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

MPAI Technical Specification

Context-based Audio Enhancement
MPAI-CAE

V2.1

WARNING

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

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

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

Context-based Audio Enhancement (MPAI-CAE) V2.1

Contents

1 Introduction

In recent years, Artificial Intelligence (AI) and related technologies, applied to a broad range of applications, have started affecting the life of millions of people and they are expected to do so even 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. Indeed, research has shown that data coding with AI-based technologies is generally *more efficient* than with existing technologies for, e.g., compression and feature-based description.

However, some AI technologies may carry inherent risks, e.g., in terms of bias toward some classes of users. Therefore, the need for standardisation is more important and urgent than ever.

The international, unaffiliated, not-for-profit MPAI – Moving Picture, Audio and Data Coding by Artificial Intelligence Standards Developing Organisation has the mission to develop *AI-enabled data coding standards*. MPAI Application Standards enable the development of AI-based products, applications, and services.

As a part of its mission, MPAI has developed standards operating procedures to enable a user of MPAI implementations to make informed decision about their applicability. Central to this is the notion of Performance, defined as a set of attributes characterising a reliable and trustworthy implementation.

Therefore, to fully achieve the MPAI mission, technical standards must be complemented by the creation and management of an ecosystem designed to underpin the life cycle of MPAI standards through the steps of specification, technical testing, assessment of product safety and security, and distribution.

In the following, Terms beginning with a capital letter are defined in *Table 1* if they are specific to this Standard and in *Table 33* if they are common to all MPAI Standards.

The MPAI Ecosystem, fully specified in [1], is composed of:

- MPAI as provider of Technical, Conformance and Performance Specifications.
- Implementers of MPAI standards.
- MPAI-appointed Performance Assessors.
- The MPAI Store which assigns Implementer identifiers (ImplementerID's) and distributes validated Implementations.

The common infrastructure enabling the implementation of MPAI Application Standards is the AI Framework (AIF) Standard (MPAI-AIF).

Figure 1 depicts the MPAI-AIF Reference Model under which Implementations of MPAI Application Standards and user-defined MPAI-AIF conforming applications operate.

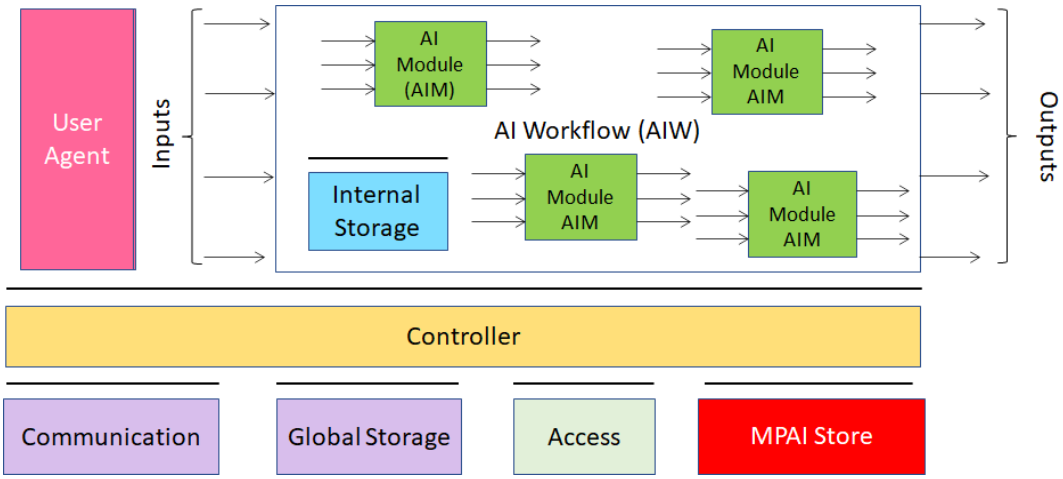

Figure 1 - The AI Framework (AIF) Reference Model and its Components

An AIF Implementation allows execution of AI Workflows (AIW), composed by basic processing elements called AI Modules (AIM).
MPAI Application Standards normatively specify Semantics and Syntax 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.
In particular, an AIM is defined by its Function and Data, but not by its internal architecture, which may be based on AI or data processing, and implemented in software, hardware or hybrid software and hardware technologies.

MPAI defines Interoperability as the ability to replace an AIW or an AIM Implementation with a functionally equivalent Implementation. MPAI also defines 3 Interoperability Levels of an AIW that executes an AIW. The AIW may have 3 Levels:
Level 1 – Implementer-specific and satisfying the MPAI-AIF Standard.
Level 2 – Specified by an MPAI Application Standard.
Level 3 – Specified by an MPAI Application Standard and certified by a Performance Assessor.
MPAI offers Users access to the promised benefits of AI with a guarantee of increased transparency, trust and reliability as the Interoperability Level of an Implementation moves from 1 to 3. Additional information on Interoperability Levels is provided in Annex 3.

The Chapters and Annexes of this Technical Specification are Normative unless they are labelled as Informative.

2 Scope

Technical Specification: Context-based Audio Enhancement (MPAI-CAE) V2.1 specifies technologies that improve the user experience for audio-related applications including entertainment, communication, teleconferencing, gaming, post-production, restoration etc. in a variety of contexts such as in the home, in the car, on-the-go, in the studio etc. using context information to act on the input audio content, and potentially deliver the processed output via an appropriate protocol. MPAI-CAE specifies four Use Cases and one Composite AIM. The Use Cases are: *Emotion Enhanced Speech (EES)*, *Audio Recording Preservation (ARP)*, *Speech Restoration System (SSR)*, and *Enhanced Audioconference Experience (EAE)*; the Composite AIM is *Audio Scene Description (ASD)*.

Each Use Case normatively defines:
1. The Functions of the AIW 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. Have the AIW Function specified in the appropriate Section of Chapter 4.2.
   b. Have all its AIMs and their Connections conforming with the AIW Reference Model specified in the appropriate Section of Chapter 4.2.
   c. The AIW and AIM input and output data should have the Formats specified in the appropriate Subsection of Section 6.
2. An *AIM*, shall:
   a. Have the AIM Function specified by the appropriate Section of Chapter 4.2.
   b. Have input and output data Formats conforming with the appropriate Subsection of Section 6.
   c. Receive as input and produce as output data having the Format specified in Section 6.
3. A data *Format*, the data shall have the Format specified in Section 6.

Users of this Technical Specification should note that:
1. This Technical Specification defines Interoperability Levels but does not mandate any.
2. Implementers are free to decide the Interoperability Level their Implementation should satisfy.
3. Implementers can use the Reference Software 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. The MPAI Ecosystem outlined in Annex 3 is governed by [1].
7. Implementers and Users should consider the notices and disclaimers of Annex 2.

MPAI-CAE V2.1 includes:
1. The Scope (This Chapter)
2. Terms and Definitions
3. Normative and Informative References
4. Use Cases
5. Audio Scene Description Composite AIM
6. Data Types
7. Annexes (general concerns, some informative)

This version of the MPAI-CAE Technical Specification has been developed by the CAE-DC Development Committee. Future Versions may revise and/or extend the Scope of the Standard.

3 Terms and Definitions

The Terms used in this standard whose first letter is capital have the meaning defined in *Table 1*. The general MPAI Terms are defined in *Table 33*.

Table 1 – Table of terms and definitions

Term	Definition
Access Copy Files	Set of files providing the information stored in an audio tape recording, including Restored Audio Files, suitable for audio information access, but not for long-term preservation.
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 64.
Audio Block	A set of consecutive Audio samples.
Audio Channel	A sequence of Audio Blocks.
Audio File	A .wav file [10].
Audio Object	Audio source which is in the audible frequency band.
Audio Scene Geometry	Spatial information for the Audio Objects which are included in an audio scene.
Audio Segment	An Audio Block with Start Time and an End Time Labels corresponding to the time of the first and last sample of the Audio Segment, respectively.
Audio-Visual File	A file containing audio and video according to the MP4 File Format [14].
Capstan	The capstan is a rotating spindle used to move recording tape through the mechanism of a tape recorder.
Damaged List	A list of strings of Texts corresponding to the Damaged Segments (if any) requiring replacement with synthetic segments.
Damaged Section	An Audio Segment which is damaged in its entirety and is contained in a Damaged Segment.
Damaged Segment	An Audio Segment containing only speech (and not containing music or other sounds) which is either damaged in its entirety or contains one or more Damaged Sections specified in the Damaged List.
Degree	Strength of a feature, specifically, with respect to Emotion, “High,” “Medium,” or “Low.”
Editing List	The description of the speed, equalisation and reading backwards corrections occurred during the restoration process.
Emotion	A Data Type representing the internal status of a human or avatar resulting from their interaction with the context or subsets of it, such as “Angry”, and “Sad”.
Emotionless Speech	An Audio File containing speech without music and other sounds, and in which little or no identifiable emotion is perceptible by native listeners.
Irregularity	An event of interest to preservation in *Table 28* and *Table 29*
Irregularity File	A JSON file containing information about Irregularities of the ARP inputs.
Irregularity Image	An image corresponding to an Irregularity.
JSON	JavaScript object notation [18].

Microphone Array Geometry	Description of the position of each microphone comprising the microphone array and specific characteristics such as microphone type, look directions, and the array type.
Model Utterance	An Audio Segment used as a model or demonstration of the Emotion to be added to Emotionless Speech in order to produce Speech with Emotion.
Multichannel Audio	A data structure containing at least 2 time-aligned interleaved Audio Channels.
Multichannel Audio Stream	A data structure containing Audio Objects packaged with Audio Scene Geometry.
Neural Network Speech Model	A Neural Network Model trained on Speech Segments for Modelling and used to synthesize replacements for the entire Damaged Segment or Damaged Sections within it.
Passthrough AIM	An AIM with the same input and output data of an AIM without executing the Function of that AIM. E.g., a Noise Cancellation AIM that does not cancel the noise.
Preservation Audio File	The input Audio File resulting from the digitisation of an audio open-reel tape to be preserved and, in case, restored.
Preservation Audio-Visual File	The input Audio-Visual File produced by a camera pointed to the playback head of the magnetic tape recorder and the synchronised Audio resulting from the tape digitisation process.
Preservation Image	A Video frame extracted from Preservation Audio-Visual File.
Preservation Master Files	Set of files providing the information stored in an audio tape recording without any restoration. As soon as the original analogue recordings is no more accessible, it becomes the new item for long-term preservation.
Restored Audio Files	Set of Audio Files derived from the Preservation Audio File, where potential speed, equalisation or reading backwards errors that occurred in the digitisation process have been corrected.
Restored Speech Segment	An Audio Segment in which the entire segment has been replaced by a synthetic speech segment, or in which each Damaged Segment has been replaced by a synthetic speech segment.
Speech Features	Descriptor representing a variety of information elements incorporated in a Speech Segment, e.g., personal identity, Personal Status, additional factors such as vocal tension, creakiness, whispery quality, etc.
Speech Segments for Modelling	A set of Audio Files containing speech segments used to train the Neural Network Speech Model.
Speech With Emotion File	An Audio File containing speech with emotional features.
Spherical Coordinate System	A coordinate system where the position of a point is specified by three numbers: the radial distance of that point from a fixed origin, its polar angle measured from a fixed zenith direction, and the azimuthal angle of its orthogonal projection on a reference plane.
Spherical Grid Resolution	The maximum spherical angle between any two neighbouring sampled points on a sphere.
Text List	List of texts to be converted into speech by the Speech Synthesis for Restoration AIM.
Time Code	Number of ms from 1970-01-01T00:00:00.000 according to [8].
Time Label	A measure of time from a context-dependent zero time expressed as HH:mm:ss.SSS.

Transform Audio	A frequency representation of Audio.
Enhanced Transform Audio	Transform Audio whose samples are Enhanced Transform Audio samples.
Useful Signal	Digital signal resulting from the A/D conversion of the analogue signal recorded in an audio tape.

4 References

4.1 Normative References

This standard normatively references the following technical specifications, both from MPAI and other standard organisations:

1. MPAI; Technical Specification: The governance of the MPAI Ecosystem (MPAI-GME) V1.1; https://mpai.community/standards/mpai-gme/
2. MPAI; Technical Specification: Artificial Intelligence Framework (MPAI-AIF) V2.0; https://mpai.community/standards/mpai-aif/
3. MPAI; Technical Specification: Connected Autonomous Vehicles (MPAI-CAV) – Architecture V1.0; https://mpai.community/standards/mpai-cav/
4. MPAI; Technical Specification: Multimodal Conversation (MPAI-MMC) V2.1; https://mpai.community/standards/mpai-mmc/
5. MPAI; Technical Specification: Object and Scene Description (MPAI-OSD); https://mpai.community/standards/mpai-osd/
6. MPAI; Technical Specification: Portable Avatar Format (MPAI-PAF); https://mpai.community/standards/mpai-paf/
7. A Universally Unique IDentifier (UUID) URN Namespace; IETF RFC 4122; July 2005.
8. Date and Time on the Internet: Time Stamps; IETF RFC 3339; July 2002.
9. Universal Coded Character Set (UCS): ISO/IEC 10646; December 2020.
10. ITU-R BS.2088-1 (10/2019) - Long-form file format for the international exchange of audio programme materials with metadata.
11. ISO/IEC 14496-10; Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding.
12. ISO/IEC 23008-2; Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 2: High Efficiency Video Coding.
13. ISO/IEC 23094-1; Information technology – General video coding – Part 1: Essential Video Coding.
14. ISO/IEC 14496-12; Information technology – Coding of audio-visual objects – Part 12: ISO base media file format.
15. ZIP format, https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT.
16. Neural Network Exchange Format; https://www.khronos.org/registry/NNEF/specs/1.0/nnef-1.0.4.pdf; Khronos.
17. Open Neural Network Exchange (ONNX) format; https://www.ONNX.ai.
18. The JavaScript Object Notation (JSON) Data Interchange Format; https://datatracker.ietf.org/doc/html/rfc8259; IETF rfc8259; December 2017.
19. BS EN 60094-1:1994, BS 6288-1: 1994, IEC 94-1:1981 - Magnetic tape sound recording and reproducing systems - Part 1: Specification for general conditions and requirements.
20. K. Bradley, IASA TC-04 Guidelines in the Production and Preservation of Digital Audio Objects: standards, recommended practices, and strategies., 2nd ed. International Association of Sound and Audiovisual Archives, (2009): 2014.
21. MPAI; The MPAI Statutes; https://mpai.community/statutes/
22. MPAI; The MPAI Patent Policy; https://mpai.community/about/the-mpai-patent-policy/.

23. Framework Licence of the Context-based Audio Enhancement Technical Specification (MPAI-CAE); https://mpai.community/standards/mpai-cae/framework-licence/
24. ITU-R BS.2088-1: Long-form file format for the international exchange of audio programme materials with metadata.
25. ITU-T T-81: Information technology — Digital compression and coding of continuous-tone still images: Requirements and guidelines.

4.2 Informative References

The references provided here are for information purpose.

26. Ekman, Paul (1999), "Basic Emotions", in Dalgleish, T; Power, M (eds.), Handbook of Cognition and Emotion (PDF), Sussex, UK: John Wiley & Sons.
27. B. Rafaely, Fundamentals of spherical array processing, Springer, 2018.

5 Use Cases

MPAI implements Use Cases with AI Workflows (AIW) conforming with Technical Specification: Artificial Intelligence Framework (MPAI-AIF) V2.1. Each AIW i Use Case includes:

1. Functions of the AIW
2. Reference Model of the AIW
3. I/O data of the AIW
4. Functions of AIMs
5. Web links to the AIW, AIMs, and JSON Metadata.

5.1 Emotion-Enhanced Speech (CAE-EES)

5.1.1 Functions

Speech carries information not only about its lexical content, but also about several other aspects including age, gender, identity, and **emotional state of the speaker**. Speech synthesis is evolving towards support of these aspects.

In many use cases, emotional force can usefully be added to speech which by default would be neutral or emotionless, possibly with grades of a particular emotion. For instance, in a human-machine dialogue, messages conveyed by the machine can be more effective if they carry emotions appropriately related to the emotions detected in the human speaker.

Emotion-Enhanced Speech (EES):

1. Enables a user to indicate a model utterance or an Emotion to obtain an emotionally charged version of a given utterance.
2. Converts an individual emotionless speech segment to a segment that has a specified emotion. Both input and output speech segments are contained in files. The desired emotion is expressed either as a tag belonging to a standard list of emotions or derived by extracting features from a model utterance. EES produces an output speech segment with emotion.

