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

MPAI Technical Specification

Context-based Audio Enhancement
MPAI-CAE

V1.4

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 1 – Notices and Disclaimers.

Context-based Audio Enhancement
Version V1.4

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.

For the aforementioned reasons, 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 21* 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 takes care of secure distribution of 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.

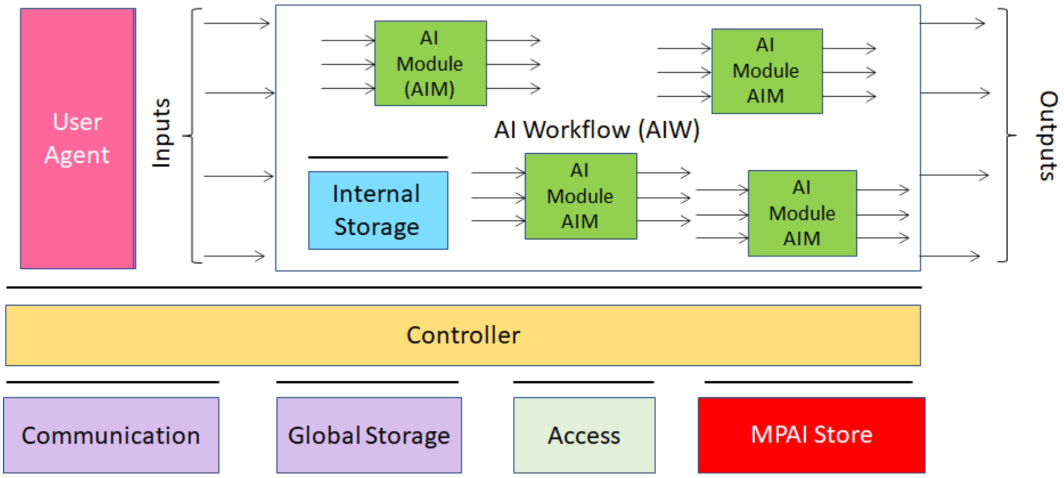

Figure 1 - The AI Framework (AIF) Reference Model and its Components

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 2*).

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.

2 Scope of standard

The common characteristic shared by the MPAI-CAE Use Cases is the improvement of 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. These use cases are *Emotion Enhanced Speech (EES), Audio Recording Preservation (ARP), Speech Restoration System (SSR),* and *Enhanced Audioconference Experience (EAE).*

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.

2.1 Emotion-Enhanced Speech (EES)

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) enables a user to indicate a model utterance or an Emotion to obtain an emotionally charged version of a given utterance.

CAE-EES implementation 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.

2.2 Audio Recording Preservation (ARP)

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 5.2) 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.

2.3 Speech Restoration System (SRS)

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.

2.4 Enhanced Audioconference Experience (EAE)

The user experience of a video/audio conference 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.

Enhanced Audioconference Experience (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.

The AIMs of the Enhanced Audioconference Experience (EAE) Use Case improve auditory experience in an audioconference, thereby reducing the effects of audioconference fatigue.

2.5 Normative content of the Use Cases

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 6.3.
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 6.3.
   c. Receive as input and produce as output data having the Format specified in Section 6.3.
3. A data *Format*, the data shall have the Format specified in Section 6.3.

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.

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 21*.

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 32.
Audio Block	A set of consecutive Audio samples.
Audio Channel	A sequence of Audio Blocks.
Audio File	A .wav file [6].

Audio Object	Direct 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 [10].
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	One of the human emotions listed in *Table 15*, or in an augmented or alternate version of this *Table 15.*
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.
Interleaved Multichannel Audio	A data structure containing at least 2 time-aligned interleaved Audio Channels.
Irregularity	An event of interest to preservation from in *Table 17* and *Table 18*
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 [13].
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 set of multiple time-aligned Audio Channels
Multichannel Audio + Audio Scene Geometry	Multichannel Audio 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 Audio 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 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.
Time Code	Number of ms from 1970-01-01T00:00:00.000 according to [4].
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
Transform Denoised Speech	Transform Audio whose samples are Denoised Speech 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; https://bit.ly/3t3SNDT.
2. MPAI; Technical Specification: Artificial Intelligence Framework (MPAI-AIF) V1; https://bit.ly/3t3SNDT.
3. A Universally Unique IDentifier (UUID) URN Namespace; IETF RFC 4122; July 2005.
4. Date and Time on the Internet: Time Stamps; IETF RFC 3339; July 2002.
5. Universal Coded Character Set (UCS): ISO/IEC 10646; December 2020.
6. WAVE PCM sound file format, http://soundfile.sapp.org/doc/WaveFormat/.
7. ISO/IEC 14496-10; Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding.

8. ISO/IEC 23008-2; Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 2: High Efficiency Video Coding.
9. ISO/IEC 23094-1; Information technology – General video coding – Part 1: Essential Video Coding.
10. ISO/IEC 14496-12; Information technology – Coding of audio-visual objects – Part 12: ISO base media file format.
11. ZIP format, https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT.
12. Neural Network Exchange Format; https://www.khronos.org/registry/NNEF/specs/1.0/nnef-1.0.4.pdf; Khronos.
13. The JavaScript Object Notation (JSON) Data Interchange Format; https://datatracker.ietf.org/doc/html/rfc8259; IETF rfc8259; December 2017.
14. 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.
15. 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.
16. MPAI; The MPAI Statutes; https://mpai.community/statutes/
17. MPAI; The MPAI Patent Policy; https://mpai.community/about/the-mpai-patent-policy/.
18. Framework Licence of the Context-based Audio Enhancement Technical Specification (MPAI-CAE); https://mpai.community/standards/mpai-cae/framework-licence/

4.2 Informative References

The references provided here are for information purpose.

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

5 Use Case Architectures

5.1 Emotion-Enhanced Speech (EES)

5.1.1 Scope of Use Case

Emotion-Enhanced Speech (EES) 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.

5.1.2 I/O data

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*.

5.1.3 Implementation Architecture

The Emotion-Enhanced Speech Reference Model depicted in *Figure 2* supports two Modes or pathways enabling addition of emotional charge to an emotionless or neutral input utterance (Emotion-less Speech).

1. Along Pathway 1 (*Figure 2*), upper and middle left), a Model Utterance 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 (*Figure 2*), middle and lower left), neutral utterance Emotion-less Speech is input along with a specification of the desired Emotion. Speech Feature Analyser2 extracts Emotionless Speech Features from Emotionless Speech, which describe its initial state. These are sent to Emotion Feature Producer, which produces (emotional) Speech Features2 that can add the desired emotional charge to Emotionless Speech. These Speech Features2 are sent to Emotion Inserter2, which combines Emotionless Speech and the (emotional) Speech Features2 set. Speech with Emotion is then produced as output.

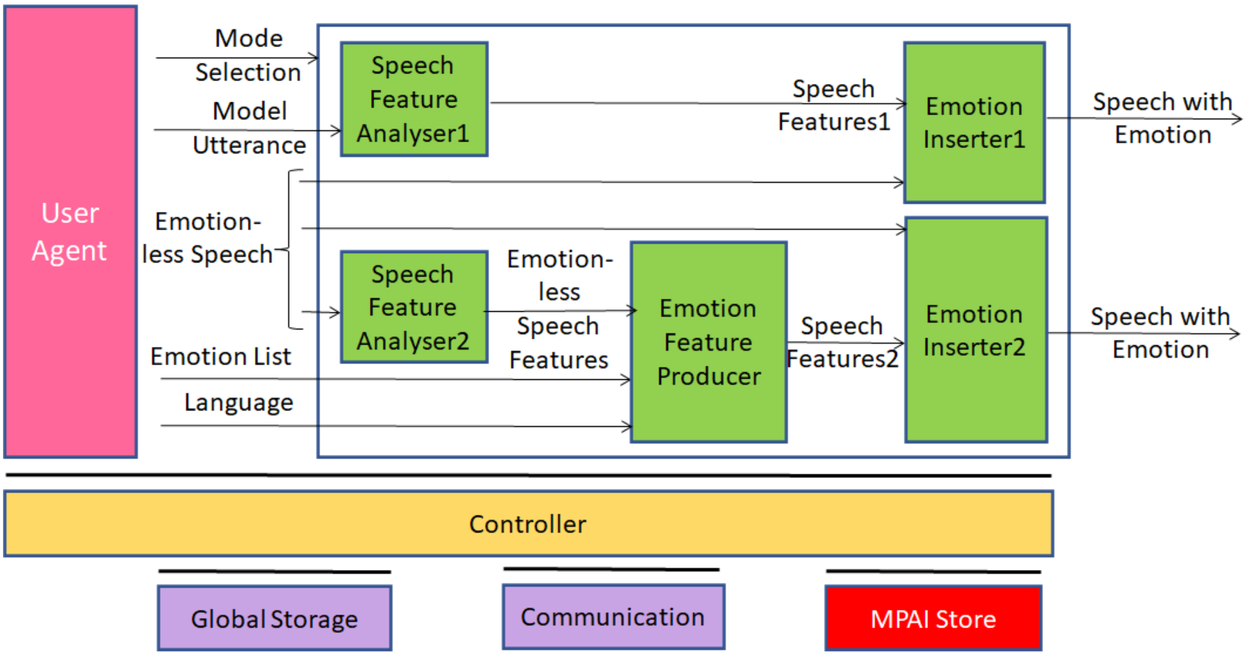

Figure 2 - Emotion-Enhanced Speech Reference Model

5.1.4 AI Modules

The AI Modules of *Figure 2* perform the functions described in *Table 3*.

Table 3 – AI Modules of Emotion-Enhanced Speech

AIM	Function
Speech Feature Analyser 1	Extracts Speech Features1 of a model emotional utterance and transfers them to the Emotion 1Inserter1.
Speech Feature Analyser2	Extracts Emotionless Speech Features of an emotionless input utterance, passing these to Emotion Feature Producer.

Emotion Feature Producer	Receives the Emotionless Speech Features produced by Speech Feature Analyser2 plus a list of Emotions to be added. (If the Degree of an Emotion is not specified, the Medium value is used.)
Emotion Inserter1	Integrates the (emotional) Speech Features1 with those of the Emotionless Speech input, yielding and delivering an emotionally modified utterance.
Emotion I2Inserter2	Integrates the (emotional) Speech Features2 with those of the Emotionless Speech input, yielding and delivering an emotionally modified utterance.

5.2 Audio Recording Preservation (ARP)

5.2.1 Scope of Use Case

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 I/O data

Table 4 gives the input and output data of Audio Recording Preservation.

Table 4 – I/O data of Audio Recording Preservation

Input	Comments
Preservation Audio File	See 6.3.16
Preservation Audio-Visual File	See 6.3.17
Output data	**Comments**
Preservation Master Files	See 6.3.18
Access Copy Files	See 6.3.1

5.2.3 Implementation Architecture

Figure 3 depicts the Audio Recording Preservation Reference Model.

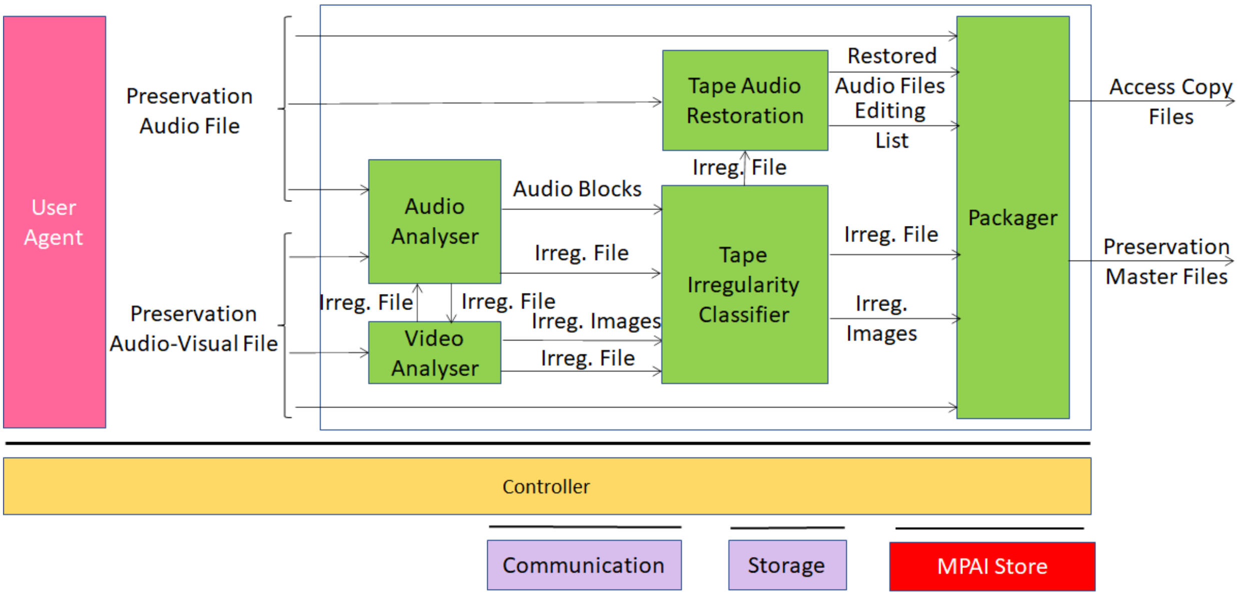

Figure 3 – Audio Recording Preservation Reference Model

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. The 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. The Video Analyser:
   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 Analyser and the offset between Preservation Audio File and the Preservation Audio-Visual File.
   d. Extracts the 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 Classifier.
4. The Audio Analyser:
   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 Analyser, extracts the Audio Blocks corresponding to each Irregularity detected and each Irregularity File received or detected.
   d. Sends the Audio Blocks and the Irregularity File related to all Irregularities to the Tape Irregularity Classifier.
5. The Tape Irregularity Classifier:
   a. Receives an Irregularity File with the corresponding Images and Audio Blocks.
   b. Classifies and selects the ones considered relevant.
   c. If the Irregularity was detected by the Video Analyser, the selected Irregularity File and the corresponding Irregularity Images are sent to the Packager.
6. The Tape Audio Restoration uses the Irregularity File to identify and restore portions of the Preservation Audio File.
7. The Packager 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.

