ETSI TS 103 871 V1.2.1 (2024-08)

Emergency Communications (EMTEL);
PEMEA Real-Time Text Extension

Reference
RTS/EMTEL-00075

Keywords
application, emergency, real-time text

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Special Committee Emergency Communications (EMTEL).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

The Pan-European Mobile Emergency Application (PEMEA) architecture provides a framework to enable applications supporting emergency calling functionality to contact emergency services while roaming. PEMEA caters for a range of extension capabilities, including Real-Time Text (RTT) which provides a text-based character by character exchange capability between the App user and the PSAP. The present document provides a specification for an RTT capability for PEMEA.

Introduction

Real-Time Text (RTT) communications are used extensively by people with hearing and speech disabilities around the world. These systems convey letters as they are typed from the source to the destination. The International Telecommunications Union (ITU) defines clear guidelines for what is required to support RTT. The present document defines an RTT protocol, complying with ITU guidelines, for use in the Pan-European Mobile Emergency Application (PEMEA) framework.

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

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

1 Scope

The present document describes the PEMEA Real-Time Text (RTT) 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.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

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

[2] ETSI TS 103 756: "Emergency Communications (EMTEL); PEMEA Instant Message Extension".

[3] Recommendation ITU-T T.140: "Protocol for multimedia application text conversation".

[4] IANA language subtag registry.

[5] IETF RFC 2617: "HTTP Authentication: Basic and Digest Access Authentication", June 1999.

[6] IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage", October 2012.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] IETF RFC 7519: "JSON Web Token (JWT)", May 2015.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

security: techniques and methods used to ensure:

- **authentication** of entities accessing resources or data;
- **authorization** of authenticated entities prior to accessing or obtaining resources and/or data;
- **privacy** of user data ensuring access only to authenticated and authorized entities;
- **secrecy** of information transferred between two authenticated and authorized entities.

trusted: As defined in ETSI TS 103 478 [1].

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
AESGCM	Advanced Encryption Standard key used with GCM
AP	Application Provider
App	Application
BEL	audible Bell sound
BS	Back Space
CPE	Customer Premises Equipment
CR	Carriage Return
DHE	Diffie-Hellman key Exchange
ECDHE	Elliptic-Curve Diffie-Hellman key Exchange
ECDSA	Elliptic Curve Digital Signature Algorithm
EDS	Emergency Data Send (message)
ESC	Escape
ETSI	European Telecommunications Standards Institute
GCM	Galios/Counter Mode
HTTP	Hyper-Text Transfer Protocol
HTTPS	Secure HTTP
IANA	Internet Assigned Numbers Authority
ID	IDentifier
IETF	Internet Engineering Task Force
INT	Interrupt character
ITU	International Telecommunications Union
JSON	JavaScript Object Notation
JWT	JSON Web Token
LF	Line Feed
MAC	Message Authentication Code
Pa	PEMEA application to AP interface
PEMEA	Pan-European Mobile Emergency Application
PIM	PSAP Interface Module
PSAP	Public Safety Answering Point
PSP	PSAP Service Provider
RSA	Rivest Shamir Aldeman public key encryption algorithm

RTT	Real-Time Text
SGR	Select Graphic Rendition
SHA	Secure Hash Algorithm
SIP	Session Initiation Protocol
SOS	Start Of String
ST	String Terminator
TLS	Transport Layer Security
tPSP	terminating PSP
UCS	Universal multiple-octet coded Character Set
URI	Uniform Resource Identifier
UTC	Coordinated Universal Timer
UTF-8	UCS Transformation Format (8 bit words)

4 PEMEA capability extensions

4.1 Overview of extension in PEMEA

PEMEA extension capabilities are defined in ETSI TS 103 478 [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 a 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 elements and apMoreInformation structures are defined in clauses 10.3.11 and 10.3.12 of ETSI TS 103 478 [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 [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 [1], protocols are left for further study and definition in subsequent specifications such as the present document. ETSI TS 103 756 [2] describes the concrete specification for PEMEA Instant Message protocol.

