType": "ParticipantID_t",  
                "Technology": "Software",  
                "Protocol": "",  
                "IsRemote": false  
            }  
        ]  
    }  
}
```

```

    {
        "Name":"AvatarDescriptor",
        "Direction":"OutputInput",
        "RecordType":"AvatarDescriptor_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"VisualSceneDescriptor",
        "Direction":"InputOutput",
        "RecordType":"VisualSceneDescriptor_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    }
],
"SubAIMs":[
],
"Topology":[
],
"Implementations":[
],
"Documentation":[
    {
        "Type":"Tutorial",
        "URI":"https://mpai.community/standards/mpai-ara/"
    }
]
}
}

```

## AudioSceneCreation

```

{
    "Identifier":{
        "ImplementerID":/* String assigned by IIDRA */,
        "Specification":{
            "Name":"ARA",
            "AIW":"CRX",
            "AIM":"AudioSceneCreation",
            "Version":"1"
        },
        "Description":"This AIM composes the Audio Scene.",
        "Types":[
        {
            "Name": "SpatialAttitude_t",
            "Type": "float32[6]"
        },
        {
            "Name": "ParticipantID_t",
            "Type": "uint8[]"
        },
        {
            "Name": "InputSpeech_t",
            "Type": "uint16[]"
        },
        {
            "Name": "AudioSceneDescriptor_t",
            "Type": "uint8[]"
        }
    ],
    "Ports":[
        {
            "Name":"SpatialAttitude2",
            "Direction":"OutputInput",
            "RecordType":"SpatialAttitude_t",
            "Technology":"Software",
            "Protocol":"",
            "IsRemote":false
        },
        {
            "Name":"VisualScene2",
            "Direction":"InputOutput",
            "RecordType":"VisualScene_t",
            "Technology":"Software",
            "Protocol":"",
            "IsRemote":false
        }
    ]
}

```

```

        {
            "Name":"ParticipantID2",
            "Direction":"OutputInput",
            "RecordType":"ParticipantID_t",
            "Technology":"Software",
            "Protocol":"",
            "IsRemote":false
        }
    {
        "Name":"InputSpeech",
        "Direction":"OutputInput",
        "RecordType":"Speech_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    },
    {
        "Name":"AudioSceneDescriptor",
        "Direction":"InputOutput",
        "RecordType":"AudioSceneDescriptor_t",
        "Technology":"Software",
        "Protocol":"",
        "IsRemote":false
    }
],
"SubAIMs":[
],
"Topology":[
],
"Implementations":[
],
"Documentation":[
{
    "Type":"Tutorial",
    "URI":"https://mpai.community/standards/mpai-ara/"
}
]
}
}

```

## AVSceneViewer

```

{
    "Identifier":{
        "ImplementerID":/* String assigned by IIDRA */,
        "Specification":{
            "Name":"MPAI-ARA",
            "AIW":"ABV-CRX",
            "AIM":"AVSceneViewer",
            "Version":"1"
        },
        "Description":"This AIM renders the Audio-Visual Scene.",
        "Types":[
        {
            "Name": "VisualSceneDescriptor_t",
            "Type": "uint8[]"
        },
        {
            "Name": "AudioSceneDescriptor_t",
            "Type": "uint8[]"
        },
        {
            "Name": "OutputAudio_t",
            "Type": "uint16[]"
        },
        {
            "Name": "OutputVisual_t",
            "Type": "uint8[]"
        }
    ],
    "Ports":[

```

```

{
  "Name":"VisualSceneDescriptor",
  "Direction":"OutputInput",
  "RecordType":" VisualSceneDescriptor_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"AudioSceneDescriptor",
  "Direction":"OutputInput",
  "RecordType":" AudioSceneDescriptor_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"PointOfView",
  "Direction":"InputOutput",
  "RecordType":" PointOfView_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"OutputAudio",
  "Direction":"InputOutput",
  "RecordType":" OutputAudio_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"OutputVisual",
  "Direction":"InputOutput",
  "RecordType":" OutputVisual_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
}
],
"SubAIMs":[
],
"Topology":[
],
"Implementations":[
],
"Documentation":[
  {
    "Type":"Tutorial",
    "URI":"https://mpai.community/standards/mpai-ara/"
  }
]
}
}

```

## Annex 8 - Metadata of Personal Status Display Composite AIM

### 1. Metadata of PersonalStatusDisplay

