Audio Video (WebRTC) Protocol for PEMEA

PEMEA-CONS-Spec-AV-001

PEMEA CONSORTIUM

Connecting internet apps to emergency services

V1.1

5 February 2024

Contributors

Name	Company
Jorge Forcada	Deveryware
James Winterbottom	Deveryware

Table of Contents

1. Executive Summary

Audio-video communications are a common communications capability in all modern communication Apps. These Apps support dedicated and multi-party video communications on a range of platforms including mobile phones, tablets, laptops and browsers. While some applications use SIP to establish communications sessions, the vast majority use WebRTC media streams, generally with proprietary signalling.

Using simple WebRTC Audio_Video communications enables equal access to emergency services for both abled and disabled citizens from almost any device, safely and easily while keeping with the modern trend towards all Web-based communications.

The specification in the present document does not preclude PEMEA from being used to support and initiate other protocols or implementations.

The present document assumes a working knowledge of PEMEA and familiarity with the PEMEA specification ETSI TS 103 478 [R.1]. Terms common to the PEMEA specification are not redefined or explained in detail in the present document.

2. Terms and Definitions

The following terms and definitions are used in this document:

AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
AESGCM	Advanced Encryption Standard key used with GCM
AP	Application Provider
App	Application
CN	Comfort Noise
CPE	Customer Premises Equipment
DHE	Diffie-Hellman key Exchange
ECDHE	Elliptic-curve Diffie-Hellman key Exchange
ECDSA	Elliptic Curve Digital Signature Algorithm
EDS	Emergency Data Send message
ETSI	European Telecommunication Standards Institute
FID	Flow Identification
GCM	Galois/Counter Mode
HTTP	Hyper-Text Transfer Protocol
HTTPS	Hyper-Text Transfer Protocol secure
IETF	Internet Engineering Task Force
IM	Instant Messenger
IP	Internet Protocol
JSON	Java Script Object Notation
JWT	JSON Web Token
MAC	Message Authentication Code
NAT	Network Address Translation
Pa	PEMEA Application to AP interface
PAV	PEMEA Audio_Video
PCMA	Pulse Code Modulation A-Law
PCMU	Pulse Code Modulation µ-Law

PEMEA	Pan-European Mobile Emergency Application framework
PIM	PSAP Interface Module (to the PEMEA network)
PSAP	Public Safety Answering Point
PSP	PSAP Service Provider
RFC	Request For Comments
RSA	Rivest Shamir Adleman public key encryption algorithm
RTC	Real-Time Communications
RTT	Real-Time Text
SDP	Session Description Protocol
SFU	Selective Forwarding Units
SIP	Session Initiation Protocol
STUN	Session Traversal Utilities for NAT
TLS	Transport Layer Security
tPSP	Terminating PSAP Service Provider
TS	Technical Specification
TURN	Traversal Using Relay around NAT
UCS	Universal multiple-octet Coded Character Set
URI	Universal Resource Identifier
UTC	Coordinated Universal Timer
UTF-8	UCS Transformation Format (8 bit words)
WMS	WebRTC MediaStream

3. References

3.1. Normative References

[R.1] ETSI TS 103 478 (V1.1.1): "Emergency Communications (EMTEL); Pan-European Mobile Emergency Application"

[R.2] IETF RFC 2617 (June 1999): "HTTP Authentication: Basic and Digest Access Authentication"

[R.3] IETF RFC 6750 (October 2012): "The OAuth 2.0 Authorization Framework: Bearer Token Usage"

[R.4] IETF RFC 6455 (December 2011): "The WebSocket Protocol"

[R.5] IETF RFC 8445 (July 2018): "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal"

[R.6] IETF RFC 8839 (January 2021): "Session Description Protocol (SDP) Offer/Answer Procedures for Interactive Connectivity Establishment (ICE)"

[R.7] IETF RFC 7742 (March 2016): "WebRTC Video Processing and Codec Requirements"

[R.8] IETF RFC 7874 (May 2016): "WebRTC Audio Codec and Processing Requirements"

[R.9] IETF RFC 6716 (September 2012): "Definition of the Opus Audio Codec"

[R.10] IETF RFC 8251 (October 2017): "Updates to the Opus Audio Codec"

[R.11] IETF RFC 7587 (June 2015): "RTP Payload Format for the Opus Speech and Audio Codec"

[R.12] IETF RFC 8838 (January 2021): "Trickle ICE: Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol"

[R.13] IETF RFC 4566 (July 2006): "SDP: Session Description Protocol"

[R.14] IETF RFC 8840 (January 2021): "A Session Initiation Protocol (SIP) Usage for Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (Trickle ICE)"

[R.15] IETF RFC 8863 (January 2021): "Interactive Connectivity Establishment Patiently Awaiting Connectivity (ICE PAC)"

[R.16] IETF RFC 5766 (April 2020): "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)"

3.2. Informative References

[IR.1] ETSI TS 103 756 (V1.1.1): "Emergency Communications (EMTEL); PEMEA Instant Message Extension"
[IR.2] ETSI TS 103 871 (V1.1.1): "Emergency Communications (EMTEL); PEMEA Real-Time Text Extension"
[IR.3] IETF RFC 7519 (May 2015): "JSON Web Token (JWT)"
[IR.4] IETF RFC 8656 (February 2020): "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)"
[IR.5] IETF RFC 3551 (July 2003): "RTP Profile for Audio and Video Conferences with Minimal Control"
[IR.6] IETF RFC 3389 (September 2002): "Real-time Transport Protocol (RTP) Payload for Comfort Noise (CN)"

4. Scope

The present document describes the PEMEA Audio_Video (PAV) capability, and the need for this functionality. The required entities and actors are identified along with the protocol, specifying message exchanges between entities. The message formats are specified and procedural descriptions of expected behaviours under different conditions are detailed.

5. PEMEA capability extensions

5.1. Overview of extension in PEMEA

PEMEA extension capabilities are defined in ETSI TS 103 478 [R.1] and are implemented through the use of "reach-back" URIs. The Application Provider (AP) node advertises capabilities as part of the initial forward message through the network, the Emergency Data Send (EDS) message, and the terminating PSAP Service Provider (PSP) or PSAP responds with the subset of capabilities that it supports, thus binding the emergency session between the AP and the terminating emergency node.

Specifically, the capabilities are sent as information elements in the apMoreInformation element of the EDS message. The information element and apMoreInformation structures are defined in clauses 10.3.11 and 10.3.12 of ETSI TS 103 478 [R.1]. An information element in a PEMEA EDS message identifies a capability and each capability is made up of three distinct parts:

- typeOfInfo: what function does the information element serve
- protocol: the specific semantics for using the function
- value: the URI through which the service is invoked

Table 10 in ETSI TS 103 478 [R.1] identifies an initial set of "typeOfInfo" values used to specify a range of capability extensions for PEMEA. However, beyond the Location_Update and SIP_Request values described in Table 11 of ETSI TS 103 478 [R.1], protocols are left for further study and definition in subsequent specifications such as the present document. ETSI TS 103 756 [IR.1] describes the concrete specification for PEMEA Instant Message protocol and ETSI TS 103 871 [IR.2] describes the concrete specification for PEMEA Real-Time Text.

5.2. Service support indication and response

5.2.1. Service definition

ETSI TS 103 478 [R.1] defines the "Audio_Video" typeOfInfo in Table 10, but does not elaborate further on protocols in Table 11. The present document provides a concrete definition of the "Audio_Video" typeOfInfo in PEMEA through the specification of a protocol value. The definition in Table 1 shall be considered as an extension to Table 11 in ETSI TS 103 478 [R.1].

Table 1: RTT service definition

Info Type Value	Protocol Token	Description
Audio_Video	WEBRTC_NATIVE	Audio_Video functionality is supported using the PEMEA WebRTC message exchange protocol

5.2.2. Service support indication

An AP needing to indicate that the Application it is serving can support real-time text using the PEMEA protocol would include the following information element in the apMoreInformation element of the EDS associated with the emergency session:

```
<information typeOfInfo="Audio_Video" protocol="WEBRTC_NATIVE">
    https://ap.example.pemea.help/37agq1cyusbo
</information>
```

5.2.3. Service support response

A terminating node that can support the "Audio_Video" "PEMEA" capability includes this capability in the apMoreInformation element returned to the AP in the onCapSupportPost. This is described in clause 11.1.4 of ETSI TS 103 478 [R.1] with the value for "Audio_Video" "PEMEA" provided in the example below.

```
<apMoreInformation xmlns="urn:pemea:apps:xml:ns:pemea:base">
    <information typeOfInfo="Audio_Video" protocol="WEBRTC_NATIVE"/>
</apMoreInformation>
```

5.2.4. Auto response service

The original intent of many emergency applications was to provide ancillary data to the PSAP that was associated with an emergency voice call that the PSAP had, or soon would, receive. As a consequence, a PIM or tPSP usually notifies the PSAP-CPE when an EDS has arrived, but does not respond to the AP until a PSAP Call-Taker has answered the call. Operating in this manner allows for smart routing solutions ensuring that only the PSAP with the call binds the PEMEA session to the AP, ensuring that the data is always available to the PSAP Call-Taker rather than it being missing because it went to the wrong PSAP.

ETSI TS 103 478 [R.1] identifies some types of capabilities, most notably the SIP_Request capabilities, as being responded to automatically, that is, the PIM or tPSP sends an immediate onCapSupportPost message with all supported capabilities if the EDS contains a SIP_Request capability. This functionality is described in clause 8 of ETSI TS 103 478 [R.1] and came about because there was no way for the App to make a voice call until it has a destination SIP URI, so there was no possible way for the data to not be available at the destination PSAP.

Another reason for auto-response is that no conventional carrier/mobile voice call will be placed as part of the emergency communication. That is, only PEMEA advanced services will be used for communicating between the Caller and the PSAP Call-Taker.

The (PAV) capability falls into this latter category of services, that is, it is used in place of a conventional carrier/mobile voice call. Consequently, a PSAP (PIM or tPSP) supporting this capability and with the capacity to handle the communication shall respond to the AP with an onCapSupportPost message immediately on receipt of an EDS containing a PAV capability. The onCapSupportPost message shall contain the PAV capability along with any other capabilities that the PSAP supports.

If the PSAP does not support the PAV capability but does support another form of proffered multi-media communication, such as IM or RTT, then the PSAP should respond with only those capabilities.

6. Architecture

6.1. Overview

The PEMEA Audio_Video (PAV) capability is WebRTC-based, so it is necessary to understand a little bit about WebRTC in order to understand what is signalled, to whom and between which entities. This also helps to understand explicitly what is normatively specified in the present document, what is semantic, and what is normatively referred to from the present document but normatively specified in other documents.

The PAV capability was realized with disability usage in mind and the usage model for this necessitates multi-party communications where the Caller and PSAP Call-Taker represent two parties, but a third-party sign-language interpreter is also, more than likely, expected to participate in the call.

6.2. Architecture and high-level WebRTC flows

WebRTC was designed to use peer-to-peer communications between web browsers, and this means that media stream connections are between peers. In an Audio_Video flow, each participant can have 2 streams to send, one audio stream from a microphone and one video stream from a camera. It is not mandatory to have always both streams, and a participant could only negotiate its audio stream and not initially negotiate its video stream.

Media streams, where possible, should be as direct as possible between the communicating parties, however, owing to a range of networking impediments, communications often do not happen in this fashion and so it is necessary to use a TURN Server in the media path to ensure connectivity. The role of the TURN Server is to provide a common contact point between different communication parties through which the media streams can flow. In practical terms, in PSAP deployments firewalls exist between the Call-Taker equipment and any incoming media streams, so a TURN Server is always required. Consequently, a TURN Server is included as a mandatory component of the PAV architecture. The TURN server specification is defined in IETF RFC 5766 [R.16].

In the core WebRTC specifications a media server may be used to aggregate streams reducing the number of required peer connections, to allow transcoding between the participants. However, demultiplexing multiple streams on single connection increases complexity which is undesirable and unwarranted in emergency calling-applications where the number of participants in the communication is small. Consequently, PAV requires stream sets to be negotiated independently between participants using an architectural approached referred to as Selective Forwarding Units (SFU). Development of PEMEA Apps has been centred around equivalent access to emergency services, and so allowing control over how to display the videos of different participants is crucial to improve the accessibility and usability of these Apps.

Figure 1 depicts the PAV service architecture. It includes not only a TURN Server to allow NAT traversal, but also a Media Server to allow transcoding and recording of conversations for audit purposes.

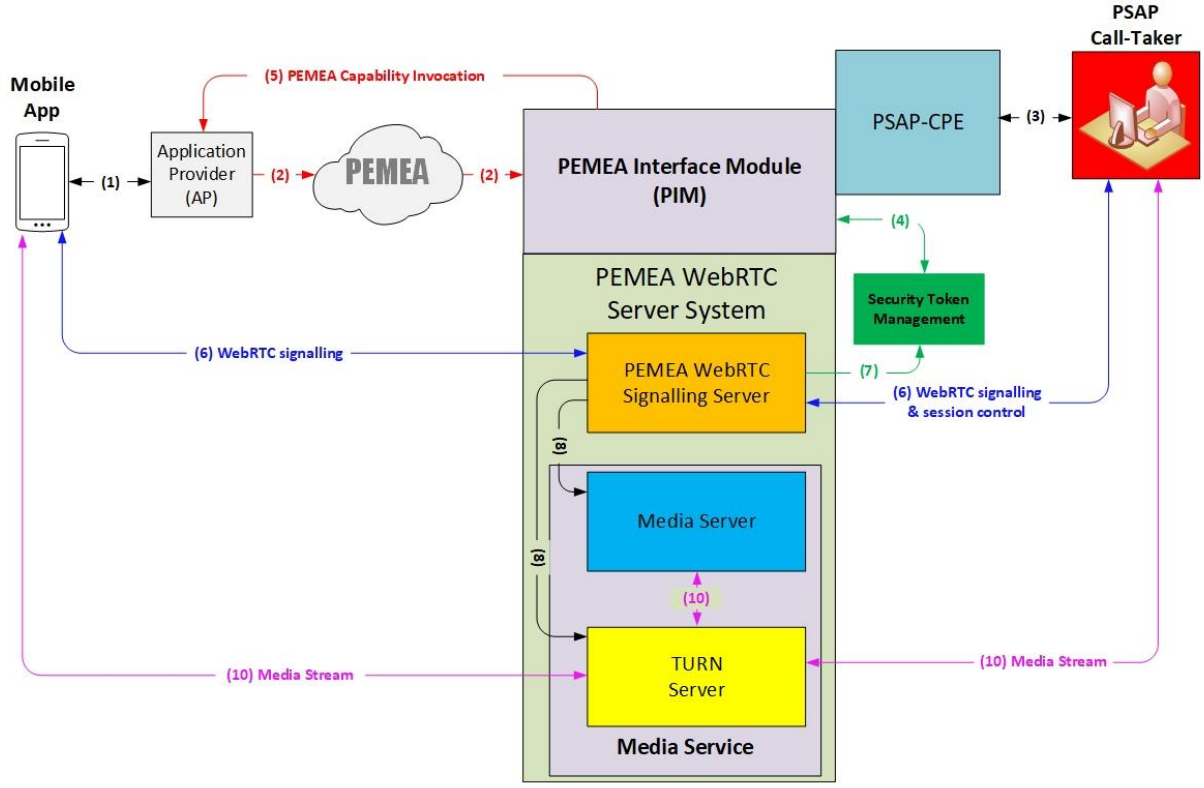


Figure 1: Audio_Video architecture for PEMEA

1. Pa interface, the application makes a call to the AP indicating the PAV capability. Invocation information is returned to the App over this interface too.
2. The AP packages the information from the App into an EDS message and sends it into the PEMEA network via the Ps interface. The EDS arrives at the PIM over the Pp interface. The PIM sends an onCapSupportPost message to the AP binding the connection between the AP and the PIM.
3. The PIM notifies the PSAP-CPE which in turn notifies the PSAP Call-Taker. The call is answered and controlled by the PSAP Call-Taker over this interface also. This includes requesting the creation of a PAV session. The PSAP Call-Taker is also able to request connection credentials for additional participants over this interface.
4. On direction from the PSAP Call-Taker (or automatically as described in clause 5.2.4) the PIM creates a new PAV signalling instance and requests Bearer tokens from the Security Token Management system. It shall generate at least a token for the PSAP Call-Taker and a token for the Caller.
5. The PIM invokes the PAV capability in the AP passing the URI for the Audio_Video signalling room as well as the Bearer token required to access it. Similar information is provided to the PSAP Call-Taker's application. The AP then passes the received information down to the App.

6. The App and the PSAP Call-Taker connect to the Audio_Video Signalling Server using the Audio_Video signalling room URI and passing in their respective security tokens. Control of their respective sessions occurs over this interface. SDP and ICE candidates are also exchanged over this interface, including the address of the TURN Server.
7. The Audio_Video Signalling Server verifies the Bearer tokens with the Security token management system before granting access to participants.
8. The Audio_Video Signalling Server obtain valid TURN server credentials for participants that will be sent over the F and G interfaces. It can also tear down connections as required.
9. The Audio_Video Signalling Server contacts the Media Server to setup the streams, it shall create peer-connections at the Media Server to which the participants will connect. A recording server could also be included in the architecture and connected with the Audio_Video Signalling Server and the Media Service, but it is not mandatory.
10. The caller and PSAP Call-Taker Apps receive the TURN Server address and credentials and connect to it. The streams are directed through the media server.

Whilst the division of the Audio_Video Server in these sub-components does not need to be strictly followed, the notion of a signalling component and a media or streaming component are important for traffic path differentiation.

6.3. Audio_Video logical components

6.3.1. Audio_Video Signalling Server

Audio_Video sessions in PEMEA are established as logical rooms through which participant applications send signalling information. It is the role of the Audio_Video Signalling Server to manage the participants in this room and the transfer of all signalling information associated with the communication between participants. The present document describes the protocol, communications and procedures for the Audio_Video Signalling server and PEMEA end-points.

6.3.2. Media Server

The media server may be used to provide recording services for the PSAP and stream transcoding for systems using codecs other than those described in clause 8.

6.3.3. TURN Server

Modern PSAP deployment isolate the PSAP Call-Taking equipment from inbound services through the use of firewalls. The firewalls control data flow through a symmetric NAT making general connectivity more secure but also more complex than direct communication. The Traversal Using Relays around NAT (TURN) protocol [IR.4] was developed specifically for allowing communications streams to work through symmetric NATs and is specifically identified as a component of the PAV architecture.

7. Security

7.1. Signalling transport security

The PEMEA Audio_Video (PAV) service is identified as an HTTPS URI that resolves to a session signalling room in the PAV server. The connection should be made using TLS 1.3 but may be made using 1.2 and this shall not support fallback below TLS 1.2. The connecting participant shall authenticate to the PEMEA signalling server using a Bearer token in the HTTP Authentication header field as described in IETF RFC 6750 [R.3]. The token should be provided to the recipient when the service is invoked. Once the connecting entity is authenticated and authorization is granted, the connection is upgraded to a websocket. The websocket is expected to remain open while the entity is "online". The protocol is resilient to connections being dropped, so an entity may reconnect as long as the EDS session remains active in the PSAP.

The lists for the TLS 1.3 and TLS 1.2 acceptable cipher suites are included in ANNEX B. These lists are informative and are based on best information at the time of writing. Older cipher suites not included in either of these lists shall not be used.

7.2. Security token usage

The HTTP Authorization header field is defined in IETF RFC 2617 [R.2] and it specifies that the usage is a scheme followed by a value, where the value may have a structure, as is the case for the digest authentication scheme.

Security token usage in the HTTP Authorization header field was originally specified for use with OAuth and is defined in IETF RFC 6750 [R.3]. Here the use of the OAuth "Bearer token" is specified so the scheme of the Authorization header field is Bearer, following the scheme a token is placed. The token shall not contain readable information unless it is encoded in base64.

Token usage in the PAV specification follows the Bearer scheme defined in IETF RFC 6750 [R.3]. Tokens issued by entities in the PAV architecture are expected also to be the validating entities, or to have ties to the validating entities, consequently, whether the tokens are opaque or follow a convention such as JSON Web Token (JWT) IETF RFC 7519 [IR.3] is not considered relevant to usage and is not specified further.