4.2 Service support indication and response

4.2.1 Service definition

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

Table 1: Extended AP Information Type Protocol Registry

Info type Value	Protocol Token	Description
RTT	PEMEA	Real-Time Text functionality is supported using the PEMEA message exchange protocol

4.2.2 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="RTT" protocol="PEMEA">
    https://ap.example.pemea.help/48sne8aopaop
</information>
```

4.2.3 Service support response

A terminating node that can support the "RTT" "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 [1] with the value for "RTT" "PEMEA" provided in the example below:

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

4.2.4 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 call-taker rather than it being missing because it went to the wrong PSAP.

ETSI TS 103 478 [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 [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 PEMEA RTT 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 upon receipt of an EDS containing a RTT PEMEA capability. The onCapSupportPost message shall contain the RTT PEMEA capability along with any other capabilities that the PSAP supports.

If the PSAP does not have the ability or capacity to support the request then it may forward the request to a neighbouring PSAP with whom it has an agreement to do so. In this situation the original PSAP shall not send an onCapSupportPost message to the originating AP. If, having forwarded the EDS, the PSAP receives an error from the destination PSAP then the originating PSAP shall send an onErrorPost event to the AP including the cause of the error.

5 Mapping to T.140

5.1 T.140 special character support

Recommendation ITU-T T.140 [3] defines requirements and procedures for RTT systems. For the most part these are mapped directly. With the movement to modern communications system however, some of the requirements in Recommendation ITU-T T.140 [3] are no longer relevant. In other cases, functionality is not provided as it is available through other PEMEA extensions or is supported implicitly through the protocol itself rather than through special characters. Table 2 indicates which functionality from clause 7 of Recommendation ITU-T T.140 [3] is supported and how.

Table 2: PEMEA RTT support for T.140 special characters

Name	Supported	Description
BEL	No	No alerting in the communication is provided
BS	Yes	Backspace character is sent as 0x08, converted to UTF-8, inside a TEXT_MESSAGE
NEW LINE	Yes	New line character is sent as 0x0A, converted to UTF-8, inside a TEXT_MESSAGE
CR LF	No	No-standard and non-preferred, not supported
INT	No	No mode negotiation is required
SGR	No	Not supported
SOS	No	Not supported
ST	No	Not supported
ESC	Yes	The present document supports the sending and receiving of the ESC (0x1B) control character, however, rendering, displaying and interpretation of control sequences is not specified
Byte order mark	No	Synchronization is not required via a Web Socket

The protocol described in the present document addresses the establishment of connections, disconnections and the transfer of data between entities, it does not attempt to address the display requirements of Recommendation ITU-T T.140 [3]. However, the intention from T.140 Appendix I shall be fulfilled. "*The display of text from the members of the conversation should be arranged so that the text from each participant is clearly readable, and its source and the relative timing of entered text is visualized in the display. Mechanisms for looking back in the contents from the current session should be provided. The text should be displayed as soon as it is received.*"

All text is transferred using UTF-8 which can represent most language character sets. The language that the user intends to communicate with is provided in the JOIN message, see clause 8.3. The language shall be specified using one of the languages provided in the language sub-tags registered with IANA [4]. The present document does not provide guidance on whether multi-lingual session participants may switch languages during the session or not though the general recommendation is against taking this action.

Text messages consist of one or more characters. Characters are transferred from the App to the AP either in real-time, as they are typed, or in batches at 0,5 second intervals so that a character is always transferred within 0,5 seconds of having been typed. This functionality is described in clause 6.1.1 of Recommendation ITU-T T.140 [3].

5.2 ESC character sequence support