```
{  
    "Identifier":{  
        "ImplementerID":/* String assigned by IIDRA */,  
        "Specification":{  
            "Name":"MPAI-ARA",  
            "AIW": "",  
            "AIM":"PersonalStatusDisplay",  
            "Version":"1"  
        },  
        "Description":"This AIM implements Personal Status Display function.",  
        "Types": [  
            {  
                "Name":"OutputSelection_t",  
                "Type":"{AvatarDescriptors_t | Avatar_t}"  
            },  
            {  
                "Name":"Text_t",  
                "Type":"uint8[] | uint16[]"  
            },  
            {  
                "Name":"PSSpeech_t",  
                "Type":"uint8[]"  
            },  
            {  
                "Name":"AvatarModel_t",  
                "Type":"uint8[]"  
            },  
            {  
                "Name":"PSFace_t",  
                "Type":"uint8[]"  
            },  
            {  
                "Name":"Speech_t",  
                "Type":"uint26[]"  
            },  
            {  
                "Name":"AvatarDescriptors_t",  
                "Type":"uint8[]"  
            },  
            {  
                "Name":"Avatar_t",  
                "Type":"uint8[]"  
            },  
        ],  
        "Ports": [  
            {  
                "Name":"OutputSelection",  
                "Direction":"InputOutput",  
                "RecordType":"Selection_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":"MachineText1",
    "Direction":"OutputInput",
    "RecordType":"Text_t",
    "Technology":"Software",
    "Protocol":"",
    "IsRemote":false
}
```

```

},
{
  "Name":"MachineSpeech2",
  "Direction":"InputOutput",
  "RecordType":"Speech_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"AvatarDescriptors",
  "Direction":"InputOutput",
  "RecordType":"AvatarDescriptors_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
},
{
  "Name":"MachineAvatar",
  "Direction":"InputOutput",
  "RecordType":"Avatar_t",
  "Technology":"Software",
  "Protocol":"",
  "IsRemote":false
}
],
"SubAIMs":[
  {
    "Name":"SpeechSynthesisPS",
    "Identifier":{
      "ImplementerID":/* String assigned by IIDRA */,
      "Specification":{
        "Standard":"MPAI-ARA ",
        "AIW":"",
        "AIM":"SpeechSynthesisPS",
        "Version":"1"
      }
    }
  },
  {
    "Name":"FaceDescription",
    "Identifier":{
      "ImplementerID":/* String assigned by IIDRA */,
      "Specification":{
        "Standard":"MPAI-ARA ",
        "AIW":"",
        "AIM":"FaceDescription",
        "Version":"2"
      }
    }
  },
  {
    "Name":"BodyDescription",
    "Identifier":{
      "ImplementerID":/* String assigned by IIDRA */,
      "Specification":{
        "Standard":"MPAI-ARA ",
        "AIW":"",
        "AIM":"BodyDescription",
        "Version":"2"
      }
    }
  },
  {
    "Name":"AvatarDescription",
    "Identifier":{
      "ImplementerID":/* String assigned by IIDRA */,
      "Specification":{
        "Standard":"MPAI-ARA ",
        "AIW":"",
        "AIM":"AvatarDescription",
        "Version":"2"
      }
    }
  }
]

```

```

},
{
  "Name":"AvatarSynthesisPS",
  "Identifier":{
    "ImplementerID":/* String assigned by IIDRA */,
    "Specification":{
      "Standard":"MPAI-ARA ",
      "AIW":"",
      "AIM":"AvatarSynthesisPS",
      "Version":"2"
    }
  }
],
"Topology":[
{
  "Output": {
    "AIMName":"",
    "PortName":"OutputSelection"
  },
  "Input": {
    "AIMName":"",
    "PortName":"OutputSelection"
  }
},
{
  "Output": {
    "AIMName":"",
    "PortName":"MachineText1"
  },
  "Input": {
    "AIMName":"",
    "PortName":"MachineText1"
  }
},
{
  "Output": {
    "AIMName":"",
    "PortName":"MachineText2"
  },
  "Input": {
    "AIMName":"SpeechSynthesisPS",
    "PortName":"MachineText2"
  }
},
{
  "Output": {
    "AIMName":"",
    "PortName":"PSSpeech"
  },
  "Input": {
    "AIMName":"SpeechSynthesisPS",
    "PortName":"PSSpeech"
  }
},
{
  "Output": {
    "AIMName":"SpeechSynthesisPS",
    "PortName":"MachineSpeech"
  },
  "Input": {
    "AIMName":"FaceDescription",
    "PortName":"MachineSpeech"
  }
},
{
  "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"
            }
        },
        {
            "Output": {
                "AIMName": "",
                "PortName": "AvatarModel3"
            },
            "Input": {
                "AIMName": "",
                "PortName": "AvatarModel3"
            }
        },
        {
            "Output": {
                "AIMName": "GestureDescription",
                "PortName": "FaceDescriptors"
            },
            "Input": {
                "AIMName": "AvatarDescription",
                "PortName": "GestureDescriptors"
            }
        },
        {
            "Output": {
                "AIMName": "AvatarDescription",
                "PortName": "AvatarDescription"
            }
        }
    ]
}
```