IETF RFC 6750 [R.3] mandates the usage of TLS for use with Bearer tokens, this usage is further defined in clause 7.1 of the present document.

8. PEMEA Audio_Video codec support

8.1. Overview

The PEMEA Audio_Video capability is built around WebRTC, which has an explicitly defined minimum set of codecs that shall be supported by all endpoints.

8.2. Audio codec support

WebRTC audio codec support is defined in IETF RFC 7874 [R.8] and it requires all end-points to support:

- Opus as defined in IETF RFC 6716 [R.9] with the payload defined in IETF RFC 7587 [R.11].
- Opus codec updates detailed in IETF RFC 8251 [R.10].
- G.711, PCMA and PCMU using payload defined in IETF RFC 3551 [IR.5] and Comfort Noise (CN) defined in IETF RFC 3389 [IR.6].

Conveyance of Opus over RTP is detailed in IETF RFC 7587 [R.11]. Other codecs in addition to Opus and G.711 may be supported, but Opus and G.711 shall be supported by all entities conforming with the present document.

Opus is the generally preferred audio codec and has inbuilt support for CN. However, Opus is more bandwidth thirsty and is based used where 8kHz or more of bandwidth is available.
G.711 is a narrow-band codec, however it generally included for interoperability with legacy communications equipment.

Applications conforming to the present document shall support both of these audio codec options.

8.3. Video codec support

WebRTC video codec support is defined IETF RFC 7742 [R.7] and to summarize, it requires all end-points to support:

- VP8.
- H.264 Constrained Baseline.

Recipients of video streams shall be able to decode video at a rate of at least 20 frames per second (fps) at a resolution not less than 320 pixels by 240 pixels. Higher frame rates and resolutions may be negotiated.

Applications conforming to the present document shall support both specified video codec and the minimum frame rate and resolution requirements.

9. Initiation procedures

9.1. Overview

This capability requires procedures to be followed by each of the App, AP and PIM nodes during the emergency session establishment. Connection management and peer media negotiations occur over the signalling interface ideally between the Audio_Video media service and each participant. A participant establishes a send-only channel with the media server and then establishes receive-only connections with the media server for each of the other participants in the Audio_Video signalling room. Notification of which participants are in the Audio_Video signalling room is communicated to participants over the Audio_Video signalling server interface.

9.2. App call initiation procedure

The App includes whatever information it normally includes in an initial data exchange with the AP at call time. In addition, it includes support for the PEMEA Audio_Video (PAV) capability.

9.3. AP call initiation procedure

On receipt of the call initiation request from the App, the AP shall follow normal session establishment and data exchange procedures with the App.

The AP shall then construct and send an EDS to its associated PSP, following standard PEMEA procedures as stipulated in ETSI TS 103 478 [R.1].

9.4. PIM emergency session establishment procedures

On receipt of an EDS containing the PAV capability, a PIM supporting the function shall immediately indicate its support to the originating AP through the onCapSupportPost procedure stipulated in ETSI TS 103 478 [R.1].

The PIM shall then notify the PSAP of the arrival of the EDS.

9.5. AP session binding procedures

On receipt of an onCapSupportPost message for a session from a valid PIM or PSP the AP shall perform all normal procedures and data exchanges with the App over Pa including notification of PSAP support for the PAV capability. The AP shall then bind the session to the PIM that sent the onCapSupportPost message and shall not accept reach-back service invocations from any node other than that PIM.

9.6. PEMEA Audio_Video invocation procedures

9.6.1. Service invocation flow

Under direction of a PSAP Call-Taker the PIM shall proceed to acquire the necessary resources to invoke the PAV capability in the AP.

The PIM shall request the creation of an Audio_Video signalling room to the Audio_Video signalling server. The Audio_Video signalling room shall be identified through a unique HTTPS URI. Initially, two security access Bearer tokens shall be minted for providing secure access to the Audio_Video signalling room.

The PIM shall return the Audio_Video signalling room URI and one security access Bearer token to the PSAP Call-Taker.

The PIM shall access the reach-back URI to invoke the PAV capability in the AP. The body of the HTTPS POST shall include the Audio_Video signalling room URI, one security access Bearer token and the token's associated expiry time.

Figure 2 provides the sequence diagram illustrating the flow of events between entities for the invocation procedure.

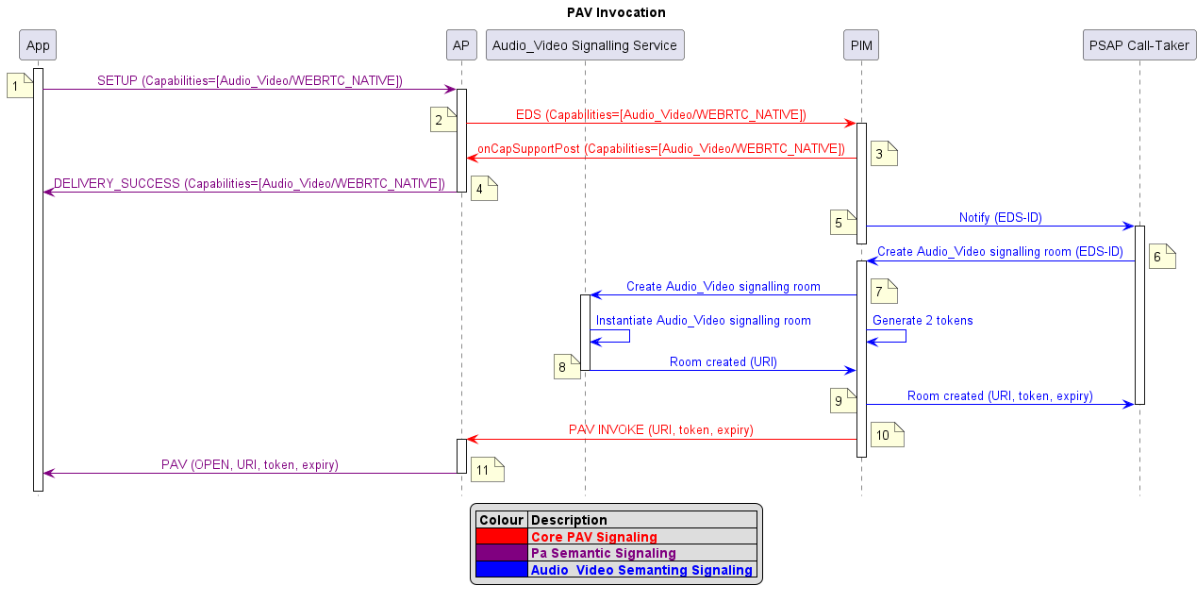


Figure 2: PAV Invocation sequence diagram

1. App initiates an emergency session with the AP over the Pa interface indicating that it can support the PAV capability.
2. The AP creates an EDS message from the data provided by the App and includes the PAV capability. The AP sends the EDS into the PEMEA network.
3. The EDS arrives at the PIM. The PIM supports the PAV capability and includes this option in the onCapSupportPost back to the AP.
4. The AP binds the emergency session to the PIM that sent the onCapSupportPost message and then signals to the App over the Pa interface the PSAP can support the PAV functionality.
5. The PIM notifies the PSAP Call-Taker that a new EDS has arrived.
6. The PSAP Call-Taker requests the PIM to initiate the creation of an Audio_Video signalling room.
7. The PIM requests that the Audio_Video Signalling Server create an Audio_Video signalling room. The room and 2 tokens are created.
8. The Audio_Video Signalling Server returns the Audio_Video signalling room URI to the PIM.
9. The PIM returns the Audio_Video signalling room URI, one token and its expiry time to the PSAP Call-Taker.
10. The PIM invokes the PAV capability in the AP using the provided reach-back URI from the EDS. The PIM includes the Audio_Video signalling room URI, token and expiry time in the body of the HTTP POST to the reach-back URI.
11. The AP signals to the App over the Pa interface that the PSAP has invoked the PAV communication capability and provides to the App the Audio_Video signalling room URI, token and expiry time.

The AP shall ignore additional invocations of the PAV capability whilst an existence instance of the capability is active. In case of errors, should the capability need to be re-invoked, the procedures in clause 9.7 shall be followed.

9.6.2. Service invocation object

The PIM invokes the PEMEA Audio_Video (PAV) service in the AP by posting to the URI provided in the Audio_Video information element included in the apMoreInformation contained in the EDS. The POST message includes a body containing a JSON object. The JSON object provides the Audio_Video signalling room URI as well as a security token and corresponding expiry time.

The JSON schema for the PAV service invocation message is provided in ANNEX A.

Property	Type	Description
url	String	The URI of the Audio_Video signalling room.
token	String	A security token used to authenticate the AP to the Audio_Video signalling room. The AP shall include the token in the HTTP Authorization header using the Bearer token scheme. The AP shall use the token each time it needs to establish or re-establish a connection to the Audio_Video signalling room for the duration of the App emergency session. The AP shall not provide the token to the App.
expiry	Integer	Specifies the expiry time of the security token. It is an integer specifying the number of second since UTC epoch, 00:00:00 1st of January 1970.

Invocation example:

```
{
    "url":"https://audio-video-server.example.com/room/534wafds21s21fdf",
    "token":"PPtzs5zzG5Pkf61KPz51",
    "expiry":1574092280231
}
```

9.6.3. Audio_Video signalling room creation and deletion

The Audio_Video signalling room is created by the Audio_Video server under direction of the PSAP Call-Taker via the PIM as described in clause 9.6.

Once the Audio_Video signalling room is created it remains active as long as the PIM maintains a context for the EDS.

9.7. Re-invocation procedures

9.7.1. Overview

In the event of PSAP communication issues, the PSAP Call-Taker may initiate a re-invocation of the PAV capability. This has implications for all participants connected to the room and should only be done in error conditions.

Under re-invocation instructions for the PAV capability from the PSAP Call-Taker the PIM shall:

1. Close the existing Audio_Video signalling room so that the associated streams and credential are invalidated.
2. On receiving the response of the termination request from the Audio_Video signalling server, the PIM requests the creation of a new Audio_Video signalling room following the invocation process in clause 9.6.
3. Create additional security tokens for third parties as so instructed by the PSAP Call-Taker following the procedures in clause 16.

9.7.2. Termination procedures

On request for Audio_Video signalling room termination from the PIM the Audio_Video signalling server shall:

1. Close opened websocket connections with all the participants. The close will be done sending a close code of 1 000 and a UTF-8 encoded reason indicating that the session has been terminated as described in the section 7 of the IETF RFC 6455 [R.4].
2. Instruct the Media Services associated with the room participants to invalidate credentials and drop any media sessions.
3. Invalidate the Audio_Video signalling room URI. The security tokens provided shall not grant access to the new Audio_Video signalling room URI that the PIM creates, therefore new tokens will be necessary.
4. Indicate room termination to the PIM.

On receiving a websocket close with the status code 1 000 from the Audio_Video signalling room, a participant shall:

1. Close all open media sessions associated with the Audio_Video call.
2. If the participant is the CALLER, the App shall consider the Audio_Video call component of the session closed.

9.7.3. Termination and Re-invocation sequences

Figure 3 provides the sequence diagram illustrating the flow of events between entities for the re-invocation procedure.

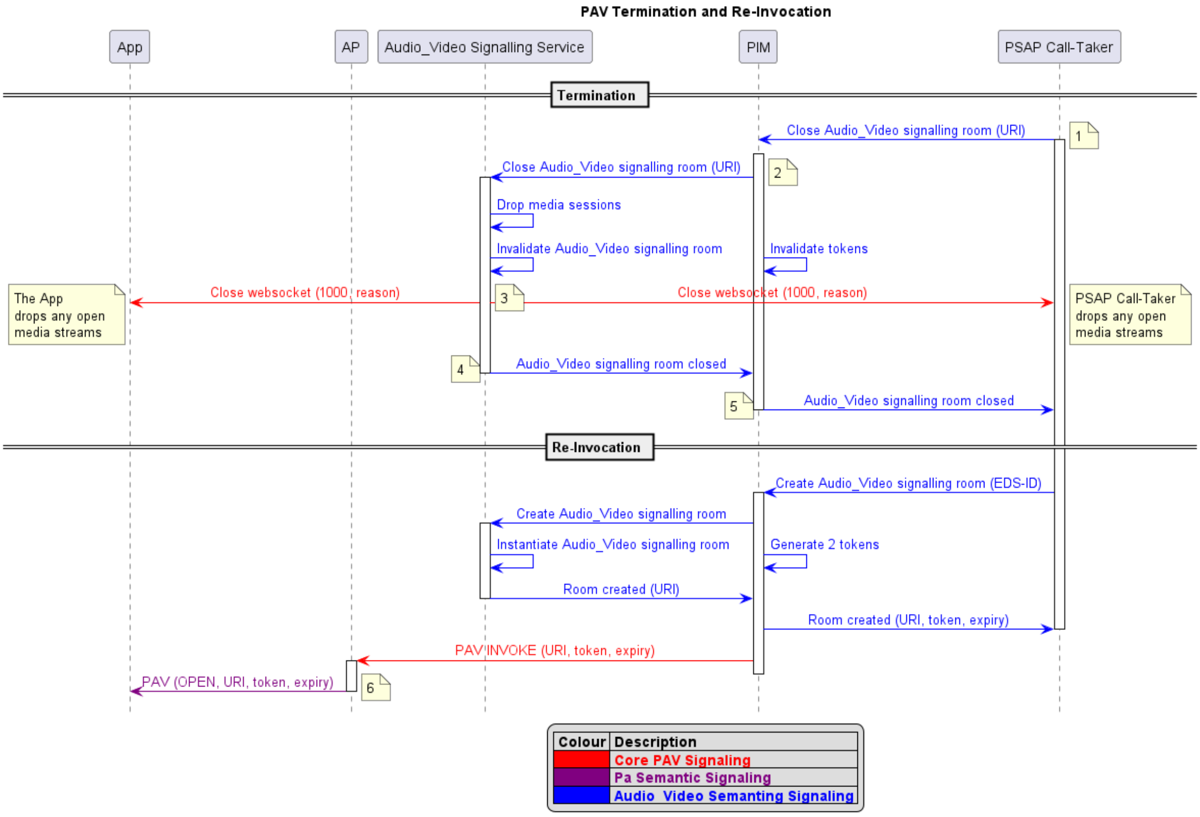


Figure 3: PAV Termination and Re-Invocation sequence diagram

1. The PSAP Call-Taker requests the PIM to close the Audio_Video signalling room.
2. The PIM requests the Audio_Video Signalling Server to close the Audio_Video signalling room. The tokens for that room are invalidated and the media sessions are dropped.
3. The Audio_Video Signalling Server closes all the websocket connections that were active in the Audio_Video signalling room. The close code shall be 1 000 and the reason should indicate that the Audio_Video session has terminated.
4. The Audio_Video Signalling Server returns a successful response to the PIM indicating that the Audio_Video signalling rom has been successfully closed.
5. The PIM returns a successful response to the PSAP Call-Taker indicating that the Audio_Video signalling rom has been successfully closed.
6. On receipt of a PAV invocation at the AP from the PIM, the AP shall notify it to the App over the Pa interface, so that the App can start again the SDP negotiation and ICE candidate exchange procedures.

The Re-Invocation follow the same procedures than the Invocation described in Figure 2.
Other participant may be re-invited to the Audio_Video signalling room following the procedures in clause 16.

10. Connection to the Audio_Video signalling room

10.1. Overview

The Audio_Video signalling room provides signalling configuration to the call participants. It also provides the means for participants to learn about the presence of other participants and to exchange signalling information that enables the establishment of media streams.

10.2. Audio_Video signalling room joining procedure

The Audio_Video signalling room shall authenticate all participants that attempt to connect to the room URI. The token in the Authentication header field shall be used to authenticate participants. If a token has expired, is not valid or is missing, then the Audio_Video signalling room shall return an HTTP 401 "Unauthorized" response to the entity attempting to connect. In case a token is correct but the participant is not authorized to access that room, the Audio_Video signalling room shall respond with an HTTP 403 "Forbidden".

Upon successful authentication of a connecting entity, the Audio_Video signalling room and the entity shall promote the connection to a websocket as described in IETF RFC 6455 [R.4].

On receipt of a JOIN message from an entity, the Audio_Video signalling room shall send a USER_LIST message to all room participants. The list shall contain the names and roles of all participants in the room and shall be structure as described in clause 19.4. The Audio_Video signalling room shall send a USER_LIST message to each room participant each time a participant enters or leaves the room.

The structure of the JOIN messages is described in clause 19.3.

10.3. PSAP Call-Taker joining procedure

To join the Audio_Video signalling room the PSAP Call-Taker equipment shall initiate an HTTPS connection to the room URI provided by the PIM. The HTTP Authentication header shall contain the security access token provided by the PIM and the token shall be placed in the Authentication header field in accordance with IETF RFC 6750 [R.3] as described in clause 7.2.

If access to the room is denied with an HTTP error response of 401 "Unauthorized" when trying to connect to the room, then the PSAP Call-Taker shall re-initiate the PEMEA Audio_Video (PAV) invocation procedure detailed in clause 9.6. If the reconnection fails, the PSAP Call-Taker shall initiate the room termination procedure detailed in clause 9.7.2 but not re-initiate the creation of the Audio_Video call. The PSAP shall invoke other communication options if provided or wait for the Caller to terminate the emergency session and re-initiate. Once authenticated to the Audio_Video signalling room, the PSAP Call-Taker equipment and the Audio_Video signalling room shall promote the connection to a websocket as described in IETF RFC 6455 [R.4].

Upon establishment of the websocket the PSAP Call-Taker equipment shall send a JOIN message with the role sub-property of the user property set to "PSAP". In the JOIN message, the PSAP Call-Taker shall indicate its user media properties, which indicates other participants how the PSAP Call-Taker will behave. The structure of the JOIN message is described in clause 19.3. Media properties are optional, and if a participant does not include some media properties, the Audio_Video signalling server shall use true as default value for each missing property for that participant.

10.4. AP joining procedure

To join the Audio_Video signalling room the App shall initiate an HTTPS connection to the room URI provided by the PIM. The HTTP Authentication header shall contain the security access token provided by the PIM and the token shall be placed in the Authentication header field in accordance with IETF RFC 6750 [R.3] as described in clause 7.2.

If access to the room is denied with an HTTP error response when trying to connect to the room as described in clause 10.2, the App should initiate alternative communication methods with the PSAP or re-initiate the emergency session.

Once authenticated to the Audio_Video signalling room, the App and the Audio_Video signalling room shall promote the connection to a websocket as described in IETF RFC 6455 [R.4].

Upon establishment of the websocket the App shall send a JOIN message with the role sub-property of the user property ser to "CALLER". In the JOIN message, the CALLER should indicate its user media properties, which indicates other participants how the CALLER will behave. The structure of the JOIN message is described in clause 19.3. Media properties are optional and, if a participant does not include some media properties, the Audio_Video signalling server shall use true as default value for each missing property for that participant.