ESC character sequences in the present document are a set of characters bounded by ESC characters (0x1B) on either side. For example 0x1B:)0x1B may display a smiley face. The present document does not define any ESC character sequences nor does it provide any guidance on rendering or interpretation beyond all characters between two ESC characters forming the escape sequence.

An entity shall ignore all escape sequence characters if an explicit escape sequence code set has not been established through some other means. The present document leaves the possibility open for a future revision of the present document to define common sets of escape sequences.

The ESC sequence, open ESC character, intermediate characters and closing ESC character shall be sent in a single message and the receiver receiving a single erase character shall erase any and all characters in the ESC sequence.

Any message containing a partial ESC sequence shall be ignored.

6 Security

6.1 Transport security

The RTT service is identified to potential room participants as an HTTPS URI. The connection is made using TLS 1.3 but may be made using TLS 1.2, but shall not fallback below TLS 1.2. The connecting participant shall authenticate to the RTT service using a Bearer token in the HTTP Authentication header field as described in IETF RFC 6750 [6]. Once the connecting entity is authenticated and authorization 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.

6.2 Security token usage

The HTTP Authorization header field is defined in IETF RFC 2617 [5] 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 [6]. 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 is a base64 encoded string.

Token usage in the RTT PEMEA specification follows the Bearer scheme defined in IETF RFC 6750 [6].

Tokens issued by entities in the RTT PEMEA 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 [i.1] is not considered relevant to usage and is not specified further.

IETF RFC 6750 [6] mandates the usage of TLS for use with Bearer tokens, this usage is further defined in clause 6.1 of the present document.

7 Procedures and signalling

7.1 Service invocation

7.1.1 Service invocation procedures

Once the terminating PSP or PSAP has responded to the AP that it can support the PEMEA RTT service then the AP shall be capable of accepting a service invocation on the provided URI at any time. The AP shall only accept an RTT service invocation from the PIM or tPSP that sent the onCapSupportPost message.

The PSAP invokes the RTT service by:

a) The call-taker initiating their willingness to use RTT to the PSAP Interface Module (PIM) in the PSAP or the tPSP.

b) The PIM/tPSP requesting the RTT server to create an RTT-session room.

c) The RTT server creating an RTT-session room and returns a URI to the PIM/tPSP.

d) The PIM/tPSP obtains Bearer tokens for the call-taker and AP.

e) The PIM/tPSP returns the URI and a Bearer token to the PSAP call-taker.

f) The call-taker connects to the RTT-session room authenticating using the provided Bearer token.

g) The PIM/tPSP calling the URI provided by the AP for the RTT-PEMEA service and including the URI for the RTT-session room and a Bearer token in this invocation. Note that the URI is the same for the call-taker and the caller, but the Bearer tokens are different.

h) The AP indicates to the App that the PSAP wishes to communicate using RTT with the user.

i) The user indicates their willingness to communicate using RTT with the PSAP to the AP.

j) The AP initiates a connection to the RTT-session room authenticating using the Bearer token.

It is important to note that it is always the AP that authenticates to the RTT-session room and consequently all messages from the App shall traverse the AP. The present document only defines the protocol between the AP and other trusted entities e.g. PSAP call-taker or First Responder, and the RTT-session room in the PSAP, it does not define the RTT Pa messaging between the App and the AP.

7.1.2 Service invocation object

The PIM/tPSP invokes the RTT PEMEA service in the AP by posting to the URI provided in the RTT 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 RTT-session room URI as well as a security token and corresponding expiry time.

The JSON schema for the RTT service invocation message is provided in annex A.

Table 3: Invocation object fields

Element Name	Presence	Description
uri	Mandatory	The URI of the RTT-session room.
token	Mandatory	A security token used to authenticate the AP to the RTT-session 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 RTT-session room for the duration of the App emergency session. The AP shall not provide the token to the App.
expiry	Mandatory	Specifies the expiry time of the security token. expiry is an integer specifying the number of seconds since UTC epoch, 00:00:00 1st of January 1970.

Invocation example:

```
{
    "uri": "https://rtt-server.example.com/room/534wafds21s21fdf",
    "token": "PPtzs5zzG5Pkf61KPz51",
    "expiry": "1590563357576"
}
```

7.2 RTT-session room creation and deletion

The RTT-session room is created by the RTT server under direction of the PSAP call-taker via the PIM or tPSP. When the RTT-session room is created, a logging function shall be created with it to scribe all messages into and out of the room. This flow is shown in Figure 1.

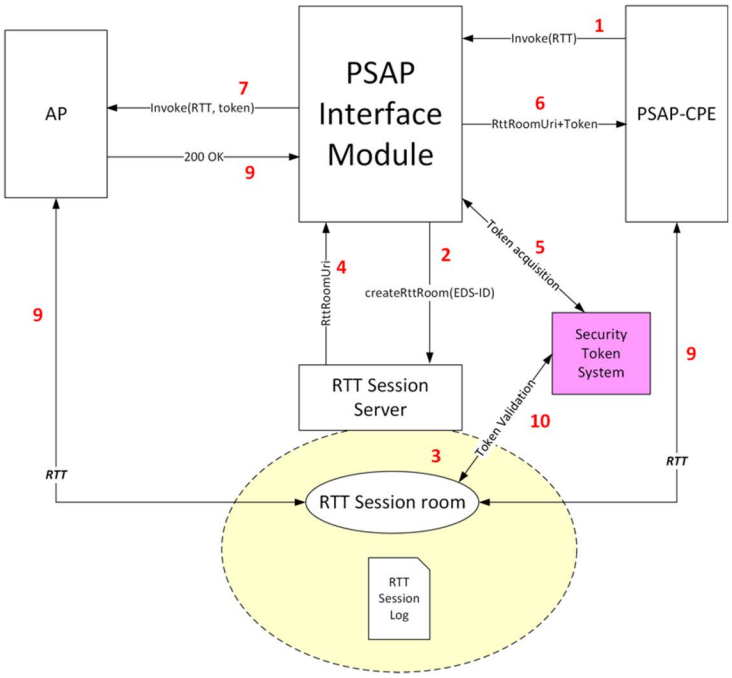

Figure 1: RTT session initiation

Once the RTT-session room is created it remains active as long as the PIM or tPSP maintains a context for the EDS. When EDS context is deleted the RTT-session room is also destroyed.

7.3 RTT-session room creation, JOIN, and TEXT_MESSAGE signalling

7.3.1 Semantics

The figures in the following sub-clauses show the signalling involved in establishing and subsequently joining a PEMEA RTT session. By necessity the diagrams show four distinctive types of signalling:

- Semantic signalling across the Pa interface between the App and the AP is explicitly not defined in PEMEA. So, while the message names and contents may not align with any specific implementation, the semantics of what the messages convey should be understood.
- Core PEMEA signalling are explicit messages defined in the PEMEA technical specification ETSI TS 103 478 [1].
- RTT semantic signalling is messaging that needs to occur between the PSAP call-taker equipment, the PIM/tPSP and the software entities and components required to establish the RTT service. These messages are intended to provide an idea of what needs to occur, not how it should be implemented. Consequently, they are informative only and not normative.
- RTT normative signalling messages and semantics are explicitly defined in the present document.

7.3.2 RTT service invocation

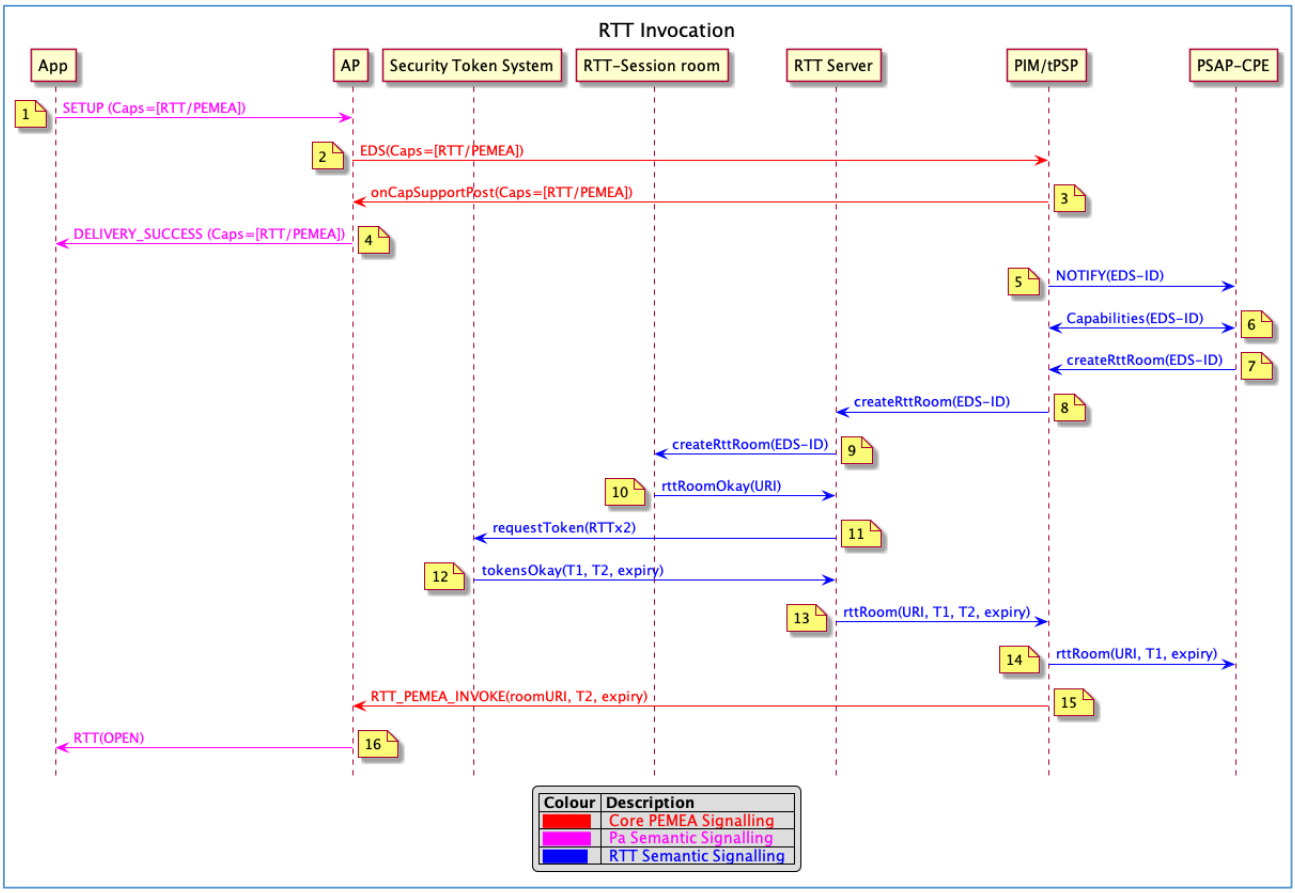

Figure 2: RTT service invocation

1) App initiates an emergency session with the AP over the Pa interface indicating that it can support the RTT/PEMEA capability.