5.2.4 AI Modules

The AIMs required by this Use Case are described in *Table 5*.

Table 5 – AI Modules of Audio Recording Preservation

AIM	Function
Audio Analyser	1. At the start, it calculates the offset between Preservation Audio and the Audio of the Preservation Audio-Visual File. 2. Detects Irregularities of the Preservation Audio File and produces the related Audio Blocks as well as the corresponding Irregularity File. 3. Sends the Irregularity File related to detected Irregularities to the Video Analyser. 4. Receives the Irregularity File detected by Video Analyser. 5. Extracts Audio Blocks corresponding to the Irregularities detected by Video Analyser. 6. At the end, it merges the produced Irregularity File with the one received from the Video Analyser, and it sends it with corresponding Audio Blocks to the Tape Irregularity Classifier.
Video Analyser	1. At the start, it detects Irregularities of the Preservation Audio-Visual File and produces the related Irregularity Images. 2. Sends the Irregularity File related to detected Irregularities to the Audio Analyser. 3. Receives the Irregularity File from the Audio Analyser. 4. Extracts the Irregularity Images corresponding to the Irregularities detected by Audio Analyser. 5. At the end, it merges the produced Irregularity File with the one received from the Audio Analyser, and it sends it with the corresponding Irregularity Images to the Tape Irregularity Classifier.
Tape Irregularities classifier	1. Receives all information from Audio Analyser and Video Analyser, classifies and selects the Irregularities of the Preservation Audio-Visual File and Preservation Audio File considered relevant. 2. Sends the Irregularity File related to the selected Irregularities and the corresponding Irregularity Images to the Packager. 3. Sends the Irregularity File related to the selected Irregularities to Tape Audio Restoration.
Audio Tape Restoration	1. Detects and corrects speed, equalisation and reading backwards errors in Preservation Audio File. 2. Sends Restored Audio Files and Editing List to Packager.
Packager	Produces Preservation Master Files and Access Copy Files.

5.3 Speech Restoration System (SRS)

5.3.1 Scope of Use Case

This Use Case addresses the need for restoration of a Damaged Segment, i.e., a segment containing speech which may be damaged in its entirety or only in part.

Restoration is carried out by synthesizing 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 I/O Data

Table 6 gives the input and output data of Speech Restoration System.

Table 6 – 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 Segment	See *Table 1*.

5.3.3 Implementation Architecture

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

.

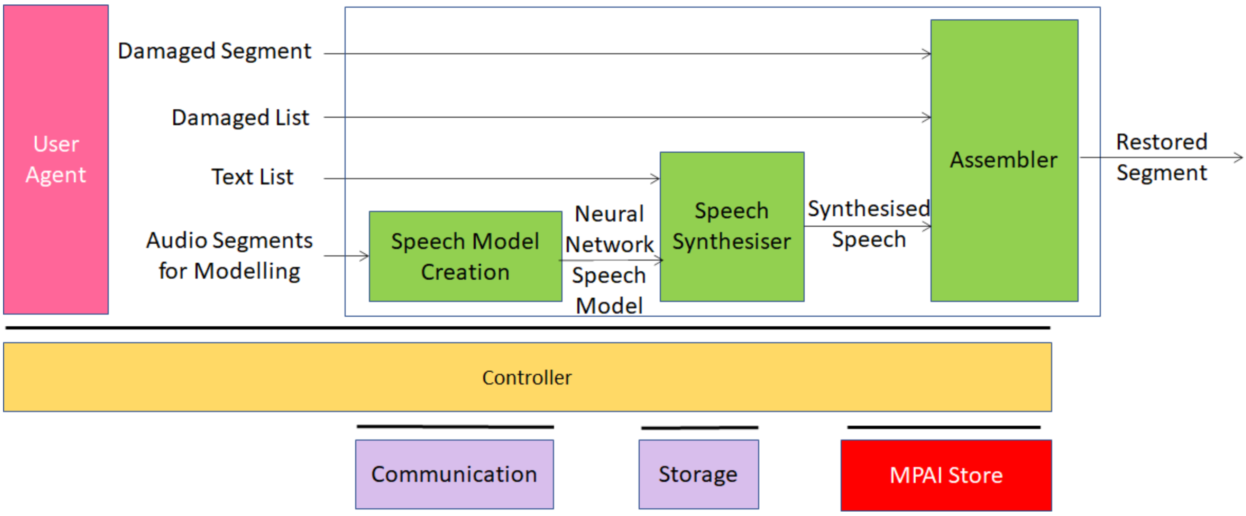

Figure 4 - Speech Restoration System (SRS) Reference Model

In the SRS use case, the entire Damaged Segment can be replaced by a synthesized 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 Synthesiser 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 Synthesiser produces synthetic replacements for each damaged section (or for Damaged Segment as a whole) and passes the replacement(s) to Assembler.
3. Assembler 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. Assembler 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.

5.3.4 AI Modules

The AIMs required by the Speech Restoration System Use Case are described in *Table 7*

Table 7 - AI Modules of Audio Recording Preservation

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 Synthesiser.
Speech Synthesiser	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 Assembler as an ordered list.
Assembler	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.

5.4 Enhanced Audioconference Experience (EAE)

5.4.1 Scope of Use Case

The EAE 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. The use case 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 quality. EAE also extracts the spatial attributes 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 attributes are represented in a well-defined Audio Scene Geometry metadata format and packaged in a format that is amenable to further processing for efficient delivery and further processing. The coding and compression of the extracted speech signals as well as their reconstruction/representation at the receiver side are outside the scope of this use case.

5.4.2 I/O data

Table 8 shows the input and output data for the Enhanced Audioconference Experience workflow.

Table 8 – I/O data of Enhanced Audioconference Experience

Inputs	**Comments**
Microphone Array Audio	See 6.3.11
Microphone Array Geometry	See 6.3.12
Outputs	**Comments**
Multi-channel Audio + Audio Scene Geometry	See *Table 1*.

5.4.3 Implementation Architecture

Figure 5 shows the Workflow for the EAE.

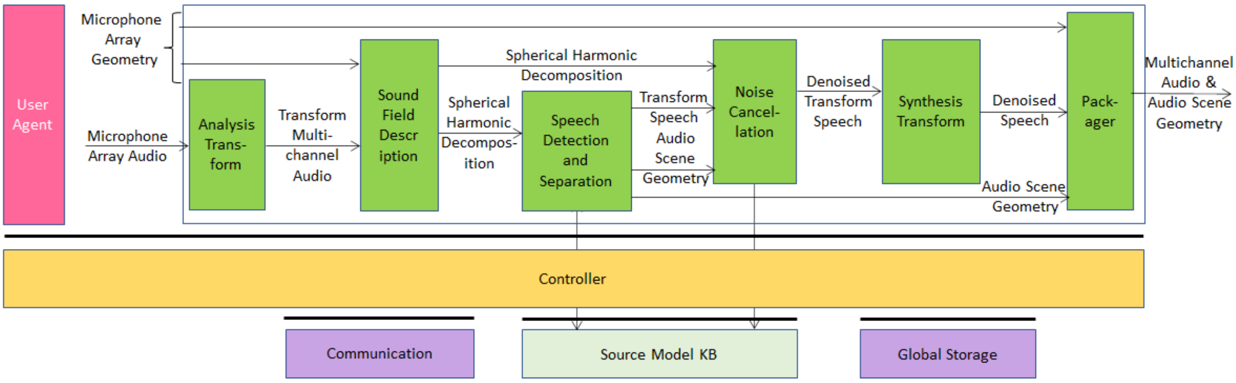
User Agent
Microphone Array Geometry
Microphone Array Audio
Analysis Trans-form
Transform Multi-channel Audio
Sound Field Descr iption
Spherical Harmonic Decomposition
Spherical Harmonic Decompos-ition
Speech Detection and Separation
Transform Speech Audio Scene Geometry
Noise Cancel-lation
Denoised Transform Speech
Synthesis Transform
Denoised Speech
Audio Scene Geometry
Pack-ager
Multichannel Audio & Audio Scene Geometry
Controller
Communication
Source Model KB
Global Storage

Figure 5 - Enhanced Audioconference Experience Reference Model

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. **Analysis Transform** transforms the Multichannel 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 Analysis Transform AIM into the spherical frequency domain [20]. 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 the Audio Scene Geometry in JSON format 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.
4. **Noise Cancellation** eliminates background noise and reverberation which reduce the audio quality. If environmental conditions do not substantially add ambient noise to the desired speech, this AIM acts as a Passthrough AIM.
    a. It receives Transform Speech from **Speech Detection and Separation** AIM and Acoustic Scene Metadata which includes attributes pertaining to the Audio Block being processed for denoising, and SHD coefficients.
    b. It uses **Source Model KB**. The output of Noise Cancellation AIM is Denoised Transform Speech as an (M × S) complex-valued data structure which will in the next stage be processed through **Synthesis Transform** AIM to obtain Denoised Speech.
5. **Synthesis Transform** receives Denoised Transform Speech and outputs Denoised Transform Speech (F × S) by applying the inverse of the analysis transform.
6. **Packager**:
    a. Receives Denoised Speech and Audio Scene Geometry.
    b. Packages the Multichannel Audio stream and the Audio Scene Geometry.
    c. Produces one interleaved stream which contains separated Multichannel Speech Streams and Audio Scene Geometry.

5.4.4 AI Modules

The AIMs required by the Enhanced Audioconference Experience are given in *Table 9*

Table 9 - AIMs of Enhanced Audioconference Experience

AIM	Function
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 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	Removes noise and/or suppresses reverberation in the Transform Speech producing Denoised Transform Speech.

Synthesis Transform	Effects inverse transform of Denoised Transform Speech producing Denoised Speech ready for packaging.
Packager	Packages Denoised Speech and the Audio Scene Geometry.

6 AIMs

6.1 AIM Interoperability

To the extent possible, AIM input and output data are specified in a way that is neutral to the technology used to implement the AIM internals. In some cases, however, AIM input and output data of strongly depend on whether the technology used is data processing or Artificial Intelligence. If an AIM is based on, e.g., a neural network, it will need either (1) a usable neural model whose training has included specifiable features, or (2) a precise specification of the features themselves plus an adequate training corpus, so that the AIM using that data can create its own usable model.

6.2 AIMs and their data

6.2.1 Emotion Enhanced Speech

Table 10 – CAE-EES AIMs and their data

AIM	Input Data	Output Data
Speech features Analyser1	Model Utterance	Speech Features1
Speech features Analyser2	Emotionless Speech	Emotionless Speech Features
Emotion Feature Producer	Emotionless Speech Features Emotion List Language	Speech Features2
Emotion Inserter1	Emotionless Speech Speech Features1	Speech with Emotion
Emotion Inserter2	Emotionless Speech Speech Features2	Speech with Emotion

6.2.2 Audio Recording Preservation (ARP)

Table 11 – CAE-ARP AIMs and their data

AIM	Input Data	Output Data
Audio Analyser	Preservation Audio File Preservation Audio-Visual File Irregularity File	Audio Blocks Irregularity File
Video analyser	Preservation Audio-Visual File Irregularity File	Irregularity File Irregularity Images
Tape Irregularity classifier	Audio Blocks Irregularity Images Irregularity File	Irregularity File Irregularity Images
Tape Audio Restoration	Irregularity File Preservation Audio File	Editing List Restored Audio Files
Packager	Preservation Audio File Restored Audio Files Editing List	Access Copy Files Preservation Master Files

	Irregularity File Irregularity Images Preservation Audio-Visual File	

6.2.3 Speech Restoration System (SRS)

Table 12 – CAE-SRS AIMs and their data

AIM	Input Data	Output Data
Speech Model Creation	Audio Segments for Modelling	Neural Network Speech Model
Speech Synthesiser	Text List Neural Network Speech Model	Synthesised Speech
Assembler	Damaged Segments Damaged List	Restored Segment

6.2.4 Enhanced Audioconference Experience (EAE)

Table 13 – CAE-EAE AIMs and their data

AIM	Input Data	Output Data
Analysis Transform	Multichannel Audio	Transform Multichannel Audio
Sound field Description	Transform Multichannel Audio Geometry Information	Spherical Harmonic Decomposition
Speech Detection and Separation	Spherical Harmonic Decomposition	Transform Speech Audio Scene Geometry
Noise Cancellation	Spherical Harmonic Decomposition Transform Speech Audio Scene Geometry	Denoised Transform Speech
Synthesis Transform	Denoised Transform Speech	Denoised Speech
Packager	Denoised Speech Audio Scene Geometry	Multichannel Audio Audio Scene Geometry

6.3 Data Formats

Table 14 lists all data formats specified in this Technical Specification.

Table 14 – Data formats

Data Format Name	Subsection	Use Case
Access Copy Files	6.3.1	ARP
Audio Scene Geometry	6.3.2	EAE
Damaged List	6.3.3	SRS
Denoised Speech	6.3.4	SRS
Editing List	6.3.5	ARP
Emotion	6.3.6	EES
Emotionless Speech	6.3.7	EES
Interleaved Multichannel Audio	6.3.8	EAE

Irregularity File	6.3.9	ARP
Irregularity Image	6.3.10	ARP
Microphone Array Audio	6.3.11	EAE
Microphone Array Geometry	6.3.12	EAE
Mode Selection	6.3.13	EES
Multichannel Audio	6.3.14	EAE
Neural Network Speech Model	6.3.15	SRS
Preservation Audio File	6.3.16	ARP
Preservation Audio-Visual File	6.3.17	ARP
Preservation Master Files	6.3.18	ARP
Source Dictionary	6.3.19	EAE
Source Model KB Query Format	6.3.20	EAE
Speech Features1	6.3.21	EES
Speech Features2	6.3.21	EES
Spherical Harmonics Decomposition	6.3.22	EAE
Transform Denoised Speech	6.3.23	EAE
Transform Speech	6.3.24	EAE
Transform Multichannel Audio	6.3.25	EAE
Video	6.3.26	ARP

6.3.1 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 [11].
4. The Irregularity File.