10.5. JOIN sequence flow

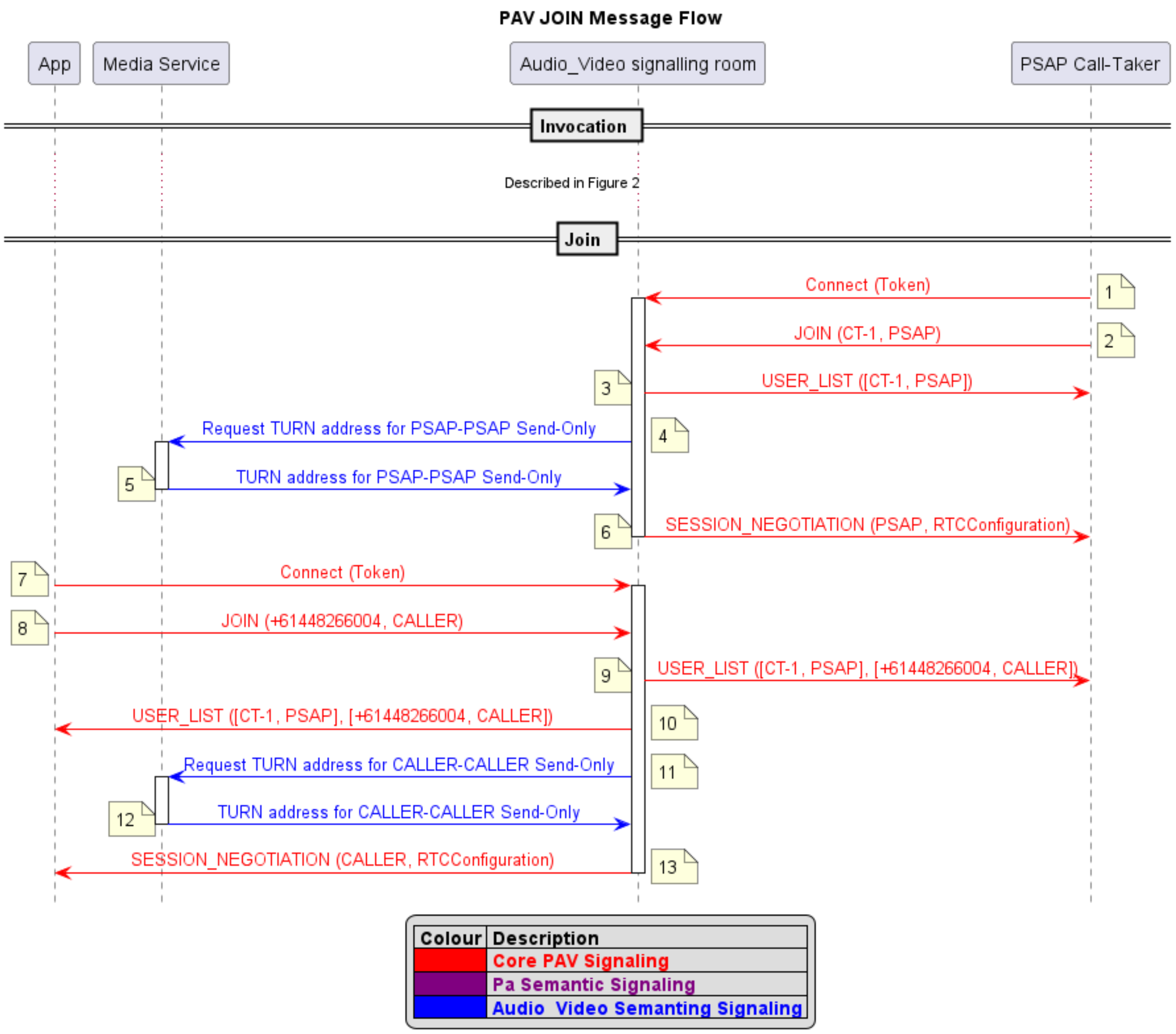


Figure 4: PAV JOIN Message sequence diagram

1. The PSAP Call-Taker connects to the Audio_Video signalling room passing in the authentication token in the Authorization HTTP header field in accordance with clause 7.2. The Audio_Video Signalling Service validates the token and upgrades the connection to a websocket.
2. The PSAP Call-Taker sends a JOIN message to the Audio_Video signalling room indicating that the connecting entity is a PSAP and includes the Call-Taker's pseudonym and optionally its user media information.
3. The Audio_Video signalling room sends to the PSAP Call-Taker a USER_LIST containing the PSAP user that was created from the previous JOIN message.
4. The Audio_Video signalling room request to the Media Service the TURN Server address that the PSAP shall use to communicate. Optionally credentials for the TURN Server can be requested for the PSAP.
5. The Media Service returns the TURN Server information requested.
6. The Audio_Video signalling room sends to the PSAP Call-Taker an RTC_SESSION_NEGOTIATIATION containing the PSAP user and the TURN Server address.

7. The App connects to the Audio_Video signalling room passing in the authentication token in the Authorization HTTP header field in accordance with clause 7.2. The Audio_Video Signalling Service validates the token and upgrades the connection to a websocket.
8. The App sends a JOIN message to the Audio_Video signalling room indicating that the connecting entity is a CALLER and includes the Caller's pseudonym and optionally its user media information.
9. The Audio_Video signalling room sends to the PSAP Call-Taker a USER_LIST containing both the PSAP and the CALLER users that were created from previous JOIN messages.
10. The Audio_Video signalling room sends to the App a USER_LIST containing both the PSAP and the CALLER users that were created from previous JOIN messages.
11. The Audio_Video signalling room request to the Media Service the TURN Server address that the CALLER shall use to communicate. Optionally credentials for the TURN Server can be requested for the CALLER.
12. The Media Service returns the TURN Server information requested.
13. The Audio_Video signalling room sends to the App an RTC_SESSION_NEGOTIATIATION containing the CALLER user and the TURN Server address.

11. List of users

11.1. Overview

After a participant successfully opens the connection with the Audio_Video signalling room and sends the JOIN message as described in clause 10, it shall receive a USER_LIST message from the Audio_Video signalling room. This message contains the list of users that are connected to the room. Each user represents a participant of the Audio_Video session. The participant that has connected shall be included in the USER_LIST.

This list is used to know the information of the participants, especially the user's name and role.

It also contains other information like the media information of each participant.

The AP shall convey the relevant information from these messages to the App so that it can display participants information. How this is performed is left to implementation.

The participants present in the USER_LIST messages are participants that have send a successful JOIN message to the Audio_Video signalling room, but the SDP negotiation described in clause 13 shall not start with these participants until an RTC_SESSION_NEGOTIATION message described in clause 19.5 has been received for that participant. This is because that participant has not yet negotiated a valid send-only stream with the Audio_Video signalling server and therefore there is no possibility to negotiate a receive-only stream from that participant.

The structure of the USER_MEDIA messages is described in clause 19.4.

11.2. Connection and disconnection of participants

The Audio_Video signalling room shall send a USER_LIST message each time a participant disconnects closing the websocket connection with the Audio_Video server and each time a participant sends a successful JOIN message to the Audio_Video signalling room.

The participants shall compare the previously received USER_LIST message with the new one to know if there is a new participant or if a participant has disappeared from the previously received USER_LIST message to the newer one.

11.3. USER_LIST sequence flow

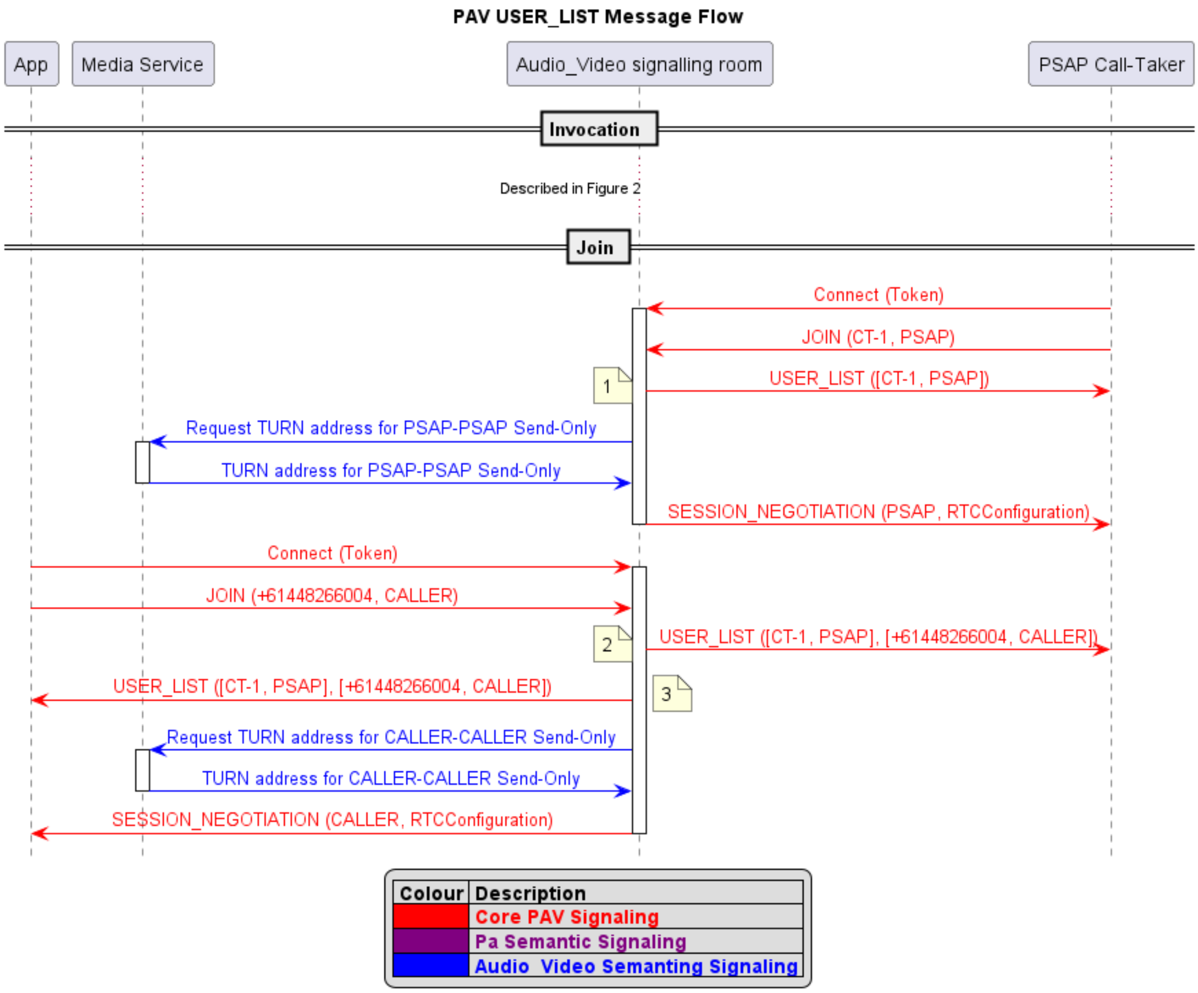


Figure 5: PAV USER_LIST Message sequence diagram

1. The Audio_Video signalling room sends to the PSAP Call-Taker a USER_LIST containing the PSAP user that was created from the previous JOIN message.
2. The Audio_Video signalling room sends to the PSAP Call-Taker a USER_LIST containing both the PSAP and the CALLER users that were created from previous JOIN messages.
3. The Audio_Video signalling room sends to the AP a USER_LIST containing both the PSAP and the CALLER users that were created from previous JOIN messages.

12. RTC session negotiation

12.1. Overview

Media streams in WebRTC are peer-to-peer, meaning that each participant in a WebRTC communication shall have a send channel that all other participants can listen on, and a listen channel for each participant so that they can hear what is being said. Communication streams will run through the TURN Server owing to firewall and access restrictions inside the PSAP.

As a consequence of this, each participant requires to know the TURN Server address in order to be able to reach a valid connectivity check with the Media Service. Optionally, the TURN Server can have authentication credentials, which should be sent to the participants in order to use them. The RTC_SESSION_NEGOTIATION message is used by the Audio_Video signalling room to provide this information and the structure of this message is detailed in clause 19.5.

The Audio_Video signalling room shall acquire TURN Server address and credentials for each participant when they join the room.

Each RTC_SESSION_NEGOTIATION message shall contain the user identification of the participant to which the message refers and the address of the TURN Server that shall be used to communicate with that participant.

RTC_SESSION_NEGOTIATION messages can also have TURN Server credentials if they are required. It is recommended to use authentication at the TURN Server, but it is not mandatory.

The structure of the RTC_SESSION_NEGOTIATION messages is described in clause 19.5.

The information received in the RTC_SESSION_NEGOTIATION messages shall be used by participants when creating peer connections with the Media Service as it is described in clauses 13 and 14.

The AP shall convey the relevant information from each RTC_SESSION_NEGOTIATION message to the App through the Pa interface. How this is performed is left to implementation.

12.2. RTC_SESSION_NEGOTIATION message for Send-Only streams

After a participant successfully opens the connection with the Audio_Video signalling room and sends the JOIN message as described in clause 10, it shall receive an RTC_SESSION_NEGOTIATION message from the Audio_Video signalling server along with the first USER_LIST message described in clause 11.

The first RTC_SESSION_NEGOTIATION that the Audio_Video signalling room sends to a participant after he sends a successful JOIN message shall contain its own user identification in the user property of the message.

This first RTC_SESSION_NEGOTIATION message contains the information that the participant needs to establish its Send-Only stream to the Audio_Video Server.

12.3. RTC_SESSION_NEGOTIATION message for Receive-Only streams

After the SDP negotiation of the Send-Only stream of a participant is done as described in clause 13.3, the Audio_Video shall send to all the connected participants of the Audio_Video signalling room an RTC_SESSION_NEGOATIATION message except to the user that made the SDP negotiation. This message shall contain the user identification of the participant that negotiated its Send-Only stream in the user property of the message and the TURN Server address and credentials that the participant shall use to create the Receive-Only stream channel to receive the stream of the participant that made the SDP negotiation.

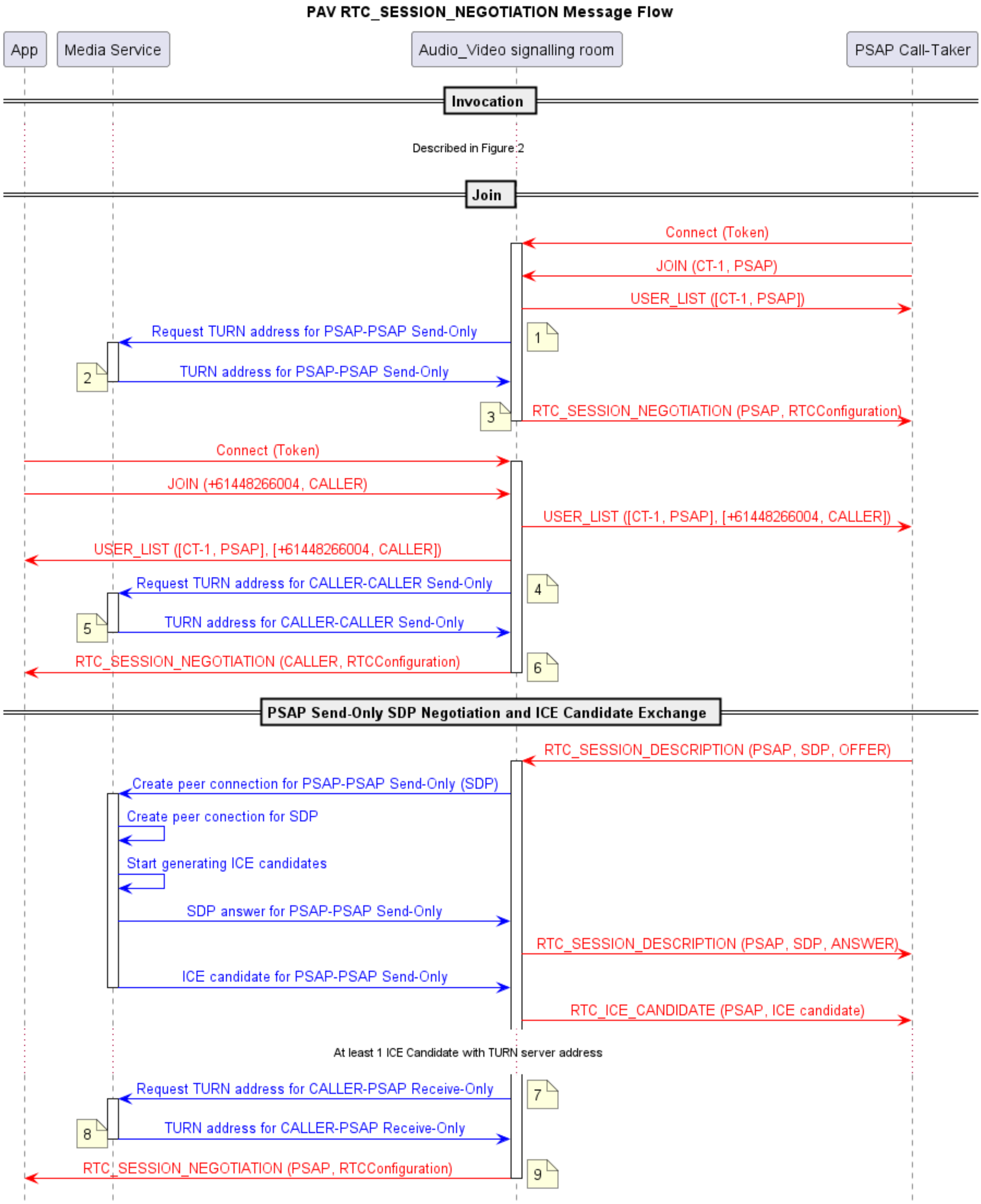
When a participant receives an RTC_SESSION_NEGOTIATION message from the Audio_Video signalling room for a participant that it's not himself, it can initiate the SDP negotiation exchange procedures described in clause 13.4 to establish the receive-only channel from that user.

If a participant re-negotiates its own send-only stream as described in clause 13.5, the Audio_Video signalling room should send a new RTC_SESSION_NEGOTIATION message to all the participants connected to the room but to the participant that re-negotiated the stream.

When a participant receives an RTC_SESSION_NEGOTION message for a participant to which he already had a peer-connection for the Receive-Only stream from that participant, it means that the participant indicated in that RTC_SESSION_NEGOTIATION message has re-negotiated its own Send-Only stream. The Media Service shall connect the new Send-Only stream with the Receive-Only streams that other participants had negotiated so that they do not need to re-negotiate the Receive-Only stream, but participant can drop the existing peer-connection associated with the Receive-Only stream from that participant, and initiate the procedures described in the clause 13.4 for that participant.

12.4. RTC_SESSION_NEGOTIATION message flow

The RTC_SESSION_NEGOTIATION message flow between 2 participants is too large, therefore it is divided in Figure 6 and Figure 7.