2) The AP creates an EDS message from the data provided by the App and includes the RTT/PEMEA protocol capability. The AP then sends the EDS into the PEMEA network.

3) The EDS arrives at the PIM/tPSP. The PIM supports the RTT/PEMEA capability and includes this option in the onCapupportPost 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 that the PSAP can support the RTT/PEMEA functionality.

5) The PIM notifies the PSAP-CPE that a new EDS has arrived.

6) The PSAP call-taker via the PSAP-CPE requests the capabilities agreed with the AP, sees the RTT/PEMEA capability.

7) The PSAP call-taker requests the PIM to initiate the creation of an RTT-Session room.

8) The PIM requests that the RTT server creates an RTT-Session room.

9) The RTT server initiates the creation of an RTT-Session room.

10) The RTT server acquires the RTT-Session room URI.

11) The RTT Server requests 2 access Bearer tokens for the RTT service from the security token system.

12) The security token system returns two access Bearer tokens for the RTT service along with their respective token expiry times to the RTT Server.

13) The RTT server returns the RTT-Session room URI, the two security access Bearer tokens and their respective expiry times to the PIM.

14) The PIM sends the RTT-Session room URI, one of the security access Bearer tokens and its expiry time to the PSAP call-taker via the PSAP-CPE.

15) The PIM invokes the RTT/PEMEA capability in the AP using the provided reach-back URI from the EDS. The PIM includes the RTT-Session room URI, security access Bearer token and expiry time in the body of the HTTP POST to the reach-back URI.

16) The AP signals to the App over the Pa interface that the PSAP has invoked the RTT/PEMEA communication capability.

7.3.3 JOIN message flow

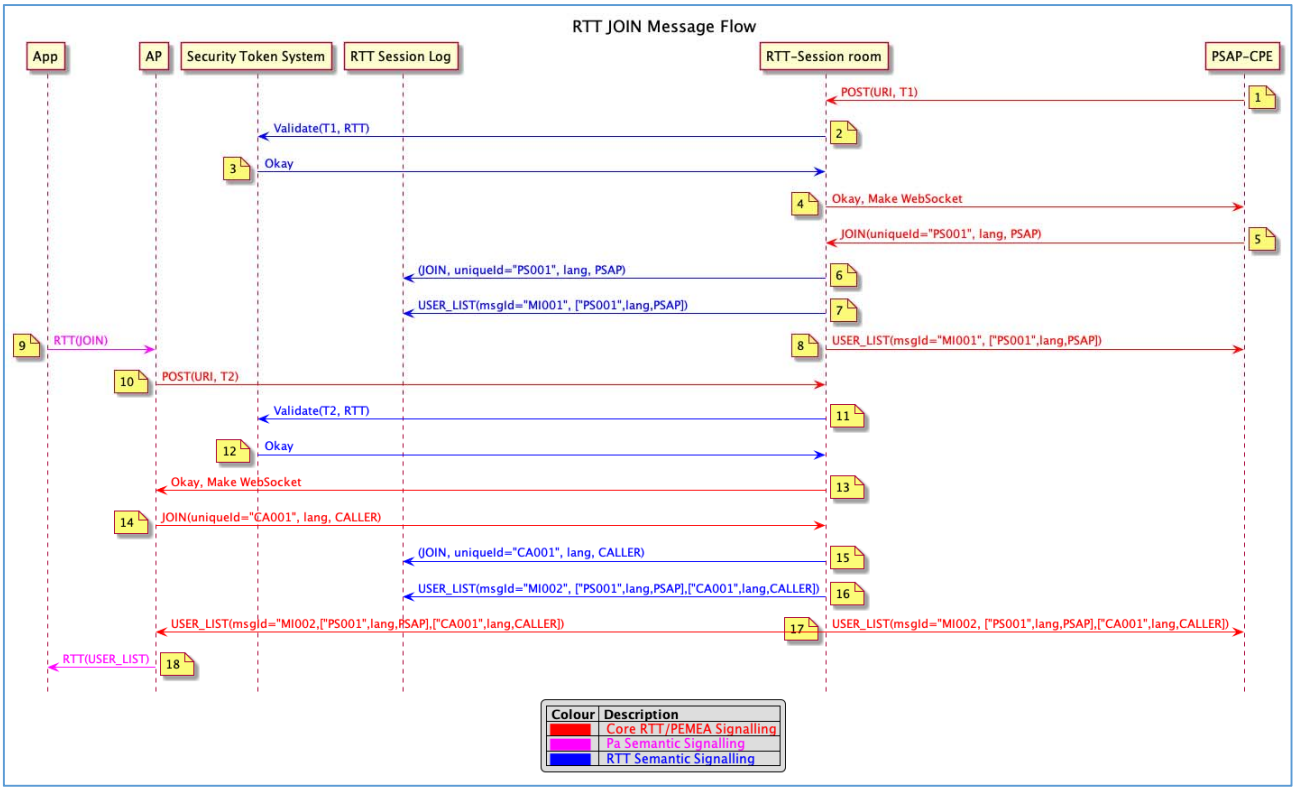

Figure 3: RTT JOIN message flow

Once the RTT/PEMEA service has been invoked in the AP, then the PSAP call-taker and Caller join the RTT-Session:

1) The PSAP call-taker connects to the RTT-Session room passing in their authentication token in the Authorization HTTP header field in accordance with clause 6.2.

2) RTT-Session room asks the Security Token System to validate the token for use with the RTT service.

3) Security Token System responds that the token is valid.

4) The RTT-Session room indicates that the token is good and initiates promoting the connection to be a secure websocket.

5) PSAP call-taker sends a JOIN message to the RTT-session room indicating that the connecting entity is a PSAP and includes the call-taker's pseudonym, communication language and uniqueId.

6) RTT-Session room writes the PSAP JOIN message to the log.

7) RTT-Session room creates a USER_LIST message including a unique message ID, and writes it to the log.

8) RTT-Session room sends the PSAP call-taker the current USER_LIST.

9) The App user indicates that they wish to join the RTT session.

10) The AP connects to the RTT-session room including the authentication token in the Authorization HTTP header field in accordance with clause 6.2.

11) RTT-Session room asks the Security Token System to validate the token for use with the RTT service.

12) Security Token System responds that the token is valid.

13) The RTT-Session room indicates that the token is good and initiates promoting the connection to be a secure websocket.

14) The AP then sends a JOIN message to the RTT-session room indicating that the connecting entity is a CALLER and includes the caller's name, communication language and uniqueId.

15) RTT-Session room writes the CALLER JOIN message to the log.

16) RTT-Session room creates a USER_LIST message including a unique message ID, and writes to the log.

17) RTT-Session room sends the PSAP call-taker and the AP the current USER_LIST.

18) The AP sends the RTT USER_LIST to the App over the Pa interface.

7.3.4 ERROR message flow

The RTT-session room sending, and user receiving, a USER_LIST message indicates a successful joining of the RTT session, a join request may also be rejected by the RTT-session room by sending an ERROR message.