```

        "PortName":"AvatarDescriptors"
    },
    "Input":{

        "AIMName":"AvatarSynthesisPS",
        "PortName":"AvatarDescriptors"
    }
},
{
    "Output":{

        "AIMName":"",
        "PortName":"MachineText"
    },
    "Input":{

        "AIMName":"PSFaceInterpretation",
        "PortName":"MachineText"
    }
},
{
    "Output":{

        "AIMName":"SpeechSynthesisPS",
        "PortName":"MachineSpeech"
    },
    "Input":{

        "AIMName":"",
        "PortName":"MachineSpeech"
    }
},
{
    "Output":{

        "AIMName":"FaceDescription",
        "PortName":"FaceDescriptors"
    },
    "Input":{

        "AIMName":"AvatarDescription",
        "PortName":"FaceDescriptors"
    }
},
{
    "Output":{

        "AIMName":"GestureDescription",
        "PortName":"GestureDescriptors"
    },
    "Input":{

        "AIMName":"AvatarDescription",
        "PortName":"GestureDescriptors"
    }
},
{
    "Output":{

        "AIMName":"AvatarDescription",
        "PortName":"AvatarDescriptors"
    },
    "Input":{

        "AIMName":"",
        "PortName":"AvatarDescriptors"
    }
},
{
    "Output":{

        "AIMName":"AvatarDescription",
        "PortName":"AvatarDescriptors"
    },
    "Input":{

        "AIMName":"AvatarSynthesisPS",
        "PortName":"AvatarDescriptors"
    }
},
{
    "Output":{

        "AIMName":"AvatarSynthesisPS",
        "PortName":"MachineAvatar"
    },
    "Input":{

        "AIMName": ""
    }
}
]
```

```

        "PortName":"MachineAvatar"
    }
],
"Implementations": [
],
"Documentation": [
{
    "Type":"Tutorial",
    "URI":"https://mpai.community/standards/mpai-ara/"
}
]
}
}
```

## 1.1 SpeechSynthesisPS

```
{
"Identifier": {
    "ImplementerID": "/* String assigned by IIDRA */",
    "Specification": {
        "Name": "MPAI-ARA",
        "AIW": "",
        "AIM": "SpeechSynthesisPS",
        "Version": "2"
    },
    "Description": "This AIM implements the Speech Synthesis with Personal Status function.",
    "Types": [
        {
            "Name": "Text_t",
            "Type": "uint8[]"
        },
        {
            "Name": "PSSpeech_t",
            "Type": "uint8[]"
        },
        {
            "Name": "Speech_t",
            "Type": "uint16[]"
        }
    ],
    "Ports": [
        {
            "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": "MachineSpeech1",
            "Direction": "InputOutput",
            "RecordType": "Speech_t",
            "Technology": "Software",
            "Protocol": "",
            "IsRemote": false
        },
        {
            "Name": "MachineSpeech2",
            "Direction": "InputOutput",
            "RecordType": "Speech_t",
            "Technology": "Software",
            "Protocol": "",
            "IsRemote": false
        }
    ]
}
```

```

        }
    ],
    "SUbAIMs": [
        ],
        "Topology": [
            ],
            "Implementations": [
                ],
                "Documentation": [
                    {
                        "Type": "Tutorial",
                        "URI": "https://mpai.community/standards/mpai-ara/"
                    }
                ]
            }
        }
    }
}

```

## 1.2 FaceDescription