Colour	Description
	Core PAV Signaling
	Pa Semantic Signaling
	Audio Video Semanting Signaling

Figure 6: PAV RTC_SESSION_NEGOTIATION Message sequence diagram part 1

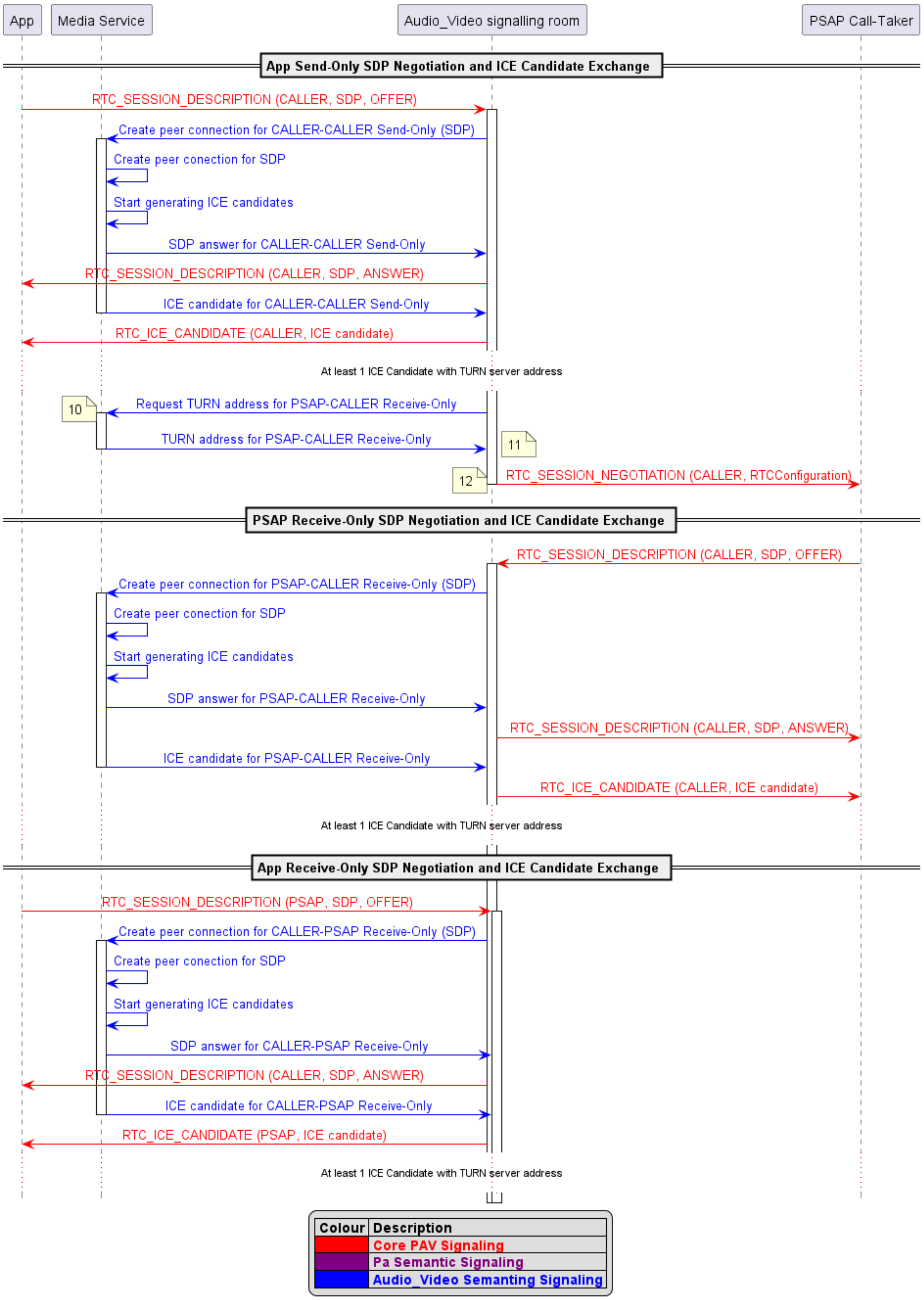


Figure 7: PAV RTC_SESSION_NEGOTIATION Message sequence diagram part 2

1. The Audio_Video signalling room requests to the Media Service the TURN Server address that the PSAP shall use for its Send-Only stream. Optionally credentials for the TURN Server can be requested for the PSAP.
2. The Media Service returns the TURN Server information requested.
3. The Audio_Video signalling room sends to the PSAP Call-Taker an RTC_SESSION_NEGOTIATIATION containing the PSAP user and the TURN Server address.
4. The Audio_Video signalling room requests to the Media Service the TURN Server address that the CALLER shall use for its Send-Only stream. Optionally credentials for the TURN Server can be requested for the CALLER.
5. The Media Service returns the TURN Server information requested.
6. The Audio_Video signalling room sends to the App an RTC_SESSION_NEGOTIATIATION containing the CALLER user and the TURN Server address.
7. After receiving the PSAP Send-Only SDP, the Audio_Video signalling room requests to the Media Service the TURN Server address that other participants shall use to communicate when creating Receive-Only stream channels to obtain the PSAP Send-Only stream. This shall be done for all the participants except for the PSAP, in this diagram, only the CALLER is connected so it is only done once.
8. The Media Service returns the TURN Server information requested.
9. The Audio_Video signalling room sends to all the participants except the PSAP an RTC_SESSION_NEGOTIATIATION containing the PSAP user and the TURN Server address for each participant. With this message each participant receives the TURN Server address that shall use to create a Receive-Only channel with the PSAP. As this diagram only has the PSAP and the CALLER, only the CALLER receives this message, but in a call with more participants, it shall be sent to all the participants except the PSAP.
10. After receiving the CALLER Send-Only SDP, the Audio_Video signalling room requests to the Media Service the TURN Server address that other participants shall use to communicate when creating Receive-Only stream channels to obtain the CALLER Send-Only stream. This shall be done for all the participants except for the CALLER, in this diagram, only the PSAP is connected so it is only done once.
11. The Media Service returns the TURN Server information requested.
12. The Audio_Video signalling room sends to all the participants except the CALLER an RTC_SESSION_NEGOTIATIATION containing the CALLER user and the TURN Server address for each participant. With this message each participant receives the TURN Server address that shall use to create a Receive-Only channel with the CALLER. As this diagram only has the CALLER and the PSAP, only the PSAP receives this message, but in a call with more participants, it shall be sent to all the participants except the CALLER.

13. SDP negotiation

13.1. Overview

WebRTC uses Real-time Transport Protocol (RTP) to enable communication between participants. To negotiate the RTP communication, Session Description Protocol (SDP) as defined in IETF RFC 4566 [R.13] shall be exchanged. The IETF RFC 8839 [R.6] specifies how the SDP negotiation shall be made, and adds the ICE connectivity checks and how they are integrated in the SDP structure.

These specifications do not define how this SDP and ICE candidate messages should be exchanged between the participants; therefore, it is described in clauses 13 and 14 of the present document.

The PEMEA Audio_Video (PAV) requires a Media Service which includes a Media Server and a TURN Server as it is described in clause 6. The participant shall create a Send-Only stream channel between them and the Media Service in order to send its media streams to the Audio_Video Server. The participants shall also create a Receive-Only stream channel between them and the Media Service for each other participant from whom they want to receive the streams that he is sending to the Media Service.

The Audio_Video Server interconnects the SDP and ICE candidate information that the participants send over the Audio_Video signalling room and the Media Service. This gives the Audio_Video Server the ability to connect in the Media Service the required connections. It receives streams from participant Send-Only streams and send streams to the participant Receive-Only streams.

The structure of the RTC_SESSION_DESCRIPTION messages is described in clause 19.6.

13.2. SDP structure considerations

The SDP "offer" generated by participants should have the property "a=sendonly" when negotiating their Send-Only stream and it should have the property "a=recvonly" when negotiating the Receive-Only stream of other participants. When the Media Service generates the SDP "answer" for the Send-Only streams of the users, it should have the property "a=recvonly", and when the Media Service generates the SDP "answer" for the Receive-Only streams of the users, it should have the property "a=sendonly". These properties should be defined in both the audio and video parts of the SDP.

As described in clause 6.3.3, the infrastructure and firewall policies on the PSAPs could be a problem when thinking in using ICE-lite implementations in the Media Service. That is why the PAV requires that all participants, including the Media Service support the ICE-full implementation as specified in IETF RFC 8445 [R.5].

The PAV uses Trickle ICE defined in IETF RFC 8838 [R.12] as a method to accelerate the initiation of the media flow between participants, as it is an emergency session and the speed of the communication establishment is crucial. This implies that the SDP of the participants shall have the property "a=ice-options:trickle" in the section of each media stream. The encoding of this option in the SDP is defined in IETF RFC 8840 [R.14].

13.3. Negotiation of Send-Only stream

Once a participant has successfully joined the Audio_Video signalling room and received the RTC_SESSION_NEGOTIATION in which the user identification is himself, the participant shall establish the RTP communication channel to send his media to the Audio_Video session.

The participant shall create a local peer-connection for his Send-Only stream to the Audio_Video Server, and generate an SDP offer for this peer-connection. It shall contain the specified audio codecs specified in clause 8.2 and the specified video codecs specified in clause 8.3. The participant shall also use the information received in the RTC_SESSION_NEGOTIATION to use the TURN Server address as a possible ICE candidate as it is described in clause 14.

The participant shall send an RTC_SESSION_DESCRIPTION message to the Audio_Video signalling room with the following properties:

1. The type property shall be set to "RTC_SESSION_DESCRIPTION".

2. The user property shall be set to the user identification. The user identification is obtained from the first RTC_SESSION_NEGOTIATION received by the user after a successful JOIN message as it is described in clause 12.2.
3. The type sub-property of the description property shall be set to "offer".
4. The sdp sub-property of the description property shall be set to the SDP offer provided by the participant.

When the Audio_Video signalling room receives the RTC_SESSION_DESCRIPTION "offer" message for the Send-Only stream from a participant, the Audio_Video signalling room shall create a peer-connection in the Media Service where the streams of the participant will be received. The SDP offer of the participant shall be added to the peer-connection and an SDP answer shall be generated by the Media Service.

The Audio_Video signalling room shall answer the participant sending him an RTC_SESSION_DESCRIPTION message with the following properties:

1. The type property shall be set to "RTC_SESSION_DESCRIPTION".
2. The user property shall be set to the user identification of the participant that sent the RTC_SESSION_DESCRIPTION "offer".
3. The type sub-property of the description property shall be set to "answer".
4. The sdp sub-property of the description property shall be obtained from the Media Service and shall be a valid SDP answer for the SDP offer sent by the participant.

When the participant receives the RTC_SESSION_DESCRIPTION "answer" message containing his user identification, the participant shall add the SDP answer to the local peer-connection.

After this process, the ICE candidate processing will start. The SDP offer and SDP answer exchange could have ICE candidates on them, but as PAV uses Trickle ICE, the ICE candidates can be exchanged using the RTC_ICE_CANDIDATE messages described in clause 14.

When the Send-Only connection of a participant is successfully made as specified in this clause, the Audio_Video signalling room shall send an RTC_SESSION_NEGOTIATION message to all the participants excluding the participant that negotiated the Send-Only stream channel as it is described in clause 12.3.

13.4. Negotiation of Receive-Only stream

When a participant wants to receive the Send-Only stream that other participant is sending to the Audio_Video Server it shall establish the RTP communication channel to receive the media streams of the selected participant.

This process shall be done with all the participants from which the participant wants to receive their media streams. If a user only wants to send his stream but does not want to receive the streams of other participants, he can avoid negotiating the streams of other participants as it is described in this clause, but he should inform other participants about it using the receiveAudio and/or receiveVideo properties with the value set to false that the PAV defines.

A participant shall not start the negotiation of the Receive-Only stream for a selected participant until the Audio_Video signalling room has sent to the participant an RTC_SESSION_NEGOTIATION message with the user property set to the value of the user identification of the selected participant.

The participant shall create a local peer-connection for his Receive-Only stream to the Audio_Video Server, and generate an SDP offer for this peer-connection. It shall contain the specified audio codecs specified in clause 8.2 and the specified video codecs specified in clause 8.3. The participant shall also use the information received in the RTC_SESSION_NEGOTIATION to use the TURN Server address as a possible ICE candidate as it is described in clause 14.

The participant shall send an RTC_SESSION_DESCRIPTION message to the Audio_Video signalling room with the following properties:

1. The type property shall be set to "RTC_SESSION_DESCRIPTION".
2. The user property shall be set to the user identification of the selected participant to whom the process described in this clause is being done.
3. The type sub-property of the description property shall be set to "offer".
4. The sdp sub-property of the description property shall be set to the SDP offer provided by the participant.

When the Audio_Video signalling room receives the RTC_SESSION_DESCRIPTION "offer" message for the Receive-Only stream from a participant to receive the stream of another selected participant, the Audio_Video signalling room shall create a peer-connection in the Media Service where the participant can receive the stream that the selected participant is sending. The Audio_Video Server shall obtain the Send-Only stream that the selected participant is sending from the Media Service, and use this stream in the Receive-Only stream that the participant requesting the negotiation will use. The SDP offer of the participant shall be added to the peer-connection and an SDP answer shall be generated by the Media Service.

If the Audio_Video signalling room receives an RTC_SESSION_DESCRIPTION "offer" message from a participant selecting a participant for whom he did not have received yet the RTC_SESSION_NEGOTIATION and thus the Audio_Video signalling room does not have yet the Send-Only stream, the Audio_Video signalling room shall send him an ERROR message described in clause 19.9 with the reasonCode property set to "streamMissing" and the reason property set to "Missing stream negotiation for user with name +34611223344 and role CALLER", being +34611223344 the name and CALLER the role of the selected user of the RTC_SESSION_DESCRIPTION "offer" message.

The Audio_Video signalling room shall answer the participant sending him an RTC_SESSION_DESCRIPTION message with the following properties:

1. The type property shall be set to "RTC_SESSION_DESCRIPTION".
2. The user property shall be set to the user identification of the selected participant to whom the process described in this clause is being done.
3. The type sub-property of the description property shall be set to "answer".
4. The sdp sub-property of the description property shall be obtained from the Media Service and shall be a valid SDP answer for the SDP offer sent by the participant.

When the participant receives the RTC_SESSION_DESCRIPTION "answer" message containing the user identification of the selected participant, the participant shall add the SDP answer to the local peer-connection for the selected participant.

After this process, the ICE candidate processing will start. The SDP offer and SDP answer exchange could have ICE candidates on them, but as PAV uses Trickle ICE, the ICE candidates can be exchanged using the RTC_ICE_CANDIDATE messages described in clause 14.

13.5. Re-Negotiation of Send-Only stream

Once a participant has made the SDP negotiation for his Send-Only channel described in clause 13.3 successfully, he is able to re-negotiate the Send-Only stream channel. This can be done if there is a problem with the streams or if the available media options change from the time at which the Send-Only stream channel was negotiated.

To do so, the participant shall remove the existing local peer-connection for his Send-Only stream and create a new local peer-connection. The participant shall generate a new SDP offer for this new peer-connection which shall contain the specified audio codecs specified in clause 8.2 and the specified video codecs specified in clause 8.3. The TURN Server address and credentials shall be the same ones that were used for the previously created Send-Only stream channel. To ensure that the TURN Server credentials are remain valid, the expiration time shall be long enough so that the re-negotiation can take place at any time during the Audio_Video session.

The participant shall send a new RTC_SESSION_DESCRIPTION message to the Audio_Video signalling room with the following properties:

1. The type property shall be set to "RTC_SESSION_DESCRIPTION".
2. The user property shall be set to the user identification. The user identification is obtained from the first RTC_SESSION_NEGOTIATION received by the user after a successful JOIN message as it is described in clause 12.2.
3. The type sub-property of the description property shall be set to "offer".
4. The sdp sub-property of the description property shall be set to the new SDP offer provided by the participant.

When the Audio_Video signalling room receives the new RTC_SESSION_DESCRIPTION "offer" message for the Send-Only stream from a participant, the Audio_Video signalling room shall delete the existing peer-connection in the Media Service and create a new peer-connection where the streams of the participant will be received. The new SDP offer of the participant shall be added to the peer-connection and a new SDP answer shall be generated by the Media Service.

The Audio_Video signalling room shall answer the participant sending him a new RTC_SESSION_DESCRIPTION message with the following properties:

1. The type property shall be set to "RTC_SESSION_DESCRIPTION".
2. The user property shall be set to the user identification of the participant that sent the RTC_SESSION_DESCRIPTION "offer".
3. The type sub-property of the description property shall be set to "answer".
4. The sdp sub-property of the description property shall be obtained from the Media Service and shall be a new valid SDP answer for the new SDP offer sent by the participant.

When the participant receives the new RTC_SESSION_DESCRIPTION "answer" message containing his user identification, the participant shall add the new SDP answer to the new local peer-connection.

After this process, the ICE candidate processing will start. The SDP offer and SDP answer exchange could have ICE candidates on them, but as PAV uses Trickle ICE, the ICE candidates can be exchanged using the RTC_ICE_CANDIDATE messages described in clause 14.

When the re-negotiation process described in this clause is successfully made, the Audio_Video signalling room should send a new RTC_SESSION_NEGOTIATION message to all the participants excluding the participant that re-negotiated the Send-Only stream channel as it is described in clause 12.3. When the participants receive an RTC_SESSION_NEGOTIATION with the user identification of a participant for whom they already have a negotiated Receive-Only channel, they can re-negotiate the Receive-Only channel for that participant as described in clause 13.6, but it is not mandatory since the Media Service should connect the new Send-Only stream with the Receive-Only stream that the participants had already negotiated with the Media Service.

13.6. Re-Negotiation of Receive-Only stream

When a participant makes the procedures described in clause 13.5, the Media Service shall reconnect the new Send-Only channel that the participant re-negotiated with all already negotiated Receive-Only streams that were already negotiated by other participants for the participant that made the re-negotiation.

The Audio_Video signalling room should send a new RTC_SESSION_NEGOTIATION message with the user identification of the participant for whom the participant had already negotiated his Receive-Only stream channel as described in clause 13.4. The participant can close the existing local peer-connection for that participant and create a new one following the procedures described in clause 13.5, but it is not mandatory since the Media Service shall interconnect the new Send-Only stream with the already negotiated Receive-Only streams.

13.7. RTC_SESSION_DESCRIPTION message flow

The RTC_SESSION_DESCRIPTION message flow between 2 participants is too large, therefore it is divided in Figure 8 and Figure 9.