A JOIN message shall be rejected by the RTT-session room if the uniqueId is already in use by an active/ONLINE user. When this occurs the websocket is also closed and the user shall create a new uniqueId and shall re-connect and send a new JOIN message. The security token shall remain valid.

The RTT-session room shall log the JOIN message and the ERROR message. However, the user list is not updated so no USER_LIST is subsequently sent to other session participants when an ERROR message has been sent in response to the JOIN message.

7.3.5 TEXT_MESSAGE flow

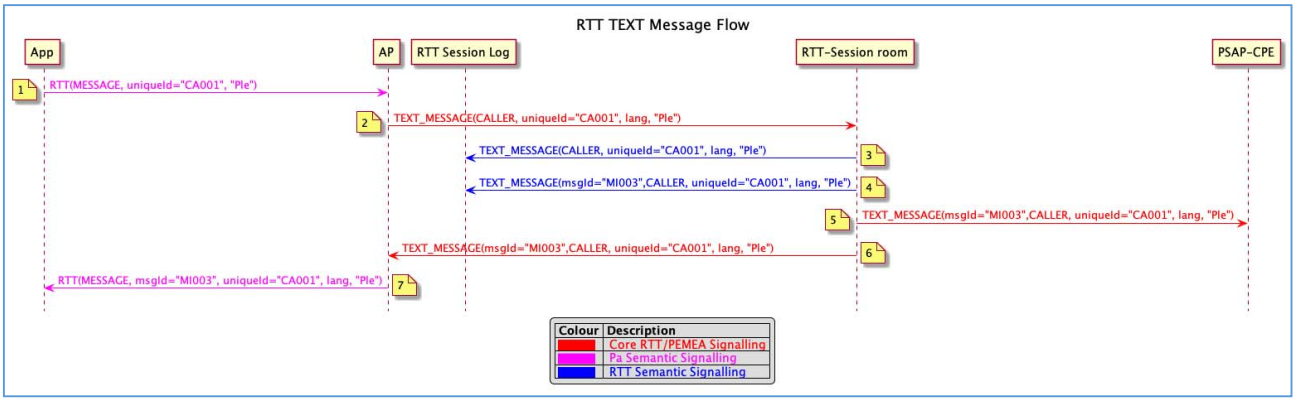

Figure 4: TEXT_MESSAGE signalling flow

Once the AP has joined the RTT-Session room it is able to send characters to the other participants in the session. This is done using the TEXT_MESSAGE. The App shall send characters one at a time to the AP. The AP may buffer characters into a single TEXT_MESSAGE, but shall send the TEXT_MESSAGE to the network no more 0,5 seconds after the first character has been received from the App. This ensures that the real-time requirement laid out in Recommendation ITU-T T.140 [3] is satisfied:

1) The App sends the characters that the user types to the AP over the Pa interface. These characters may include new line or backspace characters as identified in clause 5.

2) The AP buffers the user's characters for up to 0,5 seconds then creates a TEXT_MESSAGE for transmission to the RTT-session room and sends the TEXT_MESSAGE to the session room.

3) The RTT-session room receives the message and writes it to the session log.

4) The RTT-session room adds a unique message id to the message, along with the room identifier and a time stamp and writes this message to the session log.

5) The RTT-session room sends the new TEXT_MESSAGE to the PSAP call-taker.

6) The RTT-session room sends the new TEXT_MESSAGE to the AP.

7) The AP may relay the message to the App, indicating that the characters have been successfully delivered to the other parties in the RTT-session.

7.4 Disconnects and reconnects