6.3.2 Audio Scene Geometry

Syntax and Semantics are given below

6.3.2.1 Syntax

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Audio Scene Geometry",
  "type": "object",
  "properties": {
       "BlockIndex": {
       "type": "integer"
       },
    "BlockStart": {
       "type": "integer"
       },
    "BlockEnd": {
       "type": "integer"
    },
    "SpeechCount": {
       "type": "integer"
    },
    "SourceDetectionMask": {
       "type": "array",
        "items": {
               "type": "uint8",
        }
    },
    "SpeechList": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "SpeechID": {
            "type":"string",
            "format":"uuid"
          },
          "ChannelID": {
            "type": "integer"
          },
          "AzimuthDirection": {
            "type": "float"
          },
          "ElevationDirection": {
            "type": "float"
          },
          "DistanceFlag": {
            "type": "boolean",
          },
          "Distance": {
            "type": "float"
          }
        }
      },
      "minItems": 1,
      "uniqueItems": true,
      "required": ["SpeechID","ChannelID","AzimuthDirection","ElevationDirection",
"DistanceFlag","Distance" ]
    }
  },
  "required": ["BlockIndex","BlockStart", "BlockEnd", "SpeechCount", "SourceDetectionMask",
"SpeechList"]
}
```

6.3.2.2 Semantics

Name	*Definition*
BlockIndex	Block ID starting from 0 and incremented by 1 for each consecutive audio block processed by the system. (long integer)
BlockStart	Unix timestamp in ms from epoch. (long integer)
BlockEnd	Unix timestamp in ms from epoch. (long integer)
SpeechCount	Number of speech sources in the scene. (uint8)
SpeechList	A list containing Speech attributes.
SpeechList:Speech	A nested JSON block describing a speech source with the following elements. SpeechID: Speech source ID ([7], uuid) ChannelID: Channel ID (uint8) AzimuthDirection: Azimuth direction in degrees. (float) ElevationDirection: Elevation direction in degrees. (float) Distance: Distance in m. (float32) DistanceFlag: 0: Valid, 1: NonValid. (uint8)
SourceDetectionMask	A binary mask that represents the indices of the transform coefficients that will be used in denoising. (uint8)

6.3.3 Damaged List

Syntax and Semantics are given below.

6.3.3.1 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"]
}
```

6.3.3.2 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)

6.3.4 Denoised Speech

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

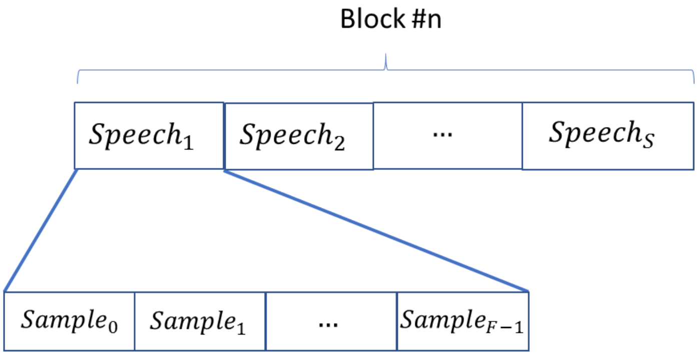

Figure 6 – Denoised speech signals after synthesis transform

6.3.5 Editing List

A JSON file encoded in UTF-8 according to [5].

6.3.5.1 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"]
}
```

6.3.5.2 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.

Name	*Definition*
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 [14,15]. The association with `OriginalSpeedStandard` shall be compliant to the values indicated in [14,15]. 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`.
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 [14,15]. The association with `AppliedSpeedStandard` shall be compliant to the values indicated in [14,15].

6.3.6 Emotion

The Syntax and Semantics of Emotion are given by the following clauses.

6.3.6.1 Syntax

Human Emotion is represented by.

```
{
   "$schema":"http://json-schema.org/draft-07/schema",
   "definitions":{
      "emotionType":{
         "type":"object",
         "properties":{
            "emotionDegree":{
```

```
            "enum": ["High", "Medium", "Low"]
          },
          "emotionName":{
             "type":"number"
          },
          "emotionSetName":{
             "type":"string"
          }
       }
    },
    "type":"object",
    "properties":{
       "primary":{
          "$ref":"#/definitions/emotionType"
       },
       "secondary":{
          "$ref":"#/definitions/emotionType"
       }
    }
 }
```

6.3.6.2 Semantics

Name	*Definition*
emotionType	Specifies the Emotion that the input carries.
emotionDegree	Specifies the Degree of Emotion as one of "Low," "Medium," and "High."
emotionName	Specifies the ID of an Emotion listed in *Table 15*
emotionSetName	Specifies the name of the Emotion set which contains the Emotion. Emotion set of *Table 15* is used as a baseline, but other sets are possible.

Emotions are expressed vocally through combinations of prosody (pitch, rhythm, and volume variations); separable speech effects (such as degrees of voice tension, breathiness, etc.); and vocal gestures (laughs, sobs, etc.).

Table 15 gives the MPAI standardised three-level Basic Emotion Set partly based on Paul Eckman [19]:

1. The EMOTION CATEGORIES column specifies the categories using nouns.
2. The GENERAL ADJECTIVAL column gives adjectival labels for general or basic emotions within a category.
3. The SPECIFIC ADJECTIVAL column gives labels for more specific (sub-categorized) emotions in the relevant category, often (but not always) representing differing degrees of the basic emotion.

Table 16 provides the semantics for each label in the GENERAL ADJECTIVAL and SPECIFIC ADJECTIVAL columns.

An Implementer wishing to extend or replace *Table 15* is requested to do the following:

1. Create a new *Table 15* where:
   a. Proposed additions are clearly marked (in case of extension).
   b. All Emotions and levels (up to 3) are listed (in case of replacement).
2. Create a new *Table 16* where:
   a. the semantics of the Emotions is added to the semantics of the existing emotions (in case of extension).

b. is provided (in case of replacement).

The semantics provided should have a level of details comparable to the semantics given in the current *Table 16*

3. Submit both tables to the MPAI Secretariat.

The appropriate MPAI Development Committee will examine the proposed extension or replacement. Only the adequacy of the proposed new tables in terms of clarity and completeness will be considered. In case the new tables are not clear or complete, a revision of the tables will be requested.

The accepted External Emotion Set will be identified as proposed by the submitter and reviewed by the appropriate MPAI Committee and posted to the MPAI web site.

Table 15 - Basic Emotion Set

EMOTION CATEGORIES	**GENERAL ADJECTIVAL**	**SPECIFIC ADJECTIVAL**
ANGER	anger	furious irritated frustrated
APPROVAL, DISAPPROVAL	admiring/approving disapproving indifferent	awed contemptuous
AROUSAL	aroused/excited/energetic	cheerful playful lethargic sleepy
ATTENTION	attentive	expectant/anticipating thoughtful distracted/absent-minded vigilant hopeful/optimistic
BELIEF	credulous	sceptical
CALMNESS	calm	peaceful/serene resigned
DISGUST	disgust	loathing
FEAR	fearful/scared	terrified anxious/uneasy
HAPPINESS	happy	joyful content delighted amused
HURT	hurt jealous	
INTEREST	interested	fascinated curious bored
PRIDE/SHAME	proud ashamed	guilty/remorseful/sorry embarrassed
SADNESS	sad	lonely grief-stricken

		discouraged depressed disappointed
SOCIAL DOMINANCE, CONFIDENCE	arrogant confident submissive	
SURPRISE	surprised	astounded startled
UNDERSTANDING	comprehending	uncomprehending bewildered/puzzled

Table 16 - Semantics of the Basic Emotion Set

ID	Emotion	Meaning
1	admiring/approving	emotion due to perception that others' actions or results are valuable
2	amused	positive emotion combined with interest (cognitive)
3	anger	emotion due to perception of physical or emotional damage or threat
4	anxious/uneasy	low or medium degree of fear, often continuing rather than instant
5	aroused/excited/energetic	cognitive state of alertness and energy
6	arrogant	emotion communicating social dominance
7	astounded	high degree of surprised
8	attentive	cognitive state of paying attention
9	awed	approval combined with incomprehension or fear
10	bewildered/puzzled	high degree of incomprehension
11	bored	not interested
12	calm	relative lack of emotion
13	cheerful	energetic combined with and communicating happiness
14	comprehending	cognitive state of successful application of mental models to a situation
15	confident	emotion due to belief in ability
16	contemptuous	high degree of disapproval
17	content	medium or low degree of happiness, continuing rather than instant
18	credulous	cognitive state of conformance to mental models of a situation
19	curious	interest due to drive to know or understand
20	delighted	high degree of happiness, often combined with surprise
21	depressed	high degree of sadness, continuing rather than instant, combined with lethargy (see AROUSAL)
22	disappointed	sadness due to failure of desired outcome
23	disapproving	not approving
24	discouraged	sadness combined with frustration
25	disgust	emotion due to urge to avoid, often due to unpleasant perception or disapproval
26	distracted/absent-minded	not attentive to present situation due to competing thoughts

27	embarrassed	shame due to consciousness of violation of social conventions
28	expectant/anticipating	attentive to (expecting) future event or events
29	fascinated	high degree of interest
30	fearful/scared	emotion due to anticipation of physical or emotional pain or other undesired event or events
31	frustrated	angry due to failure of desired outcome
32	furious	high degree of anger
33	grief-stricken	sadness due to loss of an important social contact
34	guilty/remorseful/sorry	shame due to consciousness of hurting or damaging others
35	happy	positive emotion, often continuing rather than instant
36	hopeful/optimistic	expectation of good outcomes
37	hurt	emotion due to perception that others have caused social pain or embarrassment
38	indifferent	neither approving nor disapproving
39	interested	cognitive state of attentiveness due to salience or appeal to emotions or drives
40	irritated	low or medium degree of anger
41	jealous	emotion due to perception that others are more fortunate or successful
42	joyful	high degree of happiness, often due to a specific event
43	lethargic	not aroused
44	loathing	high degree of disgust
45	lonely	sadness due to insufficient social contact
46	peaceful/serene	calm combined with low degree of happiness
47	playful	energetic and communicating willingness to play
48	proud	emotion due to perception of positive social standing
49	resigned	calm due to acceptance of failure of desired outcome, often combined with low degree of sadness
50	sad	negative emotion, often continuing rather than instant, often associated with a specific event
51	sceptical	not credulous
52	sleepy	not aroused due to need for sleep
53	startled	surprised by a sudden event or perception
54	submissive	emotion communicating lack of social dominance
55	surprised	cognitive state due to violation of expectation
56	terrified	high degree of fear
57	thoughtful	attentive to thoughts
58	uncomprehending	not comprehending
59	vigilant	high degree of expectation or attentiveness

6.3.7 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.

6.3.8 Interleaved Multichannel Audio

A data structure containing between 4 and 256 time-aligned interleaved Audio Channels and organised in blocks as depicted in *Figure 7*

.

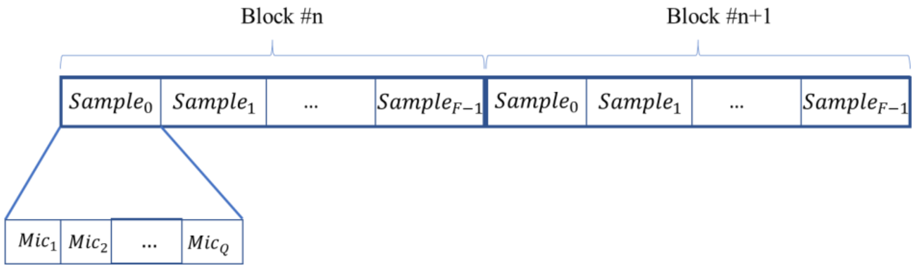

Figure 7 - Microphone Array Signals input sample ordering

6.3.9 Irregularity File

A JSON file encoded in UTF-8 according to [5].

6.3.9.1 Syntax

The JSON schema of the Irregularity ID 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"]
                    },
                    "ImageURI": {
                        "type": "string",
                        "format": "uri"
                    },
                    "AudioBlockURI": {
                        "type": "string",
                        "format": "uri"
                    }
                }
            },
            "minItems": 1,
            "uniqueItems": true,
            "required": ["IrregularityID", "Source", "TimeLabel"]
        }
    },
    "required": ["Irregularities"]
}
```

6.3.9.2 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.
AudioBlockURI	*URI* of the Audio Block related to an Irregularity. It is only used in the message between Audio Analyser and Tape Irregularity Classifier.
ImageURI	*URI* of the Image related to an Irregularity. It is only used in the messages between Audio Analyser, Tape Irregularity Classifier, and Packager.
IrregularityType	Class of an Irregularity (see values in following Tables). It is only used in the messages between Tape Irregularity Classifier, Packager and Tape Audio restoration.

Table 17 - 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.

Code	*Name*	*Definition*
`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 18 - 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.
`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.

6.3.10 Irregularity Image

An Image corresponding to an Irregularity.

6.3.11 Microphone Array Audio

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

6.3.12 Microphone Array Geometry

The Syntax and Semantics of the Microphone Array Geometry are given below.