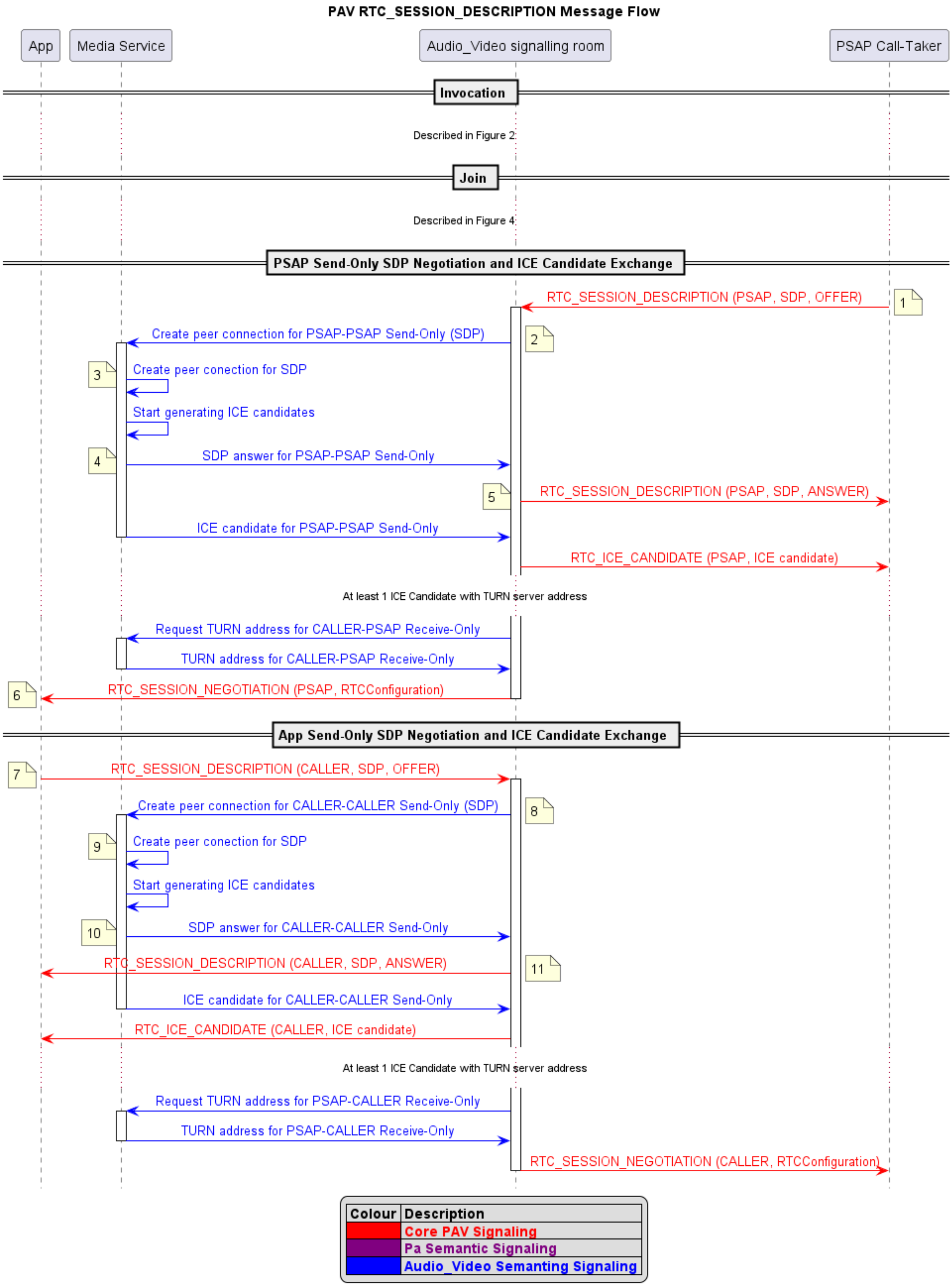


Figure 8: PAV RTC_SESSION_DESCRIPTION Message sequence diagram part 1

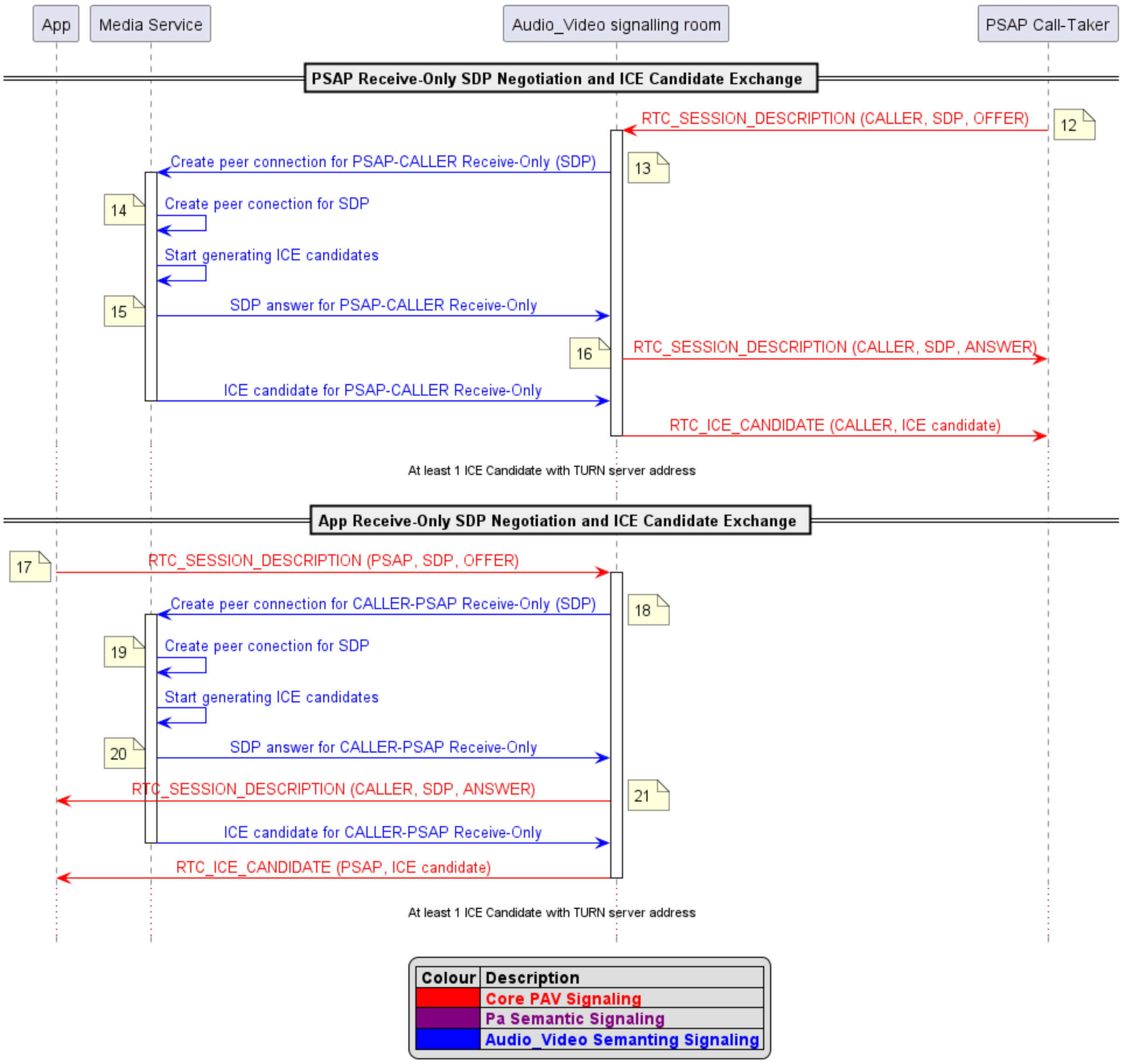


Figure 9: PAV RTC_SESSION_DESCRIPTION Message sequence diagram part 2

1. After having received the RTC_SESSION_NEGOTIATION message with the PSAP Call-Taker user identification, the PSAP Call-Taker shall create a local peer-connection for its Send-Only stream and send an RTC_SESSION_DESCRIPTION "offer" message containing the SDP offer for its Send-Only stream. The RTC_SESSION_DESCRIPTION "offer" message shall contain the PSAP user identification.
2. The Audio_Video signalling room requests to the Media Service to create the peer-connection to receive the Send-Only stream from the PSAP using the SDP offer received.
3. The Media Service creates the peer-connection using the SDP offer received and generates an SDP answer.
4. The Media Service returns the SDP answer for the PSAP Send-Only stream to the Audio_Video signalling room.
5. The Audio_Video signalling room sends an RTC_SESSION_DESCRIPTION "answer" message with the PSAP user identification to the PSAP Call-Taker. The PSAP Call-Taker shall use the SDP answer received for its local peer-connection for its Send-Only stream.
6. After having received the RTC_SESSION_NEGOTIATION message with the CALLER user identification, the App shall create a local peer-connection for its Send-Only stream and

send the associated SDP offer to the AP over the Pa interface with the CALLER user identification.

7. The App sends to the Audio_Video signalling room the RTC_SESSION_DESCRIPTION "offer" message with the SDP received from the App for its Send-Only stream. The RTC_SESSION_DESCRIPTION "offer" message shall contain the CALLER user identification.
8. The Audio_Video signalling room requests to the Media Service to create the peer-connection to receive the Send-Only stream from the CALLER using the SDP offer received.
9. The Media Service creates the peer-connection using the SDP offer received and generates an SDP answer.
10. The Media Service returns the SDP answer for the CALLER Send-Only stream to the Audio_Video signalling room.
11. The Audio_Video signalling room sends an RTC_SESSION_DESCRIPTION "answer" message with the CALLER user identification to the App. The App shall use the SDP answer received for its local peer-connection for its Send-Only stream.
12. After having received the RTC_SESSION_NEGOTIATON with the CALLER user identification, the PSAP Call-Taker can start negotiating the Receive-Only stream from the CALLER. To do so, the PSAP Call-Taker shall create a local peer-connection for the Receive-Only stream from the CALLER and send an RTC_SESSION_DESCRIPTION "offer" message containing the SDP offer for the Receive-Only stream of the CALLER. The RTC_SESSION_DESCRIPTION "offer" message shall contain the CALLER user identification.
13. The Audio_Video signalling room requests to the Media Service to create the peer-connection to send the stream that is being received from the Send-Only stream of the CALLER to the PSAP using the SDP offer received.
14. The Media Service creates the peer-connection using the SDP offer received and generates an SDP answer.
15. The Media Service returns the SDP answer to the Audio_Video signalling room for the PSAP Receive-Only stream from the CALLER.
16. The Audio_Video signalling room sends an RTC_SESSION_DESCRIPTION "answer" message with the CALLER user identification to the PSAP Call-Taker. The PSAP Call-Taker shall use the SDP answer received for its local peer-connection for its Receive-Only stream from the CALLER.
17. After having received the RTC_SESSION_NEGOTIATION with the PSAP user identification, the App can start negotiating the Receive-Only stream from the PSAP. To do so, the App shall create a local peer-connection for the Receive-Only stream from the PSAP and send an SDP offer to the Audio_Video signalling room with the PSAP user identification.
18. The Audio_Video signalling room requests to the Media Service to create the peer-connection to send the stream that is being received from the Send-Only stream of the PSAP to the CALLER using the SDP offer received.
19. The Media Service creates the peer-connection using the SDP offer received and generates an SDP answer.
20. The Media Service returns the SDP answer to the Audio_Video signalling room for the CALLER Receive-Only stream from the PSAP.
21. The Audio_Video signalling room sends an RTC_SESSION_DESCRIPTION "answer" message with the PSAP user identification to the App. The App shall use the SDP answer received for its local peer-connection for its Receive-Only stream from the PSAP.

14. ICE candidate exchange

14.1. Overview

The architecture of the PEMEA Audio_Video capability is described in clause 6 and explains that a TURN Server is needed to ensure that the stream flow between participants and the Media Service can always occur because in most of the cases participants could be behind NAT and thus not be able to establish the stream flow. To ensure interoperability, a TURN Server is mandatory.

The Interactive Connectivity Establishment (ICE) protocol is defined in IETF RFC 8445 [R.5] and updated in IETF RFC 8863 [R.15]. This protocol defines how the address where the stream flow will occur shall be generated and used between RTC peers in their SDP exchange. PAV uses also Trickle ICE which is defined in IETF RFC 8838 [R.12] to accelerate the start of the conversation as it is critical in emergency calls. Trickle ICE defines a mechanism to exchange ICE candidates after the SDP negotiation so that peer do not need to wait until all the candidates are generated in order to start the stream flow. IETF RFC 8838 [R.12] defines how the messages shall be generated and used by the RTC peers, but it does not define how to signal these messages.

IETF RFC 8839 [R.6] defines how ICE should be used in the exchange of SDP offer and SDP answer between peers, but it does not specify how the signalling of the SDP should be done between them. The signalling of these SDP in PAV is described in clause 13.

14.2. ICE candidate structure considerations

All participants shall support ICE as defined in IETF RFC 8445 [R.5] and IETF RFC 8863 [R.15]. The implementation shall be a full ICE implementation since interoperability is the most important thing and a full ICE implementation allow interworking with both ICE and ICE-Lite implementations.

All participants shall support Trickle ICE as defined in IETF RFC 8838 [R.12].

14.3. Exchange of ICE candidates

Clauses 13.3, 13.4, 13.5 and 13.6 describe how the participants shall generate a SDP offer from a local peer-connection and exchange it with the Audio_Video Server depending on what stream they want to negotiate, and how the Audio_Video Server shall create a peer-connection in the Media Server for that SDP offer, and generate a SDP answer that shall be sent to the participant.

When a participant makes the SDP negotiation for a selected participant, the participant shall create a local peer-connection and its associated SDP offer. The participant shall start the ICE candidate gathering process associated for that peer-connection. The participant shall send to the Audio_Video signalling room the SDP offer with an RTC_SESSION_DESCRIPTION message as described in clauses 13.3, 13.4, 13.5 and 13.6.

For each ICE candidate generated for the SDP offer, the participant shall send to the Audio_Video signalling room an RTC_ICE_CANDIDATE message with the following properties:

1. The type property shall be set to "RTC_ICE_CANDIDATE".
2. The user property shall be set to the user identification of the selected participant to whom the SDP negotiation is being done.
3. The candidate sub-property of the candidate property shall be set to the ICE candidate.

4. The sdpMLineIndex sub-property of the ICE candidate property shall be set to the identification tag of the media stream with which the candidate is associated.
5. Optionally, the sdpMid sub-property of the ICE candidate property should be set to the zero-based index of the m-line within the SDP offer with which the candidate is associated.
6. Optionally, the usernameFragment sub-property of the ICE candidate property should be set to the username fragment ("ufrag") that uniquely identifies a single ICE interaction session.

When the Audio_Video signalling room receives an RTC_ICE_CANDIDATE message from a participant, the Audio_Video signalling room shall add the ICE candidate to the peer-connection in the Media Service associated with that participant. The Audio_Video signalling room shall establish to which peer-connection the ICE candidate is associated using the websocket established with the participant to know who sent the message and the user property of the RTC_ICE_CANDIDATE message to identity to which user was the participant making the negotiation.

If the Audio_Video signalling room receives an RTC_ICE_CANDIDATE message from a participant selecting a participant for whom he did not have sent previously a valid RTC_SESSION_DESCRIPTION "offer" message, the Audio_Video signalling room shall send him an ERROR message described in clause 19.9 with the reasonCode property value set to "sdpMissing" and the reason property set to "Missing RTC_SESSION_DESCRIPTION offer for user with name +34611223344 and role CALLER", being +34611223344 the name and CALLER the role of the selected user of the RTC_ICE_CANDIDATE message.

When the Audio_Video signalling room creates the peer-connection in the Media Service as a result of receiving an RTC_SESSION_DESCRIPTION message from a participant and generates the SDP offer associated, the Media Service shall start the ICE candidate gathering process associated for that peer-connection. The Audio_Video signalling room shall send to the participant the SDP answer with an RTC_SESSION_DESCRIPTION message as described in clauses 13.3, 13.4, 13.5 and 13.6.

For each ICE candidate generated for the SDP answer, the Audio_Video signalling room shall send to the participant an RTC_ICE_CANDIDATE message with the following properties:

1. The type property shall be set to "RTC_ICE_CANDIDATE".
2. The user property shall be set to the user identification of the selected participant to whom the process described in this clause is being done.
3. The candidate sub-property of the candidate property shall be set to the ICE candidate.
4. The sdpMLineIndex sub-property of the ICE candidate property shall be set to the identification tag of the media stream with which the candidate is associated.
5. Optionally, the sdpMid sub-property of the ICE candidate property should be set to the zero-based index of the m-line within the SDP offer with which the candidate is associated.
6. Optionally, the usernameFragment sub-property of the ICE candidate property should be set to the username fragment ("ufrag") that uniquely identifies a single ICE interaction session.

When the participant receives an RTC_ICE_CANDIDATE message containing the user identification of the selected participant, the participant shall add the ICE candidate to the local peer-connection for the selected participant.

The structure of the RTC_ICE_CANDIDATE messages is described in clause 19.7.

14.4. End-of-Candidates indication

As it is defined in IETF RFC 8838 12, an end-of-candidates indication should be sent between the RTC peers, but it does not specify how to signal this indication.

When a participant and the Media Service have ended the process of generating ICE candidates for a certain peer-connection, they shall send an extra RTC_ICE_CANDIDATE message with the candidate sub-property of the candidate property set to an empty string (""). Receiving an RTC_ICE_CANDIDATE message with the candidate sub-property of the candidate property set to an empty string ("") indicates that the end-of-candidates process has been indicated for the associated peer-connection.

14.5. RTC_ICE_CANDIDATE message flow

The RTC_ICE_CANDIDATE message flow between 2 participants is too large, therefore it is divided in Figure 10 and Figure 11.

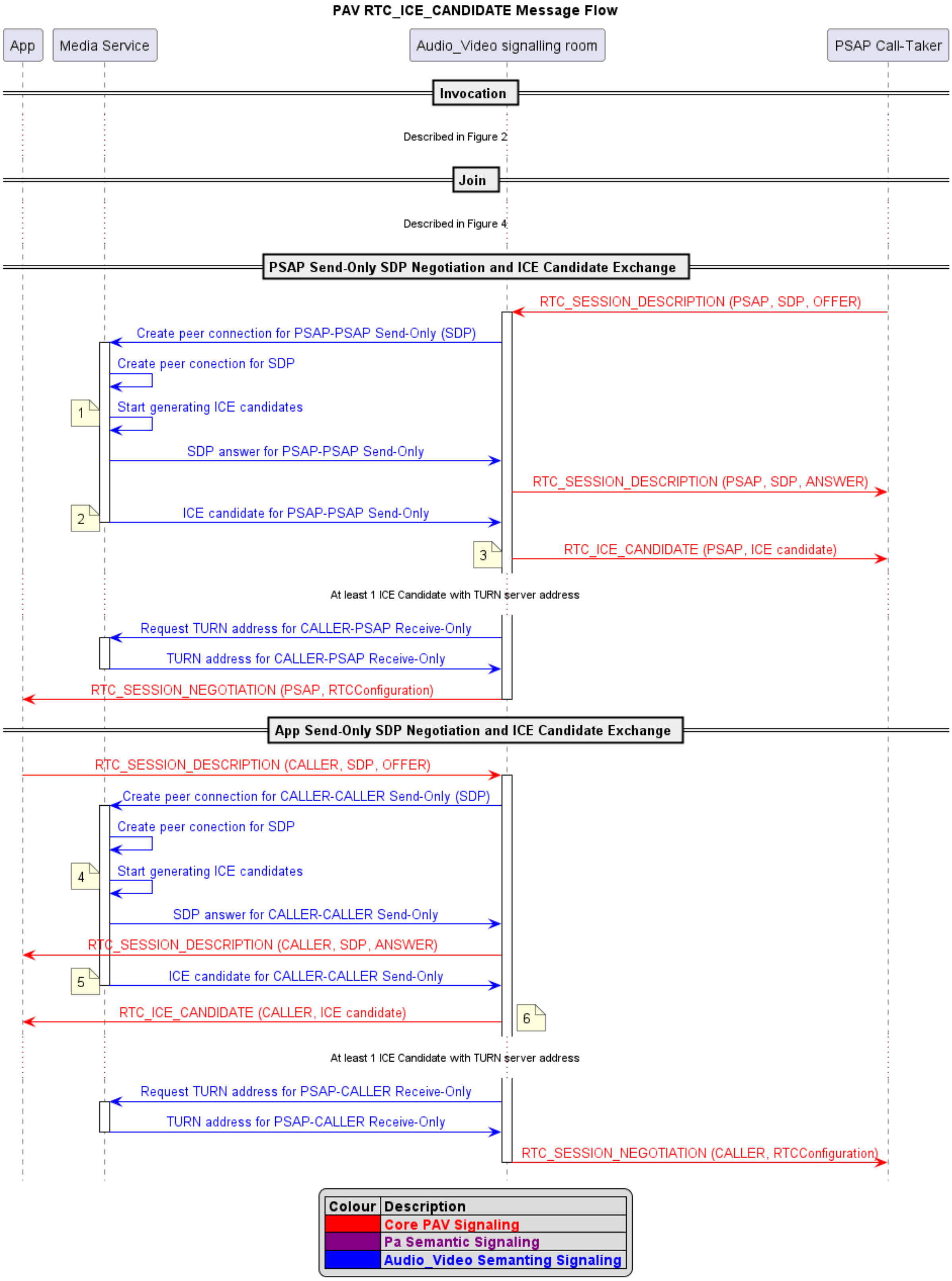
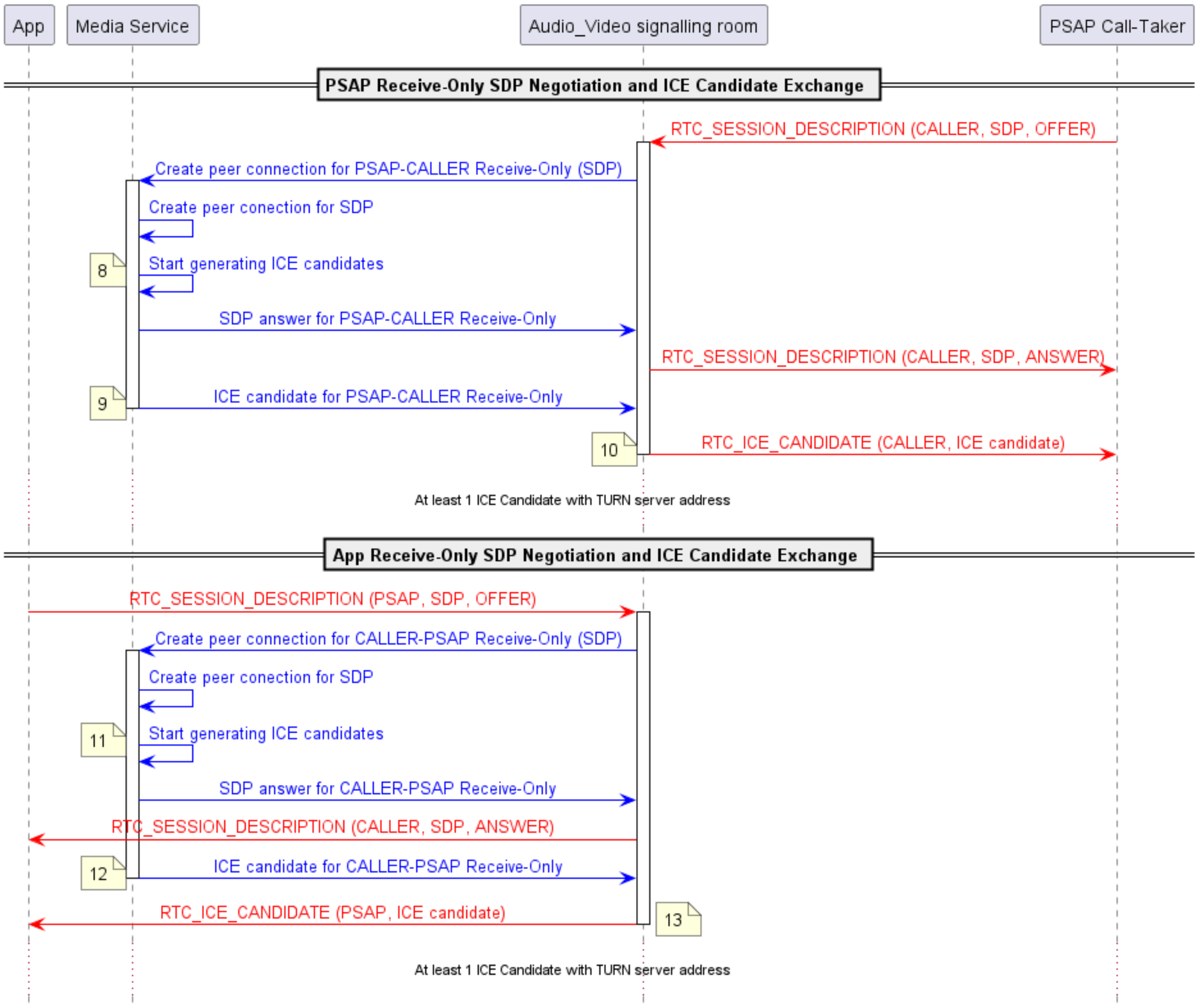


Figure 10: PAV RTC_ICE_CANDIDATE Message sequence diagram part 1



Colour	Description
	Core PAV Signaling
	Pa Semantic Signaling
	Audio_Video Semanting Signaling

Figure 11: PAV RTC_ICE_CANDIDATE Message sequence diagram part 2

1. After the creation of the peer-connection for the Send-Only stream of the PSAP using its SDP offer, the Media Service should start generating ICE candidates for that peer-connection. At least a valid ICE candidate with the TURN server address shall be generated.
2. The Media Service sends the ICE candidates to the Audio_Video signalling room.
3. For each ICE candidate received, the Audio_Video signalling room sends an RTC_ICE_CANDIDATE message to the PSAP Call-Taker containing the ICE candidate and the PSAP user identification. The PSAP Call-Taker shall use the ICE candidates received for its local peer-connection for its Send-Only stream.
4. After the creation of the peer-connection for the Send-Only stream of the CALLER using its SDP offer, the Media Service should start generating ICE candidates for that peer-connection. At least a valid ICE candidate with the TURN server address shall be generated.
5. The Media Service sends the ICE candidates to the Audio_Video signalling room.
6. For each ICE candidate received, the Audio_Video signalling room sends an RTC_ICE_CANDIDATE message to the App containing the ICE candidate and the CALLER user identification. The App shall use the ICE candidates received for its local peer-connection for its Send-Only stream.