CAE-EES implementations can be used to create virtual agents communicating as naturally as possible, and thus improve the quality of human-machine interaction by bringing it closer to human-human interchange.

5.1.2 Reference Model

The Emotion-Enhanced Speech Reference Model depicted in ***Error! Reference source not found.*** s upports two Modes or pathways enabling addition of emotional charge to an emotionless or neutral input utterance (Emotion-less Speech).

1. Along Pathway 1 (***Error! Reference source not found.***), upper and middle left), a Model U tterance is input together with the neutral utterance Emotion-less Speech, so that features of the former can be captured and transferred to the latter.

2. Alternatively, along Pathway 2 (***Error! Reference source not found.***), middle and lower left), n eutral utterance Emotionless Speech is input along with a specification of the desired Emotion. Speech Feature Analysis2 extracts Emotionless Speech Features from Emotionless Speech, which describe its initial state. These are sent to Emotion Feature Production, which produces (emotional) Neural Speech Features that can add the desired emotional charge to Emotionless Speech. These Neural Speech Features are sent to Neural Emotion Insertion, which combines Emotionless Speech and the (emotional) Neural Speech Features set. Speech with Emotion is then produced as output.

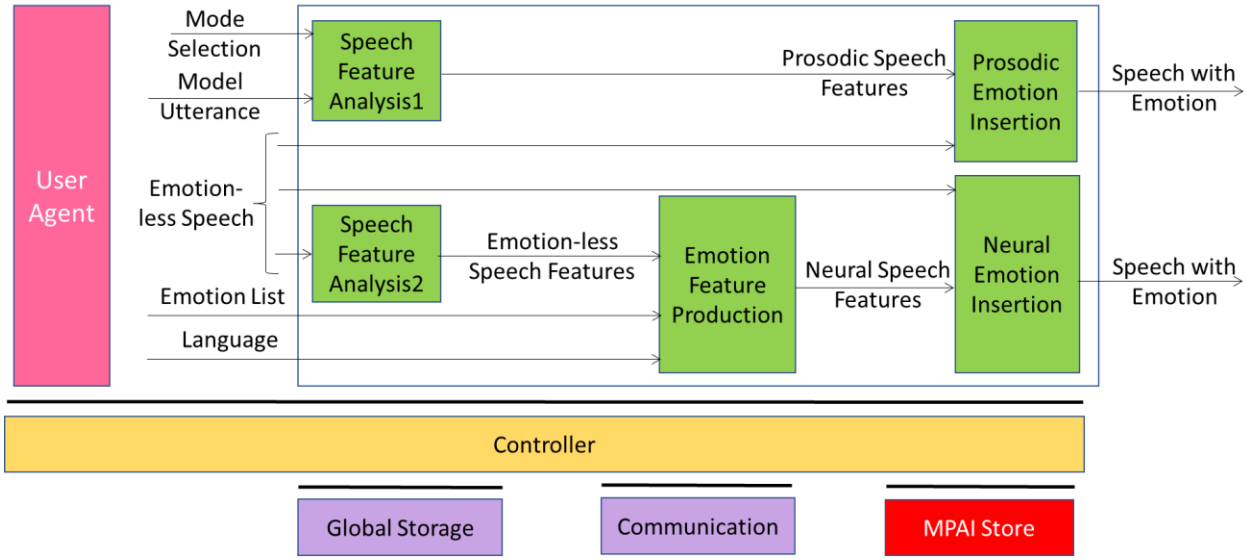

Figure 2 - Emotion-Enhanced Speech Reference Model

5.1.3 I/O data of AI Workflow

Table 2 gives the input and output data of Emotion-Enhanced Speech.

Table 2 – I/O data of Emotion-Enhanced Speech

Input data	Comments
Emotionless Speech	See definition in *Table 1*.
Emotion	See definition in *Table 1*.
Model Utterance	See definition in *Table 1*.
Output data	**Comments**
Speech with Emotion	See definition in *Table 1*.

5.1.4 Functions of AI Modules

The AI Modules of ***Error! Reference source not found.*** perform the functions described in *Table 3*.

Table 3 – AI Modules of Emotion-Enhanced Speech

AIM	Function
Speech Feature Analysis 1	Extracts Neural Speech Features of a model emotional utterance and transfers them to the Prosodic Emotion Insertion AIM.
Speech Feature Analysis 2	Extracts Emotionless Speech Features of an emotionless input utterance, passing these to the Emotion Feature Production AIM.

Emotion Feature Production	Receives the Emotionless Speech Features produced by Speech Feature Analysis2 plus a list of Emotions to be added. (If the Degree of an Emotion is not specified, the Medium value is used.)
Prosodic Emotion Insertion	Integrates the (emotional) Prosodic Speech Features with those of the Emotionless Speech input, yielding and delivering an emotionally modified utterance.
Neural Emotion Insertion	Integrates the (emotional) Neural Speech Features with those of the Emotionless Speech input, yielding and delivering an emotionally modified utterance.

5.1.5 I/O Data of AI Modules

Table 4 – CAE-EES AIMs and their data

AIM	Input Data	Output Data
Speech Features Analysis1	Model Utterance	Prosodic Speech Features
Speech Features Analysis2	Emotionless Speech	Emotionless Speech Features
Emotion Feature Production	Emotionless Speech Features Emotion List Language	Neural Speech Features
Prosodic Emotion Insertion	Emotionless Speech Prosodic Speech Features	Speech with Emotion
Neural Emotion Insertion	Emotionless Speech Neural Speech Features	Speech with Emotion

5.1.6 AIW, AIMs, and JSON Metadata

Table 5 – AIW, AIMs, and JSON Metadata

AIW	AIMs	Name	JSON
CAE-EES		Emotion Enhanced Speech	File
	CAE-SF1	Speech Feature Analysis 1	File
	CAE-SF2	Speech Feature Analysis 2	File
	CAE-EFP	Emotion Feature Production	File
	CAE-PEI	Prosodic Emotion Insertion	File
	CAE-NEI	Neural Emotion Insertion	File

5.2 Audio Recording Preservation (CAE-ARP)

5.2.1 Functions

Preservation of audio assets recorded on analogue media is an important activity for a variety of application domains, in particular cultural heritage. Preservation goes beyond mere A/D conversion. For instance, the magnetic tape of an open reel may hold important information: it can carry annotations (by the composer or by the technicians) or it can include multiple splices and/or display several types of Irregularities (e.g., corruptions of the carrier, tape of different colour or chemical composition). This information shall be preserved for a correct playback. Nevertheless, some errors can occur during the digitisation as well as the digitisation could be partial because of the corruption of the carrier. These errors shall be restored to make the content listenable. The ARP Use Case (see 5.1.5) concerns the creation of a digital copy of the digitized audio of open reel

magnetic tapes for long-term preservation and of an access copy (restored, if necessary) for correct playback of the digitized recording.

In this Audio Recording Preservation Use Case, two files are fed into a preservation system:

1. A Preservation Audio File obtained by digitising the analogue tape audio recording composed of music, soundscape or speech read from a magnetic tape.
2. A Preservation Audio-Visual File produced by a camera pointed to the playback head of the magnetic tape recorder.

The following is not required:

1. Alignment of the start and end times of the two files. However, the maximum tolerated misalignment is 10s.
2. Presence of signal at the start and the end of the two files.
3. Alignment of the Useful Signal on both files.
4. The same time base for both files. However, the time difference between the same samples in two files shall not be more than 30ms for a 1-hour audio tape.

The output of the restoration process is composed by:

1. Preservation Master Files.
2. Access Copy Files.

5.2.2 Reference Model

Figure 3 depicts the Audio Recording Preservation Reference Model.

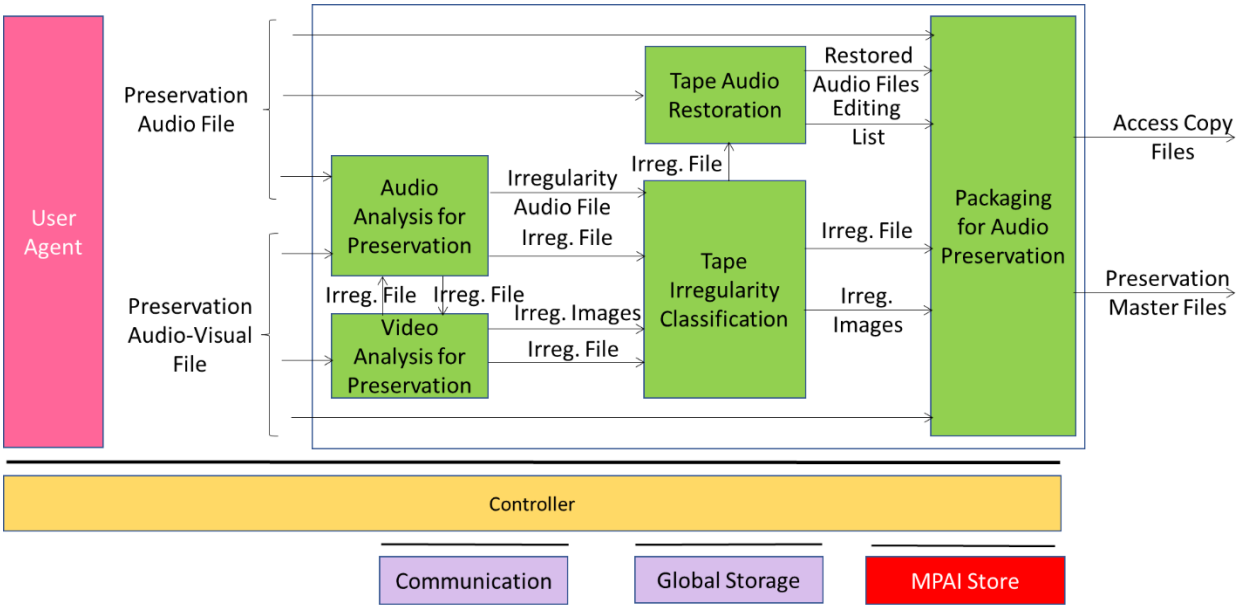

Figure 3 – Audio Recording Preservation Reference Model

The sequence of operations of the Audio Recording Preservation unfolds as follows:

1. The analogue audio signal from the open-reel tape recorder is digitised as Preservation Audio File.
2. Preservation Audio-Visual File is the combination of:
    a. The video camera pointed at the playback head of the open-reel tape recorder.
    b. The analogue audio signal digitised with the same video clock.
3. Audio Analysis for Preservation:
    a. Detects Irregularities.
    b. Assigns IDs to them that are unique to the analysed open-reel tape.
    c. Receives an Irregularity File from the Video Analysis for Recording
    d. Extracts the Audio Files corresponding to each Irregularity received or detected.

    e. Sends the Audio Files and the Irregularity File related to all Irregularities to the Tape Irregularity Classification.
4. Video Analysis for Preservation:
    a. Detects Irregularities.
    b. Assigns IDs to them that are unique to the analysed open-reel tape.
    c. Receives an Irregularity File from the Audio Analysis for Recording and the offset between Preservation Audio File and the Preservation Audio-Visual File.
    d. Extracts the Irregularity Images corresponding to each Irregularity received or detected.
    e. Sends the Irregularity Images and the Irregularity File related to all Irregularities to the Tape Irregularity Classification.
5. Tape Irregularity Classification:
    a. Receives an Irregularity File with the corresponding Images and Audio Files.
    b. Classifies and selects the ones considered relevant.
    c. If the Irregularity was detected by the Video Analysis for Recording, the selected Irregularity File and the corresponding Irregularity Images are sent to the Packaging for Audio Recording.
6. The Tape Audio Restoration uses the Irregularity File to identify and restore portions of the Preservation Audio File.
7. The Packaging for Audio Preservation collects the Preservation Audio File, Restored Audio Files, the Editing List, the Irregularity File and corresponding Irregularity Images if detected by the Video Analyser, and the Preservation Audio-Visual File and then it produces the Preservation Master Files and Access Copy Files.

5.2.3 I/O data of AI Workflow

Table 6 gives the input and output data of Audio Recording Preservation.

Table 6 – I/O data of Audio Recording Preservation

Input	Comments
Preservation Audio File	A Preservation Audio File obtained by digitising the analogue tape audio recording composed of music, soundscape or speech read from a magnetic tape.
Preservation Audio-Visual File	A Preservation Audio-Visual File produced by a camera pointed to the playback head of the magnetic tape recorder.
Output data	**Comments**
Preservation Master Files	Set of files providing the information stored in an audio tape recording without any restoration. As soon as the original analogue recordings is no more accessible, it becomes the new item for long-term preservation.
Access Copy Files	Set of Audio Files derived from the Preservation Audio File, where potential speed, equalisation or reading backwards errors that occurred in the digitisation process have been corrected.

5.2.4 Functions of AI Modules

The AIMs required by this Use Case are described in *Table 7*.

Table 7 – Functions of AI Modules of Audio Recording Preservation

AIM	Function
Audio Analysis for Preservation	1. At the start, it calculates the offset between Preservation Audio and the Audio of the Preservation Audio-Visual File.

	2. Sends Audio Irregularity File to and receives Video Irregularity Files from Video Analysis for Preservation. 3. Extracts the Audio Files corresponding to the Irregularities identified in both Irregularity Files. 4. Sends the Irregularity merged from the Audio and Video Irregularity Files to Tape Irregularity Classification with the corresponding Audio Files.
Video Analysis for Preservation	1. Detects and enters the Video Irregularities of the Preservation Audio-Visual File in the Video Irregularity File. 2. Sends Video Irregularity File to and receives Audio Irregularity Files from Audio Analysis for Preservation. 3. Extracts the Images corresponding to the Irregularities of both Irregularity Files. 4. Sends the Irregularity merged from the Audio and Video Irregularity Files to Tape Irregularity Classification with the corresponding Video Files.
Tape Irregularities Classification	1. Receives Irregularity File (Audio) and Audio Files from Audio Analysis for Preservation. 2. Receives Irregularity File (Video) and Irregularity Images from Video Analysis for Preservation. 3. Classifies and selects the relevant Irregularities of the Preservation Audio-Visual File and Preservation Audio File. 4. Sends the Irregularity File related to the selected Irregularities to Tape Audio Restoration. 5. Sends the Irregularity Files related to the selected Irregularities and the corresponding Irregularity Images to Packaging for Audio Recording.
Tape Audio Restoration	1. Detects and corrects speed, equalisation and reading backwards errors in Preservation Audio File. 2. Sends Restored Audio Files and Editing List to Packaging for Audio Preservation
Packaging for Audio Recording	Produces Preservation Master Files and Access Copy Files.

5.2.5 I/O Data of AI Modules

Table 8 – CAE-ARP AIMs and their data

AIM	Input Data	Output Data
Audio Analysis for Preservation	Preservation Audio File Preservation Audio-Visual File Irregularity File	Audio Files Audio Irregularity File
Video Analysis for Preservation	Preservation Audio-Visual File Audio Irregularity File	Video Irregularity File Irregularity Images
Tape Irregularities Classification	Irregularity Audio Files Audio Irregularity File Irregularity Images Video Irregularity File	Irregularity File Irregularity Images
Tape Audio Restoration	Irregularity File Preservation Audio File	Editing List Restored Audio Files

Packaging for Audio Preservation	Preservation Audio File Restored Audio Files Editing List Irregularity File Irregularity Images Preservation Audio-Visual File	Access Copy Files Preservation Master Files

5.2.6 AIW, AIMs, and JSON Metadata

Table 9 - Acronyms and URs of JSON Metadata

AIW	AIMs	Name	JSON
CAE-ARP		Audio Recording Preservation	File
	CAE-AAP	Audio Analysis for Preservation	File
	CAE-VAP	Video Analysis for Preservation	File
	CAE-TIC	Tape Irregularity Classification	File
	CAE-TAR	Tape Audio Restoration	File
	CAE-PAP	Packaging for Audio Preservation	File

5.3 Speech Restoration System (CAE-SRS)

5.3.1 Functions

The goal of this use case is to restore a Damaged Segment of an Audio Segment containing only speech from a single speaker. The damage may affect the entire segment, or only part of it.
Restoration will not involve filtering or signal processing. Instead, *replacements* for the damaged vocal elements will be *synthesised* using a speech model. The latter is a component or set of components, normally including one or more neural networks, which accepts text and possibly other specifications, and delivers audible speech in a specified format – here, the speech of the required replacement or replacements. If the damage affects the entire segment, an entirely new segment is synthesized; if only parts are affected, corresponding segments will be synthesized individually to enable later integration into the undamaged parts of the Damaged Segment, with reference to appropriate Time Labels.
The speech segments necessary for the creation of the speech model can be flexibly resourced from undamaged parts of the input segment or from other recording sources that are consistent with the original segment's acoustic environment.
Restoration is carried out by synthesising replacements for the damaged vocal elements as follows:

1. If the damage affects the entire segment, restoration will be carried out by synthesizing an entirely new segment version.
2. If the damage affects only parts of the segment, then those parts will be synthesized individually, and then integrated into the undamaged parts of the Damaged Segment in a final step, as indicated by appropriate Time Labels.