6.3.12.1 Syntax

```
{
        "$schema": "http://json-schema.org/draft-07/schema#",
        "title": "Microphone Array Geometry",
        "type": "object",
        "properties": {
                "MicrophoneArrayType": {
                        "type": "integer"
                },
                "MicrophoneArrayScat": {
                        "type": "integer"
                },
                "MicrophoneArrayFilterURI": {
                        "type": "string",
                        "format": "uri"
                },
                "SamplingRate": {
                        "type": "integer"
                },
                "SampleType": {
                        "type": "integer"
                },
                "BlockSize": {
                        "type": "integer"
                },
                "NumberofMicrophones": {
                        "type": "integer"
                },
                "MicrophoneList": {
                        "type": "array",
                        "items": {
                                "type": "object",
                                "properties": {
                                        "xCoord": {
                                                "type": "float"
                                        },
                                        "yCoord": {
                                                "type": "float"
                                        },
                                        "zCoord": {
                                                "type": "float"
                                        },
                                        "directivity": {
                                                "type": "integer"
                                        },
                                        "micxLookCoord": {
                                                "type": "float"
                                        },
                                        "micyLookCoord": {
                                                "type": "float"
                                        },
                                        "miczLookCoord": {
                                                "type": "float"
                                        }
                                }
                        },
                        "minItems": 4,
                        "uniqueItems": true,
                        "required": ["xCoord", "yCoord", "zCoord", "directivity", "micxLookCoord",
"micyLookCoord", "miczLookCoord"]
                },
                "MicrophoneArrayLookCoord": {
                        "type": "object",
                        "properties": {
                                "xLookCoord": {
                                        "type": "float"
                                },
                                "yLookCoord": {
                                        "type": "float"
                                },
                                "zLookCoord": {
                                        "type": "float"
                                }
                        },
                        "uniqueItems": true,
                        "required": ["xLookCoord", "yLookCoord", "zLookCoord"]
                }
        },
        "required": ["MicrophoneArrayType", "MicrophoneArrayScat", "MicrophoneArrayFilterURI",
"SamplingRate", "SampleType", "BlockSize", "NumberofMicrophones", "MicrophoneList",
"MicrophoneArrayLookCoord"]
}
```

6.3.12.2 Semantics

Name	*Definition*
MicrophoneArrayType	Indicates the type of microphone array positioning such as 0:Spherical, 1:Circular, 2:Planar, 3:Linear, 4:Other. (uint8)
MicrophoneArrayScat	Indicates the type of the microphone array (0:Rigid, 1:Open, 2:Other). (uint8)
MicrophoneArrayFilter URI	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)
SamplingRate	Sampling rate used by the microphone array (0:16kHz, 1:24kHz, 2:32kHz, 3:44.1kHz, 4:48kHz, 5:64kHz, 6:96kHz, 7:192kHz). (uint8)
SampleType	Sample type (0:16bit, 1:24bit, 2:32bit). (uint8)
BlockSize	Block Size (0:64,1:128,2:256,3:512,4:1024,5:2048, 6:4096). (uint8)
NumberofMicrophones	Represents the number of Microphones. (uint8)

Name	*Definition*
MicrophoneList	A list containing Microphone attributes.
MicrophoneList:Microphone	A nested JSON block describing a single microphone element with the following elements. xCoord: x position of the microphone in m. (float) yCoord: y position of the microphone in m.(float) zCoord: z position of the microphone in m. (float) directivity: The directivity pattern of the specific microphone, 0: omnidirectional, 1: figure of eight, 2: cardioid, 3: supercardioid, 4: hypercardioid, 5: other (uint8) micxLookCoord: x component of the vector representing the look direction of the microphone in m. (float) micyLookCoord: y component of the vector representing the look direction of the microphone in m. (float) miczLookCoord: z component of the vector representing the look direction of the microphone. (float)
MicrophoneArrayLookCoord	xLookCoord: x component of the vector representing the look direction of the microphone array. (float) yLookCoord: y component of the vector representing the look direction of the microphone array. (float) zLookCoord: z component of the vector representing the look direction of the microphone array. (float)

6.3.13 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 Analyser1 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 Analyser2 module will analyse Emotionless Speech and pass extracted Emotionless Speech Features along with a specification of the desired emotion to Emotion Feature Producer. These modules will produce (emotional) Speech Features2 and pass them to an Emotion Inserter2 module capable of combining Emotionless Speech and (emotional) Speech Features2 to produce Speech with Emotion. See Section 5.1.3.

6.3.14 Multichannel Audio Stream

Interleaved Multichannel Audio packaged with Time Code according to the structure of *Figure 8* specified in *Table 19*.

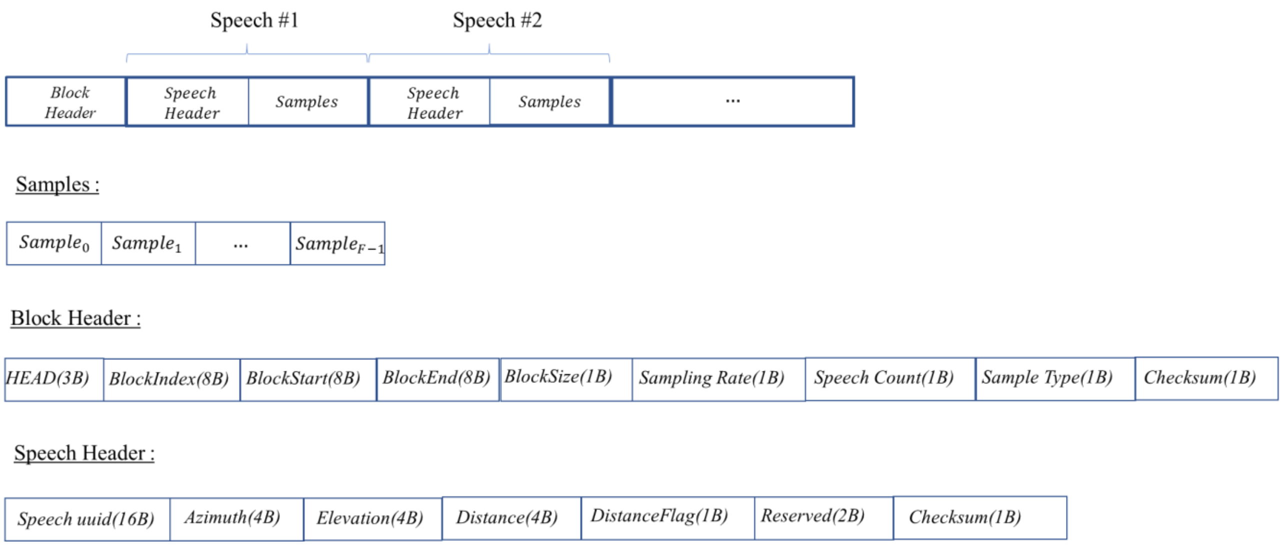

Figure 8 – Multichannel speech stream packages

Table 19 – Multichannel separated speech signals packaging

Label	**Size**	**Description**
HEAD	3 Bytes	Comprises 3 characters ‘EAE’.
BlockIndex	8 Bytes	Copy of the value BlockIndex in Metadata, indicating the timing order of the output block.
BlockStart	8 Bytes	Copy of the value BlockStart in Metadata.
BlockEnd	8 Bytes	Copy of the value BlockEnd in Metadata.
BlockSize	1 Byte	Copy of the value BlockSize in Geometry Information.
Sampling Rate	1 Byte	Copy of the value SamplingRate in Geometry Information.
Speech Count	1 Byte	Copy of the value SpeechCount in Metadata.
Sample Type	1 Byte	Copy of the value SampleType in Geometry Information.
Checksum	1 Byte	Checksum is calculated by summing the block and speech header bytes and taking its modulo by 256.
Speech uuid	16 Bytes	Copy of the value Speech uuid in Metadata.
Azimuth	4 Bytes	Copy of the value Speech:Azimuth in Metadata.
Elevation	4 Bytes	Copy of the value Speech:Elevation in Metadata.
Distance	4 Bytes	Copy of the value Speech:Distance in Metadata.
DistanceFlag	1 Byte	Copy of the value Speech:DistanceFlag in Metadata.

6.3.15 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 Khronos Neural Network Exchange Format [12].

6.3.16 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.

6.3.17 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.

6.3.18 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.

6.3.19 Source Dictionary

A dictionary of real-valued functions sampled on the sphere corresponding to plane wave acoustic sources [20]. The plane waves are localized at the nodes of a spherical grid with the first *Spherical Grid Resolution,* evaluated at the nodes of the spherical grid with the second *Spherical Grid Resolution.*

6.3.20 Source Model KB Query Format

The Source Model KB is a 2D Source Dictionary. It is queried with *Spherical Grid Resolutions*. The response is a 2D *Source Dictionary.*

6.3.21 Speech Features

Speech Features1 and Speech Features2 are digitally represented as follows.

```
{
    "$schema": "http://json-schema.org/draft-07/schema",
    "title": "SpeechFeatures1",
    "type": "object",
    "properties": {
        "intonations": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "pitch": {
                        "type": "number"
                    },
                    "intensity": {
                        "type": "number"
                    },
                    "duration": {
                        "type": "number"
                    }
                },
                "required": ["pitch", "intensity", "duration"]
            }
        },
        "unit": {
            "type": "string"
        },
    },
    "required": ["intonations", "unit"]
}
```

```
{
    "$schema": "http://json-schema.org/draft-07/schema",
    "title": "SpeechFeatures2",
    "type": "array",
    "items": {
        "type": "number"
    }
}
```

6.3.21.1 Semantics

Name	*Definition*
SpeechFeatures1	Indicates intonation elements extracted from the input speech, specifically pitch, duration, and intensity.
SpeechFeatures2	Indicates specifically neural-network-based characteristic elements extracted from the input speech by a 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* 20 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 (6.3.5.2).

Table 20 – 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

6.3.22 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 coefficient of order l and degree m corresponding to the k-th transform coefficient respectively.

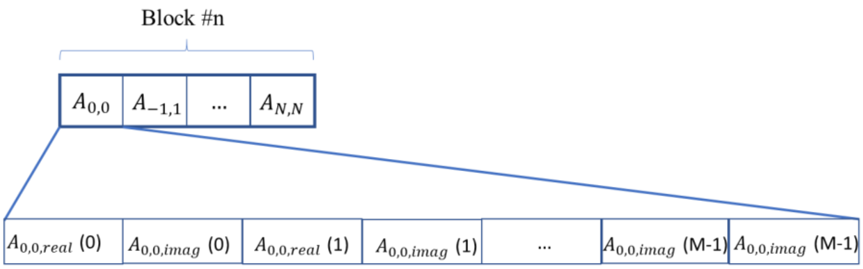

Figure 9 – Spherical Harmonics Decomposition of sound field

6.3.23 Transform Denoised Speech

Transform Audio whose samples are Denoised Speech samples.

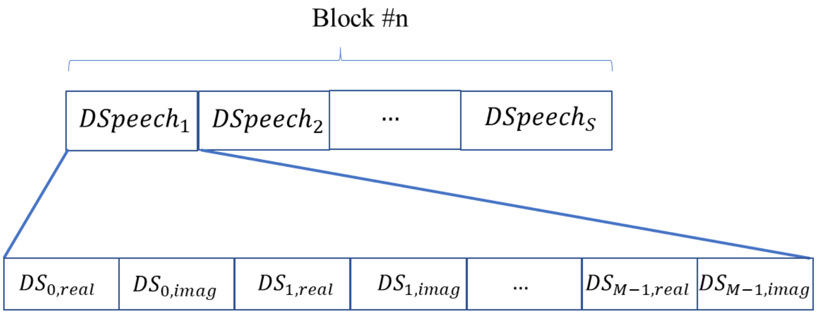

Figure 10 – Denoised transform domain speech signals

6.3.24 Transform Speech

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 floating point values.

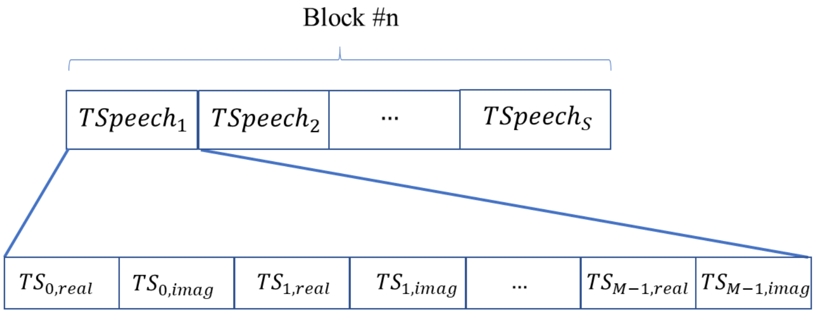

Figure 11 - Transform domain separated speech signals

6.3.25 Transform Multichannel Audio

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

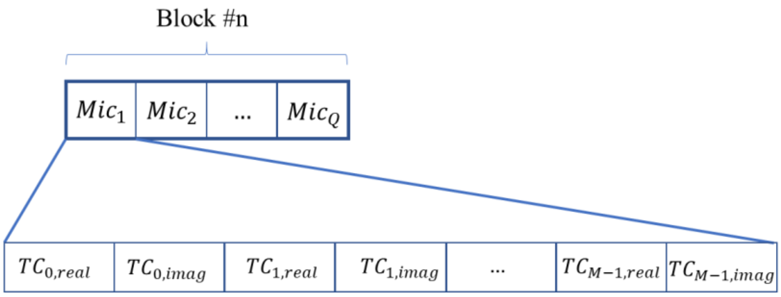

Figure 12 – Transform Multichannel Audio

6.3.26 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 50-120 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 [7], MPEG-H HEVC [8], MPEG-5 EVC [9].

Annex 1 - MPAI-wide terms and definitions (Normative)

The Terms used in this standard whose first letter is capital and are not already included in *Table 1* are defined in *Table 21*.

Table 21 – 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.
Component	One of the 7 AIF elements: Access, Communication, Controller, Internal Storage, Global Storage, MPAI Store, and User Agent.
Conformance	The attribute of an Implementation of being a correct technical Implementation of a Technical Specification.
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.
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

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

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

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

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

Copyright

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

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

Annex 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 is a high-level description of the MPAI ecosystem operation applicable to fully conforming MPAI implementations:

1. MPAI establishes and controls the not-for-profit MPAI Store (step 1).
2. MPAI appoints Performance Assessors (step 2).
3. MPAI publishes Standards (step 3).
4. Implementers submit Implementations to Performance Assessors (step 4).
5. If the Implementation Performance is acceptable, Performance Assessors inform Implementers (step 5a) and MPAI Store (step 5b).
6. Implementers submit Implementations to the MPAI Store (step 6); The Store Tests Conformance and security of the Implementation.
7. Users download Implementations (step 7).