```

{
    "Identifier": {
        "ImplementerID": "/* String assigned by IIDRA */",
        "Specification": {
            "Name": "MPAI-ARA",
            "AIW": "",
            "AIM": "FaceDescription",
            "Version": "2"
        },
        "Description": "This AIM implements the Face Description function.",
        "Types": [
            {
                "Name": "Speech_t",
                "Type": "uint16[]"
            },
            {
                "Name": "AvatarModel_t",
                "Type": "uint8[]"
            },
            {
                "Name": "PSFace_t",
                "Type": "uint8[]"
            },
            {
                "Name": "FaceDescriptors_t",
                "Type": "uint8[]"
            }
        ],
        "Ports": [
            {
                "Name": "MachineSpeech2",
                "Direction": "InputOutput",
                "RecordType": "Speech_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":"FaceDescriptors",
            "Direction":"OutputInput",
            "RecordType":"FaceDescriptors_t",
            "Technology":"Software",
            "Protocol":"",
            "IsRemote":false
        }
    ],
    "SUbAIMs":[
        ],
        "Topology":[
        ],
        "Implementations":[
        ],
        "Documentation":[
            {
                "Type":"Tutorial",
                "URI":"https://mpai.community/standards/mpai-ara/"
            }
        ]
    }
}

```

### 1.3 BodyDescription

```

{
    "Identifier":{
        "ImplementerID":/* String assigned by IIDRA */,
        "Specification":{
            "Name":"MPAI-ARA",
            "AIW":"",
            "AIM":"BodyDescription",
            "Version":"1"
        },
        "Description":"This AIM implements the Body Description function.",
        "Types":[
            {
                "Name":"Text_t",
                "Type":"{uint8[] | uint16[]}"
            },
            {
                "Name":"AvatarModel_t",
                "Type":"uint8[]"
            },
            {
                "Name":"PSGesture_t",
                "Type":"uint8[]"
            },
            {
                "Name":"GestureDescriptors_t",
                "Type":"uint8[]"
            }
        ],
        "Ports":[
            {
                "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":"GestureDescriptors",
            "Direction":"OutputInput",
            "RecordType":"GestureDescriptors_t",
            "Technology":"Software",
            "Protocol":"",
            "IsRemote":false
        }
    ],
    "SUbAIMs":[
        ],
        "Topology":[
        ],
        "Implementations":[
        ],
        "Documentation":[
            {
                "Type":"Tutorial",
                "URI":"https://mpai.community/standards/mpai-ara/"
            }
        ]
    }
}

```

## 1.4 AvatarDescription

```

{
    "Identifier":{
        "ImplementerID":/* String assigned by IIDRA */,
        "Specification":{
            "Name":"MPAI-ARA",
            "AIW":"",
            "AIM":"AvatarDescription",
            "Version":"1"
        },
        "Description":"This AIM implements the Avatar Description function.",
        "Types":[
            {
                "Name":"FaceDescriptors_t",
                "Type":"uint8[]"
            },
            {
                "Name":"GestureDescriptors_t",
                "Type":"uint8[]"
            },
            {
                "Name":"AvatarDescriptors_t",
                "Type":"uint8[]"
            }
        ],
        "Ports":[
            {
                "Name":"FaceDescriptors",
                "Direction":"InputOutput",

```

```

        "RecordType": "FaceDescriptors_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "GestureDescriptors",
        "Direction": "InputOutput",
        "RecordType": "GestureDescriptors_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
    {
        "Name": "AvatarDescriptors2",
        "Direction": "OutputInput",
        "RecordType": "AvatarDescriptors_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    }
],
"SubAIMs": [
],
"Topology": [
],
"Implementations": [
],
"Documentation": [
{
    "Type": "Tutorial",
    "URI": "https://mpai.community/standards/mpai-ara/"
}
]
}
}

```

## 1.5 AvatarSynthesisPS

```

{
    "Identifier": {
        "ImplementerID": "/* String assigned by IIDRA */",
        "Specification": {
            "Name": "MPAI-ARA",
            "AIW": "",
            "AIM": "AvatarSynthesisPS",
            "Version": "2"
        },
        "Description": "This AIM implements the Avatar Synthesis with Personal Status function.",
        "Types": [
            {
                "Name": "AvatarDescriptors_t",
                "Type": "uint8[]"
            },
            {
                "Name": "Avatar_t",
                "Type": "uint8[]"
            }
        ],
        "Ports": [
            {
                "Name": "AvatarDescriptors",
                "Direction": "InputOutput",
                "RecordType": "AvatarDescriptors_t",
                "Technology": "Software",
                "Protocol": "",
                "IsRemote": false
            },
            {
                "Name": "MachineAvatar",
                "Direction": "OutputInput",
                "RecordType": "MachineAvatar_t",
                "Technology": "Software",
                "Protocol": "",
                "IsRemote": false
            }
        ]
    }
}

```

```
        "RecordType": "Avatar_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },
],
"SUbAIMs": [
],
"Topology": [
],
"Implementations": [
],
"Documentation": [
{
    "Type": "Tutorial",
    "URI": "https://mpai.community/standards/mpai-ara/"
}
]
}
}
```