The Speech Segments for Modelling – Audio Segments necessary for the creation of the Neural Network Speech Model – may be obtained from any undamaged parts of the input speech segment; however, other Audio Segments consistent with the original segment's sound environment can also be used.

5.3.2 Reference Model

The Reference Model of the Speech Restoration System is given by *Figure 4*

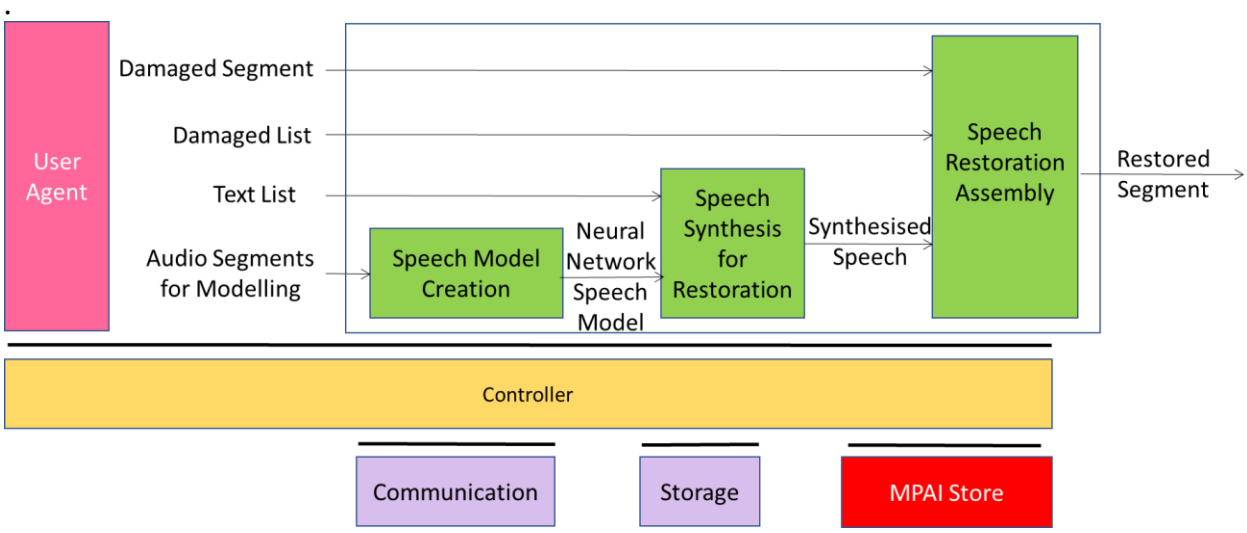

Figure 4 - Speech Restoration System (SRS) Reference Model

In the SRS use case, the entire Damaged Segment can be replaced by a synthesised segment, or parts within it can be synthesized to enable integration of the replaced segments.
The sequence of events in this Use Case is as follows:

1. Speech Model Creation receives Audio Segments for Modelling, a set of recordings composing a corpus that will be used to train a Neural Network Speech Model in Speech Model Creation.
2. That Neural Network Speech Model is passed to the Speech Synthesis for Restoration AIM, which also receives a Text List as input. Each element of Text List is a string specifying the text of a damaged section of Damaged Segment (or of Damaged Segment as a whole). Speech Synthesis for Restoration produces synthetic replacements for each damaged section (or for Damaged Segment as a whole) and passes the replacement(s) to Speech Restoration Assembly.
3. Speech Restoration Assembly receives as input the entire Damaged Segment, plus Damaged List, a list indicating the locations of any damaged sections within Damaged Segment. The list will be null if Damaged Segment in its entirety was replaced.
4. Speech Restoration Assembly produces as output Restored Segment, in which any repaired sections have been replaced by synthetic sections, or in which the entire Damaged Segment has been replaced.

5.3.3 I/O Data of AI Workflow

Table 10 gives the input and output data of Speech Restoration System.

Table 10 – I/O data of Audio Recording Preservation

Input	Comments
Speech Segments for Modelling	See *Table 1*.
Text List	See *Table 1*.
Damaged List	See *Table 1*.
Damaged Segment	See *Table 1*.
Output	**Comments**
Restored Speech Segment	See *Table 1*.

5.3.4 Functions of AI Modules

The AIMs required by the Speech Restoration System Use Case are described in *Table 11*.

Table 11 - AI Modules of Speech Recording System

AIM	Function
Speech Model Creation	1. Receives in separate files the Audio Segments for Modelling, adequate for model creation. 2. Creates the current Neural Network Speech Model. 3. Sends that Neural Network Speech Model to the Speech Synthesis for Restoration.
Speech Synthesis for Restoration	1. Receives the current Neural Network Speech Model. 2. Receives Damaged List as a data structure: a. Containing one element if Damaged Segment is damaged throughout or b. Representing a list in which each element specifies via Time Labels the start and end of a damaged section within Damaged Segment. 3. Synthesizes each Damaged Section in Damaged List. 4. Sends the newly synthesised segments to the Speech Restoration Assembly as an ordered list.
Speech Restoration Assembly	1. Receives the Damaged Segment. 2. Receives the ordered list of synthetic segments. 3. Receives Damaged List Time Labels, indicating where the synthesized segments should be inserted in left-to-right order. In case Damaged Segment as a whole was damaged, the list contains one entry. 4. Assembles the final version of the Restored Segment.

5.3.5 I/O Data of AI Modules

Table 12 – CAE-SRS AIMs and their I/O Data

AIM	Input Data	Output Data
Speech Model Creation	Audio Segments for Modelling	Neural Network Speech Model
Speech Synthesis for Restoration	Text List Neural Network Speech Model	Synthesised Speech
Speech Restoration Assembly	Damaged Segments Damaged List	Restored Segment

5.3.6 AIW, AIMs and JSON Metadata

Table 13 – AIMs and JSON Metadata

AIW	AIMs	Names	JSON
CAE-SRS		Speech Restoration System	File
	CAE-SMC	Speech Model Creation	File
	CAE-SSR	Speech Synthesis for Restoration	File
	CAE-SRA	Speech Restoration Assembly	File

5.4 Enhanced Audioconference Experience (CAE-EAE)

5.4.1 Functions

The Enhanced Audioconference Experience Use Case addresses the situation where one or more speakers are active in a noisy meeting room and are trying to communicate with one or more interlocutors using speech over a network. In this situation, the user experience is very often far from satisfactory due to multiple competing speakers, non-ideal acoustical properties of the physical spaces that the speakers occupy and/or background noise. These can lead to a reduction in intelligibility of speech resulting in participants not fully understanding what their interlocutors are saying, in addition to creating a distraction and eventually leading to what is known as *audioconference fatigue*. When microphone arrays are used to capture the speakers, most of the described problems can be resolved by appropriate processing of the captured signals. The speech signals from multiple speakers can be separated from each other, the non-ideal acoustics of the space can be reduced, and any background noise can be substantially suppressed.

CAE-EAE is concerned with extracting from microphone array recordings the speech signals from individual speakers as well as reducing the background noise and the reverberation that reduce speech intelligibility. CAE-EAE also extracts the Spatial Attitudes of the speakers with respect to the position of the microphone array to facilitate the spatial representation of the speech signals at the receiver side if necessary. These Spatial Attitudes are represented in the Audio Scene Geometry format and packaged in a format that is amenable to further processing for efficient delivery and further processing. Data reduction of the extracted speech signals as well as their reconstruction/representation at the receiver side are outside the scope of this Use Case.

CAE-EAE aims to provide a complete solution to process speech signals recorded by microphone arrays to provide clear speech signals substantially free from background noise and acoustics-related artefacts to improve the auditory quality of audioconference experience. Thus, CAE-EAE improves auditory experience in an audioconference, thereby reducing the effects of audioconference fatigue.

5.4.2 Reference Model

Figure 5 shows the Reference Model for the CAE-EAE.

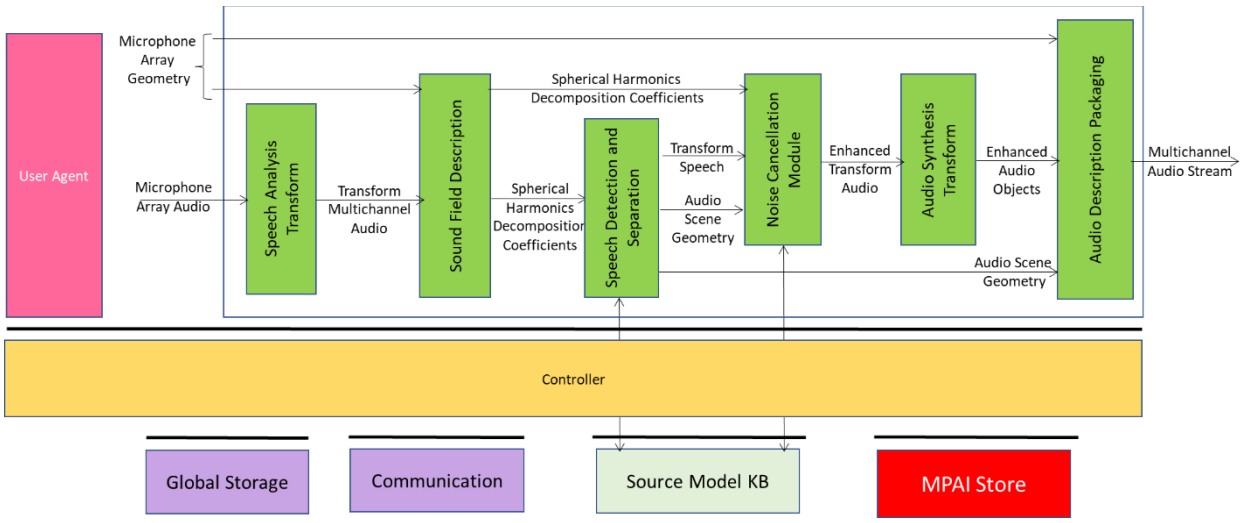

Figure 5 - Enhanced Audioconference Experience Reference Model

5.4.3 I/O data of AI Workflow

Table 14 shows the input and output data for the Enhanced Audioconference Experience workflow.

Table 14 – I/O data of Enhanced Audioconference Experience

Inputs	Comments
Microphone Array Geometry	A Data Type representing the position of each microphone comprising a Microphone Array and specific characteristics such as microphone type, look directions, and the array type.
Microphone Array Audio	A Data Type whose structure contains between 4 and 256 time-aligned interleaved Audio Channels organised in blocks.
Outputs	**Comments**
Multichannel Audio Stream	Interleaved Multichannel Audio packaged with Time Code as specified in Multichannel Audio-Stream.

The Enhanced Audio Experience AIW:

1. Receives:

1.1. Microphone Array Geometry which describes the number, positioning, and configuration of the microphone(s). Using this information, the system can detect the relative directions of the active speakers according to the microphone array and separate relevant audioconference speech sources from each other and from other spurious sounds. Since audio conferencing is a real-time application scenario, the use case operates on Audio Blocks.

1.2. Microphone Array Audio which is input to EAE as short Multichannel Audio Blocks comprising real valued time domain audio samples where the number of audio samples in each Audio Block is the same for all the microphones.

2. Produces Multichannel Audio Stream.

5.4.4 Functions of AI Modules

The AIMs required by the Enhanced Audioconference Experience are given in *Table 15*.

Table 15 - AIMs of Enhanced Audioconference Experience

AIM	Function
Audio Analysis Transform	Represents the input Multichannel Audio in a new form amenable to further processing by the subsequent AIMs in the architecture.
Sound Field Description	Produces Spherical Harmonic Decomposition Coefficients of the Transformed Multichannel Audio.
Speech Detection and Separation	Separates speech and non-speech signals in the Spherical Harmonic Decomposition producing Transform Speech and Audio Scene Geometry.
Noise Cancellation Module	Removes noise and/or suppresses reverberation in the Transform Speech producing Enhanced Transform Audio.
Audio Synthesis Transform	Effects inverse transform of Enhanced Transform Audio producing Enhanced Audio Objects ready for packaging.
Audio Description Packaging	Multiplexes Enhanced Audio Objects and the Audio Scene Geometry.

The EAE use case receives Microphone Array Audio and Microphone Array Geometry which describes the number, positioning, and configuration of the microphone(s). Using this information,

the system can detect the relative directions of the active speakers according to the microphone array and separate relevant audioconference speech sources from each other and from other spurious sounds. Since audio conferencing is a real-time application scenario, the use case operates on Audio Blocks.

The Multichannel Audio is input to EAE as short Multichannel Audio Blocks comprising real valued time domain audio samples where the number of audio samples in each audio block is the same for all the microphones.

The sequence of operations of the EAE use case is the following:

1. **Audio Analysis Transform** transforms the Microphone Array Audio into frequency bands via a Fast Fourier Transform (FFT). The following operations are carried out in discrete frequency bands. When such a configuration is used a 50% overlap between subsequent audio blocks needs to be employed. The output is a data structure comprising complex valued audio samples in the frequency domain.
2. **Sound Field Description** converts the output from the Speech Analysis Transform AIM into the spherical frequency domain [27]. If the microphone array used in capturing the scene is a spherical microphone array, Spherical Fourier Transform (SFT) can be used to obtain the Spherical Harmonic Decomposition (SHD) coefficients that represent the captured sound field in the spatial frequency domain. For other types of arrays, more elaborate processing might be necessary. The output of this AIM is (M × (N+1)2) complex valued data frame comprising the SHD coefficients up to an order which depends on the number of individual microphones in the array.
3. **Speech Detection and Separation** receives the SHD coefficients of the sound field to detect directions of active sound sources and to separate them. Each separated source can either be a speech or a non-speech signal. Speech detection is carried out on an Audio Block basis by using on each separated source an appropriate voice activity detector (VAD) that is a part of this AIM. This AIM will output speech as an (M × S) Audio Block comprising transform domain speech signals and block-by-block Audio Scene Geometry comprising auxiliary information which contains a (M × 1) binary mask indicating the channels of the transform domain SHD coefficients that would be used by the Noise Cancellation AIM for denoising. Speech Detection and Separation AIM uses the **Source Model KB** which contains discrete-time and discrete-valued simple acoustic source models that are used in source separation. The format such acoustic source models is not standardised as it is part of the Speech Detection and Separation AIM.
4. **Noise Cancellation Module** .
    a. Receives Transform Audio from **Speech Detection and Separation** AIM and Audio Scene Geometry which includes attributes pertaining to the Audio Block being processed for denoising, and SHD coefficients.
    b. Uses **Source Model KB** to produce Enhanced Transform Audio as an (M × S) complex-valued data structure which will in the next stage be processed through **Audio Synthesis Transform** AIM to obtain Enhanced Audio Objects.
5. **Audio Synthesis Transform** receives Enhanced Transform Audio and outputs Enhanced Audio Objects (F × S) by applying the inverse of the analysis transform.
6. **Audio Description Packaging**:
    a. Receives Microphone Array Geometry, Enhanced Audio Objects and Audio Scene Geometry.
    b. Packages Sampling Rate and Sample Type from Microphone Array Geometry, Enhanced Audio Object, and the Audio Scene Geometry.
    c. Produces one interleaved stream which contains Multichannel Audio Streams.