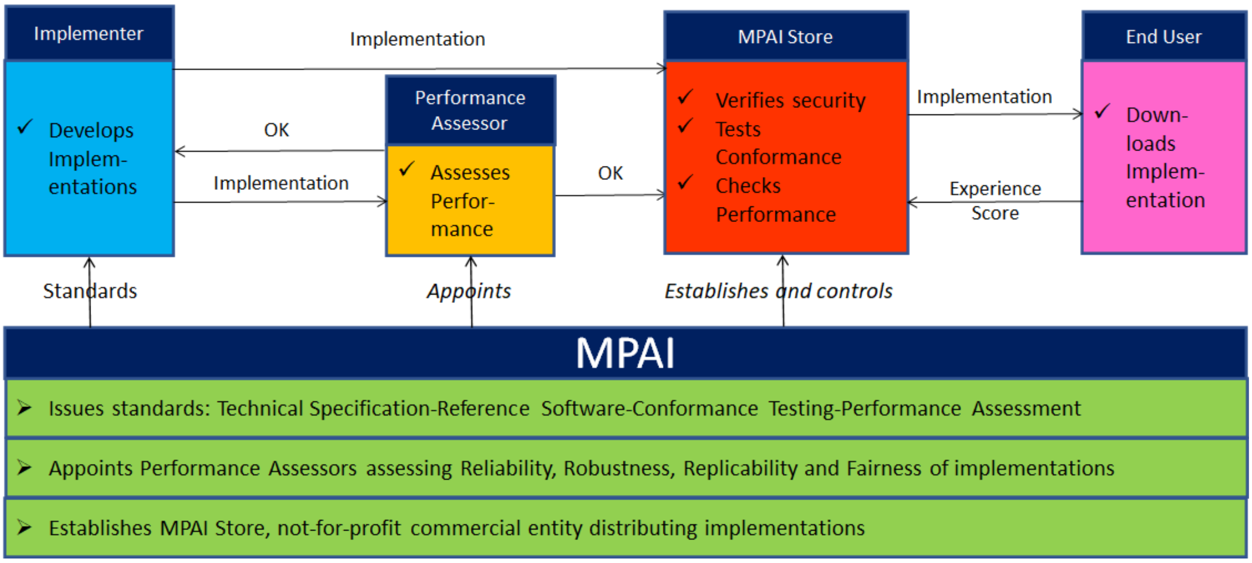
Implementer
✓ Develops Implem-entations
Implementation
OK
Implementation
Performance Assessor
✓ Assesses Perfor-mance
OK
MPAI Store
✓ Verifies security
✓ Tests Conformance
✓ Checks Performance
Implementation
Experience Score
End User
✓ Down-loads Implem-entation
Standards
Appoints
Establishes and controls
MPAI
➢ Issues standards: Technical Specification-Reference Software-Conformance Testing-Performance Assessment
➢ Appoints Performance Assessors assessing Reliability, Robustness, Replicability and Fairness of implementations
➢ Establishes MPAI Store, not-for-profit commercial entity distributing implementations

Figure 13 – The MPAI ecosystem operation

Annex 4 – Patent Declarations

The MPAI Context-based Audio Enhancement (MPAI-CAE) Technical Specification has been developed according to the process outlined in the MPAI Statutes [16] and the MPAI Patent Policy [17].
The following entities have agreed to license their standard essential patents reading on the MPAI Context-based Audio Enhancement (MPAI-CAE) Technical Specification according to the MPAI-CAE Framework License [18]:

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"
        },
        {
                "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"
        },
        {
                "IrregulatityID": "3cdc2973-e95e-4125-acb7-121ad89067ef",
                "Source": "a",
                "TimeLabel": "00:04:89.020",
                "IrregularityType": "esv"
        }]
}
```

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 Speech Features 1

```
{
    "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 Speech Features 2

```
[
    1.456,
    5.1289,
    0.12,
    12345.54378,
    12389943.2837,
    58.29
]
```

Annex 6 - AIW and AIM Metadata of CAE-EES

6.1 AIW Metadata

```
{
        "$schema": "https://json-schema.org/draft/2020-12/schema",
        "$id": "https://mpai.community/standards/resources/MPAI-AIF/V1/AIW-AIM-
metadata.schema.json",
        "title": "EES v1 AIW/AIM metadata",
        "Identifier": {
                "ImplementerID": 100,
                "Specification": {
                "Standard": "MPAI-CAE",
                "AIW": "CAE-EES",
                "AIM": "CAE-EES",
                "Version": "1"
                }
        },
        "Description":"This AIW implements EES application of MPAI-CAE",
        "Types":[
                {
                        "Name": "Speech_t",
                        "Type": "uint32[]"
                },
                {
                        "Name": "Emotion_t",
                        "Type": "uint8"
                },
                {
                        "Name": "EmotionList_t",
                        "Type": "Emotion_t[]"
                },
                {
                        "Name": "Text_t",
                        "Type": "{byte[] One_Byte_Text | uint16[] Two_Byte_Text}"
                },
                {
                        "Name": "SpeechFeatures1_t",
                        "Type": "float[]"
                },
                {
                        "Name": "SpeechFeatures2_t",
                        "Type": "float[]"
                }
        ],
        "Ports":[
                {
                        "Name":"ModeSelection",
                        "Direction":"InputOutput",
                        "RecordType":"Text_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"ModelUtterance",
                        "Direction":"InputOutput",
                        "RecordType":"Speech_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"EmotionlessSpeech1",
                        "Direction":"InputOutput",
                        "RecordType":"Speech_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"EmotionlessSpeech2",
                        "Direction":"InputOutput",
                        "RecordType":"Speech_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },

                {
                        "Name":"EmotionList",
                        "Direction":"InputOutput",
                        "RecordType":"Text_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"Language",
                        "Direction":"InputOutput",
                        "RecordType":"Text_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"SpeechWithEmotion",
                        "Direction":"OutputInput",
                        "RecordType":"Speech_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                }
        ],
        "SubAIMs":[
                {
                        "Name": "SpeechFeatureAnalyser1",
                        "Identifier": {
                                "ImplementerID": 100,
                                "Specification": {
                                        "Standard": "MPAI-CAE",
                                        "AIW": "CAE-EES",
                                        "AIM": "SpeechFeatureAnalyser1",
                                        "Version": "1"
                                }
                        }
                },
                {
                        "Name": "SpeechFeatureAnalyser2",
                        "Identifier": {
                                "ImplementerID": 100,
                                "Specification": {
                                        "Standard": "MPAI-CAE",
                                        "AIW": "CAE-EES",
                                        "AIM": "SpeechFeatureAnalyser2",
                                        "Version": "1"
                                }
                        }
                },
                {
                        "Name": "EmotionFeatureProducer",
                        "Identifier": {
                                "ImplementerID": 100,
                                "Specification": {
                                        "Standard": "MPAI-CAE",
                                        "AIW": "CAE-EES",
                                        "AIM": "EmotionFeatureProducer",
                                        "Version": "1"
                                }
                        }
                },
                {
                        "Name": "EmotionInserter1",
                        "Identifier": {
                                "ImplementerID": 100,
                                "Specification": {
                                        "Standard": "MPAI-CAE",
                                        "AIW": "CAE-EES",
                                        "AIM": "EmotionInserter1",
                                        "Version": "1"
                                }
                        }
                },
                {
                        "Name": "EmotionInserter2",
                        "Identifier": {
                                "ImplementerID": 100,
                                "Specification": {
                                        "Standard": "MPAI-CAE",
                                        "AIW": "CAE-EES",
                                        "AIM": "EmotionInserter2",
                                        "Version": "1"
                                }
                        }
                }
        ],
        "Topology":[
                {
                        "Output":{
                                "AIMName":"SpeechFeatureAnalyser1",
                                "PortName":"SpeechFeatures1"
                        },
                        "Input":{
                                "AIMName":"EmotionInserter1",
                                "PortName":"SpeechFeatures1"
                        }
                },
                {
                        "Output":{
                                "AIMName":"SpeechFeatureAnalyser2",
                                "PortName":"EmotionlessSpeechFeatures"
                        },
                        "Input":{
                                "AIMName":"EmotionFeatureProducer",
                                "PortName":"EmotionlessSpeechFeatures"
                        }
                },
                {
                        "Output":{
                                "AIMName":"EmotionFeatureProducer",
                                "PortName":"SpeechFeatures2"
                        },
                        "Input":{
                                "AIMName":"EmotionInserter2",
                                "PortName":"SpeechFeatures2"
                        }

                }
        ]
}
```

6.2 AIM Metadata

6.2.1 Speech Feature Analyser1

```
{
        "Identifier":{
                "ImplementerID":100,
                "Specification":{
                        "Name": "CAE",
                        "AIW": "EES",
                        "AIM": "SpeechFeatureAnalyser1",
                        "Version":"1"
                }
        },
        "Description": "This AIM implements speech feature analyser 1 function for CAE-EES that
extracts Speech Features1 of a model emotional utterance and transfers them to the Emotion
Inserter1.",
        "Types":[
                {
                        "Name": "Speech_t",
                        "Type": "uint32[]"
                },
                {
                        "Name": "SpeechFeatures1_t",
                        "Type": "float[]"
                }
        ],
        "Ports":[
                {
                        "Name":"ModelUtterance",
                        "Direction":"InputOutput",
                        "RecordType":"Speech_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"SpeechFeatures1",
                        "Direction":"OutputInput",
                        "RecordType":"SpeechFeatures1_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                }
        ]
}
```

6.2.2 Speech Feature Analyser2

```
{
        "Identifier":{
                "ImplementerID":100,
                "Specification":{
                        "Name": "CAE",
                        "AIW": "EES",
                        "AIM": "SpeechFeatureAnalyser2",
                        "Version":"1"
                }
        },
        "Description": "This AIM implements speech feature analyser 2 function for CAE-EES that
extracts Speech Features2 of an emotionless input utterance, passing these to Emotion Feature
Inserter2.",
        "Types":[
                {
                        "Name": "Speech_t",
                        "Type": "uint32[]"
                },
                {
                        "Name": "SpeechFeatures2_t",
                        "Type": "float[]"
                }
        ],
        "Ports":[
                {
                        "Name":"EmotionlessSpeech_2",
                        "Direction":"InputOutput",
                        "RecordType":"Speech_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"EmotionlessSpeechFeatures",
                        "Direction":"OutputInput",
                        "RecordType":"SpeechFeatures2_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                }
        ]
}
```

6.2.3 Emotion Feature Producer

```
{
        "Identifier":{
                "ImplementerID":100,
                "Specification":{
                        "Name": "CAE",
                        "AIW": "EES",
                        "AIM": "EmotionFeatureProducer",
                        "Version": "1"
                }
        },
        "Description":"This AIM implements emotion feature Producer function for CAE-EES that
receives the Speech Features produced by Speech Feature Analyser2 plus a list of Emotions to be
added.",
        "Types":[
                {
                        "Name": "SpeechFeatures2_t",
                        "Type": "float[]"
                },
                {
                        "Name": "Emotion_t",
                        "Type": "uint8"
                },
                {
                        "Name": "EmotionList_t",
                        "Type": "Emotion_t[]"
                },
                {
                        "Name": "Text_t",
                        "Type": "{byte[] One_Byte_Text | uint16[] Two_Byte_Text}"
                }
        ],
        "Ports":[
                {
                        "Name":"EmotionlessSpeechFeatures",
                        "Direction":"InputOutput",
                        "RecordType":"SpeechFeatures2_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"EmotionList",
                        "Direction":"InputOutput",
                        "RecordType":"EmotionList_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"Language",
                        "Direction":"InputOutput",
                        "RecordType":"Text_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"SpeechFeatures2",
                        "Direction":"OutputInput",
                        "RecordType":"SpeechFeatures2_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                }
        ]
}
```

6.2.4 Emotion Inserter1

```
{
        "Identifier":{
                "ImplementerID":100,
                "Specification":{
                        "Name": "CAE",
                        "AIW": "EES",
                        "AIM": "EmotionInserter",
                        "Version":"1"
                }
        },
        "Description":"This AIM implements emotion inserter1 function for CAE-EES that integrates
the Speech Features with those of the Emotionless Speech input, yielding and delivering an
emotionally modified utterance.",
        "Types":[
                {
                        "Name": "Speech_t",
                        "Type": "uint32[]"
                },
                {
                        "Name": "Emotion_t",
                        "Type": "uint8"
                },
                {
                        "Name": "EmotionList_t",
                        "Type": "Emotion_t[]"
                },
                {
                        "Name": "SpeechFeatures1_t",
                        "Type": "float[]"
                }
        ],
        "Ports":[

                {
                        "Name":"SpeechFeatures1",
                        "Direction":"InputOutput",
                        "RecordType":"SpeechFeatures1_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"EmotionlessSpeech",
                        "Direction":"InputOutput",
                        "RecordType":"Speech_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"SpeechWithEmotion",
                        "Direction":"OutputInput",
                        "RecordType":"Speech_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                }
        ]
}
```

6.2.5 Emotion Inserter2

```
{
        "Identifier":{
                "ImplementerID":100,
                "Specification":{
                        "Name": "CAE",
                        "AIW": "EES",
                        "AIM": "EmotionInserter",
                        "Version":"1"
                }
        },
        "Description":"This AIM implements emotion inserter2 function for CAE-EES that integrates
the Speech Features with those of the Emotionless Speech input, yielding and delivering an
emotionally modified utterance.",
        "Types":[
                {
                        "Name": "Speech_t",
                        "Type": "uint32[]"
                },
                {
                        "Name": "Emotion_t",
                        "Type": "uint8"
                },
                {
                        "Name": "EmotionList_t",
                        "Type": "Emotion_t[]"
                },
                {
                        "Name": "SpeechFeatures2_t",
                        "Type": "float[]"
                }
        ],
        "Ports":[

                {
                        "Name":"SpeechFeatures2",
                        "Direction":"InputOutput",
                        "RecordType":"SpeechFeatures2_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"EmotionlessSpeech",
                        "Direction":"InputOutput",
                        "RecordType":"Speech_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"SpeechWithEmotion",
                        "Direction":"OutputInput",
                        "RecordType":"Speech_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                }
        ]
}
```