7. After the creation of the peer-connection for the Receive-Only stream of the PSAP from the CALLER using its SDP offer, the Media Service should start generating ICE candidates for that peer-connection. At least a valid ICE candidate with the TURN server address shall be generated.
8. The Media Service sends the ICE candidates to the Audio_Video signalling room.
9. For each ICE candidate received, the Audio_Video signalling room sends an RTC_ICE_CANDIDATE message to the PSAP Call-Taker containing the ICE candidate and the CALLER user identification. The PSAP Call-Taker shall use the ICE candidates received for its local peer-connection for its Receive-Only stream for the CALLER.
10. After the creation of the peer-connection for the Receive-Only stream of the CALLER from the PSAP using its SDP offer, the Media Service should start generating ICE candidates for that peer-connection. At least a valid ICE candidate with the TURN server address shall be generated.
11. The Media Service sends the ICE candidates to the Audio_Video signalling room.
12. For each ICE candidate received, the Audio_Video signalling room sends an RTC_ICE_CANDIDATE message to the App containing the ICE candidate and the PSAP user identification. The App shall use the ICE candidates received for its local peer-connection for its Receive-Only stream for the PSAP.

15. User media

15.1. Overview

The PEMEA Audio_Video (PAV) architecture is described in clause 6 and includes a Media Server to which the participants establish their Send-Only streams to send its streams to the Audio_Video Server and their Receive-Only channels to receive individually the streams that are being sent by other participants to the Audio_Video Server. Once the SDP negotiation described in clause 13 and the ICE candidate exchange described in clause 14 have completed, media stream connectivity between the Apps and the media server is established.

Session participants communication is through the Audio_Video Server. Therefore, a participant only knows if other participants join or leave the Audio_Video signalling room as described in clause 11.2 and if other participants have negotiated their Send-Only stream with the Audio_Video Server as described in clause 13.3.

Improving the awareness of what other participants are doing is important in emergency calls. If a user is in a dangerous situation where he cannot make any noise, he could make a call and share its video, but the App will not emit any sound. Without the ability to know that the CALLER is not negotiating the Receive-Only streams, the PSAP Call-Taker could be confused about why the user is not answering back its questions. The audio, video, receiveAudio and receiveVideo properties defined in Table 3, Table 19 and Table 20 can be used to exchange information about what the participants are doing locally during the PAV session. These properties are exchanged initially when sending the JOIN message and can be changed during the call with USER_MEDIA messages.

15.2. User media properties

The audio property indicates whether the audio channel of the Send-Only stream that the participant has negotiated with the Audio_Video Server is being muted locally or not. True means that the stream has useful information, false means that it is being muted locally by the participant.

The video property indicates whether the video channel of the Send-Only stream that the participant has negotiated with the Audio_Video Server is being muted locally or not. True means that the stream has useful information, false means that it is being muted locally by the participant.

The receiveAudio property indicates whether the participant will play the audio channel of the Receive-Only streams of other participants. The participant can negotiate the Receive-Only channels or not, but the property tells other participants that the participant is not presenting to the user the audio streams. If this property is sent to false by a CALLER user, it means that the call is a silent-call, and the PSAP-CPE can know it as soon as the CALLER user has joined the Audio_Video signalling room.

The receiveVideo property indicates whether the participant will play the video channel of the Receive-Only streams of other participants. The participant can negotiate the Receive-Only channels or not, but the property tells other participants that the participant is not presenting to the user the video streams.

15.3. USER_MEDIA message flow

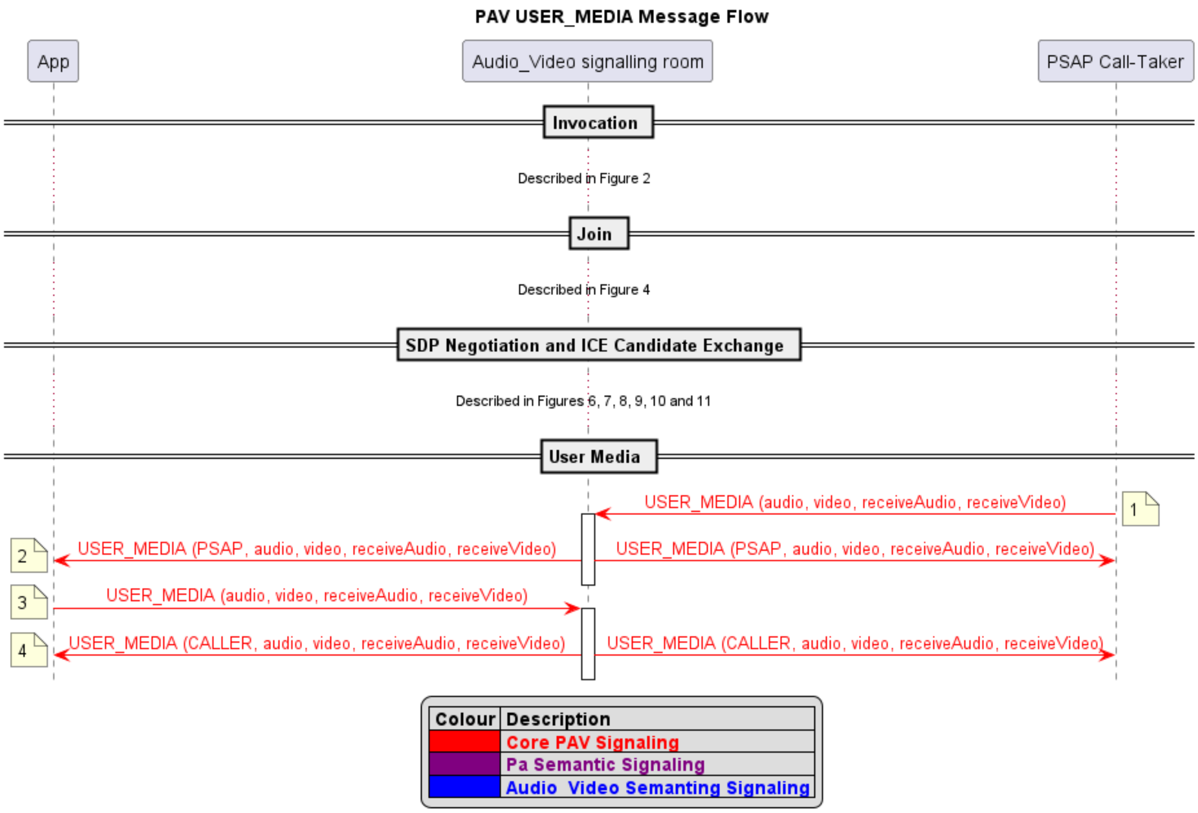


Figure 12: PAV USER_MEDIA Message sequence diagram

1. If the PSAP Call-Taker wants to communicate some modification in its media, it shall send a USER_MEDIA message to the Audio_Video signalling room.
2. When the Audio_Video signalling room receives a USER_MEDIA message it shall broadcast it to all the opened websocket connections that had performed the JOIN procedure described in clause 10.

3. If the App wants to communicate some modification in its media, it shall send a USER_MEDIA message to the Audio_Video signalling room.
4. When the Audio_Video signalling room receives a USER_MEDIA message it shall broadcast it to all the opened websocket connections that had performed the JOIN procedure described in clause 10.

16. Add participants to the call

The ability to add participants into the call is important as it brings additional expertise to the emergency situation that is able to assist in providing a successful outcome. The PEMEA Audio_Video (PAV) capability is based around WebRTC and the general architecture is provided in Figure 1. The general concept for linking in third-parties is that the PSAP Call-Taker instructs the PIM to obtain additional security access tokens for the Audio_Video signalling room and that the PSAP Call-Taker provides these, and the Audio_Video signalling URI, to the relevant third-parties. The subsequent JOIN procedure described in clause 10.2 and session establishment procedures described in clauses 13 and 14 are clearly normatively defined in the present document.

Acquisition and dissemination of the Audio_Video signalling room URI and additional security access tokens may vary widely in implementation based on the type of third-party and their Call-Taker equipment and whether policy, such as language or disability, is also applied at the third-party receiver to link in the correct personnel. Consequently, the present document does not go into further details on how this occurs, only that the system shall support the ability of the PSAP Call-Taker to request the additional tokens and provide a facility for the dissemination of this contact information to the relevant third-parties.

17. Audio_Video signalling room closure

The Audio_Video signalling room can be closed though the PIM by the PSAP-CPE. The present document does not instruct when the PIM should close the Audio_Video session and this decision shall be made by decision of the PSAP-CPE.

In most of the cases it can be done when the PSAP-Call-Taker considers the call to be terminated or when the App instructs the AP that the Caller wants to terminate the call, which will result in the AP invalidating the reach-back URI as described in clause 14.1.3 of ETSI TS 103 478 [R.1], but with the confirmation of the PSAP-CPE.

When the Audio_Video signalling room is closed by the PIM, the URI to access the Audio_Video signalling room shall be invalidated and the Audio_Video Server shall free all the resources associated with the corresponding Audio_Video signalling room.

18. Leaving the Audio_Video session

When a participant wants to leave the PAV session, he only needs to close the websocket connection with the Audio_Video signalling room and drop all media streams.

When the Audio_Video signalling room notices that a participant has left the room closing his websocket connection, it shall invalidate in the Media Server all the peer-connections related to that participant. It shall also send a USER_LIST message removing the user identification of the participant from the list of participants to let other participants know that he has left the Audio_Video session.

19. PEMEA Audio_Video message and type definitions

19.1. Overview

The PEMEA Audio_Video (PAV) protocol messages are defined as a series of JSON documents exchanged between the AP or PEMEA terminating node and an Audio_Video signalling room established inside the secure emergency network. The Audio_Video signalling room is established solely for communications with a single emergency session. Each emergency session requiring the use of the PAV service has its own Audio_Video signalling room created. Service and message exchanges between the AP and the App are not defined in the present document and are left to application implementers.

The JSON specifications for the messages are provided in ANNEX A. The subsequent sub-clauses in clauses 10, 11, 12, 13, 14 and 15 of the present document describe each of the PAV messages, its function, elements and any key constraints. Messages exchanges and procedures are specified in clause 7.

19.2. Data types

19.2.1.MessageType

The MessageType enumerable defines the "type" of message being sent.

Table 2: PEMEA Audio_Video message type values

Value	Description
JOIN	Message sent to the Audio_Video signalling room when the user wants to join the Audio_Video session.
USER_LIST	Message sent from the Audio_Video signalling room to all participants containing all users whenever a user enters or leaves the Audio_Video session.
RTC_SESSION_NEGOTIATION	Message from the Audio_Video signalling room to all participants with the information needed to communicate with other users.
RTC_SESSION_DESCRIPTION	Message sent either from a participant to the Audio_Video signalling room, or from the Audio_Video signalling room to all participants containing an SDP of a user.
RTC_ICE_CANDIDATE	Message sent either from a participant to the Audio_Video signalling room, or from the Audio_Video signalling room to all participants containing an ICE candidate of a user.
USER_MEDIA	Message sent either from a participant to the Audio_Video signalling room, or from the Audio_Video signalling room to all participants containing an update of a user's media information.
ERROR	Sent by the Audio_Video signalling room in case a JOIN request is received containing a name and role already in use by another connected user.

The participants leave the Audio_Video session by breaking their websocket connection to the Audio_Video signalling room, so no explicit leave message is defined for this protocol.

19.2.2.User

Defines a user in the Audio_Video signalling room. It is used in the USER_LIST messages to indicate how the participants are using the media information.

Table 3: User properties

Property	Type	Description
user	UserId	The participant information. Refer to Table 4.
audio	Boolean	Whether the user is sending audio stream or not. Can be useful to let other participants know that a user negotiated an audio stream but it is muting the stream by any reason.
video	Boolean	Whether the user is sending video stream or not. Can be useful to let other participants know that a user negotiated a video stream but it is muting the stream by any reason.
receiveAudio	Boolean	Whether the user's device is displaying the audio streams of other participants or not. Can be useful to inform Call-Takers if the Caller is in a danger situation and he is performing a silent call.
receiveVideo	Boolean	Whether the user's device is displaying the video streams of other participants or not. Can be useful to inform other participants that their video will not be displayed in the user's device.

19.2.3.UserId

Defines a user identification in the Audio_Video signalling room. It is used to identify the users in the room.

Table 4: UserId properties

Property	Type	Description
name	String	The name that the user wants to use. It could be a name "George", or a phone number "+34666554433", for example.
role	String	The role defines the type of user that is associated with the name. The recommended values are provided in Table 5.

Table 5: Role values

Value	Description
CALLER	The value sent by the AP to the Audio_Video signalling room and used to identify the user initiating the emergency communication to all other participants in the Audio_Video session.
PSAP	The value sent by the PSAP Call-Taker to the Audio_Video signalling room and used to identify the Call-Taker to all other participants in the Audio_Video session.
POLICE	If the police are linked into the Audio_Video signalling room then this value is sent by them to identify that police are in the session to all other participants in the Audio_Video session.
FIREFIGHTER	If the fire department are linked into the Audio_Video signalling room then this value is sent by them to identify that firefighters are in the session to all other participants in the Audio_Video session.
MED	If the ambulance or medical services are linked into the Audio_Video signalling room then this value is sent by them to identify that they are in the session to all other participants in the Audio_Video session.

19.2.4.RtcConfiguration

Defines the configuration of the TURN Server to configure peer connections.

Table 6: RtcConfiguration properties

Property	Type	Description
iceServers	Array<RtcIceServer>	The list of elements that describe TURN Servers that shall be used to ensure connectivity with a participant. Refer to Table 8.
iceTransportPolicy	String	The current ICE transport policy. The allowed values are provided in Table 7.

Table 7: IceTransportPolicy values

Value	Description
all	All ICE candidates will be considered.
relay	Only ICE candidates whose IP addresses are being relayed, such as those being passed through a TURN Server, will be considered. This is the recommended value to use, since a TURN Server would be needed in most scenarios due to PSAP network restrictions.

19.2.5.RtcIceServer

Defines a TURN Server address and credentials.

Table 8: RtcIceServer properties

Property	Type	Description
urls	Array<String>	The list of URLs that shall be used to connect to the TURN Server.
username	String	Optional. The username to use during the authentication process.
credential	String	Optional. The credential to use when logging into the TURN Server.

19.2.6.RtcSessionDescription

Defines the SDP negotiation between peers. Each RtcSessionDescription consists of a description type indicating which part of the offer/answer negotiation process it describes and of the SDP descriptor of the media session.

Table 9: RtcSessionDescription properties

Property	Type	Description
type	RtcSdpType	Describes the SDP type. The allowed values are provided in Table 10.
sdp	String	The Session Description Protocol (SDP) as specified in IETF RFC 4566 [R.13] and in IETF RFC 8839 [R.6].

Table 10: RtcSdpType values

Value	Description
offer	The SDP describes the initial proposal in an offer/answer exchange.
answer	The SDP describes the agreed-upon configuration, and is being sent to finalize the negotiation.

19.2.7.RtcIceCandidate

Defines the SDP negotiation between peers. Each RtcSessionDescription consists of a description type indicating which part of the offer/answer negotiation process it describes and of the SDP descriptor of the media session.

Table 11: RtcIceCandidate properties

Property	Type	Description
candidate	String	Describes the Interactive Connectivity Establishment (ICE) candidate as specified in IETF RFC 8445 [R.5] and using Trickle ICE which is specified in IETF RFC 8838 [R.12].
sdpMLineIndex	Integer	The zero-based index of the m-line within the SDP of the media description (as defined in IETF RFC 4566 [R.13]) with which the ICE candidate is associated.
sdpMid	String	Optional. The identification tag of the media stream with which the ICE candidate is associated.
usernameFragment	String	Optional. The username fragment ("ufrag") that uniquely identifies a single ICE interaction session.

19.3. JOIN message

19.3.1.Message overview

The JOIN message is the message sent from the participants to the Audio_Video signalling room when the user wants to join the session. This may be the AP, the PSAP-CPE or another trusted user.

Table 12: JOIN message properties

Property	Type	Description
type	String	Type of the message. The value is "JOIN". Refer to Table 2.
user	PartialUserId	The information A list of User elements containing the information of the users in the Audio_Video signalling room. Refer to Table 3.
audio	Boolean	Whether the user is sending audio stream or not. Can be useful to let other participants know that a user negotiated an audio stream but it is muting the stream by any reason.
video	Boolean	Whether the user is sending video stream or not. Can be useful to let other participants know that a user negotiated a video stream but it is muting the stream by any reason.
receiveAudio	Boolean	Whether the user's device is displaying the audio streams of other participants or not. Can be useful to inform Call-Takers if the Caller is in a danger situation and he is performing a silent call.
receiveVideo	Boolean	Whether the user's device is displaying the video streams of other participants or not. Can be useful to inform other participants that their video will not be displayed in the user's device.
timestamp	Integer	Time at which the message was created by the participant. Number of milliseconds from epoch (00:00:00:00 1st January 1970).

19.3.2.Examples

```
{
    "type":"JOIN",
    "user":{
        "name":"PSAP 1",
        "role":"PSAP"
    },
    "audio":true,
    "video":true,
    "receiveAudio":true,
    "receiveVideo":true,
    "timestamp":1683893671026
}
```

19.4. USER_LIST message

19.4.1.Message overview

The USER_LIST message is sent to all participants in the Audio_Video signalling room whenever a user enters and leaves the Audio_Video signalling room.

It is considered that a user enters the room after having successfully opened the websocket connection and successfully sent a JOIN message.

It is considered that a user leaves the Audio_Video signalling room after the websocket connection is closed as defined in IETF RFC 6455 [R.4].

Table 13: USER_LIST message properties

Property	Type	Description
type	String	Type of the message. The value is "USER_LIST". Refer to Table 2.
users	Array<User>	A list of User elements containing the information of the users in the Audio_Video signalling room. Refer to Table 3.
timestamp	Integer	Time at which the message was created at the Audio_Video signalling room. Number of milliseconds from epoch (00:00:00:00 1st January 1970).