5.4.5 I/O Data of AI Modules

Table 16 – CAE-EAE AIMs and their data

AIM	Input Data	Output Data
Audio Analysis Transform	Microphone Array Audio	Transform Multichannel Audio
Sound Field Description	Transform Multichannel Audio	Spherical Harmonic Decomposition Coefficients
Speech Detection and Separation	Spherical Harmonic Decomposition Coefficients	Transform Audio Audio Scene Geometry
Noise Cancellation Module	Spherical Harmonic Decomposition Coefficients Transform Audio Audio Scene Geometry	Enhanced Transform Audio
Audio Synthesis Transform	Enhanced Transform Audio	Enhanced Audio Objects
Audio Description Packaging	Enhanced Audio Objects Audio Scene Geometry	Multichannel Audio Stream

5.4.6 AIW, AIMs, and JSON Metadata

Table 17 – AIW, AIMs, and JSON Metadata

AIW	AIMs	Names	JSON
CAE-EAE		Enhanced Audioconference Experience	File
	CAE-AAT	Audio Analysis Transform	File
	CAE-SFD	Sound Field Description	File
	CAE-SDS	Speech Detection and Separation	File
	CAE-NCM	Noise Cancellation Module	File
	CAE-AST	Audio Synthesis Transform	File
	CAE-ADP	Audio Description Packaging	File

5.5 Human-Connected Autonomous Vehicle (CAV) Interaction

Note: this Use Case is not specified by MPAI-CAE but by [4]. The MMC-HCI Use Case initial elements – Functions, Reference Model, and I/O Data – are reported here because the MMC-HCI AIW uses the Audio Scene Description Composite AIM.

5.5.1 Functions of Use Case

A group of humans approach a Connected Autonomous Vehicle (CAV) in a noisy environment. At least one human should be recognised by his/her voice. All humans may hold a conversation with the CAV through the Human-CAV Interaction Subsystem (HCI), e.g., to request to be taken somewhere.

After they are let into the cabin, the humans sit on the seat. During the travel, they converse between themselves and potentially with the CAV. The HCI function separates the different speech sources to be able to participate in the conversation, e.g., to answer specific questions. The cabin, too, is assumed to be noisy.

This use case is part of the Connected Autonomous Vehicle (CAV) – Architecture Technical Specification [3]. A CAV is a system able to execute a command to move itself based on 1) analysis and interpretation of the data sensed by a range of onboard sensors exploring the

environment and 2) information transmitted by other sources in range, e.g., other CAVs, traffic lights and roadside units.
Figure 6 depicts the four subsystems of a CAV.

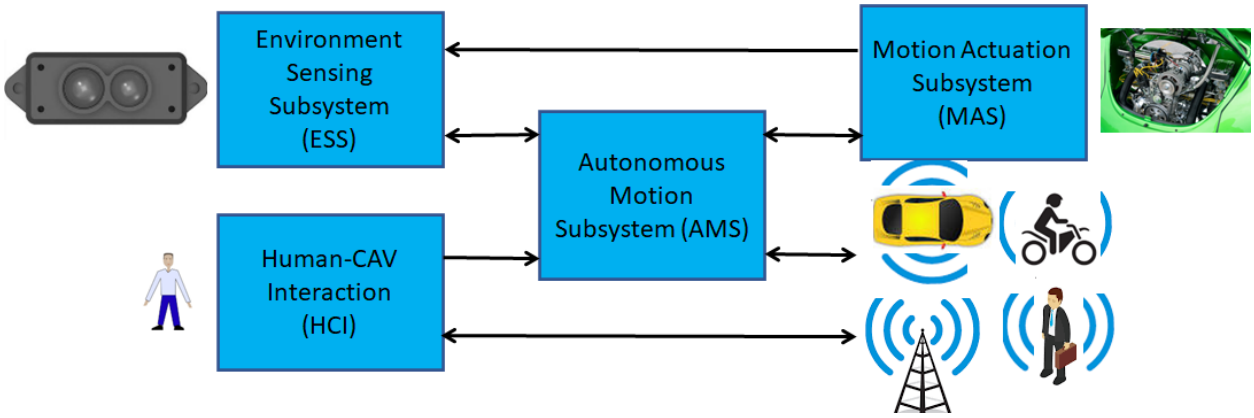

Figure 6 – The Connected Autonomous Vehicle Reference Model

1. **Human-CAV interaction (HCI)** recognises the human owner or renter, responds to humans' commands and queries, converses with humans during the travel and may activate other Subsystems in response to humans' requests. The data exchanged between the HCI, and the Autonomous Motion Subsystem (AMS) is depicted in Figure 7 but the requirements of the format of the data exchanged between HCI and AMS are not part of this document.
2. **Environment Sensing Subsystem (ESS)** acquires information from the Environment via a variety of sensors and produces a representation of the Environment (Basic Environment Representation), i.e., its best understanding of the Environment based on the sensed data.
3. **Autonomous Motion Subsystem (AMS)** computes the route to destination, uses different sources of information – CAV sensors, other CAVs and transmitting units – to produce a Full Environment Representation and issues commands to drive the CAV to the intended destination.
4. **Motion Actuation Subsystem (MAS)** provides non-electromagnetic and non-acoustical environment information¸ and receives and actuates motion commands in the physical world.

The CAV in Human-CAV Interaction is represented by an avatar with the following perceptible features:
1. Visual: head, face, and shoulders.
2. Audio: speech.

Both visual and audio features convey as much as possible the Personal Status that would be displayed by a human driver in similar conditions.
In the following the Reference Model and the I/O Data of the Human-CAV Interaction Subsystem will be reported.

5.5.2 Reference Model

Figure 7 represents the Human-CAV Interaction (HCI) Reference Model.

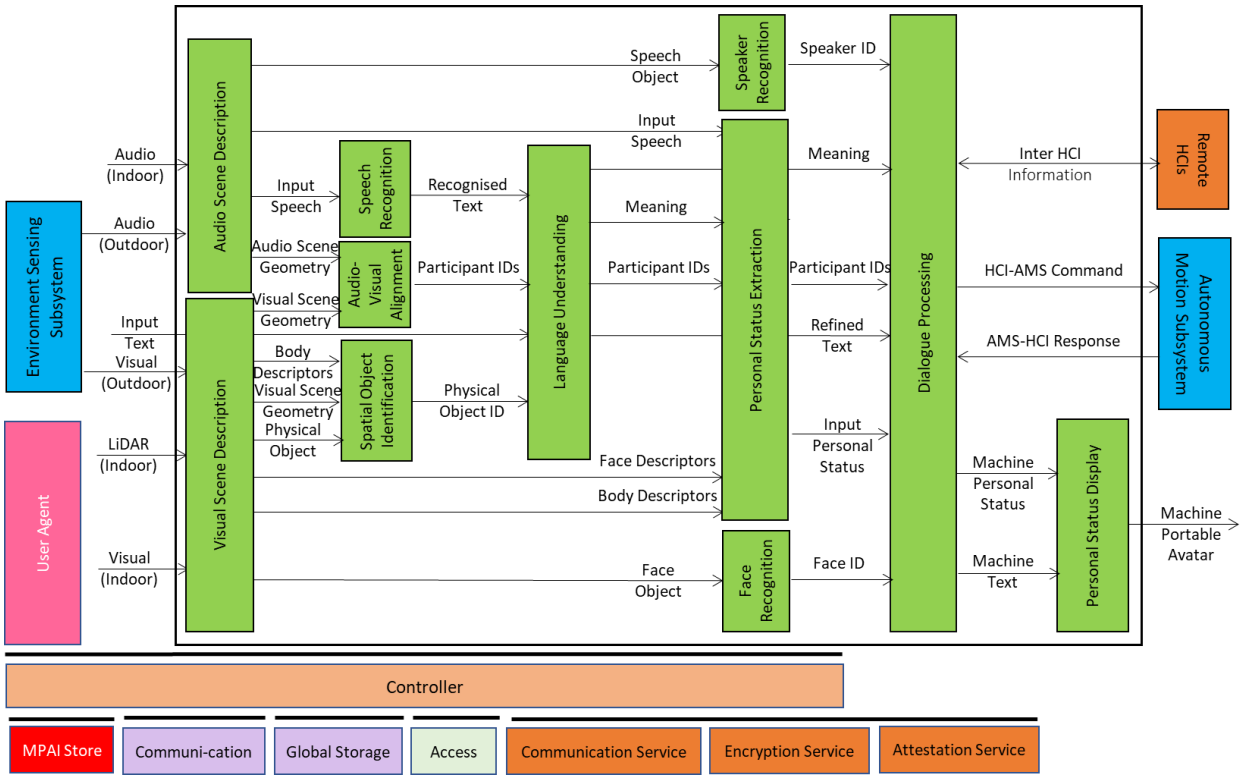

Figure 7 – Human-CAV Interaction Reference Model

The operation of HCI involves the following functions:

1. A group of humans approaches the CAV <u>outside the CAV</u>:
    a. The Audio Scene Description AIM creates the Audio Scene Description in the form of Audio (Speech) Objects corresponding to each speaking human in the Environment (close to the CAV).
    b. The Visual Scene Description creates the Visual Scene Descriptors in the form of Human Objects with the possibility of extracting the Head and Face corresponding to each human in the Environment (close to the CAV).
    c. The Speaker Recognition and Face Recognition AIMs authenticate the humans that the HCI is interacting with using Speech and Face Descriptors.
    d. The Speech Recognition AIM recognises the speech of each human.
    e. The Personal Status Extraction AIM extracts the Personal Status of the humans.
    f. The Language Understanding AIM extracts Meaning and produces the refined Text (Language Understanding).
    g. The Dialogue Processing AIM validates the human Identities, produces the response and displays the HCI Personal Status, and issues commands to the Autonomous Motion Subsystem.
2. A group of humans sits in the seats <u>inside the CAV</u>:
    a. The Audio Scene Description AIM creates the Audio Scene Descriptions in the form of Audio (Speech) Objects corresponding to each speaking human in the cabin.
    b. The Visual Scene Description creates the Visual Scene Descriptors in the form of Human Objects with the possibility of extracting the Head and Face corresponding to each human in the cabin.
    c. The Speaker Recognition and Face Recognition AIMs identify the humans the HCI is interacting with using Speech and Face Descriptors.
    d. The Speech Recognition AIM recognises the speech of each human.
    e. The Personal Status Extraction AIM extracts the Personal Status of the humans.

f. The Language Understanding AIM extracts Meaning and produces the refined Text (Language Understanding).
g. The Dialogue Processing AIM recognises the human Identities, produces the response and displays the HCI Personal Status, and issues commands to the Autonomous Motion Subsystem.

3. The HCI interacts with the humans in the cabin in several ways:
    a. By responding to commands/queries from one or more humans at the same time, e.g.:
        i. Commands to go to a waypoint, park at a place, etc.
        ii. Commands with an effect in the cabin, e.g., turn off air conditioning, turn on the radio, call a person, open window or door, search for information etc.
        Note: For completeness, Figure 7 includes the conversion of human commands and responses from the CAV. However, this document does not address the format in which the HCI interacts with the Autonomous Motion Subsystem.
    b. By conversing with and responding to questions from one or more humans at the same time about travel-related issues (in-depth domain-specific conversation), e.g.:
        i. Humans request information, e.g., time to destination, route conditions, weather at destination, etc.
        ii. CAV offers alternatives to humans, e.g., long but safe way, short but likely to have interruptions.
        iii. Humans ask questions about objects in the cabin.
    c. By following the conversation on travel matters held by humans in the cabin if 1) the passengers allow the HCI to do so, and 2) the processing is carried out inside the CAV.

5.5.3 I/O Data of HCI AI Workflow

Table 18 gives the input/output data of the Human-CAV Interaction Subsystem.

Table 18 - I/O data of Human-CAV Interaction

Input data	From	Description
Input Audio (Outdoor)	Environment Sensing Subsystem	User authentication User command User conversation
Input Audio (Indoor)	Cabin Passengers	User's social life Commands/interaction with HCI
Input Visual (Outdoor)	Environment Sensing Subsystem	Commands/interaction with HCI
Input Visual (Indoor)	Cabin Passengers	User's social life Commands/interaction with HCI
AMS-HCI Message	Autonomous Motion Subsystem	Includes response to HCI-AMS Message
Inter HCI Information	Remote HCI	HCI-to-HCI information
Output data	**To**	**Comments**
Inter HCI Information	Remote HCI	HCI-to-HCI information
HCI-AMS Message	Autonomous Motion Subsystem	HCI-to-AMS Message
Machine Portable Avatar	Cabin Passengers	HCI's avatar.

5.5.4 AIW, AIMs, and JSON Metadata

Table 19 – AIW, AIM, and JSON Metadata

AIW	AIM		Name	JSON
MMC-HCI			Human-CAV Interaction	File
	CAE-ASD		Audio Scene Description	File
		CAE-AAT	Audio Analysis Transform	File
		CAE-ASL	Audio Source Localisation	File
		CAE-ASE	Audio Separation and Enhancement	File
		CAE-AST	Audio Synthesis Transform	File
		CAE-AMX	Audio Descriptor Multiplexing	File
		OSD-VSD	Visual Scene Description	File
	MMC-ASR		Automatic Speech Recognition	File
	OSD-AVA		Audio-Visual Alignment	File
	OSD-VOI		Visual Object Identification	File
		OSD-VDI	Visual Direction Identification	File
		OSD-VOE	Visual Object Extraction	File
		OSD-VII	Visual Instance Identification	File
	MMC-NLU		Natural Language Understanding	File
	MMC-SIR		Speaker Identity Recognition	File
	MMC-PSE		Personal Status Extraction	File
		MMC-ITD	Input Text Description	File
		MMC-ISD	Input Speech Description	File
		PAF-IFD	Input Face Description	File
		PAF-IBD	Input Body Description	File
		MMC-PTI	PS-Text Interpretation	File
		MMC-PSI	PS-Speech Interpretation	File
		PAF-PFI	PS-Face Interpretation	File
		PAF-PGI	PS-Gesture Interpretation	File
		MMC-PMX	Personal Status Multiplexing	File
	MMC-EDP		Entity Dialogue Processing	File
	PAF-FIR		Face Identity Recognition	File
	PAF-PSD		Personal Status Display	File
		MMC-TTS	Text-to-Speech	File
		PAF-IFD	Input Face Description	File
		PAF-IBD	Input Body Description	File
		PAF-PMX	Portable Avatar Multiplexing	File

6 Audio Scene Description Composite AIM

6.1 Functions of Audio Scene Description

Audio Scene Description (CAE-ASD):

1. Receives the Audio Scene composed of:
1.1. Microphone Array Geometry.
1.2. Multichannel Audio, i.e., the output of the Microphone Array.
2. Separates Audio Objects in the scene.
3. Produces Audio Scene Descriptors containing:

6.2 Reference Model of Audio Scene Description

Figure 8 depicts the Reference Model of CAE-ASD.

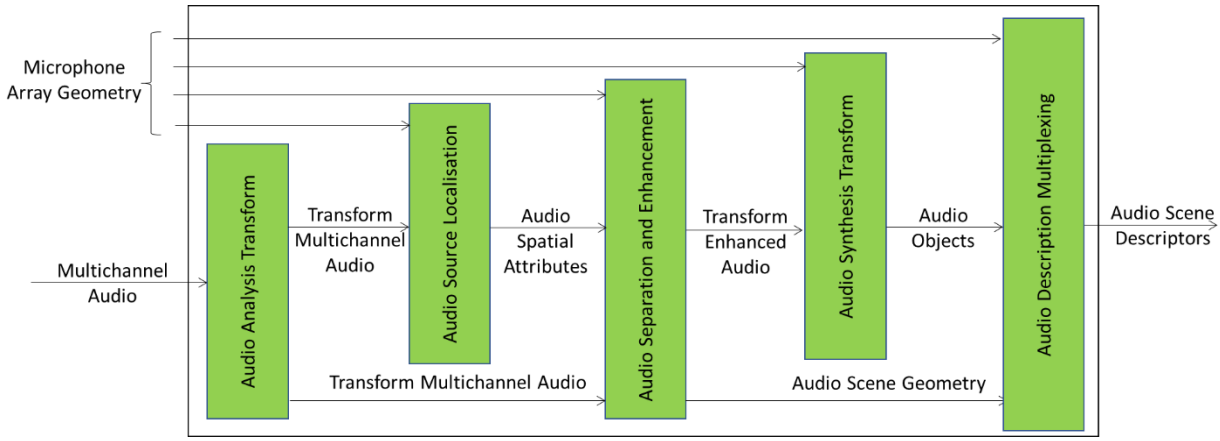

Figure 8 - Reference Model of Audio Scene Description Composite AIM

6.3 I/O Data of Audio Scene Description

Table 20 gives the Input/Output data of Audio Scene Description.

Table 20 – I/O data of Audio Scene Description

Input data	Comment
Microphone Array Geometry	The description of the spatial microphone arrangement.
Multichannel Audio	The Audio output of the Microphone Array.
Output data	**Comments**
Scene Descriptors	The Descriptors of the Audio Scene.

6.4 Functions of AI Modules of Audio Scene Description

Table 21 gives the list of the AIMs with their functions. Note that Audio Analysis Transform and Audio Synthesis Transform are the same AIMs of the Enhanced Audioconference Experience Use Case.