Annex 7 - AIW and AIM of ARP

7.1 AIW metadata

```
{
    "Identifier": {
        "ImplementerID": 100,
        "Specification": {
            "Standard": "CAE",
            "AIW": "ARP",
            "AIM": "ARP",
            "Version": "1.00"
        }
    },
    "Description": "This AIW implements ARP application of MPAI-CAE",
    "Types": [{
        "Name": "Audio_t",
        "Type": "uint32[]"
    },{
        "Name": "AudioFileArray_t",
        "Type": "Audio_t[]"
    },{
        "Name": "Image_t",
        "Type": "uint64[]"
    },{
        "Name": "IrregularityImages_t",
        "Type": "Image_t[]"
    },{
        "Name": "Video_t",
        "Type": "{int32 frameNumber; int16 x; int16 y; byte[] frame}"
    },{
        "Name": "JSON_t",
        "Type": "{byte[] oneByteText | uint16[] twoByteText}"
    },{
        "Name": "AccessCopyFiles_t",
        "Type": "{AudioFileArray_t RestoredAudioFiles; JSON_t EditingList; IrregularityImages_t
IrregularityImages; JSON_t IrregularityFile}"
    },{
        "Name": "PreservationMasterFiles_t",
        "Type": "{Audio_t PreservationAudioFile; Video_t PreservationAudioVisualFile;
IrregularityImages_t IrregularityImages; JSON_t IrregularityFile}"
    }],
    "Ports": [{
        "Name": "PreservationAudioFile",
        "Direction": "InputOutput",
        "RecordType": "Audio_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "PreservationAudioVisualFile",
        "Direction": "InputOutput",
        "RecordType": "Frame_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "AccessCopyFiles",
        "Direction": "OutputInput",
        "RecordType": "AccessCopy_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "PreservationMasterFiles",
        "Direction": "OutputInput",
        "RecordType": "PreservationMasterFiles_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
```

```
}],
"SubAIMs": [{
    "Name": "AudioAnalyser",
    "Identifier": {
        "ImplementerID": 101,
        "Specification": {
            "Standard": "CAE",
            "AIW": "ARP",
            "AIM": "AudioAnalyser",
            "Version": "1.00"
        }
    }
},{
    "Name": "VideoAnalyser",
    "Identifier": {
        "ImplementerID": 102,
        "Specification": {
            "Standard": "CAE",
            "AIW": "ARP",
            "AIM": "VideoAnalyser",
            "Version": "1.00"
        }
    }
},{
    "Name": "TapeAudioRestoration",
    "Identifier": {
        "ImplementerID": 103,
        "Specification": {
            "Standard": "CAE",
            "AIW": "ARP",
            "AIM": "TapeAudioRestoration",
            "Version": "1.00"
        }
    }
},{
    "Name": "TapeIrregularityClassifier",
    "Identifier": {
        "ImplementerID": 104,
        "Specification": {
            "Standard": "CAE",
            "AIW": "ARP",
            "AIM": "TapeIrregularityClassifier",
            "Version": "1.00"
        }
    }
},{
    "Name": "Packager",
    "Identifier": {
        "ImplementerID": 105,
        "Specification": {
            "Standard": "CAE",
            "AIW": "ARP",
            "AIM": "Packager",
            "Version": "1.00"
        }
    }
}],
"Topology": [{
    "Output": {
        "AIMName": "",
        "PortName": "PreservationAudioFile"
    },
    "Input": {
        "AIMName": "AudioAnalyser",
        "PortName": "PreservationAudioFile"
    }
},{
    "Output": {
        "AIMName": "",
        "PortName": "PreservationAudioFile"
    },
    "Input": {
        "AIMName": "TapeAudioRestoration",
        "PortName": "PreservationAudioFile"
```

```
            }
        },{
            "Output": {
                "AIMName": "",
                "PortName": "PreservationAudioFile"
            },
            "Input": {
                "AIMName": "Packager",
                "PortName": "PreservationAudioFile"
            }
        },{
            "Output": {
                "AIMName": "",
                "PortName": "PreservationAudioVisualFile"
            },
            "Input": {
                "AIMName": "AudioAnalyser",
                "PortName": "PreservationAudioVisualFile"
            }
        },{
            "Output": {
                "AIMName": "",
                "PortName": "PreservationAudioVisualFile"
            },
            "Input": {
                "AIMName": "VideoAnalyser",
                "PortName": "PreservationAudioVisualFile"
            }
        },{
            "Output": {
                "AIMName": "",
                "PortName": "PreservationAudioVisualFile"
            },
            "Input": {
                "AIMName": "Packager",
                "PortName": "PreservationAudioVisualFile"
            }
        },{
            "Output": {
                "AIMName": "AudioAnalyser",
                "PortName": "IrregularityFileOutput_1"
            },
            "Input": {
                "AIMName": "VideoAnalyser",
                "PortName": "IrregularityFileInput"
            }
        },{
            "Output": {
                "AIMName": "VideoAnalyser",
                "PortName": "IrregularityFileOutput_1"
            },
            "Input": {
                "AIMName": "AudioAnalyser",
                "PortName": "IrregularityFileInput"
            }
        },{
            "Output": {
                "AIMName": "AudioAnalyser",
                "PortName": "IrregularityFileOutput_2"
            },
            "Input": {
                "AIMName": "TapeIrregularityClassifier",
                "PortName": "IrregularityFileInput_1"
            }
        },{
            "Output": {
                "AIMName": "VideoAnalyser",
                "PortName": "IrregularityFileOutput_2"
            },
            "Input": {
                "AIMName": "TapeIrregularityClassifier",
                "PortName": "IrregularityFileInput_2"
            }
        },{
```

```
    "Output": {
        "AIMName": "TapeIrregularityClassifier",
        "PortName": "IrregularityFileOutput_1"
    },
    "Input": {
        "AIMName": "TapeAudioRestoration",
        "PortName": "IrregularityFile"
    }
},{
    "Output": {
        "AIMName": "TapeIrregularityClassifier",
        "PortName": "IrregularityFileOutput_2"
    },
    "Input": {
        "AIMName": "Packager",
        "PortName": "IrregularityFile"
    }
},{
    "Output": {
        "AIMName": "AudioAnalyser",
        "PortName": "AudioBlocks"
    },
    "Input": {
        "AIMName": "TapeIrregularityClassifier",
        "PortName": "AudioBlocks"
    }
},{
    "Output": {
        "AIMName": "VideoAnalyser",
        "PortName": "IrregularityImages"
    },
    "Input": {
        "AIMName": "TapeIrregularityClassifier",
        "PortName": "IrregularityImagesInput"
    }
},{
    "Output": {
        "AIMName": "TapeIrregularityClassifier",
        "PortName": "IrregularityImagesOutput"
    },
    "Input": {
        "AIMName": "Packager",
        "PortName": "IrregularityImages"
    }
},{
    "Output": {
        "AIMName": "TapeAudioRestoration",
        "PortName": "RestoredAudioFiles"
    },
    "Input": {
        "AIMName": "Packager",
        "PortName": "RestoredAudioFiles"
    }
},{
    "Output": {
        "AIMName": "TapeAudioRestoration",
        "PortName": "EditingList"
    },
    "Input": {
        "AIMName": "Packager",
        "PortName": "EditingList"
    }
},{
    "Output": {
        "AIMName": "Packager",
        "PortName": "AccessCopyFiles"
    },
    "Input": {
        "AIMName": "",
        "PortName": "AccessCopyFiles"
    }
},{
    "Output": {
        "AIMName": "Packager",
            "PortName": "PreservationMasterFiles"
        },
        "Input": {
            "AIMName": "",
            "PortName": "PreservationMasterFiles"
        }
    }],
    "Implementations": []
}
```

7.2 AIM metadata

7.2.1 Audio Analyser

```
{
    "Identifier": {
        "ImplementerID": 101,
        "Specification": {
            "Standard": "CAE",
            "AIW": "ARP",
            "AIM": "AudioAnalyser",
            "Version": "1.00"
        }
    },
    "Description": "This AIM implements the Audio Analyser.",
    "Types": [{
        "Name": "Audio_t",
        "Type": "uint32[]"
    },{
        "Name": "AudioFileArray_t",
        "Type": "Audio_t[]"
    },{
        "Name": "Video_t",
        "Type": "{int32 frameNumber; int16 x; int16 y; byte[] frame}"
    },{
        "Name": "JSON_t",
        "Type": "{byte[] oneByteText | uint16[] twoByteText}"
    }],
    "Ports": [{
        "Name": "PreservationAudioFile",
        "Direction": "InputOutput",
        "RecordType": "Audio_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "PreservationAudioVisualFile",
        "Direction": "InputOutput",
        "RecordType": "Video_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityFileInput",
        "Direction": "InputOutput",
        "RecordType": "JSON_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityFileOutput_1",
        "Direction": "OutputInput",
        "RecordType": "JSON_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityFileOutput_2",
        "Direction": "OutputInput",
        "RecordType": "JSON_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "AudioBlocks",
        "Direction": "OutputInput",
        "RecordType": "AudioFileArray_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    }],
    "SubAIMs": [],
    "Topology": [],
    "Implementations": []
}
```

7.2.2 Video Analyser

```
{
    "Identifier": {
        "ImplementerID": 102,
        "Specification": {
            "Standard": "CAE",
            "AIW": "ARP",
            "AIM": "VideoAnalyser",
            "Version": "1.00"
        }
    },
    "Description": "This AIM implements the Video Analyser.",
    "Types": [{
        "Name": "Image_t",
        "Type": "uint64[]"
    },{
        "Name": "IrregularityImages_t",
        "Type": "Image_t[]"
    },{
        "Name": "Video_t",
        "Type": "{int32 frameNumber; int16 x; int16 y; byte[] frame}"
    },{
        "Name": "JSON_t",
        "Type": "{byte[] oneByteText | uint16[] twoByteText}"
    }],
    "Ports": [{
        "Name": "PreservationAudioVisualFile",
        "Direction": "InputOutput",
        "RecordType": "Video_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityFileInput",
        "Direction": "InputOutput",
        "RecordType": "JSON_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityFileOutput_1",
        "Direction": "OutputInput",
        "RecordType": "JSON_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityFileOutput_2",
        "Direction": "OutputInput",
        "RecordType": "JSON_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityImages",
        "Direction": "OutputInput",
        "RecordType": "IrregularityImages_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    }],
    "SubAIMs": [],
    "Topology": [],
    "Implementations": []
}
```

7.2.3 Tape Irregularity classifier

```
{
    "Identifier": {
        "ImplementerID": 103,
        "Specification": {
            "Standard": "CAE",
            "AIW": "ARP",
            "AIM": "TapeIrregularityClassifier",
            "Version": "1.00"
        }
    },
    "Description": "This AIM implements the Tape Irregularity Classifier.",
    "Types": [{
        "Name": "Audio_t",
        "Type": "uint32[]"
    },{
        "Name": "AudioFileArray_t",
        "Type": "Audio_t[]"
    },{
        "Name": "Image_t",
        "Type": "uint64[]"
    },{
        "Name": "IrregularityImages_t",
        "Type": "Image_t[]"
    },{
        "Name": "JSON_t",
        "Type": "{byte[] oneByteText | uint16[] twoByteText}"
    }],
    "Ports": [{
        "Name": "AudioBlocks",
        "Direction": "InputOutput",
        "RecordType": "AudioFileArray_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityFileInput_1",
        "Direction": "InputOutput",
        "RecordType": "JSON_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityFileInput_2",
        "Direction": "InputOutput",
        "RecordType": "JSON_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityImagesInput",
        "Direction": "InputOutput",
        "RecordType": "IrregularityImages_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityFileOutput_1",
        "Direction": "OutputInput",
        "RecordType": "JSON_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityFileOutput_2",
        "Direction": "OutputInput",
        "RecordType": "JSON_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityImagesOutput",
        "Direction": "OutputInput",
        "RecordType": "IrregularityImages_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    }],
    "SubAIMs": [],
    "Topology": [],
    "Implementations": []
}
```

7.2.4 Tape Audio Restoration

```
{
    "Identifier": {
        "ImplementerID": 104,
        "Specification": {
            "Standard": "CAE",
            "AIW": "ARP",
            "AIM": "TapeAudioRestoration",
            "Version": "1.00"
        }
    },
    "Description": "This AIM implements the Tape Audio Restoration.",
    "Types": [{
        "Name": "Audio_t",
        "Type": "uint32[]"
    },{
        "Name": "AudioFileArray_t",
        "Type": "Audio_t[]"
    },{
        "Name": "JSON_t",
        "Type": "{byte[] oneByteText | uint16[] twoByteText}"
    }],
    "Ports": [{
        "Name": "PreservationAudioFile",
        "Direction": "InputOutput",
        "RecordType": "Audio_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityFile",
        "Direction": "InputOutput",
        "RecordType": "JSON_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "RestoredAudioFiles",
        "Direction": "OutputInput",
        "RecordType": "AudioFileArray_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "EditingList",
        "Direction": "OutputInput",
        "RecordType": "JSON_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    }],
    "SubAIMs": [],
    "Topology": [],
    "Implementations": []
}
```

7.2.5 Packager

```
{
    "Identifier": {
        "ImplementerID": 105,
        "Specification": {
            "Standard": "CAE",
            "AIW": "ARP",
            "AIM": "Packager",
            "Version": "1.00"
        }
    },
    "Description": "This AIM implements the Packager.",
    "Types": [{
        "Name": "Audio_t",
        "Type": "uint32[]"
    },{
        "Name": "AudioFileArray_t",
        "Type": "Audio_t[]"
    },{
        "Name": "Image_t",
        "Type": "uint64[]"
    },{
        "Name": "IrregularityImages_t",
        "Type": "Image_t[]"
    },{
        "Name": "Video_t",
        "Type": "{int32 frameNumber; int16 x; int16 y; byte[] frame}"
    },{
        "Name": "JSON_t",
        "Type": "{byte[] oneByteText | uint16[] twoByteText}"
    },{
        "Name": "AccessCopyFiles_t",
        "Type": "{AudioFileArray_t RestoredAudioFiles; JSON_t EditingList; IrregularityImages_t
IrregularityImages; JSON_t IrregularityFile}"
    },{
        "Name": "PreservationMasterFiles_t",
        "Type": "{Audio_t PreservationAudioFile; Video_t PreservationAudioVisualFile;
IrregularityImages_t IrregularityImages; JSON_t IrregularityFile}"
    }],
    "Ports": [{
        "Name": "PreservationAudioFile",
        "Direction": "InputOutput",
        "RecordType": "Audio_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "RestoredAudioFiles",
        "Direction": "InputOutput",
        "RecordType": "AudioFileArray_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "EditingList",
        "Direction": "InputOutput",
        "RecordType": "JSON_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityFile",
        "Direction": "InputOutput",
        "RecordType": "JSON_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "IrregularityImages",
        "Direction": "InputOutput",
        "RecordType": "IrregularityImages_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "PreservationAudioVisualFile",
        "Direction": "InputOutput",
        "RecordType": "Video_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "AccessCopyFiles",
        "Direction": "OutputInput",
        "RecordType": "AccessCopyFiles_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    },{
        "Name": "PreservationMasterFiles",
        "Direction": "OutputInput",
        "RecordType": "PreservationMasterFiles_t",
        "Technology": "Software",
        "Protocol": "",
        "IsRemote": false
    }],
    "SubAIMs": [],
    "Topology": [],
    "Implementations": []
}
```