Each participant in the Audio_Video signalling room is required to keep a list of the participants so that it knows when participants join and leave the session.

19.4.2. Examples


```
{
    "type":"USER_LIST",
    "users":[
        {
            "user":{
                "name":"+34611223344",
                "role":"CALLER"
            },
            "audio":true,
            "video":true,
            "receiveAudio":true,
            "receiveVideo":true
        },
        {
            "user":{
                "name":"PSAP 1",
                "role":"PSAP"
            },
            "audio":true,
            "video":true,
            "receiveAudio":true,
            "receiveVideo":true
        }
    ],
    "timestamp":1574092280231
}
```


19.5. RTC_SESSION_NEGOTIATION message

19.5.1. Message overview

The RTC_SESSION_NEGOTIATION message is sent from the Audio_Video signalling room to the participants. It is used to exchange the TURN Server address so that it is used during the ICE connectivity checks in order to reach a common point where the participants can communicate.

When a participant sends a successful JOIN message to the Audio_Video signalling room, the Audio_Video server shall obtain the TURN address that the user shall use for his Send-Only stream channel and send the information to the participant with an RTC_SESSION_NEGOTIATION message.

Each time that other participants negotiate their Send-Only stream channels, the Audio_Video signalling room shall send an RTC_SESSION_NEGOTIATION message to the rest of the users so that them can configure their Receive-Only channels for the new stream.

If a participant receives an RTC_SESSION_NEGOTIATION message for a user for which he already had a Receive-Only connection, he can close the connection and negotiate a new Receive-Only stream, but it is not mandatory because the Media Service shall interconnect the new Send-Only stream with the Receive-Only streams for that participant that other participants has already negotiated.

Table 14: RTC_SESSION_NEGOTIATION message properties

Property	Type	Description
type	String	Type of the message. The value is "RTC_SESSION_NEGOTIATION". Refer to Table 2.
user	UserId	The user for whom the configuration received in the message shall be used when making the SDP negotiation and the ICE candidates exchange. Refer to Table 4.
configuration	RtcConfiguration	The TURN Server address that shall be used when generating the SDP and the ICE candidates for the peer-connection for the user of the message.
timestamp	Integer	Time at which the message was created at the Audio_Video signalling room. Number of milliseconds from epoch (00:00:00:00 1st January 1970).

19.5.2.Examples


```
{
    "type":"RTC_SESSION_NEGOTIATION",
    "configuration":{
        "iceServers":[
            {
                "credential":"NYRkaIcQ3vJAa50M7uWClHvYpqM=",
                "urls":[
                    "turn:36.181.54.52:3478?transport=udp"
                ],
                "username":"1683893970:Jorge_PSAP"
            }
        ],
        "iceTransportPolicy":"relay"
    },
    "timestamp":1683893670757,
    "user": {
        "name":"PSAP 1",
        "role":"PSAP"
    }
}
```


19.6. RTC_SESSION_DESCRIPTION message

19.6.1.Message overview

RTC_SESSION_DESCRIPTION messages are sent from the participants to the Audio_Video signalling room when using type "offer", and are sent in response from the Audio_Video signalling room back to the participants using type "answer".

They are used to exchange the SDP from the participants to the Media Server to negotiate the Send-Only and Receive-Only streams. When the Audio_Video signalling room receives a successful RTC_SESSION_DESCRIPTION "offer" message from a participant indicating for which user the peer-connection want to be established, it shall create a peer-connection in the Media Service, generate an SDP answer, and send it in an RTC_SESSION_DESCRIPTION "answer" message.

Table 15: Participant RTC_SESSION_DESCRIPTION message properties

Property	Type	Description
type	String	Type of the message. The value is "RTC_SESSION_DESCRIPTION". Refer to Table 2.
user	UserId	The user with whom the SDP negotiation shall be done. Refer to Table 4.
description	RtcSessionDescription	The Session Description Protocol (SDP) that is being sent with the message. Refer to Table 9.
timestamp	Integer	Time at which the message was created by the participant. Number of milliseconds from epoch (00:00:00:00 1st January 1970).

When the Audio_Video signalling room answers an RTC_SESSION_DESCRIPTION "offer", it shall include the time when the RTC_SESSION_DESCRIPTION "answer" is created.

Table 16: Audio_Video signalling room CHANGE_PERMISSIONS message properties

Property	Type	Description
type	String	Type of the message. The value is "RTC_SESSION_DESCRIPTION". Refer to Table 2.
user	UserId	The user with whom the SDP negotiation shall be done. Refer to Table 4.
description	RtcSessionDescription	The Session Description Protocol (SDP) that is being sent with the message. Refer to Table 9.
timestamp	Integer	Time at which the message was created at the Audio_Video signalling room. Number of milliseconds from epoch (00:00:00:00 1st January 1970).

19.6.2.Examples

Message sent from a participant to the Audio_Video signalling room:


```
{
    "type":"RTC_SESSION_DESCRIPTION",
    "user":{
        "name":"PSAP 1",
        "role":"PSAP"
    },
    "description":{
        "type":"offer",
        "sdp":"v=0\r\no=- 3289333057432362461 2 IN IP4 127.0.0.1\r\ns=-\r\nt=0
0\r\na=group:BUNDLE 0 1\r\na=extmap-allow-mixed\r\na=msid-semantic: WMS cc102b2a-82f5-4764-
9f7c-6413e91795e8\r\nm=audio 9 UDP/TLS/RTP/SAVPF 111 63 9 0 8 13 110 126\r\nc=IN IP4
0.0.0.0\r\na=rtcp:9 IN IP4 0.0.0.0\r\na=ice-ufrag:Ylsq\r\na=ice-
pwd:Ao+gJF32TQPVigZoqsfNjCIu\r\na=ice-options:trickle\r\na=fingerprint:sha-256
B0:71:50:15:2C:87:35:14:1A:8D:E6:1B:E5:F3:7A:5D:93:00:0C:75:6E:0C:75:1C:23:AB:AD:1E:27:56:92:
0C\r\na=setup:actpass\r\na=mid:0\r\na=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-
level\r\na=extmap:2 http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time\r\na=extmap:3
http://www.ietf.org/id/draft-holmer-rmcat-transport-wide-cc-extensions-01\r\na=extmap:4
urn:ietf:params:rtp-hdrext:sdes:mid\r\na=sendonly\r\na=msid:cc102b2a-82f5-4764-9f7c-
6413e91795e8 eaea309b-b962-48fd-a212-9d84d7998b27\r\na=rtcp-mux\r\na=rtpmap:111
opus/48000/2\r\na=rtcp-fb:111 transport-cc\r\na=fmtp:111
minptime=10;useinbandfec=1\r\na=rtpmap:63 red/48000/2\r\na=fmtp:63 111/111\r\na=rtpmap:9
G722/8000\r\na=rtpmap:0 PCMU/8000\r\na=rtpmap:8 PCMA/8000\r\na=rtpmap:13
CN/8000\r\na=rtpmap:110 telephone-event/48000\r\na=rtpmap:126 telephone-
event/8000\r\na=ssrc:776783554 cname:DG2HEjCJnqk12I+N\r\na=ssrc:776783554 msid:cc102b2a-82f5-
4764-9f7c-6413e91795e8 eaea309b-b962-48fd-a212-9d84d7998b27\r\nm=video 9 UDP/TLS/RTP/SAVPF 96
97 102 103 104 105 106 107 108 109 127 125 39 40 45 46 98 99 100 101 112 113 114\r\nc=IN IP4
0.0.0.0\r\na=rtcp:9 IN IP4 0.0.0.0\r\na=ice-ufrag:Ylsq\r\na=ice-
pwd:Ao+gJF32TQPVigZoqsfNjCIu\r\na=ice-options:trickle\r\na=fingerprint:sha-256
B0:71:50:15:2C:87:35:14:1A:8D:E6:1B:E5:F3:7A:5D:93:00:0C:75:6E:0C:75:1C:23:AB:AD:1E:27:56:92:
0C\r\na=setup:actpass\r\na=mid:1\r\na=extmap:14 urn:ietf:params:rtp-
hdrext:toffset\r\na=extmap:2 http://www.webrtc.org/experiments/rtp-hdrext/abs-send-
time\r\na=extmap:13 urn:3gpp:video-orientation\r\na=extmap:3 http://www.ietf.org/id/draft-
holmer-rmcat-transport-wide-cc-extensions-01\r\na=extmap:5
http://www.webrtc.org/experiments/rtp-hdrext/playout-delay\r\na=extmap:6
http://www.webrtc.org/experiments/rtp-hdrext/video-content-type\r\na=extmap:7
http://www.webrtc.org/experiments/rtp-hdrext/video-timing\r\na=extmap:8
http://www.webrtc.org/experiments/rtp-hdrext/color-space\r\na=extmap:4 urn:ietf:params:rtp-
hdrext:sdes:mid\r\na=extmap:10 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id\r\na=extmap:11
urn:ietf:params:rtp-hdrext:sdes:repaired-rtp-stream-id\r\na=sendonly\r\na=msid:cc102b2a-82f5-
4764-9f7c-6413e91795e8 20cbfead-f1b8-4205-9ae8-9f502047c302\r\na=rtcp-mux\r\na=rtcp-
rsize\r\na=rtpmap:96 VP8/90000\r\na=rtcp-fb:96 goog-remb\r\na=rtcp-fb:96 transport-
cc\r\na=rtcp-fb:96 ccm fir\r\na=rtcp-fb:96 nack\r\na=rtcp-fb:96 nack pli\r\na=rtpmap:97
rtx/90000\r\na=fmtp:97 apt=96\r\na=rtpmap:102 H264/90000\r\na=rtcp-fb:102 goog-
remb\r\na=rtcp-fb:102 transport-cc\r\na=rtcp-fb:102 ccm fir\r\na=rtcp-fb:102 nack\r\na=rtcp-
fb:102 nack pli\r\na=fmtp:102 level-asymmetry-allowed=1;packetization-mode=1;profile-level-
id=42001f\r\na=rtpmap:103 rtx/90000\r\na=fmtp:103 apt=102\r\na=rtpmap:104
H264/90000\r\na=rtcp-fb:104 goog-remb\r\na=rtcp-fb:104 transport-cc\r\na=rtcp-fb:104 ccm
fir\r\na=rtcp-fb:104 nack\r\na=rtcp-fb:104 nack pli\r\na=fmtp:104 level-asymmetry-
allowed=1;packetization-mode=0;profile-level-id=42001f\r\na=rtpmap:105
rtx/90000\r\na=fmtp:105 apt=104\r\na=rtpmap:106 H264/90000\r\na=rtcp-fb:106 goog-
remb\r\na=rtcp-fb:106 transport-cc\r\na=rtcp-fb:106 ccm fir\r\na=rtcp-fb:106 nack\r\na=rtcp-
fb:106 nack pli\r\na=fmtp:106 level-asymmetry-allowed=1;packetization-mode=1;profile-level-
id=42e01f\r\na=rtpmap:107 rtx/90000\r\na=fmtp:107 apt=106\r\na=rtpmap:108
```

```
H264/90000\r\na=rtcp-fb:108 goog-remb\r\na=rtcp-fb:108 transport-cc\r\na=rtcp-fb:108 ccm
fir\r\na=rtcp-fb:108 nack\r\na=rtcp-fb:108 nack pli\r\na=fmtp:108 level-asymmetry-
allowed=1;packetization-mode=0;profile-level-id=42e01f\r\na=rtpmap:109
rtx/90000\r\na=fmtp:109 apt=108\r\na=rtpmap:127 H264/90000\r\na=rtcp-fb:127 goog-
remb\r\na=rtcp-fb:127 transport-cc\r\na=rtcp-fb:127 ccm fir\r\na=rtcp-fb:127 nack\r\na=rtcp-
fb:127 nack pli\r\na=fmtp:127 level-asymmetry-allowed=1;packetization-mode=1;profile-level-
id=4d001f\r\na=rtpmap:125 rtx/90000\r\na=fmtp:125 apt=127\r\na=rtpmap:39
H264/90000\r\na=rtcp-fb:39 goog-remb\r\na=rtcp-fb:39 transport-cc\r\na=rtcp-fb:39 ccm
fir\r\na=rtcp-fb:39 nack\r\na=rtcp-fb:39 nack pli\r\na=fmtp:39 level-asymmetry-
allowed=1;packetization-mode=0;profile-level-id=4d001f\r\na=rtpmap:40 rtx/90000\r\na=fmtp:40
apt=39\r\na=rtpmap:45 AV1/90000\r\na=rtcp-fb:45 goog-remb\r\na=rtcp-fb:45 transport-
cc\r\na=rtcp-fb:45 ccm fir\r\na=rtcp-fb:45 nack\r\na=rtcp-fb:45 nack pli\r\na=rtpmap:46
rtx/90000\r\na=fmtp:46 apt=45\r\na=rtpmap:98 VP9/90000\r\na=rtcp-fb:98 goog-remb\r\na=rtcp-
fb:98 transport-cc\r\na=rtcp-fb:98 ccm fir\r\na=rtcp-fb:98 nack\r\na=rtcp-fb:98 nack
pli\r\na=fmtp:98 profile-id=0\r\na=rtpmap:99 rtx/90000\r\na=fmtp:99 apt=98\r\na=rtpmap:100
VP9/90000\r\na=rtcp-fb:100 goog-remb\r\na=rtcp-fb:100 transport-cc\r\na=rtcp-fb:100 ccm
fir\r\na=rtcp-fb:100 nack\r\na=rtcp-fb:100 nack pli\r\na=fmtp:100 profile-
id=2\r\na=rtpmap:101 rtx/90000\r\na=fmtp:101 apt=100\r\na=rtpmap:112
red/90000\r\na=rtpmap:113 rtx/90000\r\na=fmtp:113 apt=112\r\na=rtpmap:114
ulpfec/90000\r\na=ssrc-group:FID 660247603 365749183\r\na=ssrc:660247603
cname:DG2HEjCJnqk12I+N\r\na=ssrc:660247603 msid:cc102b2a-82f5-4764-9f7c-6413e91795e8
20cbfead-f1b8-4205-9ae8-9f502047c302\r\na=ssrc:365749183
cname:DG2HEjCJnqk12I+N\r\na=ssrc:365749183 msid:cc102b2a-82f5-4764-9f7c-6413e91795e8
20cbfead-f1b8-4205-9ae8-9f502047c302\r\n"
    },
    "timestamp":1683893671026
}
```


Message sent from the Audio_Video signalling room to a participant:


```
{
    "type":"RTC_SESSION_DESCRIPTION",
    "user":{
        "name":"PSAP 1",
        "role":"PSAP"
    },
    "description":{
        "sdp":"v=0\r\no=- 3892882470 3892882470 IN IP4 0.0.0.0\r\ns=Kurento Media
Server\r\nc=IN IP4 0.0.0.0\r\nt=0 0\r\na=extmap-allow-mixed:\r\na=msid-semantic: WMS
cc102b2a-82f5-4764-9f7c-6413e91795e8\r\na=group:BUNDLE 0 1\r\nm=audio 1 UDP/TLS/RTP/SAVPF 111
0\r\na=extmap:2 http://www.webrtc.org/experiments/rtp-hdrext/abs-send-
time\r\na=recvonly\r\na=mid:0\r\na=rtcp:9 IN IP4 0.0.0.0\r\na=rtpmap:111
opus/48000/2\r\na=rtpmap:0 PCMU/8000\r\na=setup:active\r\na=rtcp-mux\r\na=fmtp:111
minptime=10;useinbandfec=1\r\na=ssrc:393676185 cname:user364753389@host-19fe0ed2\r\na=ice-
ufrag:b1dw\r\na=ice-pwd:OkZapSr9R64sYbtQD/uKXf\r\na=fingerprint:sha-256
A4:F0:50:57:63:22:CF:DC:13:71:83:9C:EB:C4:A6:44:17:B9:69:52:F5:0F:62:3A:BF:02:10:4E:29:BA:07:
00\r\nm=video 1 UDP/TLS/RTP/SAVPF 96 102 104 106 108 127 39\r\na=extmap:2
http://www.webrtc.org/experiments/rtp-hdrext/abs-send-
time\r\na=recvonly\r\na=mid:1\r\na=rtcp:9 IN IP4 0.0.0.0\r\na=rtpmap:96
VP8/90000\r\na=rtpmap:102 H264/90000\r\na=rtpmap:104 H264/90000\r\na=rtpmap:106
H264/90000\r\na=rtpmap:108 H264/90000\r\na=rtpmap:127 H264/90000\r\na=rtpmap:39
H264/90000\r\na=rtcp-fb:96 goog-remb\r\na=rtcp-fb:96 ccm fir\r\na=rtcp-fb:96 nack\r\na=rtcp-
fb:96 nack pli\r\na=rtcp-fb:102 goog-remb\r\na=rtcp-fb:102 ccm fir\r\na=rtcp-fb:102
nack\r\na=rtcp-fb:102 nack pli\r\na=rtcp-fb:104 goog-remb\r\na=rtcp-fb:104 ccm fir\r\na=rtcp-
fb:104 nack\r\na=rtcp-fb:104 nack pli\r\na=rtcp-fb:106 goog-remb\r\na=rtcp-fb:106 ccm
fir\r\na=rtcp-fb:106 nack\r\na=rtcp-fb:106 nack pli\r\na=rtcp-fb:108 goog-remb\r\na=rtcp-
fb:108 ccm fir\r\na=rtcp-fb:108 nack\r\na=rtcp-fb:108 nack pli\r\na=rtcp-fb:127 goog-
remb\r\na=rtcp-fb:127 ccm fir\r\na=rtcp-fb:127 nack\r\na=rtcp-fb:127 nack pli\r\na=rtcp-fb:39
goog-remb\r\na=rtcp-fb:39 ccm fir\r\na=rtcp-fb:39 nack\r\na=rtcp-fb:39 nack
pli\r\na=setup:active\r\na=rtcp-mux\r\na=fmtp:102 level-asymmetry-allowed=1;packetization-
mode=1;profile-level-id=42001f\r\na=fmtp:104 level-asymmetry-allowed=1;packetization-
mode=0;profile-level-id=42001f\r\na=fmtp:106 level-asymmetry-allowed=1;packetization-
mode=1;profile-level-id=42e01f\r\na=fmtp:108 level-asymmetry-allowed=1;packetization-
mode=0;profile-level-id=42e01f\r\na=fmtp:127 level-asymmetry-allowed=1;packetization-
mode=1;profile-level-id=4d001f\r\na=fmtp:39 level-asymmetry-allowed=1;packetization-
mode=0;profile-level-id=4d001f\r\na=ssrc:2145314644 cname:user364753389@host-
19fe0ed2\r\na=ice-ufrag:b1dw\r\na=ice-pwd:OkZapSr9R64sYbtQD/uKXf\r\na=fingerprint:sha-256
A4:F0:50:57:63:22:CF:DC:13:71:83:9C:EB:C4:A6:44:17:B9:69:52:F5:0F:62:3A:BF:02:10:4E:29:BA:07:
00\r\n,
        "type":"answer"
    },
    "timestamp":1683893670891
}
```

19.7. RTC_ICE_CANDIDATE message

19.7.1.Message overview

RTC_ICE_CANDIDATE messages are sent from the participants to the Audio_Video signalling room after having sent an RTC_SESSION_DESCRIPTION "offer" message, and are sent from the Audio_Video signalling room after having answered with an RTC_SESSION_DESCRIPTION "answer" message.

They are used to exchange the ICE candidates from the participants to the Media Server and vice-versa. These ICE candidates shall be used in conjunction with the associated SDP sent over an RTC_SESSION_DESCRIPTION message to reach a common connectivity point where the media flow will occur.

The PEMEA Audio_Video capability uses Trickle ICE as defined in IETF RFC 8838 [R.12] to accelerate the ICE connectivity checks in order to establish the communication channels faster. To do that, the SDP and the ICE candidates are exchanged over different messages, and have to be paired together in their associated peer-connections. After a participant sends an RTC_SESSION_DESCRIPTION message to negotiate the stream with another participant, it shall start collecting the ICE candidates for that connection and sending them over RTC_ICE_CANDIDATE messages to the Audio_Video signalling room using the same user property in both RTC_SESSION_DESCRIPTION "offer" and RTC_ICE_CANDIDATE messages to allow the Audio_Video signalling room to correlate the data.