Table 21 – AI Modules of Audio Scene Description

AIM	Function
Audio Analysis Transform	Transforms the Microphone Array Audio into frequency bands via a Fast Fourier Transform (FFT). The following operations are carried out in discrete frequency bands. When such a configuration is used, a 50% overlap between subsequent audio blocks needs to be employed. The output is a data structure comprising complex valued audio samples in the frequency domain.
Audio Source Localisation	Detects the Audio Objects in the Audio Scene with their Spatial Attitudes. It receives Transform Multichannel Audio, and Microphone Array Geometry. Its output is Spatial Attitudes of the Audio Objects.
Audio Separation and Enhancement	Separates the Audio Objects by using their Spatial Attitudes. It receives Transform Multichannel Audio, Audio Object Spatial

	Attributes and Microphone Array Geometry. Its outputs are Transform Enhanced Audio and Audio Scene Geometry.
Audio Synthesis Transform	Transforms the Transform Enhanced Source into time domain via an Inverse Fast Fourier Transform (IFFT). It receives Transform Enhanced Audio and outputs Enhanced Audio by applying the inverse of the Audio Analysis Transform.
Audio Description Multiplexing	Receives Enhanced Audio, Microphone Array Geometry, and Audio Scene Geometry. It multiplexes the Enhanced Audio and the Audio Scene Geometry and then produces Audio Scene Descriptors.

6.5 I/O Data of AI Modules of Audio Scene Description

Table 22 – Audio Scene Description and their data

AIM	Input Data	Output Data
Audio Analysis Transform	Multichannel Audio	Transform Multichannel Audio
Audio Source Localisation	Transform Multichannel Audio Microphone Array Geometry	Audio Spatial Attitudes
Audio Separation and Enhancement	Audio Spatial Attitudes Transform Multichannel Audio Microphone Array Geometry	Transform Enhanced Audio Audio Scene Geometry
Audio Synthesis Transform	Transform Enhanced Audio	Enhanced Audio
Audio Description Multiplexing	Enhanced Audio Audio Scene Geometry Microphone Array Geometry	Audio Scene Descriptors

6.6 Specification of Audio Scene Description AIW, AIMs, and JSON Metadata

Table 23 – AIM and JSON Metadata

AIW	AIMs	Names	JSON
CAE-ASD		Audio Scene Description	File
	CAE-AAT	Audio Analysis Transform	File
	CAE-ASL	Audio Source Localisation	File
	CAE-ASE	Audio Separation and Enhancement	File
	CAE-AST	Audio Synthesis Transform	File
	CAE-ADM	Audio Description Multiplexing	File

7 Data Types

Table 24 lists all data formats specified in this Technical Specification.

Table 24 – Data Types

Data Type Name	Subsection	Use Case
Access Copy Files	7.1	ARP

Audio Block	7.2	EAE
Audio File	7.3	ARP
Audio Object	7.4	ASD
Audio Scene Descriptors	7.5	ASD
Audio Scene Geometry	7.6	EAE
Audio Segment	7.7	SRS
Damaged List	7.8	SRS
Editing List	7.9	ARP
Emotion	7.10	EES
Emotionless Speech	7.11	EES
Enhanced Audio	7.12	ASD
Enhanced Transform Audio	7.13	EAE, ASD
Irregularity File	7.14	ARP
Irregularity Image	7.15	ARP
Microphone Array Geometry	7.17	EAE, ASD
Mode Selection	7.18	EES
Multichannel Audio	7.19	ASD
Multichannel Audio Stream	7.20	EAE
Neural Network Speech Model	7.21	SRS
Preservation Audio File	7.22	ARP
Preservation Audio-Visual File	7.23	ARP
Preservation Master Files	7.24	ARP
Speech Features	7.25	EES
Spherical Harmonics Decomposition	7.26	EAE
Transform Audio	7.27	EAE
Transform Multichannel Audio	7.28	EAE, ASD
Video	7.29	ARP

7.1 Access Copy Files

The following set of files:

1. The Restored Audio Files.
2. Editing List.
3. The set of Irregularity Images in a .zip file [15].
4. The Irregularity File.

7.2 Audio Block

A set of consecutive samples without time code.

7.3 Audio File

A wave file conforming to WAV RF64 file format [19].

7.4 Audio Object

7.4.1 Definition

Audio Object is a Data Type digitally representing either:

1. An object in the real world that a human can hear, or
2. A synthetically generated Object that a human can hear when rendered.

The Format of an Audio Object is signalled by FormatID.

7.4.2 Syntax

```
{
   "$schema":"http://json-schema.org/draft-07/schema#",
   "title":"AudioObject",
   "type":"object",
   "properties":{
      "Header":{
         "type":"object",
         "properties":{
            "Standard":{
               "type":"string"
            },
            "Version":{
               "type":"integer"
            },
            "Subversion":{
               "type":"integer"
            }
         }
      },
      "AOBID":{
         "type":"string"
      },
      "AudioObjectsData":{
         "type":"object",
         "properties":{
            "SamplingRate":{
               "type":"number"
            },
            "SamplingType":{
               "type":"number"
            },
            "AudioObject":{
               "type":"object",
               "properties":{
                  "FormatID":{
                     "type":"integer"
                  },
                  "ObjectLength":{
                     "type":"integer"
                  },
                  "DataInObject":{
                     "$ref":"https://schemas.mpai.community/CAE/V2.1/data/AudioObject.json"
                  }
               }
            }
         }
      }
   }
}
```

7.4.3 Semantics

Table 25 – Audio Object Semantics

Label	Size	Description
HEADER	9 Bytes	
• Standard	7 Bytes	The string CAE-ASD
• Version	1 Byte	Major version
• Subversion	1 Byte	Minor version
AOBID	16 Bytes	UUID Identifier of the Audio Object.
AudioObjectData	N1 Bytes	Data associated to each Audio Object.
• SamplingRate	0-3 bits	0:8, 1:16, 2: 22.05, 3:24, 4:32, 5:44.1, 6:48, 7: 96, 8: 192 (all kHz)

• SampleType	4-6 bits	(aka, sample precision) 0:8, 1:16, 2:24, 3:32, 4:64 (bits/sample)
• Reserved	7 bit	
• AudioObject	N2 Bytes	
○ FormatID	1 Byte	Audio Object Format Identifier
○ ObjectLength	4 Bytes	Number of Bytes in Audio Object
○ DataInObject	N3 Bytes	Data of Audio Object

7.5 Audio Scene Descriptors

7.5.1 Definition

A Data Type that includes the arrangement and the Objects of an Audio Scene.

7.5.2 Syntax

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Audio Scene Descriptors",
  "type": "object",
  "properties": {
    "Header": {
      "type": "object",
      "properties": {
        "Standard": {
          "type": "string"
        },
        "Version": {
          "type": "integer"
        },
        "Subversion": {
          "type": "integer"
        }
      }
    },
    "ASDID": {
      "type": "string"
    },
    "Time": {
      "type": "object",
      "properties": {
        "TimeType": {
          "type": "boolean"
        },
        "StartTime": {
          "type": "number"
        },
        "EndTime": {
          "type": "number"
        }
      }
    },
    "AudioObjectCount": {
      "type": "integer"
    },
    "AudioObjectsData": {
      "type": "object",
      "properties": {
        "AudioObjectID": {
          "type": "string"
        },
        "SamplingRate": {
          "type": "number"
        },
        "SamplingType": {
          "type": "number"
        },
        "SpatialAttitude": {
          "$ref": "https://schemas.mpai.community/OSD/V1.0/data/SpatialAttitude.json"
```

```
        },
        "AudioObject": {
          "type": "object",
          "properties": {
            "FormatID": {
              "type": "integer"
            },
            "ObjectLength": {
              "type": "integer"
            },
            "DataInObject": {
              "$ref": "https://schemas.mpai.community/CAE/V2.1/data/AudioObject.json"
            }
          }
        }
      }
    }
  }
}
```

7.5.3 Semantics

Table 26 provides the semantics of Audio Scene Descriptors.

Table 26 – Audio Scene Descriptors Semantics

Label	Size	Description
HEADER	9 Bytes	
• Standard	7 Bytes	The string CAE-ASD
• Version	1 Byte	Major version
• Subversion	1 Byte	Minor version
ASDID	16 Bytes	UUID Identifier of Audio Scene Descriptors set.
Time	17 Bytes	Collects various data expressed with bits
• TimeType	0 bit	0=Relative: time starts at 0000/00/00T00:00 1=Absolute: time starts at 1970/01/01T00:00.
• Reserved	1-7 bits	reserved
• StartTime	8 Bytes	Start of current Audio Scene Descriptors (in µs).
• EndTime	8 Bytes	End of current Audio Scene Descriptors (in µs).
AudioObjectCount	1 Byte	Number of Audio Objects in the Audio Scene.
AudioObjectsData	N1 Bytes	Data associated to each Audio Object.
AudioObjectID	1 Byte	ID of a specific Audio Object in the Audio Scene.
• SamplingRate	0-3 bits	0:8, 1:16, 2: 22.05, 3:24, 4:32, 5:44.1, 6:48, 7: 96, 8: 192 (all kHz)
• SampleType	4-6 bits	(aka, sample precision) 0:8, 1:16, 2:24, 3:32, 4:64 (bits/sample)
• Reserved	7 bit	
• Spatial Attitude	N2 Bytes	Spatial Attitude of Audio Object.
• AudioObject	N3 Bytes	Set of Audio Object Data.
• FormatID	1 Byte	Format Identifier Audio Object.
○ ObjectLength	4 Bytes	Number of Bytes in Audio Object.
○ DataInObject	N4 Bytes	Data of Audio Object.
○		

7.6 Audio Scene Geometry

7.6.1 Definition

The digital representation of the spatial arrangement of the Audio Scene Objects.

7.6.2 Syntax

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Audio Scene Geometry",
  "type": "object",
  "properties": {
    "Header": {
      "type": "object",
      "properties": {
        "Standard": {
          "type": "string"
        },
        "Version": {
          "type": "integer"
        },
        "Subversion": {
          "type": "integer"
        }
      }
    },
    "ASGID": {
      "type": "string"
    },
    "Time": {
      "type": "object",
      "properties": {
        "TimeType": {
          "type": "boolean"
        },
        "StartTime": {
          "type": "number"
        },
        "EndTime": {
          "type": "number"
        }
      }
    },
    "AudioObjectCount": {
      "type": "integer"
    },
    "AudioObjectsData": {
      "type": "object",
      "properties": {
        "AudioObjectID": {
          "type": "string"
        },
        "SpatialAttitude": {
          "$ref": "https://schemas.mpai.community/OSD/V1.0/data/SpatialAttitude.json"
        }
      }
    }
  }
}
```

7.6.3 Semantics

Table 27 provides the semantics of the Audio Scene Geometry.

Table 27 – Audio Scene Geometry Semantics

Label	Size	Description
HEADER	9 Bytes	
• Standard	7 Bytes	The string CAE-ASD

• Version	1 Byte	Major version
• Subversion	1 Byte	Minor version
ASDID	16 Bytes	UUID Identifier of Audio Scene Descriptors set.
Time	17 Bytes	Collects various data expressed with bits
• TimeType	0 bit	0=Relative: time starts at 0000/00/00T00:00 1=Absolute: time starts at 1970/01/01T00:00.
• Reserved	1-7 bits	
• StartTime	8 Bytes	Start of current Audio Scene Descriptors (in µs).
• EndTime	8 Bytes	End of current Audio Scene Descriptors (in µs).
AudioObjectCount	1 Byte	Number of Audio Objects in the Audio Scene.
AudioObjectsData	N1 Bytes	Data associated to each Audio Object.
• AudioObjectID	1 Byte	ID of a specific Audio Object in the Audio Scene.
• Reserved	6-7 bits	
• Spatial Attitude	N2 Bytes	Spatial Attitude of Audio Object.

7.7 Audio Segment

An Audio Block with Time Labels.

7.8 Damaged List

7.8.1 Definition

A list of strings of Texts corresponding to the Damaged Segments (if any) requiring replacement with synthetic segment.

7.8.2 Syntax

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Damaged list",
  "type": "object",
  "properties": {
    "DamagedSections": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "SegmentStart": {
            "type": "string",
            "pattern": "[0-9]{2}:[0-5][0-9]:[0-5][0-9]\\.[0-9]{3}"
          },
          "SegmentEnd": {
            "type": "string",
            "pattern": "[0-9]{2}:[0-5][0-9]:[0-5][0-9]\\.[0-9]{3}"
          }
        }
      },
      "minItems": 1,
      "uniqueItems": true,
      "required": [
        "SegmentStart",
        "SegmentEnd"
      ]
    }
  },
  "required": [
    "DamagedSections"
  ]
}
```

7.8.3 Semantics

Name	*Definition*
`DamagedSections`	A JSON array containing metadata description of Audio Segments within the given Damaged Segments.
`SectionStart`	Time Label of the beginning of the DamagedSection. `(string)`
`SectionEnd`	Time Label of the of the end of the DamagedSection. `(string)`

7.9 Editing List

7.9.1 Definition

The description of corrections for the speed, equalisation, and reverse playback that have been made during the restoration process.

7.9.2 Syntax

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Editing List",
  "type": "object",
  "properties": {
    "OriginalSpeedStandard": {
      "enum": [
        0.9375,
        1.875,
        3.75,
        7.5,
        15,
        30
      ]
    },
    "OriginalEqualisationStandard": {
      "enum": [
        "IEC",
        "IEC1",
        "IEC2"
      ]
    },
    "OriginalSamplingFrequency": {
      "type": "integer"
    },
    "Restorations": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "RestorationID": {
            "type": "string",
            "format": "uuid"
          },
          "PreservationAudioFileStart": {
            "type": "string",
            "pattern": "[0-9]{2}:[0-5][0-9]:[0-5][0-9]\\.[0-9]{3}"
          },
          "PreservationAudioFileEnd": {
            "type": "string",
            "pattern": "[0-9]{2}:[0-5][0-9]:[0-5][0-9]\\.[0-9]{3}"
          },
          "RestoredAudioFileURI": {
            "type": "string",
            "format": "uri"
          },
          "ReadingBackwards": {
            "type": "boolean"
```

```
          },
          "AppliedSpeedStandard": {
            "enum": [
              0.9375,
              1.875,
              3.75,
              7.5,
              15,
              30
            ]
          },
          "AppliedSamplingFrequency": {
            "type": "integer"
          },
          "AppliedEqualisationStandard": {
            "enum": [
              "IEC",
              "IEC1",
              "IEC2"
            ]
          }
        }
      },
      "minItems": 1,
      "uniqueItems": true,
      "required": [
        "RestorationID",
        "RestoredAudioFileURI",
        "PreservationAudioFileStart",
        "PreservationAudioFileEnd",
        "AppliedSamplingFrequency",
        "ReadingBackwards"
      ]
    }
  },
  "required": [
    "Restorations",
    "OriginalSamplingFrequency"
  ]
}
```

7.9.3 Semantics

Name	*Definition*
`OriginalSpeedStandard`	Speed standard applied to the tape recorder during the digitisation of an open-reel tape. It can be one of the following values: 0.9375, 1.875, 3.75, 7.5, 15, 30. These values are in inch per seconds (ips). This field is optional.
`OriginalEqualisationStandard`	Equalisation standard applied to the tape recorder during the digitisation of an open-reel tape. It can be one of the following values: "IEC", "IEC1", "IEC2". The notation refers to documents [19,20]. The association with `OriginalSpeedStandard` shall be compliant to the values indicated in [19,20]. This field is optional.
`OriginalSamplingFrequency`	UUID [3] that identifies a Restoration.
`Restorations`	List of restorations objects. Each object shall have at least the following fields: `RestorationID, RestoredAudioFileURI, PreservationAudioFileStart, PreservationAudioFileEnd, AppliedSamplingFrequency, ReadingBackwards.`

Name	*Definition*
`RestorationID`	UUID [7] that identifies a Restoration.
`PreservationAudioFileStart`	Time Label indicating the instant of the Preservation Audio File when the restoration starts.
`PreservationAudioFileEnd`	Time Label indicating the instant of the Preservation Audio File when the restoration ends.
`RestoredAudioFileURI`	URI of a Restored Audio File.
`ReadingBackwords`	Boolean value indicating if the audio signal direction has been inverted during the restoration process.
`AppliedSpeedStandard`	Speed standard applied during the restoration process. It can be one of the following values: 0.9375, 1.875, 3.75, 7.5, 15, 30. These values are in inch per seconds (ips). This field is optional.
`AppliedSamplingFrequency`	Specifies the sampling frequency of the Restored Audio File. This field is mandatory.
`AppliedEqualisationStandard`	Equalisation standard applied during the restoration process. It can be one of the following values: "IEC", "IEC1", "IEC2". The notation refers to documents [19,20]. The association with `AppliedSpeedStandard` shall be compliant to the values indicated in [19,20].