Annex 8 - AIW and AIM of SRS

8.1 AIW metadata

```
{
	"$schema": "https://json-schema.org/draft/2020-12/schema",
	"$id": "https://mpai.community/standards/resources/MPAI-AIF/V1/AIW-AIM-
	metadata.schema.json",
	"title": "SRS AIF v1 AIW/AIM metadata",
	"Identifier": {
		"ImplementerID": 100,
		"Specification": {
			"Standard": "MPAI-CAE",
			"AIW": "CAE-SRS",
			"AIM": "CAE-SRS",
			"Version": "1"
		}
	},

	"APIProfile": "Main",
	"Description":"This AIW implements SRS application of MPAI-CAE",
	"Types":[
		{
			"Name": "Speech_t",
			"Type": "uin32[]"
		},
		{
			"Name": "AudioSegments_t",
			"Type": "Speech_t[]"
		},
		{
			"Name": "JSON_t",
			"Type": "{byte[] One_Byte_Text | uint16[] Two_Byte_Text}"
		}
	],
	"Ports":[
		{
			"Name":"DamagedSegments",
			"Direction":"InputOutput",
			"RecordType":"Speech_t",
			"Technology":"Software",
			"Protocol":"",
			"IsRemote": false
		},
		{
			"Name":"DamagedList",
			"Direction":"InputOutput",
			"RecordType":"JSON_t",
			"Technology":"Software",
			"Protocol":"",
			"IsRemote": false
		},
		{
			"Name":"TextList",
			"Direction":"InputOutput",
			"RecordType":"JSON_t",
			"Technology":"Software",
			"Protocol":"",
			"IsRemote": false
		},
		{
			"Name":"AudioSegmentsForModelling",
			"Direction":"InputOutput",
			"RecordType":"AudioSegments_t",
			"Technoogy":"Software",
			"Protocol":"",
			"IsRemote": false
		},
		{
			"Name":"RestoredSegment",
				"Direction":"OutputInput",
				"RecordType":"Speech_t",
				"Technology":"Software",
				"Protocol":"",
				"IsRemote": false
			}
		],
		"SubAIMs":[
			{
				"Name": "SpeechModelCreation",
				"Identifier": {
					"ImplementerID": 100,
					"Specification": {
						"Standard": "MPAI-CAE",
						"AIW": "CAE-SRS",
						"AIM": "SpeechModelCreation",
						"Version": "1"
					}
				}
			},
			{
				"Name": "SpeechSynthesiser",
				"Identifier": {
					"ImplementerID": 100,
					"Specification": {
						"Standard": "MPAI-CAE",
						"AIW": "CAE-SRS",
						"AIM": "SpeechSynthesiser",
						"Version": "1"
					}
				}
			},
			{
				"Name": "Assembler",
				"Identifier": {
					"ImplementerID": 100,
					"Specification": {
						"Standard": "MPAI-CAE",
						"AIW": "CAE-SRS",
						"AIM": "Assembler",
						"Version": "1"
					}
				}
			}
		],
		"Topology":[
			{
				"Output":{
					"AIMName":"SpeechModelCreation",
					"PortName":"NeuralNetworkSpeechModel"
				},
				"Input":{
					"AIMName":"SpeechSynthesiser",
					"PortName":"NeuralNetworkSpeechModel"
				}
			},
			{
				"Output":{
					"AIMName":"SpeechSynthesiser",
					"PortName":"SynthesisedSpeech"
				},
				"Input":{
					"AIMName":"Assembler",
					"PortName":"SynthesisedSpeech"
				}
			}
		]
	}
```

8.2 AIM metadata

8.2.1 Speech Model Creation

```
{
        "Identifier":{
                "ImplementerID":100,
                "Specification":{
                        "Name": "CAE",
                        "AIW": "SRS",
                        "AIM": "SpeechModelCreation",
                        "Version":"1
                }
        },
        "Description":"This AIM implements Speech Model Creation function for CAE-SRS that
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.",
        "Types":[
                {
                        "Name": "Speech_t",
                        "Type": "uin32[]"
                },
                {
                        "Name": "AudioSegments_t",
                        "Type": "Speech_t[]"
                },
                {
                        "Name": "JSON_t",
                        "Type": "{byte[] One_Byte_Text | uint16[] Two_Byte_Text}"
                }
        ],
        "Ports":[
                {
                        "Name":"AudioSegmentsForModelling",
                        "Direction":"InputOutput",
                        "RecordType":"AudioSegments_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"NeuralNetworkSpeechModel",
                        "Direction":"OutputInput",
                        "RecordType":"Text_t",
                        "Type":"Software",
                        "Protocol":"",
                        "IsRemote": false
                }
        ]
}
```

8.2.2 Speech Synthesiser

```
{
        "Identifier":{
                "ImplementerID":100,
                "Specification":{
                        "Name": "CAE",
                        "AIW": "SRS",
                        "AIM": "SpeechSynthesiser",
                        "Version":"1
                }
        },
        "Description":"This AIM implements Speech Synthesiser function for CAE-SRS. The Neural
Network Speech Model is passed to the Speech Synthesiser 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 Synthesiser produces synthetic replacements
for each damaged section (or for Damaged Segment as a whole) and passes the replacement(s) to
Assembler.",
        "Types":[
                {
                        "Name": "Speech_t",
                        "Type": "uin32[]"
                },
                {
                        "Name": "AudioSegments_t",
                        "Type": "Speech_t[]"
                },
                {
                        "Name": "Text_t",
                        "Type": "{byte[] One_Byte_Text | uint16[] Two_Byte_Text}"
                }
                {
                        "Name": "JSON_t",
                        "Type": "{byte[] One_Byte_Text | uint16[] Two_Byte_Text}"
                }
        ],
        "Ports":[
                {
                        "Name":"TextList",
                        "Direction":"InputOutput",
                        "RecordType":"JSON_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },{
                        "Name":"NeuralNetworkSpeechModel",
                        "Direction":"InputOutput",
                        "RecordType":"Text_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },{
                        "Name":"SynthesisedSpeech",
                        "Direction":"OutputInput",
                        "RecordType":"AudioSegments_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                }
        ]
}
```

8.2.3 Assembler

```
{
        "Identifier":{
                "ImplementerID":100,
                "Specification":{
                        "Name": "CAE",
                        "AIW": "SRS",
                        "AIM": "Assembler",
                        "Version":"1
                }
        },
        "Description":"This AIM implements Assembler function for CAE-SRS. Assembler receives as
input the entire Damaged Segment, plus Damaged List Time Labels, 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. Assembler 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.",
        "Types":[
                {
                        "Name": "Speech_t",
                        "Type": "uin32[]"
                },
                {
                        "Name": "AudioSegments_t",
                        "Type": "Speech_t[]"
                },
                {
                        "Name": "Text_t",
                        "Type": "{byte[] One_Byte_Text | uint16[] Two_Byte_Text}"
                }
        {
            "Name": "JSON_t",
            "Type": "{byte[] One_Byte_Text | uint16[] Two_Byte_Text}"
        }
    ],
    "Ports":[
        {
            "Name":"DamagedSegments",
            "Direction":"InputOutput",
            "RecordType":"Text_t",
            "Technology":"Software",
            "Protocol":"",
            "IsRemote": false
        },
        {
            "Name":"DamagedList",
            "Direction":"InputOutput",
            "RecordType":"JSON_t",
            "Technology":"Software",
            "Protocol":"",
            "IsRemote": false
        },
        {
            "Name":"SynthesisedSpeech",
            "Direction":"InputOutput",
            "RecordType":"AudioSegments_t",
            "Technology":"Software",
            "Protocol":"",
            "IsRemote": false
        },
        {
            "Name":"RestoredSegment",
            "Direction":"OutputInput",
            "RecordType":"Speech_t",
            "Technology":"Software",
            "Protocol":"",
            "IsRemote": false
        }
    ]
}
```

Annex 9 - AIW and AIM of EAE

9.1 AIW Metadata

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://mpai.community/standards/resources/MPAI-AIF/V1/AIW-AIM-metadata.schema.json",
  "title": "EAE AIF v1 AIW/AIM metadata",
    "Identifier": {
      "ImplementerID": 100,
      "Specification": {
        "Standard": "MPAI-CAE",
        "AIW": "CAE-EAE",
        "AIM": "CAE-EAE",
        "Version": "1"
      }
    },
    "APIProfile": "Main",
    "Description": "This AIF is used to call the AIW of EAE",
     "Types": [
      {
                              "Name":"Audio_t",
                              "Type":"uint16[]",
                      },
        {
                              "Name":"Array_Audio_t",
                              "Type":"Audio_t[]",
                      },
        {
                              "Name":"TransformArray_Audio_t",
                              "Type":"Array_Audio_t[]",
                      },
        {
                              "Name":"Text_t",
                              "Type":"uint8[]",
                      }
     ],
    "Ports": [
      {
                              "Name":"MicrophoneArrayAudio",
                              "Direction":"InputOutput",
                              "RecordType":"Array_Audio_t",
                              "Technology":"Software",
                              "Protocol":"",
                              "IsRemote": false
                      },
                      {
                              "Name":"TransformMultichannelAudio",
                              "Direction":"OutputInput",
                              "RecordType":"TransformArray_Audio_t",
                              "Technology":"Software",
                              "Protocol":"",
                              "IsRemote": false
                      },
                      {
                              "Name":"TransformMultichannelAudio",
                              "Direction":"InputOutput",
                              "RecordType":"TransformArray_Audio_t",
                              "Technology":"Software",
                              "Protocol":"",
                              "IsRemote": false
                      },
               {
                              "Name":"MicrophoneArrayGeometry",
                              "Direction":"InputOutput",
                              "RecordType":"Text_t",
                              "Technology":"Software",
                              "Protocol":"",
                              "IsRemote": false
```

```
                },
                {
                        "Name":"SphericalHarmonicsDecomposition",
                        "Direction":"OutputInput",
                        "RecordType":"TransformArray_Audio_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"SphericalHarmonicsDecomposition",
                        "Direction":"InputOutput",
                        "RecordType":"TransformArray_Audio_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
        {
                        "Name":"TransformSpeech",
                        "Direction":"OutputInput",
                        "RecordType":"TransformArray_Audio_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"AudioSceneGeometry",
                        "Direction":"OutputInput",
                        "RecordType":"Text_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"SphericalHarmonicsDecomposition",
                        "Direction":"InputOutput",
                        "RecordType":"TransformArray_Audio_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"TransformSpeech",
                        "Direction":"InputOutput",
                        "RecordType":"TransformArray_Audio_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"AudioSceneGeometry",
                        "Direction":"InputOutput",
                        "RecordType":"Text_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"DenoisedTransformSpeech",
                        "Direction":"OutputInput",
                        "RecordType":"TransformArray_Audio_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
                        "Name":"DenoisedTransformSpeech",
                        "Direction":"InputOutput",
                        "RecordType":"TransformArray_Audio_t",
                        "Technology":"Software",
                        "Protocol":"",
                        "IsRemote": false
                },
                {
```

```
                              "Name":"DenoisedSpeech",
                              "Direction":"OutputInput",
                              "RecordType":"Array_Audio_t",
                              "Technology":"Software",
                              "Protocol":"",
                              "IsRemote": false
                      }
        ],
    "SubAIMs": [
        {
            "Name": "AnalysisTransform",
            "Identifier": {
              "ImplementerID": 100,
              "Specification": {
                "Standard": "MPAI-CAE",
                "AIW": "CAE-EAE",
                "AIM": "AnalysisTransform",
                "Version": "1"
              }
            }
        },
        {
            "Name": "SoundFieldDescription",
            "Identifier": {
              "ImplementerID": 100,
              "Specification": {
                "Standard": "MPAI-CAE",
                "AIW": "CAE-EAE",
                "AIM": "SoundFieldDescription",
                "Version": "1"
              }
            }
        },
        {
            "Name": "SpeechDetectionandSeparation",
            "Identifier": {
              "ImplementerID": 100,
              "Specification": {
                "Standard": "MPAI-CAE",
                "AIW": "CAE-EAE",
                "AIM": "SpeechDetectionandSeparation",
                "Version": "1"
              }
            }
        },
        {
            "Name": "NoiseCancellation",
            "Identifier": {
              "ImplementerID": 100,
              "Specification": {
                "Standard": "MPAI-CAE",
                "AIW": "CAE-EAE",
                "AIM": "NoiseCancellation",
                "Version": "1"
              }
            }
        },
        {
            "Name": "SynthesisTransform",
            "Identifier": {
              "ImplementerID": 100,
              "Specification": {
                "Standard": "MPAI-CAE",
                "AIW": "CAE-EAE",
                "AIM": "SynthesisTransform",
                "Version": "1"
              }
            }
        },
        {
            "Name": "Packager",
            "Identifier": {
              "ImplementerID": 100,
              "Specification": {
```