Table 17: Participant RTC_ICE_CANDIDATE message properties

Property	Type	Description
type	String	Type of the message. The value is "RTC_ICE_CANDIDATE". Refer to Table 2.
user	UserId	The user with whom the ICE exchange shall be done. Refer to Table 4.
candidate	RtcConfiguration	The Interactive Connectivity Establishment (ICE) candidate that is being sent with the message. Refer to Table 11.
timestamp	Integer	Time at which the message was created by the participant. Number of milliseconds from epoch (00:00:00:00 1st January 1970).

When the Audio_Video signalling room answers a participant's RTC_SESSION_DESCRIPTION "offer" message with an RTC_SESSION_DESCRIPTION "answer", the Audio_Video signalling room shall request the Media Service to start generating ICE candidates for that peer-connection. These ICE candidates shall be sent to the participant that sent the RTC_SESSION_DESCRIPTION "offer" message and shall use the same user property in both RTC_SESSION_DESCRIPTION "answer" and RTC_ICE_CANDIDATE messages to allow the participant to correlate the data.

Table 18: Audio_Video signalling room RTC_ICE_CANDIDATE message properties

Property	Type	Description
type	String	Type of the message. The value is "RTC_ICE_CANDIDATE". Refer to Table 2.
user	UserId	The user with whom the ICE exchange shall be done. Refer to Table 4.
candidate	RtcConfiguration	The Interactive Connectivity Establishment (ICE) candidate that is being sent with the message. Refer to Table 11.
timestamp	Integer	Time at which the message was created at the Audio_Video signalling room. Number of milliseconds from epoch (00:00:00:00 1st January 1970).

19.7.2.Examples

Message sent from a participant to the Audio_Video signalling room:

```
{
    "type":"RTC_ICE_CANDIDATE",
    "user":{
```

```
        "name":"PSAP 1",
        "role":"PSAP"
    },
    "candidate":{
        "candidate":"candidate:943110355 1 udp 41885695 36.181.54.52 51484 typ relay raddr
0.0.0.0 rport 0 generation 0 ufrag Ylsq network-id 1 network-cost 10",
        "sdpMLineIndex":0,
        "sdpMid":"0"
    },
    "timestamp":1683893671026
}
```

Message sent from the Audio_Video signalling room to a participant:

```
{
    "type":"RTC_ICE_CANDIDATE",
    "user":{
        "name":"PSAP 1",
        "role":"PSAP"
    },
    "candidate":{
        "candidate": "candidate:7 1 UDP 505413887 36.181.54.52 61335 typ relay raddr
10.50.0.239 rport 5719",
        "sdpMLineIndex":0,
        "sdpMid":"0"
    },
    "timestamp":1683893671026
}
```

19.8. USER_MEDIA message

19.8.1.Message overview

The USER_MEDIA message is sent from the participants to the Audio_Video signalling room, and when the Audio_Video signalling room receives a USER_MEDIA message it broadcasts it to all the participants of the room.

It is used to inform other participant about what a participant is doing with the streams of the Audio_Video session. Without this message, there is no way of knowing if a participant of the room is not using the video streams because he only wants to display audio, or to inform other users that a user is muting its already negotiated Send-Only video stream locally.

It is purely informative information that participants can use to be better aware of the situation in the call.

Table 19: Participant USER_MEDIA message properties

Property	Type	Description
type	String	Type of the message. The value is "USER_MEDIA". Refer to Table 2.
audio	Boolean	Whether the user is sending audio stream or not. Can be useful to let other participants know that the user negotiated an audio stream but it is muting the stream by any reason.
video	Boolean	Whether the user is sending video stream or not. Can be useful to let other participants know that the user negotiated a video stream but it is muting the stream by any reason.
receiveAudio	Boolean	Whether the user's device is displaying the audio streams of other participants or not. Can be useful to inform Call-Takers if the Caller is in a danger situation and he is performing a silent call.
receiveVideo	Boolean	Whether the user's device is displaying the video streams of other participants or not. Can be useful to inform other participants that their video will not be displayed in the user's device.
timestamp	Integer	Time at which the message was created by the participant. Number of milliseconds from epoch (00:00:00:00 1st January 1970).

The Audio_Video signalling room is responsible to send the received USER_MEDIA messages to the connected room participants though their websocket connections. It adds to the messages the user that sent the message, which identity is known by the Audio_Video signalling room as it authenticated the connection with the provided token. The Audio_Video signalling room also add the time at which the message was received.

Table 20: Audio_Video signalling room USER_MEDIA message properties

Property	Type	Description
type	String	Type of the message. The value is "USER_MEDIA". Refer to Table 2.
audio	Boolean	Whether the user is sending audio stream or not. Can be useful to let other participants know that the user negotiated an audio stream but it is muting the stream by any reason.
video	Boolean	Whether the user is sending video stream or not. Can be useful to let other participants know that the user negotiated a video stream but it is muting the stream by any reason.
receiveAudio	Boolean	Whether the user's device is displaying the audio streams of other participants or not. Can be useful to inform Call-Takers if the Caller is in a danger situation and he is performing a silent call.
receiveVideo	Boolean	Whether the user's device is displaying the video streams of other participants or not. Can be useful to inform other participants that their video will not be displayed in the user's device.
user	UserId	The user that sent the message. Refer to Table 4.
timestamp	Integer	Time at which the message was received at the Audio_Video signalling room. Number of milliseconds from epoch (00:00:00:00 1st January 1970).

19.8.2.Examples

Message sent from a participant to the Audio_Video signalling room:

```
{
    "type":"USER_MEDIA",
    "audio":true,
    "video":false,
    "receiveAudio":true,
    "receiveVideo":true,
    "timestamp":1683893671026
}
```

Message sent from the Audio_Video signalling room to a participant:

```
{
    "type":"USER_MEDIA",
    "user":{
        "name":"PSAP 1",
        "role":"PSAP"
    },
    "audio":true,
    "video":true,
    "receiveAudio":true,
    "receiveVideo":true,
    "timestamp":1683893671707
}
```

19.9. ERROR message

19.9.1.Message overview

The ERROR message is sent by the Audio_Video signalling room when the Audio_Video signalling server identifies a problem in the session. The message is not intended to be visible to the end-user but to be received by the systems in order to take the appropriate actions depending on the error.

Table 21: Error message properties

Property	Type	Description
type	String	Type of the message. The value is "ERROR". Refer to Table 2.
code	Integer	Status code of the error.
reason	String	Property containing text describing the problem.

19.9.2. Error message example


```
{
    "type":"ERROR",
    "code":400
    "reason":"\"user.role\" is required"
}
```

ANNEX A PEMEA Audio_Video JSON schema

A.1 General

This normative annex includes all of the JSON schema necessary to implement the present document.

A.2 Audio_Video invocation schema

This schema is used by the PIM/tPSP to invoke the Audio_Video capability in the AP.

```
{
	"$schema": "http://json-schema.org/draft-07/schema",
	"type": "object",
	"title": "Audio_Video invocation schema",
	"properties": {
		"token": {
			"type": "string"
		},
		"uri": {
			"type": "string",
			"format": "uri"
		},
		"expiry": {
			"type": "number"
		}
	},
	"required": ["token", "uri", "expiry"]
}
```

A.3 JOIN schema

This schema specifies the JOIN message.

```
{
	"$schema": "http://json-schema.org/draft-07/schema",
	"type": "object",
	"title": "Audio_Video JOIN message Schema",
	"properties": {
		"type": {
			"const": "JOIN"
		},
		"user": {
			"$ref": "#/definitions/userId"
		},
		"audio": {
			"type": "boolean",
			"default": true
		},
		"video": {
			"type": "boolean",
			"default": true
		},
		"receiveAudio": {
			"type": "boolean",
			"default": true
		},
		"receiveVideo": {
			"type": "boolean",
			"default": true
```

```
			},
			"timestamp": {
				"type": "number"
			}
		},
		"definitions": {
			"userId": {
				"type": "object",
				"properties": {
					"name": {
						"type": "string"
					},
					"role": {
						"type": "string"
					}
				},
				"required": ["name", "role"]
			}
		},
		"required": ["type", "user", "timestamp"]
}
```

A.4 USER_LIST schema

This schema specifies the USER_LIST message.

```
{
		"$schema": "http://json-schema.org/draft-07/schema",
		"type": "object",
		"title": "Audio_Video USER_LIST message Schema",
		"properties": {
			"type": {
				"const": "USER_LIST"
			},
			"users": {
				"type": "array",
				"items": {
					"$ref": "#/definitions/user"
				}
			},
			"timestamp": {
				"type": "number"
			}
		},
		"definitions": {
			"user": {
				"type": "object",
				"properties": {
					"user": {
						"$ref": "#/definitions/userId"
					},
					"audio": {
						"type": "boolean"
					},
					"video": {
						"type": "boolean"
					},
					"receiveAudio": {
						"type": "boolean"
					},
					"receiveVideo": {
						"type": "boolean"
					}
				},
				"required": ["user", "audio", "video"]
```

```
            },
            "userId": {
                  "type": "object",
                  "properties": {
                        "name": {
                              "type": "string"
                        },
                        "role": {
                              "type": "string"
                        }
                  },
                  "required": ["name", "role"]
            }
      },
      "required": ["type", "users", "timestamp"]
}
```

A.5 RTC_SESSION_NEGOTIATION schema

This schema specifies the RTC_SESSION_NEGOTIATION message.

```
{
      "$schema": "http://json-schema.org/draft-07/schema",
      "type": "object",
      "title": "Audio_Video message Schema",
      "properties": {
            "type": {
                  "const": "RTC_SESSION_NEGOTIATION"
            },
            "user": {
                  "$ref": "#/definitions/userId"
            },
            "configuration": {
                  "$ref": "#/definitions/rtcConfiguration"
            },
            "timestamp": {
                  "type": "number"
            }
      },
      "definitions": {
            "userId": {
                  "type": "object",
                  "properties": {
                        "name": {
                              "type": "string"
                        },
                        "role": {
                              "type": "string"
                        }
                  },
                  "required": ["name", "role"]
            },
            "rtcConfiguration": {
                  "type": "object",
                  "properties": {
                        "iceServers": {
                              "type": "array",
                              "items": {
                                    "$ref": "#/definitions/rtcIceServer"
                              }
                        },
                        "iceTransportPolicy": {
                              "type": "string",
                              "enum": ["all", "relay"]
                        }
```

```
                    },
                    "required": ["iceServers", "iceTransportPolicy"]
            },
            "rtcIceServer": {
                    "type": "object",
                    "properties": {
                            "url": {
                                    "type": "array",
                                    "items": {
                                            "type": "string",
                                            "format": "uri"
                                    }
                            },
                            "username": {
                                    "type": "string"
                            },
                            "credential": {
                                    "type": "string"
                            }
                    },
                    "required": ["url"]
            }
    },
    "required": ["type", "user", "configuration", "timestamp"]
}
```

A.6 RTC_SESSION_DESCRIPTION from participants schema

This schema defines the RTC_SESSION_DESCRIPTION messages sent by participants to the Audio_Video signalling room.

```
{
    "$schema": "http://json-schema.org/draft-07/schema",
    "type": "object",
    "title": "Audio_Video RTC_SESSION_DESCRIPTION message Schema for
participant",
    "properties": {
            "type": {
                    "const": "RTC_SESSION_DESCRIPTION"
            },
            "description": {
                    "$ref": "#/definitions/rtcSessionDescription"
            },
            "user": {
                    "$ref": "#/definitions/userId"
            },
            "timestamp": {
                    "type": "number"
            }
    },
    "definitions": {
            "rtcSessionDescription": {
                    "type": "object",
                    "properties": {
                            "sdp": {
                                    "type": "string"
                            },
                            "type": {
                                    "const": "offer"
                            }
                    },
                    "required": ["sdp", "type"]
            },
            "userId": {
                    "type": "object",
```

```
                    "properties": {
                            "name": {
                                    "type": "string"
                            },
                            "role": {
                                    "type": "string"
                            }
                    },
                    "required": ["name", "role"]
            }
    },
    "required": ["type", "description", "user", "timestamp"]
}
```

A.7 RTC_SESSION_DESCRIPTION from Audio_Video signalling room schema

This schema defines the RTC_SESSION_DESCRIPTION messages sent by the Audio_Video signalling room to participants.

```
{
    "$schema": "http://json-schema.org/draft-07/schema",
    "type": "object",
    "title": "Audio_Video RTC_SESSION_DESCRIPTION message Schema for
Audio_Video-server",
    "properties": {
            "type": {
                    "const": "RTC_SESSION_DESCRIPTION"
            },
            "description": {
                    "$ref": "#/definitions/rtcSessionDescription"
            },
            "user": {
                    "$ref": "#/definitions/userId"
            },
            "timestamp": {
                    "type": "number"
            }
    },
    "definitions": {
            "rtcSessionDescription": {
                    "type": "object",
                    "properties": {
                            "sdp": {
                                    "type": "string"
                            },
                            "type": {
                                    "const": "answer"
                            }
                    },
                    "required": ["sdp", "type"]
            },
            "userId": {
                    "type": "object",
                    "properties": {
                            "name": {
                                    "type": "string"
                            },
                            "role": {
                                    "type": "string"
                            }
                    },
                    "required": ["name", "role"]
            }
    },
    "required": ["type", "description", "user", "timestamp"]
```

```
}
```

A.8 RTC_ICE_CANDIDATE from participants schema

This schema defines the RTC_ICE_CANDIDATE messages sent by participants to the Audio_Video signalling room.


```
{
	"$schema": "http://json-schema.org/draft-07/schema",
	"type": "object",
	"title": "Audio_Video RTC_ICE_CANDIDATE message Schema for participant",
	"properties": {
		"type": {
			"const": "RTC_ICE_CANDIDATE"
		},
		"candidate": {
			"$ref": "#/definitions/rtcIceCandidate"
		},
		"user": {
			"$ref": "#/definitions/userId"
		},
		"timestamp": {
			"type": "number"
		}
	},
	"definitions": {
		"rtcIceCandidate": {
			"type": "object",
			"properties": {
				"candidate": {
					"type": "string"
				},
				"sdpMLineIndex": {
					"type": "number"
				},
				"sdpMid": {
					"type": "string"
				},
				"usernameFragment": {
					"type": "string"
				}
			},
			"required": ["candidate", "sdpMLineIndex"]
		},
		"userId": {
			"type": "object",
			"properties": {
				"name": {
					"type": "string"
				},
				"role": {
					"type": "string"
				}
			},
			"required": ["name", "role"]
		}
	},
	"required": ["type", "rtcIceCandidate", "user", "timestamp"]
}
```


A.9 RTC_ICE_CANDIDATE from Audio_Video signalling room schema

This schema defines the RTC_ICE_CANDIDATE messages sent by the Audio_Video signalling room to participants.

```
{
	"$schema": "http://json-schema.org/draft-07/schema",
	"type": "object",
	"title": "Audio_Video RTC_ICE_CANDIDATE message Schema for Audio_Video-
server",
	"properties": {
		"type": {
			"const": "RTC_ICE_CANDIDATE"
		},
		"candidate": {
			"$ref": "#/definitions/rtcIceCandidate"
		},
		"user": {
			"$ref": "#/definitions/userId"
		},
		"timestamp": {
			"type": "number"
		}
	},
	"definitions": {
		"rtcIceCandidate": {
			"type": "object",
			"properties": {
				"candidate": {
					"type": "string"
				},
				"sdpMLineIndex": {
					"type": "number"
				},
				"sdpMid": {
					"type": "string"
				},
				"usernameFragment": {
					"type": "string"
				}
			},
			"required": ["candidate", "sdpMLineIndex"]
		},
		"userId": {
			"type": "object",
			"properties": {
				"name": {
					"type": "string"
				},
				"role": {
					"type": "string"
				}
			},
			"required": ["name", "role"]
		}
	},
	"required": ["type", "rtcIceCandidate", "user", "timestamp"]
}
```

A.10 USER_MEDIA from participants schema

This schema defines the USER_MEDIA messages sent by participants to the Audio_Video signalling room.

```
{
	"$schema": "http://json-schema.org/draft-07/schema",
```

```
	"type": "object",
	"title": "Audio_Video USER_MEDIA message Schema for participant",
	"properties": {
		"type": {
			"const": "USER_MEDIA"
		},
		"audio": {
			"type": "boolean"
		},
		"video": {
			"type": "boolean"
		},
		"receiveAudio": {
			"type": "boolean"
		},
		"receiveVideo": {
			"type": "boolean"
		},
		"user": {
			"$ref": "#/definitions/userId"
		},
		"timestamp": {
			"type": "number"
		}
	},
	"definitions": {
		"userId": {
			"type": "object",
			"properties": {
				"name": {
					"type": "string"
				},
				"role": {
					"type": "string"
				}
			},
			"required": ["name", "role"]
		}
	},
	"required": ["type", "audio", "video", "user", "timestamp"]
}
```

A.11 USER_MEDIA from Audio_Video signalling room schema

This schema defines the RTC_ICE_CANDIDATE messages sent by the Audio_Video signalling room to participants.

```
{
	"$schema": "http://json-schema.org/draft-07/schema",
	"type": "object",
	"title": "Audio_Video USER_MEDIA message Schema for Audio_Video-server",
	"properties": {
		"type": {
			"const": "USER_MEDIA"
		},
		"audio": {
			"type": "boolean"
		},
		"video": {
			"type": "boolean"
		},
		"receiveAudio": {
			"type": "boolean"
		},
		"receiveVideo": {
```

```
                    "type": "boolean"
            },
            "user": {
                    "$ref": "#/definitions/userId"
            },
            "timestamp": {
                    "type": "number"
            }
    },
    "definitions": {
            "userId": {
                    "type": "object",
                    "properties": {
                            "name": {
                                    "type": "string"
                            },
                            "role": {
                                    "type": "string"
                            }
                    },
                    "required": ["name", "role"]
            }
    },
    "required": ["type", "audio", "video", "user", "timestamp"]
}
```


A.12 ERROR schema

This schema is used by the PIM/tPSP to invoke the Audio_Video capability in the AP.


```
{
    "$schema": "http://json-schema.org/draft-07/schema",
    "type": "object",
    "title": "Audio_Video ERROR message Schema",
    "properties": {
            "type": {
                    "const": "ERROR"
            },
            "reason": {
                    "type": "string"
            },
            "code": {
                    "type": "integer"
            }
    },
    "required": ["type", "reason", "code"]
}
```

ANNEX B Recommended TLS cipher suits

This annex provides a recommended set of cipher suites for use with this protocol.

Table 22: Recommended TLS 1.3 cipher suites

Cipher	TLS version	Encryption	MAC
`TLS_AES_128_GCM_SHA256`	1.3	`AESGCM(128)`	AEAD
`TLS_AES_256_GCM_SHA384`	1.3	`AESGCM(256)`	AEAD
`TLS_CHACHA20_POLY1305_SHA256`	1.3	`CHACHA20/POLY1305(256)`	AEAD

Table 23: Acceptable TLS 1.2 cipher suites

Cipher	TLS version	Encryption	MAC
`ECDHE-ECDSA-AES128-GCM-SHA256`	1.2	`AESGCM(128)`	AEAD
`ECDHE-RSA-AES128-GCM-SHA256`	1.2	`AESGCM(128)`	AEAD
`ECDHE-ECDSA-AES256-GCM-SHA384`	1.2	`AESGCM(256)`	AEAD
`ECDHE-RSA-AES256-GCM-SHA384`	1.2	`AESGCM(256)`	AEAD
`ECDHE-ECDSA-CHACHA20-POLY1305`	1.2	`CHACHA20/POLY1305(256)`	AEAD
`ECDHE-RSA-CHACHA20-POLY1305`	1.2	`CHACHA20/POLY1305(256)`	AEAD
`DHE-RSA-AES128-GCM-SHA256`	1.2	`AESGCM(128)`	AEAD
`DHE-RSA-AES256-GCM-SHA384`	1.2	`AESGCM(256)`	AEAD

HISTORY

Document history		
V1.0	10 Jan 2023	Stable version
V1.1	5 February 2024	Reviewed