7.10 Emotion

The Syntax and Semantics of Emotion are specified by [4].

7.11 Emotionless Speech

An Audio File containing only speech in which music and other sounds are absent, and in which little or no identifiable emotion is perceptible by native listeners.

7.12 Enhanced Audio

Interleaved Multichannel Audio where each channel contains time aligned Enhanced Audio samples digitally represented with at least single precision floating point.

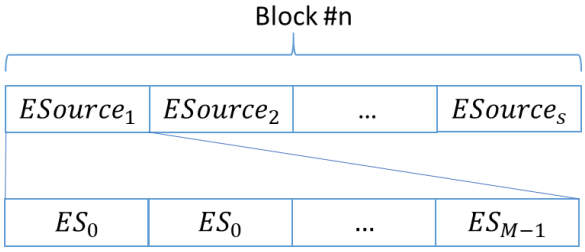

Figure 9 –Enhanced Audio

7.13 Enhanced Transform Audio

Transform Audio whose samples are Enhanced Transform Audio samples.

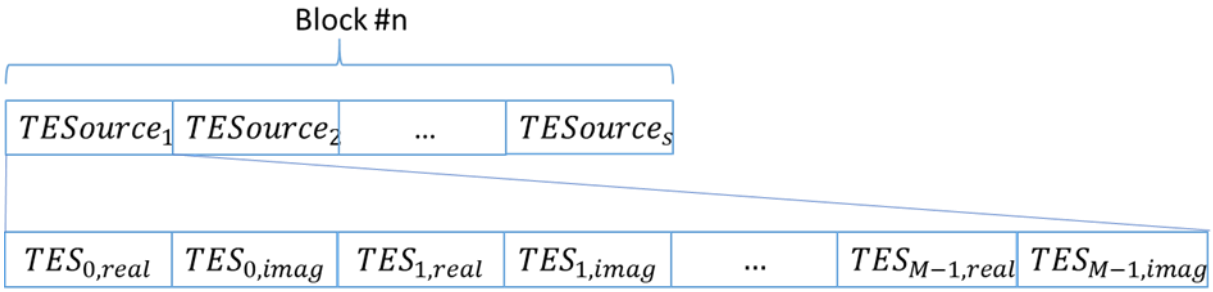

Figure 10 – Transform Enhanced Audio

7.14 Irregularity File

7.14.1 Definition

A file containing information about Irregularities of the Preservation Audio File and Audio-Visual Preservation File.

7.14.2 Syntax

The JSON schema of the Irregularity File is:

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Irregularity File",
  "type": "object",
  "properties": {
    "Offset": {
      "type": "integer"
    },
    "Irregularities": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "IrregularityID": {
            "type": "string",
            "format": "uuid"
          },
          "Source": {
            "enum": [
              "a",
              "v",
              "b"
            ]
          },
          "TimeLabel": {
            "type": "string",
            "pattern": "[0-9]{2}:[0-5][0-9]:[0-5][0-9]\\.[0-9]{3}"
          },
          "IrregularityType": {
            "enum": [
              "sp",
              "b",
              "sot",
              "eot",
              "da",
              "di",
              "m",
              "s",
              "wf",
              "pps",
              "ssv",
              "esv",
              "sb"
            ]
```

```
          },
          "IrregularityProperties": {
            "type": "object",
            "properties": {
              "ReadingSpeedStandard": {
                "enum": [
                  0.9375,
                  1.875,
                  3.75,
                  7.5,
                  15,
                  30
                ]
              },
              "ReadingEqualisationStandard": {
                "enum": [
                  "IEC",
                  "IEC1",
                  "IEC2"
                ]
              },
              "WritingSpeedStandard": {
                "enum": [
                  0.9375,
                  1.875,
                  3.75,
                  7.5,
                  15,
                  30
                ]
              },
              "WritingEqualisationStandard": {
                "enum": [
                  "IEC",
                  "IEC1",
                  "IEC2"
                ]
              }
            }
          },
          "ImageURI": {
            "type": "string",
            "format": "uri"
          },
          "AudioFileURI": {
            "type": "string",
            "format": "uri"
          }
        }
      },
      "minItems": 1,
      "uniqueItems": true,
      "required": [
        "IrregularityID",
        "Source",
        "TimeLabel"
      ]
    }
  },
  "required": [
    "Irregularities"
  ]
}
```

7.14.3 Semantics

Name	*Definition*
`Offset`	Integer value indicating the time offset (in milliseconds) between Preservation Audio File and Preservation Audio-Visual File. The time reference is the Preservation Audio File.
`Irregularities`	Array of Irregularities. Each Irregularity shall have at least an `IrregularityID`, `TimeLabel` and `TimeReference`.
`IrregularityID`	*UUID* [7] that identifies an Irregularity.
`Source`	“a”: if the Irregularity is detected by the Audio Analyser. “v”: if the Irregularity is detected by the Video Analyser. “b”: if the Irregularity is detected by both Audio Analyser and Video Analyser.
`TimeLabel`	Time Label indicating the timing of an Irregularity. The time reference is the Preservation Audio File.
`AudioFileURI`	*URI* of the Audio File related to an Irregularity. It is only used in the message between Audio Analyser and Tape Irregularity Classifier.
`IrregularityType`	Class of an Irregularity (see values in following Tables).
`IrregularityProperties`	Optional object containing additional specifications about the current Irregularity.
`ReadingSpeedStandard`	Speed standard applied during the digitisation phase. It can be one of the following values: 0.9375, 1.875, 3.75, 7.5, 15, 30. These values are in inch per seconds (ips). This field is optional.
`ReadingEqualisationStandard`	Equalisation standard applied during the digitisation phase. It can be one of the following values: "IEC", "IEC1", "IEC2". The notation refers to documents [14,15]. The association with `ReadingSpeedStandard` shall be compliant to the values indicated in [14,15]. This field is optional.
`WritingSpeedStandard`	Speed standard applied during the recording phase. It can be one of the following values: 0.9375, 1.875, 3.75, 7.5, 15, 30. These values are in inch per seconds (ips). This field is optional.
`WritingEqualisationStandard`	Equalisation standard applied during the recording phase. It can be one of the following values: "IEC", "IEC1", "IEC2". The notation refers to documents [14,15]. The association with `WritingSpeedStandard` shall be compliant to the values indicated in [14,15]. This field is optional.
`ImageURI`	*URI* of the Image related to an Irregularity. It is only used in the messages between Audio Analyser, Tape Irregularity Classifier, and Packager.

Table 28 - Extended list of Irregularities that can be detected by the Video Analyser

Code	*Name*	*Definition*
`sp`	***Splice***	Splice of magnetic tape to magnetic tape, or leader tape to magnetic tape (or vice versa).
`b`	***Brands on tape***	Most of the brands consist of the full name of the tape manufacturer, logo, or tape model codes. The brand changes in size, shape, and colour, depending on the tape used.
`sot`	***Start of tape***	It refers to what happens when the tape playback starts, at which point it is neither under tension nor in contact with the capstan and pinch roller. The distinguishing visual characteristic of this class is the tape coming in tension and in contact with the capstan and pinch roller. This happens at the beginning of the Preservation Audio-Visual File.
`eot`	***Ends of tape***	It refers to what happens when the tape reaches its end of playback, at which point it is neither under tension nor in contact with the capstan and pinch roller. The distinguishing visual characteristic of this class is the tape coming free or completely detached from the capstan. This happens at the end of the Preservation Audio-Visual File.
`da`	***Damaged tape***	It groups all kinds of damages on the surface of the tape and alterations of the tape shape. This class includes: 1. Ripples: this is formally known in the cataloguing rules as “kink” or “wrinkle”, these may be a single crease on a layer of tape or multiple creases in the tape. 2. Cupping: an abnormal flexure of the tape surface across or along its width, due to different rates of shrinkage along the substrate and recording layers. 3. Damage to tape edges, occurring when the edges do not appear flat or straight.
`di`	***Dirt***	Tape contamination and dirt: presence of mould, powder, crystals, other biological contaminations, or similar sullying.
`m`	***Marks***	Marks, signs or words written on the back of the tape (i.e., the nonmagnetic side) or on the adhesive tape of splices.
`s`	***Shadows***	The class contains frames in which shadows or reflections are temporarily cast on the tape by external objects in motion.
`wf`	***Wow and flutter***	Pitch variation due to the recording or playback equipment. If this effect is due to recording equipment it is detectable only on the Preservation Audio File and not on the Preservation Audio-Visual File.

Table 29 - List of Irregularities that can be detected only on the Preservation Audio File

Code	*Name*	*Definition*
`pps`	***Play, pause and stop***	Sound audio effects derived by play, pause or stop buttons during the recording. In a single tape several recordings from different sources can be recorded. This kind of irregularities cannot be identified in the digital video.

Code	*Name*	*Definition*
`ssv`	***Speed standard variation***	Instant when the recording has a variation of the speed (and, in case, of the equalization) standard.
`esv`	***Equalization standard variation***	Instant when the recording has a variation of the equalization standard without a change of the speed.
`sb`	***Signal backward***	Instant when a recording start playback audio signal backwards. This could happen in case of incorrect signal recording or digitization.

The Irregularities that could be identified in both audio and video are: `sp`, `sot`, `eot`, `da`, `di`, and `wf`.
Considering that **Brands on tape** are usually very frequent and repetitive, only one occurrence (usually the first one) is considered as a valid Irregularity by the Tape Irregularity Classifier.
Shadows has no impact on the signal. They should be considered because they can have an important impact on the classification, but they should not be included in the Preservation Master File.

7.15 Irregularity Image

JPEG file corresponding to an Irregularity conforming to [20].

7.16 Microphone Array Audio

Interleaved Multichannel Audio whose channels are sampled at a minimum of 5.33 ms (i.e., 256 samples at 48 kHz) to a maximum of 85.33 ms (i.e., 4096 samples at 48 kHz) and each sample is in single or double precision float.

7.17 Microphone Array Geometry

7.17.1 Definition

A Data Type representing the position of each microphone comprising a Microphone Array and specific characteristics such as microphone type, look directions, the array type, sampling rate and sample type.

7.17.2 Syntax

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Microphone Array Geometry",
  "type": "object",
  "properties": {
    "Header": {
      "type": "object",
      "properties": {
        "Standard": {
          "type": "string"
        },
        "Version": {
          "type": "integer"
        },
        "Subversion": {
          "type": "integer"
        }
      }
    },
    "MAGID": {
      "type": "string"
    },
```

```
"MicrophoneFeatures": {
  "type": "object",
  "properties": {
    "ArrayType": {
      "type": "integer"
    },
    "ArrayScat": {
      "type": "integer"
    },
    "ArrayFilterURI": {
      "type": "string",
      "format": "uri"
    }
  }
},
"SamplingFeatures": {
  "type": "object",
  "properties": {
    "SamplingRate": {
      "type": "integer"
    },
    "SampleType": {
      "type": "integer"
    }
  }
},
"BlockSize": {
  "type": "integer"
},
"NumberofMicrophones": {
  "type": "integer"
},
"Microphoneattributes": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "xCoord": {
        "type": "number"
      },
      "yCoord": {
        "type": "number"
      },
      "zCoord": {
        "type": "number"
      },
        "directivity": {
          "type": "integer"
        },
        "micxLookCoord": {
          "type": "number"
        },
        "micyLookCoord": {
          "type": "number"
        },
        "miczLookCoord": {
          "type": "number"
        }
      }
    },
    "minItems": 4,
    "uniqueItems": true,
    "required": [
      "xCoord",
      "yCoord",
      "zCoord",
      "directivity",
      "micxLookCoord",
      "micyLookCoord",
      "miczLookCoord"
    ]
  },
  "MicrophoneArrayLookCoord": {
    "type": "object",
```

```
        "properties": {
          "xLookCoord": {
            "type": "number"
          },
          "yLookCoord": {
            "type": "number"
          },
          "zLookCoord": {
            "type": "number"
          }
        },
        "uniqueItems": true,
        "required": [
          "xLookCoord",
          "yLookCoord",
          "zLookCoord"
        ]
      }
    },
    "required": [
      "MicrophoneArrayType",
      "MicrophoneArrayScat",
      "MicrophoneArrayFilterURI",
      "SamplingRate",
      "SampleType",
      "BlockSize",
      "NumberofMicrophones",
      "MicrophoneList",
      "MicrophoneArrayLookCoord"
    ]
  }
```

7.17.3 Semantics

Table 30 gives the Semantics of Microphone Array Geometry.

Table 30 – Semantics of Microphone Array Geometry

Label	Size	Description
HEADER	9 Bytes	
• Standard	7 Bytes	The CAE-MAG string
• Version	1 Byte	Major MPAI-CAE version
• Subversion	1 Byte	Minor MPAI-CAE version
MAGID	16 Bytes	UUID Identifier of the Microphone Array Geometry.
Microphone features		
• ArrayType	bit 0-1	Indicates the type of microphone array positioning such as 00:Spherical, 01:Circular, 10:Planar, 11:Linear. (uint8)
• ArrayScat	bit 2	Indicates the type of the microphone array (0:Rigid, 1:Open). (uint8)
• Reserved	bit 2-7	
• ArrayFilterURI	N Bytes	A uniform resource identifier (URI) string identifying the path to a local or remote file containing specific filter coefficients of the microphone array to be used for equalisation. (string)
Sampling features		
• SamplingRate	0-3 bits	0:8, 1:16, 2: 22.05, 3:24, 4:32, 5:44.1, 6:48, 7: 96, 8: 192 (all kHz)

• SampleType	4-6 bits	(aka sample precision)0:8, 1:16, 2:24, 3:32, 4:64 (bits/sample)
• Reserved	bit 7	
BlockSize	4 Bytes	Minimum BlockSize: ≥ 256.
NumberOfMicrophones	1 Byte	
MicrophoneAttributes		A list containing Microphone attributes.
• MicrophoneID	1 Byte	
• xCoord	4 Bytes	x position of the microphone in m. (number)
• yCoord	4 Bytes	y position of the microphone in m.(number)
• zCoord	4 Bytes	z position of the microphone in m. (number)
• directivity	bit 0-2	The directivity pattern of the specific microphone, 000: omnidirectional, 001: figure of eight, 010: cardioid, 011: supercardioid, 100: hypercardioid (uint8)
• Reserved	Bit 3-7	
• Channel map	1 Byte	Indicates the number of Audio channel
• micxLookCoord	4 Bytes	x component of the vector representing the look direction of the microphone in m. (number)
• micyLookCoord	4 Bytes	y component of the vector representing the look direction of the microphone in m. (number)
• miczLookCoord	4 Bytes	z component of the vector representing the look direction of the microphone. (number)
MicrophoneArrayLookCoord		
xLookCoord	4 Bytes	x component of the vector representing the look direction of the microphone array. (number)
yLookCoord	4 Bytes	y component of the vector representing the look direction of the microphone array. (number)
zLookCoord	4 Bytes	z component of the vector representing the look direction of the microphone array. (number)

7.18 Mode Selection

In the EES use case, one of “Mode-1” or “Mode-2” indicating that Pathway 1 or Pathway 2, respectively, will be followed in adding emotion to Emotionless Speech. In Mode-1, a suitably configured Speech Feature Analysis1 module will capture emotional features from Model Utterance and transfer them to Emotionless Speech, thus producing Speech with Emotion. By contrast, in Mode-2, a suitable Speech Feature Analysis2 module will analyse Emotionless Speech and pass extracted Emotionless Speech Features along with a specification of the desired emotion to Emotion Feature Production. These modules will produce (emotional) Neural Speech Features and pass them to a Neural Emotion Insertion module capable of combining Emotionless Speech and (emotional) Neural Speech Features to produce Speech with Emotion. See Section 5.1.3.

7.19 Multichannel Audio

A data structure containing between 4 and 256 time-aligned interleaved Audio Channels and organised in blocks as depicted in *Figure 11*.