```
            "Standard": "MPAI-CAE",
            "AIW": "CAE-EAE",
            "AIM": "Packager",
            "Version": "1"
          }
        }
    }
  ],
"Topology": [
                   {
                                   "Output":{
                                          "AIMName":"",
                                          "PortName":"MicrophoneArrayAudio"
                                   },
                                   "Input":{
                                          "AIMName":"AnalysisTransform",
                                          "PortName":"MicrophoneArrayAudio"
                                   }
                           },
  {
                                   "Output":{
                                          "AIMName":"",
                                          "PortName":"MicrophoneArrayGeometry_1"
                                   },
                                   "Input":{
                                          "AIMName":"SoundFieldDescription",
                                          "PortName":" MicrophoneArrayGeometry_1"
                                   }
                           },
  {
                                   "Output":{
                                          "AIMName":"",
                                          "PortName":"MicrophoneArrayGeometry_2"
                                   },
                                   "Input":{
                                          "AIMName":"Packager",
                                          "PortName":" MicrophoneArrayGeometry_2"
                                   }
                           },
  {
                                   "Output":{
                                          "AIMName":"AnalysisTransform",
                                          "PortName":"TransformMultiChannelAudio"
                                   },
                                   "Input":{
                                          "AIMName":"SoundFieldDescription",
                                          "PortName":"TransformMultiChannelAudio"
                                   }
                           },
  {
                                   "Output":{
                                          "AIMName":"SoundFieldDescription",
                                          "PortName":"SphericalHarmonicsDecomposition_1"
                                   },
                                   "Input":{
                                          "AIMName":"SpeechDetectionandSeparation",
                                          "PortName":"SphericalHarmonicsDecomposition_1"
                                   }
                           },
  {
                                   "Output":{
                                          "AIMName":"SoundFieldDescription",
                                          "PortName":"SphericalHarmonicsDecomposition_2"
                                   },
                                   "Input":{
                                          "AIMName":"SpeechDetectionandSeparation",
                                          "PortName":"SphericalHarmonicsDecomposition_2"
                                   }
                           },
  {
                                   "Output":{
                                          "AIMName":"SpeechDetectionandSeparation",
                                          "PortName":"TransformSpeech"
                                   },
                                   "Input":{
                                          "AIMName":"NoiseCancellation",
                                          "PortName":"TransformSpeech"
                                   }
                           },
  {
                                   "Output":{
                                          "AIMName":"SpeechDetectionandSeparation",
                                          "PortName":"AudioSceneGeometry_1"
                                   },
                                   "Input":{
                                          "AIMName":"NoiseCancellation",
                                          "PortName":"AudioSceneGeometry_1"
                                   }
                           },
  {
                                   "Output":{
                                          "AIMName":"SpeechDetectionandSeparation",
                                          "PortName":"AudioSceneGeometry_2"
                                   },
                                   "Input":{
                                          "AIMName":"Packager",
                                          "PortName":"AudioSceneGeometry_2"
                                   }
                           },
  {
                                   "Output":{
                                          "AIMName":"NoiseCancellation",
                                          "PortName":"DenoisedTransformSpeech"
                                   },
                                   "Input":{
                                          "AIMName":"SynthesisTransform",
                                          "PortName":"DenoisedTransformSpeech"
                                   }
                           },
  {
                                   "Output":{
                                          "AIMName":"SynthesisTransform",
                                          "PortName":"DenoisedSpeech"
                                   },
                                   "Input":{
                                          "AIMName":"Packager",
                                          "PortName":"DenoisedSpeech"
                                   }
                           }
            ],
"Implementations": [{
      "BinaryName": "eae.exe",
      "Architecture": "x64",
      "OperatingSystem": "Windows",
      "Version": "v0.1",
      "Source": "AIMStorage",
      "Destination": ""
  }
    ],
"ResourcePolicies": [
      {
            "Name": "Memory",
            "Minimum": "50000",
            "Maximum": "100000",
            "Request": "75000"
  },
      {
            "Name": "CPUNumber",
            "Minimum": "1",
            "Maximum": "2",
            "Request": "1"
  },
      {
            "Name": "CPU:Class",
            "Minimum": "Low",
            "Maximum": "High",
            "Request": "Medium"
  },
          {
               "Name": "GPU:CUDA:FrameBuffer",
               "Minimum": "11GB_GDDR5X",
               "Maximum": "8GB_GDDR6X",
               "Request": "11GB_GDDR6"
      },
          {
               "Name": "GPU:CUDA:MemorySpeed",
               "Minimum": "1.60GHz",
               "Maximum": "1.77GHz",
               "Request": "1.71GHz"
      },
          {
               "Name": "GPU:CUDA:Class",
               "Minimum": "SM61",
               "Maximum": "SM86",
               "Request": "SM75"
      },
          {
               "Name": "GPU:Number",
               "Minimum": "1",
               "Maximum": "1",
               "Request": "1"
      }
    ],
               "Documentation":[
                       {
                               "Type":"Tutorial",
                               "URI":"https://mpai.community/standards/mpai-cae/"
                       }
               ]
}
```

9.2 AIM Metadata

9.2.1 Metadata of CAE-EAE Analysis Transform AIM

```
{
       "Identifier":{
               "ImplementerID":100,
               "Specification":{
                       "Name": "CAE",
                       "AIW": "EAE",
                       "AIM": "AnalysisTransform",
                       "Version":"1"
               }
    },
       "Description":"This AIM implements analysis transform function for CAE-EAE that converts
microphone array audio into transform multichannel audio.",
       "Types":[
               {
                       "Name": "Audio_t",
                       "Type": "uint16[]"
        },
               {
                       "Name": "Array_Audio_t",
                       "Type": "Audio_t[]"
        },
               {
                       "Name": "Transform_Array_Audio_t",
                       "Type": "Array_Audio_t[]"
        }
       ],
       "Ports":[
               {
                       "Name":"MicrophoneArrayAudio",
                       "Direction":"InputOutput",
                       "RecordType":"Array_Audio_t",
                       "Technology":"Software",
                       "Protocol":"",
                       "IsRemote": false
               },
               {
                       "Name":"TransformMultichannelAudio",
                       "Direction":"OutputInput",
                       "RecordType":"TransformArray_Audio_t",
                       "Technology":"Software",
                       "Protocol":"",
                       "IsRemote": false
        }
       ],
       "SubAIMs":[],
       "Topology":[],
    "Implementations": [],
       "Documentation":[
               {
                       "Type":"Tutorial",
                       "URI":"https://mpai.community/standards/mpai-cae/"
               }
       ]
}
```

9.2.2 Metadata of CAE-EAE Sound Field Description AIM

```
{
        "AIM":{
                "ImplementerID": 100,
                "Standard":{
                        "Name": "CAE",
                        "AIW": "EAE",
                        "AIM": "SoundFieldDescription",
                        "Version":"1"
                },
                "Description":"This AIM implements sound field description function for CAE-EAE
that converts transform multichannel audio into spherical harmonics decomposition.",
                "Types":[
{
                        "Name": "Text_t",
                        "Type": "uint8[]"
                },
                        {
                        "Name": "Audio_t",
                        "Type": "uint16[]"
                },
                {
                        "Name": "Array_Audio_t",
                        "Type": "Audio_t[]"
                },
                {
                        "Name": "Transform_Array_Audio_t",
                        "Type": "Array_Audio_t[]"
                }
                ],
                "Ports":[
                        {
                                "Name":"TransformMultichannelAudio",
                                "Direction":"InputOutput",
                                "RecordType":"TransformArray_Audio_t",
                                "Technology":"Software",
                                "Protocol":"",
                                "IsRemote": false
                        },
{
                                "Name":"MicrophoneArrayGeometry",
                                "Direction":"InputOutput",
                                "RecordType":"Text_t",
                                "Technology":"Software",
                                "Protocol":"",
                                "IsRemote": false
                        },
                        {
                                "Name":"SphericalHarmonicsDecomposition",
                                "Direction":"OutputInput",
                                "RecordType":"TransformArray_Audio_t",
                                "Technology":"Software",
                                "Protocol":"",
                                "IsRemote": false
                        }
                ],
                "SubAIMs":[],
                "Topology":[],
                "Documentation":[
                        {
                                "Type":"tutorial",
                                "URI":"https://mpai.community/standards/mpai-cae/"
                        }
                ]
        }
}
```

9.2.3 Metadata of CAE-EAE Speech Detection and Separation AIM

```
{
        "AIM":{
                "ImplementerID": 100,
                "Standard":{
                        "Name": "CAE",
                        "AIW": "EAE",
                        "AIM": "SpeechDetectionandSeparation",
                        "Version":"1"
},
                "Description":"This AIM implements speech detection and separation function for
CAE-EAE that converts spherical harmonics coefficients into transform speech and Audio Scene
Geometry.",
"Types":[
{
                        "Name": "Text_t",
                        "Type": "uint8[]"
                },
{
                        "Name": "Audio_t",
                        "Type": "uint16[]"
                },
                {
                        "Name": "Array_Audio_t",
                        "Type": "Audio_t[]"
                },
                {
                        "Name": "Transform_Array_Audio_t",
                        "Type": "Array_Audio_t[]"
                }
                ],
                "Ports":[
                        {
                                "Name":"SphericalHarmonicsDecomposition",
                                "Direction":"InputOutput",
                                "RecordType":"TransformArray_Audio_t",
                                "Technology":"Software",
                                "Protocol":"",
                                "IsRemote": false

                        },
{
                                "Name":"TransformSpeech",
                                "Direction":"OutputInput",
                                "RecordType":"TransformArray_Audio_t",
                                "Technology":"Software",
                                "Protocol":"",
                                "IsRemote": false

                        },
                        {
                                "Name":"AudioSceneGeometry",
                                "Direction":"OutputInput",
				"RecordType":"Text_t",
				"Technology":"Software",
				"Protocol":"",
				"IsRemote": false

			}
		],
		"AIMs":[],
		"Topology":[],
		"Documentation":[
			{
				"Type":"tutorial",
				"URI":"https://mpai.community/standards/mpai-cae/"
			}
		]
	}
}
```

9.2.4 Metadata of CAE-EAE Noise Cancellation AIM

```
{
	"AIM":{
		"ImplementerID": 100,
		"Standard":{
			"Name": "CAE",
			"AIW": "EAE",
			"AIM": "NoiseCancellation",
			"Version":"1"
		},
		"Description":"This AIM implements noise cancellation function for CAE-EAE that
converts transform speech into denoised transform speech.",
"Types":[
{
			"Name": "Text_t",
			"Type": "uint8[]"
		},
{
			"Name": "Audio_t",
			"Type": "uint16[]"
		},
		{
			"Name": "Array_Audio_t",
			"Type": "Audio_t[]"
		},
		{
			"Name": "Transform_Array_Audio_t",
			"Type": "Array_Audio_t[]"
		}
		],
		"Ports":[
			{
				"Name":"SphericalHarmonicsDecomposition",
				"Direction":"InputOutput",
				"RecordType":"TransformArray_Audio_t",
				"Technology":"Software",
				"Protocol":"",
				"IsRemote": false

			},
{
				"Name":"TransformSpeech",
				"Direction":"InputOutput",
				"RecordType":"TransformArray_Audio_t",
				"Technology":"Software",
				"Protocol":"",
				"IsRemote": false

			},
			{
				"Name":"AudioSceneGeometry",
				"Direction":"InputOutput",
				"RecordType":"Text_t",
				"Technology":"Software",
                                "Protocol":"",
                                "IsRemote": false

                        },
{
                                "Name":"DenoisedTransformSpeech",
                                "Direction":"OutputInput",
                                "RecordType":"TransformArray_Audio_t",
                                "Technology":"Software",
                                "Protocol":"",
                                "IsRemote": false

                        }

                ],
                "AIMs":[

                ],
                "Topology":[

                ],

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

9.2.5 Metadata of CAE-EAE Synthesis Transform AIM

```
{
        "AIM":{
                "ImplementerID": 100,
                "Standard":{
                        "Name": "CAE",
                        "AIW": "EAE",
                        "AIM": "SynthesisTransform",
                        "Version":"1"
                },
                "Description":"This AIM implements synthesis transform function for CAE-EAE that
converts denoised transform speech into denoised speech.",
                "Types":[
{
                        "Name": "Audio_t",
                        "Type": "uint16[]"
                },
                {
                        "Name": "Array_Audio_t",
                        "Type": "Audio_t[]"
                },
                {
                        "Name": "Transform_Array_Audio_t",
                        "Type": "Array_Audio_t[]"
                }

                ],

                "Ports":[
                        {
                                "Name":"DenoisedTransformSpeech",
                                "Direction":"InputOutput",
                                "RecordType":"TransformArray_Audio_t",
                                "Technology":"Software",
                                "Protocol":"",
"IsRemote": false

                        },
{
                                "Name":"DenoisedSpeech",
                                "Direction":"OutputInput",
                                "RecordType":"Array_Audio_t",
                                "Technology":"Software",
                                "Protocol":"",
"IsRemote": false

                        }

                ],
                "AIMs":[

                ],
                "Topology":[

                ],

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

9.2.6 Metadata of CAE-EAE Packager AIM

```
{
        "AIM":{
                "ImplementerID": 100,
                "Standard":{
                        "Name": "CAE",
                        "AIW": "EAE",
                        "AIM": "Packager",
                        "Version":"1"
                },
                "Description":"This AIM implements packager function for CAE-EAE that converts
denoised speech into Multichannel Audio + Audio Scene Geometry.",
"Types":[
{
                        "Name": "Text_t",
                        "Type": "uint8[]"
                },
{
                        "Name": "Audio_t",
                        "Type": "uint16[]"
                },
                {
                        "Name": "Array_Audio_t",
                        "Type": "Audio_t[]"
                }
                ],
                "Ports":[
                        {
                                "Name":"DenoisedSpeech",
                                "Direction":"InputOutput",
                                "RecordType":"Array_Audio_t",
                                "Technology":"Software",
                                "Protocol":"",
                                "IsRemote": false

                        },
{
                                "Name":"AudioSceneGeometry",
                                "Direction":"InputOutput",
                                "RecordType":"Text_t",
                                "Technology":"Software",
                                "Protocol":"",
                                "IsRemote": false

                        },
{
```

```
                                "Name":"MultichannelAudioandAudioSceneGeometry",
                                "Direction":"OutputInput",
                                "RecordType":"Array_Audio_t",
                                "Technology":"Software",
                                "Protocol":"",
                                "IsRemote": false
                        }

                ],
                "AIMs":[],
                "Topology":[   ],
                "Documentation":[
                        {
                                "Type":"tutorial",
                                "URI":"https://mpai.community/standards/mpai-cae/"
                        }
                ]
        }
}
```