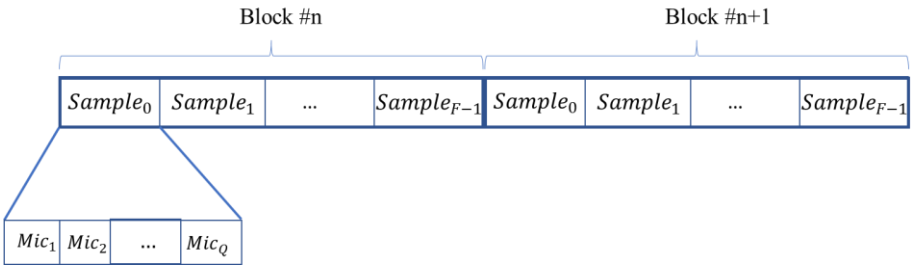

Figure 11 - Microphone Array Signals input sample ordering

7.20 Multichannel Audio-Stream

7.20.1 Definition

A data structure containing Audio Objects packaged with Audio Scene Geometry and Time Code according to the structure specified in *Table 31*.

7.20.2 Syntax

```
{
  "$id": "https://schemas.mpai.community/CAE/V2.1/MultichannelAudioStream.json",
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "MultichannelAudioStream",
  "type": "object",
  "properties": {
    "BlockHeader": {
      "type": "object",
      "properties": {
        "HEAD": {
          "type": "string"
        },
        "BlockIndex": {
          "type": "integer"
        },
        "BlockStart": {
          "type": "integer"
        }
      }
    },
    "MASID": {
      "type": "string"
    },
    "BlockInfo": {
      "type": "object",
      "properties": {
        "BlockIndex": {
          "type": "integer"
        },
        "BlockStart": {
          "type": "integer"
        },
        "BlockEnd": {
          "type": "integer"
        },
        "Checksum": {
          "type": "integer"
        }
      }
    },
    "AudioObjectCount": {
      "type": "integer"
    },
    "AudioObjectsData": {
      "type": "object",
      "properties": {
        "AudioObjectID": {
          "type": "string"
```

```
      },
      "SamplingRate": {
        "type": "number"
      },
      "SampleType": {
        "type": "number"
      },
      "Reserved": {
        "type": "number"
      },
      "SpatialAttitude": {
        "$ref": "https://schemas.mpai.community/OSD/V1.0/data/SpatialAttitude.json"
      }
    },
    "required": [
      "BlockHeader",
      "MAPID",
      "BlockInfo",
      "AudioObjectCount"
    ]
  }
 }
}
```


7.20.3 Semantics

Table 31 – Multichannel Audio Stream Semantics

Label	Size	Description
HEADER	9 Bytes	
Standard	7 Bytes	The CAE-MAS string
Version	1 Byte	Major MPAI-CAE version
Subversion	1 Byte	Minor MPAI-CAE version
MASID	16 Bytes	UUID Identifier of the Multichannel Audio Stream.
BlockInfo		
• BlockIndex	8 Bytes	Indicates the timing order of the output block. Derived from Audio Scene Geometry.
• BlockStart	8 Bytes	Derived from Audio Scene Geometry.
• BlockEnd	8 Bytes	Derived from Audio Scene Geometry.
• BlockSize	1 Byte	Derived from Audio Scene Geometry.
• Checksum	1 Byte	Checksum is calculated by summing the block and speech header bytes modulo 256.
AudioObjectCount	1 Byte	AudioObjectCount of Audio Scene Geometry.
AudioObjectsData	N1 Bytes	
• AudioObjectID	16 Bytes	AudioObjectID in Audio Object.
• Sampling Rate	0-3 bits	SamplingRate of Audio Scene Descriptors.
• Sample Type	4-6 bits	(aka, sample precision) 0:8, 1:16, 2:24, 3:32, 4:64 (bits/sample)
• Reserved	7 bit	
• Spatial Attitude	N2 Bytes	

7.21 Neural Network Speech Model

A Neural Network Model trained on Speech Segments for Modelling and used to synthesize replacements for the entire Damaged Segment or Damaged Sections within it.

The Neural Network Speech Model is passed to Speech Synthesiser as a data set with the following signalling:

1. 0: Khronos Neural Network Exchange Format (NNEF) [16].

2. 1: Open Neural Network Exchange (ONNX) format [17].

7.22 Preservation Audio File

An Audio File containing Audio sampled at one of the following values 44.1, 48, 96, 192 kHz with 16 or 24 bits/sample.

7.23 Preservation Audio-Visual File

An Audio-Visual File containing:
1. Video.
2. Audio sampled at one of the following values 32, 44.1, 48 kHz with 16 or 24 bits/sample.

7.24 Preservation Master Files

The following set of files:
1. Preservation Audio File.
2. Preservation Audio-Visual File where the audio has been replaced with the Audio of the Preservation Audio File fully synchronised with the video.
3. The set of Irregularity Images in a .zip file [11].
4. The Irregularity File listing all detected Irregularities.

7.25 Speech Descriptors

7.25.1 Definition

Data representing various features of a Speech Segment, including speaker identity, prosody, and additional vocal elements including tension, whispery quality, or creaky voice.

7.25.2 Syntax

```
{
  "$id": "https://schemas.mpai.community/MMC/V2.1/SpeechDescriptors.json",
  "$schema": "http://json-schema.org/draft-07/schema",
  "title": "SpeechDescriptors",
  "type": "object",
  "properties": {
    "SpeechFeatures": {
      "type": "object",
      "properties": {
        "pitch": {
          "type": "number"
        },
        "tone": {
          "type": "object",
          "properties": {
            "toneName": {
              "type": "string"
            },
            "toneSetName": {
              "type": "string"
            }
          }
        },
        "intonation": {
          "type": "object",
          "properties": {
            "pitch": {
              "type": "number"
            },
            "speed": {
              "type": "number"
            },
            "intensity": {
              "type": "number"
            }
```

```
          },
          "emotion": {
            "$ref": "https://schemas.mpai.community/MMC/V2.0/data/Emotion.json"
          },
          "NNSpeechFeatures": {
            "type": "array",
            "items": {
              "type": "number"
            }
          }
        }
      }
    }
  }
}
```

7.25.3 Semantics

Name	*Definition*
SpeechFeatures	Characteristic elements extracted from the input speech, specifically pitch, tone, intonation, intensity, speed, emotion, and NNspeechFeatures.
NNSpeechFeatures	Specifically neural-network-based characteristic elements extracted from the input speech by Neural Network
intonations	Vector representing an ordered sequence of elements, where each element is a triplet specifying the pitch, duration, and intensity of one linguistic *unit*. This vector starts at 0.0 ms.
pitch	Member of an element of *intonations* indicating the fundamental frequency in Hz (Hertz) of linguistic *unit*.
intensity	Member of an element of *intonations* indicating the energy of the linguistic *unit* perceived as loudness. Intensity is expressed as a real number in dBs (decibels).
duration	Member of an element of *intonations* indicating the length of linguistic *units* measured in milliseconds expressed as a real number.
unit	Specifies the linguistic unit. Here we are considering only "phonemes".

Note: *Table* 32 lists some Basic Tones, e.g., "formal" or "informal," with semantic characterisations of each. Elements can be added to the Basic Tone Set or new sets can be defined via the registration procedure specified in 7.9.3.

Table 32 – Basic Tones

TONE CATEGORIES	ADJECTIVAL	Semantics
FORMALITY	formal informal	serious, official, polite everyday, relaxed, casual
ASSERTIVENESS	assertive factual hesitant	certain about content neutral about content uncertain about content
REGISTER (per situation or use case)	conversational directive	appropriate to informal speech related to commands or requests for action

7.26 Spherical Harmonic Decomposition

The complex-valued spherical harmonics coefficients for each Transform Audio Block. $A_{l,m,real}(k)$ and $A_{l,m,imag}(k)$ represent the real and imaginary parts of the Spherical Harmonics Decomposition coefficients of order l and degree m corresponding to the k-th transform coefficient respectively.

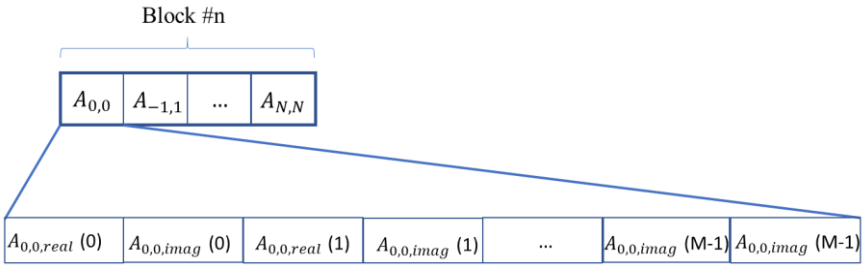

Figure 12 – Spherical Harmonics Decomposition of sound field

7.27 Transform Audio

A data structure obtained by transforming Multichannel Audio containing speech and where the real and imaginary parts of the transformed data are represented as single or double precision numbering point values.

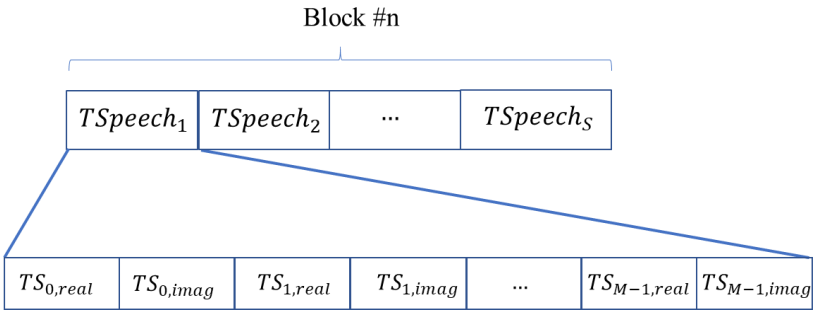

Figure 13 - Transform domain separated speech signals

7.28 Transform Multichannel Audio

A data structure obtained from the transformation of Microphone Array Audio.

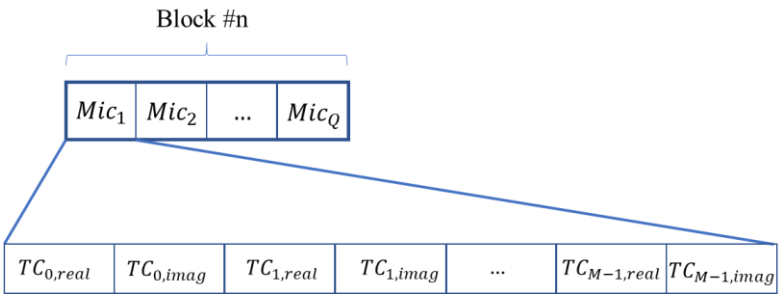

Figure 14 – Transform Multichannel Audio

7.29 Video

Video satisfies the following specifications:

1. Pixel shape: square.

2. Bit depth: 8 or 10 bits/pixel.
3. Aspect ratio: 4/3 or 16/9.
4. 640 < # of horizontal pixels < 1920.
5. 480 < # of vertical pixels < 1080.
6. Frame frequency 24-120 Hz.
7. Scanning: progressive or interlaced.
8. Colorimetry: ITU-R BT709 or BT2020.
9. Colour format: RGB or YUV.
10. Compression, either:
    a. Uncompressed.
    b. Compressed according to one of the following standards: MPEG-4 AVC [11], MPEG-H HEVC [12], MPEG-5 EVC [13].

Annex 1 - MPAI-wide terms and definitions

The Terms used in this standard whose first letter is capital and are not already included in *Table 1* are defined in *Table 33.*

Table 33 – MPAI-wide Terms

Term	Definition
Access	Static or slowly changing data that are required by an application such as domain knowledge data, data models, etc.
AI Framework (AIF)	The environment where AIWs are executed.
AI Workflow (AIW)	An organised aggregation of AIMs implementing a Use Case receiving AIM-specific Inputs and producing AIM-specific Outputs according to its Function.
AI Module (AIM)	A processing element receiving AIM-specific Inputs and producing AIM-specific Outputs according to according to its Function.
Application Standard	An MPAI Standard designed to enable a particular application domain.
Channel	A connection between an output port of an AIM and an input port of an AIM. The term "connection" is also used as synonymous.
Communication	The infrastructure that implements message passing between AIMs.
Composite AIM	An AIM aggregating more than one AIM.
Component	One of the 7 AIF elements: Access, Communication, Controller, Internal Storage, Global Storage, MPAI Store, and User Agent.
Composite AIM	
Conformance	The attribute of an Implementation of being a correct technical Implementation of a Technical Specification.
Conformance Tester	An entity authorised by MPAI to Test the Conformance of an Implementation.
Conformance Testing	The normative document specifying the Means to Test the Conformance of an Implementation.
Conformance 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 Format	The standard digital representation of data.
Data Semantics	The meaning of data.
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.
Explainability	The ability to trace the output of an Implementation back to the inputs that have produced it.
Fairness	The attribute of an Implementation whose extent of applicability can be assessed by making the training set and/or network open to testing for bias and unanticipated results.

Function	The operations effected by an AIW or an AIM on input data.
Global Storage	A Component to store data shared by AIMs.
Internal Storage	A Component to store data of the individual AIMs.
Identifier	A name that uniquely identifies an Implementation.
Implementation	1. An embodiment of the MPAI-AIF Technical Specification, or 2. An AIW or AIM of a particular Level (1-2-3) conforming with a Use Case of an MPAI Application Standard.
Implementer	A legal entity implementing MPAI Technical Specifications.
ImplementerID (IID)	A unique name assigned by the ImplementerID Registration Authority to an Implementer.
ImplementerID Registration Authority (IIDRA)	The function within the MPAI Store to assign ImplementerID's to Implementers.
Interoperability	The ability to functionally replace an AIM with another AIM having the same Interoperability Level.
Interoperability Level	The attribute of an AIW and its AIMs to be executable in an AIF Implementation and to: 1. Be proprietary (Level 1). 2. Pass the Conformance Testing (Level 2) of an Application Standard. 3. `Pass the Performance Testing (Level 3) of an Application Standard.
Knowledge Base	Structured and/or unstructured information made accessible to AIMs via MPAI-specified interfaces.
Message	A sequence of Records transported by Communication through Channels.
Normativity	The set of attributes of a technology or a set of technologies specified by the applicable parts of an MPAI standard.
Performance	The attribute of an Implementation of being Reliable, Robust, Fair and Replicable.
Performance Assessment	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.
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.
Profile	A particular subset of the technologies used in MPAI-AIF or an AIW of an Application Standard and, where applicable, the classes, other subsets, options and parameters relevant to that subset.
Record	A data structure with a specified structure.
Reference Model	The AIMs and theirs Connections in an AIW.
Reference Software	A technically correct software implementation of a Technical Specification containing source code, or source and compiled code.
Reliability	The attribute of an Implementation that performs as specified by the Application Standard, profile and version the Implementation refers to, e.g., within the application scope, stated limitations, and for the period of time specified by the Implementer.
Replicability	The attribute of an Implementation whose Performance, as Assessed by a Performance Assessor, can be replicated, within an agreed level, by another Performance Assessor.

Robustness	The attribute of an Implementation that copes with data outside of the stated application scope with an estimated degree of confidence.
Service Provider	An entrepreneur who offers an Implementation as a service (e.g., a recommendation service) to Users.
Standard	The ensemble of Technical Specification, Reference Software, Conformance Testing and Performance Assessment of an MPAI application Standard.
Technical Specification	(Framework) the normative specification of the AIF. (Application) the normative specification of the set of AIWs belonging to an application domain along with the AIMs required to Implement the AIWs that includes: 1. The formats of the Input/Output data of the AIWs implementing the AIWs. 2. The Connections of the AIMs of the AIW. 3. The formats of the Input/Output data of the AIMs belonging to the AIW.
Testing Laboratory	A laboratory accredited by MPAI to Assess the Grade of Performance of Implementations.
Time Base	The protocol specifying how Components can access timing information.
Topology	The set of AIM Connections of an AIW.
Use Case	A particular instance of the Application domain target of an Application Standard.
User	A user of an Implementation.
User Agent	The Component interfacing the user with an AIF through the Controller.
Version	A revision or extension of a Standard or of one of its elements.
Zero Trust	A model of cybersecurity primarily focused on data and service protection that assumes no implicit trust.

Annex 2 - 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

Annex 3 - The Governance of the MPAI Ecosystem (Informative)

Level 1 Interoperability

With reference to *Figure 1*, MPAI issues and maintains a standard – called MPAI-AIF – whose components are:

1. An environment called AI Framework (AIF) running AI Workflows (AIW) composed of inter-connected AI Modules (AIM) exposing standard interfaces.
2. A distribution system of AIW and AIM Implementation called MPAI Store from which an AIF Implementation can download AIWs and AIMs.

A Level 1 Implementation shall be an Implementation of the MPAI-AIF Technical Specification executing AIWs composed of AIMs able to call the MPAI-AIF APIs.

Implementers' benefits	Upload to the MPAI Store and have globally distributed Implementations of - AIFs conforming to MPAI-AIF. - AIWs and AIMs performing proprietary functions executable in AIF.
Users' benefits	Rely on Implementations that have been tested for security.
MPAI Store	- Tests the Conformance of Implementations to MPAI-AIF. - Verifies Implementations' security, e.g., absence of malware. - Indicates unambiguously that Implementations are Level 1.

Level 2 Interoperability

In a Level 2 Implementation, the AIW shall be an Implementation of an MPAI Use Case and the AIMs shall conform with an MPAI Application Standard.

Implementers' benefits	Upload to the MPAI Store and have globally distributed Implementations of - AIFs conforming to MPAI-AIF. - AIWs and AIMs conforming to MPAI Application Standards.
Users' benefits	- Rely on Implementations of AIWs and AIMs whose Functions have been reviewed during standardisation. - Have a degree of Explainability of the AIW operation because the AIM Functions and the data Formats are known.
Market's benefits	- Open AIW and AIM markets foster competition leading to better products. - Competition of AIW and AIM Implementations fosters AI innovation.
MPAI Store's role	- Tests Conformance of Implementations with the relevant MPAI Standard. - Verifies Implementations' security. - Indicates unambiguously that Implementations are Level 2.

Level 3 Interoperability

MPAI does not generally set standards on how and with what data an AIM should be trained. This is an important differentiator that promotes competition leading to better solutions. However, the performance of an AIM is typically higher if the data used for training are in greater quantity and more in tune with the scope. Training data that have large variety and cover the spectrum of all cases of interest in breadth and depth typically lead to Implementations of higher "quality".

For Level 3, MPAI normatively specifies the process, the tools and the data or the characteristics of the data to be used to Assess the Grade of Performance of an AIM or an AIW.

Implementers' benefits	May claim their Implementations have passed Performance Assessment.

Users' benefits	Get assurance that the Implementation being used performs correctly, e.g., it has been properly trained.
Market's benefits	Implementations' Performance Grades stimulate the development of more Performing AIM and AIW Implementations.
MPAI Store's role	- Verifies the Implementations' security - Indicates unambiguously that Implementations are Level 3.

The MPAI ecosystem

The following *Figure 15* is a high-level description of the MPAI ecosystem operation applicable to fully conforming MPAI implementations as specified in the Governance of the MPAI Ecosystem Specification [1]:

1. MPAI establishes and controls the not-for-profit MPAI Store.
2. MPAI appoints Performance Assessors.
3. MPAI publishes Standards.
4. Implementers submit Implementations to Performance Assessors.
5. If the Implementation Performance is acceptable, Performance Assessors inform Implementers and MPAI Store.
6. Implementers submit Implementations to the MPAI Store
7. MPAI Store verifies security and Tests Conformance of Implementation.
8. Users download Implementations and report their experience to MPAI.

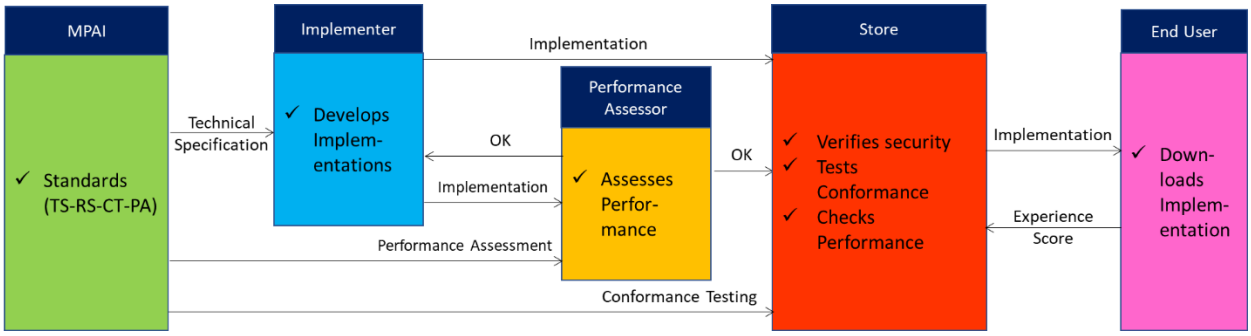

Figure 15 – The MPAI ecosystem operation

Annex 4 – Patent Declarations

Technical Specification: Context-based Audio Enhancement (MPAI-CAE) V2.1 has been developed according to the process outlined in the MPAI Statutes [21] and the MPAI Patent Policy [22].
The following entities have agreed to license their standard essential patents reading on the MPAI-CAE V1.4 according to the MPAI-CAE Framework License [23]:

Entity	Email address
ASELSAN A. Ş.	*Mert Burkay Çöteli MBCoteli@aselsan.com.tr*
Middle East Technical University (METU)	*Huseyin Hacihabiboglu hhuseyin@metu.edu.tr*
Speech Morphing, Inc.	*Fathy Yassa fathy@speechmorphing.com*

Annex 5 - Examples (Informative)

3.1 Audio Scene Geometry

An example of Audio Scene Geometry.


```
{
  "BlockIndex": 1,
  "BlockStart": 1631536788000,
  "BlockEnd": 1631536788063,
  "SpeechCount": 2,
  "SpeechList": [
                {
                        "SpeechID": "09859d16-3c73-4bb0-9c74-91b451e34925",
                        "ChannelID": 1,
                        "AzimuthDirection": 90.0,
                        "ElevationDirection": 30.0,
                        "Distance": 2.0,
                        "DistanceFlag": false
                },
                {
                        "SpeechID": "3cdc2973-e95e-4125-acb7-121ad89067ef",
                        "ChannelID": 2,
                        "AzimuthDirection": 180.0,
                        "ElevationDirection": 30.0,
                        "Distance": 1.27,
                        "DistanceFlag": false
                }
  ],
  "SourceDetectionMask": [0,1]
}
```


3.2 Damaged List

An example of a damaged list JSON file:


```
{
        "DamagedSections": [
    {
        "SegmentStart": "00:00:01.351",
         "SegmentEnd": "00:01:55.654",
     },
     {
        "SegmentStart": "00:01:55.654",
        "SegmentEnd": "00:02:35.168",
     }
        ]
}
```


3.3 Editing List

Example of a complete Editing List with two elements: the first related to reading backwards error, whereas the second to speed and equalisation errors.


```
{
    "OriginalSpeedStandard": 15,
    "OriginalEqualisationStandard": "IEC1",
    "OriginalSampleFrequency": 96000,
    "Restorations": [{
        "RestorationID": "09859d16-3c73-4bb0-9c74-91b451e34925",
        "PreservationAudioFileStart": "00:00:00.000",
        "PreservationAudioFileEnd": "00:00:05.125",
        "RestoredAudioFileURI": "http://www.place_to_be_defined.com/restored_1",
        "ReadingBackwords": true,
        "AppliedSpeedStandard": 15,
        "AppliedSampleFrequency": 96000,
        "OriginalEqualisationStandard": "IEC1"
    },
```

```
    {
        "RestorationID": "3cdc2973-e95e-4125-acb7-121ad89067ef ",
        "PreservationAudioFileStart": "00:00:05.125",
        "PreservationAudioFileEnd": "00:00:15.230",
        "RestoredAudioFileURI": "http://www.place_to_be_defined.com/restored_2",
        "ReadingBackwords": false,
        "AppliedSpeedStandard": 7.5,
        "AppliedSampleFrequency": 48000,
        "OriginalEqualisationStandard": "IEC2"
    }]
}
```


3.4 Irregularity File

An example of Irregularity File from Audio Analyser to Video Analyser is:


```
{
    "Offset": 150,
    "Irregularities": [{
        "IrregularityID": "09859d16-3c73-4bb0-9c74-91b451e34925",
        "Source": "a",
        "TimeLabel": "00:02:45.040"
    },{
        "IrregularityID": "3cdc2973-e95e-4125-acb7-121ad89067ef",
        "Source": "a",
        "TimeLabel": "00:04:89.020"
    }]
}
```


An example of Irregularity File from Video Analyser to Audio Analyser is:


```
{
    "Irregularities": [{
        "IrregularityID": "09859d16-3c73-4bb0-9c74-91b451e34925",
        "Source": "v",
        "TimeLabel": "00:02:45.040"
    },{
        "IrregularityID": "3cdc2973-e95e-4125-acb7-121ad89067ef",
        "Source": "v",
        "TimeLabel": "00:04:89.020"
    }]
}
```


An example of Irregularity File from Audio Analyser to Tape Irregularity Classifier is:


```
{
    "Offset": 150,
    "Irregularities": [{
        "IrregularityID": "09859d16-3c73-4bb0-9c74-91b451e34925",
        "Source": "a",
        "TimeLabel": "00:02:45.040",
        "AudioSegmentURI": "http://www.place_to_be_defined.com/audio_segment_1",
        "IrregularityType": "ssv",
        "IrregularityProperties: {
            "ReadingSpeedStandard": 15,
            "ReadingEqualisationStandard": "IEC1",
            "WritingSpeedStandard": 7.5,
            "WritingEqualisationStandard": "IEC2"
        }
    },{
        "IrregularityID": "3cdc2973-e95e-4125-acb7-121ad89067ef",
        "Source": "v",
        "TimeLabel": "00:04:89.020",
        "AudioSegmentURI": "http://www.place_to_be_defined.com/audio_segment_2"
    }]
}
```


An example of Irregularity File from Video Analyser to Tape Irregularity Classifier is:


```
{
    "Offset": 150,
    "Irregularities": [{
        "IrregularityID": "09859d16-3c73-4bb0-9c74-91b451e34925",
```

```
        "Source": "a",
        "TimeLabel": "00:02:45.040",
        "ImageURI": "http://www.place_to_be_defined.com/image_1"
    },{
        "IrregularityID": "3cdc2973-e95e-4125-acb7-121ad89067ef",
        "Source": "v",
        "TimeLabel": "00:04:89.020",
        "ImageURI": "http://www.place_to_be_defined.com/image_2"
    }]
}
```


An example of Irregularity File from Tape Irregularity Classifier to Tape Audio Restoration is:


```
{
    "Irregularities": [{
        "IrregulatityID": "09859d16-3c73-4bb0-9c74-91b451e34925",
        "Source": "a",
        "TimeLabel": "00:02:45.040",
        "IrregularityType": "ssv",
        "IrregularityProperties: {
            "ReadingSpeedStandard": 15,
            "ReadingEqualisationStandard": "IEC1",
            "WritingSpeedStandard": 7.5,
            "WritingEqualisationStandard": "IEC2"
        }
    },{
        "IrregulatityID": "3cdc2973-e95e-4125-acb7-121ad89067ef",
        "Source": "a",
        "TimeLabel": "00:04:89.020",
        "IrregularityType": "esv",
        "IrregularityProperties: {
            "ReadingSpeedStandard": 7.5,
            "ReadingEqualisationStandard": "IEC2",
            "WritingSpeedStandard": 7.5,
            "WritingEqualisationStandard": "IEC1"
        }
    }]
}
```


An example of Irregularity File from Tape Irregularity Classifier to Packager is:


```
{
    "Offset": 150,
    "Irregularities": [{
        "IrregulatityID": "09859d16-3c73-4bb0-9c74-91b451e34925",
        "Source": "v",
        "TimeLabel": "00:02:45.040",
        "IrregularityType": "sot",
        "ImageURI": "http://www.place_to_be_defined.com/image_1"
    },{
        "IrregulatityID": "3cdc2973-e95e-4125-acb7-121ad89067ef",
        "Source": "b",
        "TimeLabel": "00:04:89.020",
        "IrregularityType": "sp",
        "ImageURI": "http://www.place_to_be_defined.com/image_2"
    }]
}
```


3.5 Microphone Array Geometry


```
{
  "MicrophoneArrayType": 0,
  "MicrophoneArrayScat": 0,
  "MicrophoneArrayFilterURI": "https://mpai.community/standards/mpai-cae/",
  "SamplingRate": 4,
  "SampleType": 0,
  "BlockSize": 3,
  "NumberofMicrophones": 4,
  "MicrophoneList": [
              {
                      "xCoord": 1.0,
                      "yCoord": 2.0,
```

```
                        "zCoord": 3.0,
                        "directivity": 0,
                        "micxLookCoord": 70.2,
                        "micyLookCoord": 75.5,
                        "miczLookCoord": 87.3
                },
                {
                        "xCoord": 5.3,
                        "yCoord": 5.6,
                        "zCoord": 74.3,
                        "directivity": 1,
                        "micxLookCoord": 67.9,
                        "micyLookCoord": 75.2,
                        "miczLookCoord": 90.0
                },
                {
                        "xCoord": 34.2,
                        "yCoord": 65.2,
                        "zCoord": 56.9,
                        "directivity": 2,
                        "micxLookCoord": 56.8,
                        "micyLookCoord": 87.9,
                        "miczLookCoord": 78.3
                },
                {
                        "xCoord": 34.9,
                        "yCoord": 29.7,
                        "zCoord": 89.8,
                        "directivity": 3,
                        "micxLookCoord": 56.9,
                        "micyLookCoord": 65.4,
                        "miczLookCoord": 72.9
                }
  ],
  "MicrophoneArrayLookCoord": [{
    "xLookCoord": 56.0,
    "yLookCoord": 90.0,
    "zLookCoord": 86.3
  }]
}
```


3.6 Prosodic Speech Features


```
{
    "intonations": [{
        "pitch": 300,
        "intensity": 88.7,
        "duration”:100.0
    },{
        "pitch": 180,
        "intensity": 85.2,
        "duration”:98.0
    },{
        "pitch": 280,
        "intensity": 92.5,
        "duration”:92.0
    },{
        "pitch": 230,
        "intensity": 81.9,
        "duration”:98.0
    },{
        "pitch": 150,
        "intensity": 78.3,
        "duration”:98.0
    }],
    "unit": "phoneme"
}
```


3.7 Neural Speech Features


```
[
    1.456,
```

```
    5.1289,
    0.12,
    12345.54378,
    12389943.2837,
    58.29
]
```

Annex 6 - Communication Among AIM Implementors (Informative)

A core design principle of MPAI is modularity: AI Modules or AIMs and their interfaces must be defined so that each AIM can be built by an independent implementor, without damage to the function of the relevant AI Workflow (AIW) as an ensemble. Accordingly, to the extent possible, AIM input and output data are specified so that the inner implementation of an AIM need not be known or considered by AIMs cooperating in an AIW or in a Composite AIM. In other words, so far as possible, cooperating AIMs are designed to interact as black boxes. However, AIMs based upon the neural network technology currently prevalent in AI systems will sometimes require closer cooperation – in effect, greater transparency.

A neural-network-based AIM may sometimes deliver its output to downstream AIMs in a mutually intelligible format such as text, so that the receiving AIMs can obtain training corpora with relative ease. (Auxiliary programs may sometimes be available for translation of the relevant output into the comprehensible format.) Alternatively, a precise specification of the syntax and semantics of the output may meet the training needs of the downstream AIM.

Sometimes, however, the delivery may be in the form of neural vectors; and in this case, some assistance in processing these will be required. For training purposes, the downstream AIM will need either a sufficient corpus of output vectors from the upstream AIM or the actual neural network model used to train the upstream AIM, which can then be used to produce a new training corpus.

The Emotion Enhanced Speech workflow provides an example. It is designed to enable modification of the Translated Speech (that is, of the target language or output speech) using Speech Features extracted from the input, or source language, speech. This modification can enable the spoken translation to express the original emotion, or to employ the original speaker's voice quality to give the impression that he or she is pronouncing the translation. For these purposes, a Speech Feature Extraction AIM can extract relevant speech features from the input speech and pass them to the Text-To-Speech (Features) AIM. However, while the two AIMs can indeed be independently implemented, the downstream (receiving) Text-to-Speech (Features) AIM will need to process the received speech features appropriately. If Speech Feature Extraction employs neural network technology and passes the resulting features as vectors, then Text-To-Speech (Features) will need either a sufficient corpus of output vectors from Speech Feature Extraction or the actual neural network model used to train that AIM.

There are comparable considerations for the Conversation with Emotion (CWE) use case. And again, they will obtain for any AIMs that exchange neural information. In explicitly providing for such communication among artificial machine learning models and components, MPAI is not only recognising practical requirements for cooperation among such modules, but also acknowledging an analogy with communication among biological neural subsystems.