Despite communications networks being reliable, accidental disconnects owing to temporary issues do still occur. PEMEA does not define how the AP and the App communicate, though some high-level semantics for RTT are described in the present document. The present document describes communication between the authorized entities and the RTT-session room, most commonly the AP and the PSAP-CPE.

If the RTT-session room terminates for an unexpected reason, then the websockets used for communication between the participants and the RTT-session room will close. Should this occur, then the participants should attempt to reconnect, with an ever-increasing exponential back-off. Participants shall reconnect using the same uniqueId (see clause 7.3.3). Failure to reconnect after a configurable period should result in the participant not attempting to continue to retry. When this occurs for the PSAP call-taker, the PSAP call-taker may request a new session to be created and this will then follow the creation process described in clause 7.2.

When this occurs, the new session may have a new URI, so a new RTT-session invocation is sent to the AP including the new RTT-session room URI. The new invocation shall include a new token and expiry time. When this occurs then a participant shall connect using the same uniqueId as was used for the previous session (see clause 7.3.3), this ensures that logs can be aligned.

Since the session log provides a transcript of what information has been exchanged between room participants, it shall be persistent, so in the event of a new room creation due to a system failure of some kind, the same session log shall continue to be used in the new RTT-session room.

On receipt of the new RTT invocation, the AP shall auto join the newly provided RTT-session room URI and communication is re-established. The AP does not need to report to the App the loss of connectivity to the RTT-session room until it determines that the connectivity cannot be restored. A simplified version of this flow is provided in Figure 5.

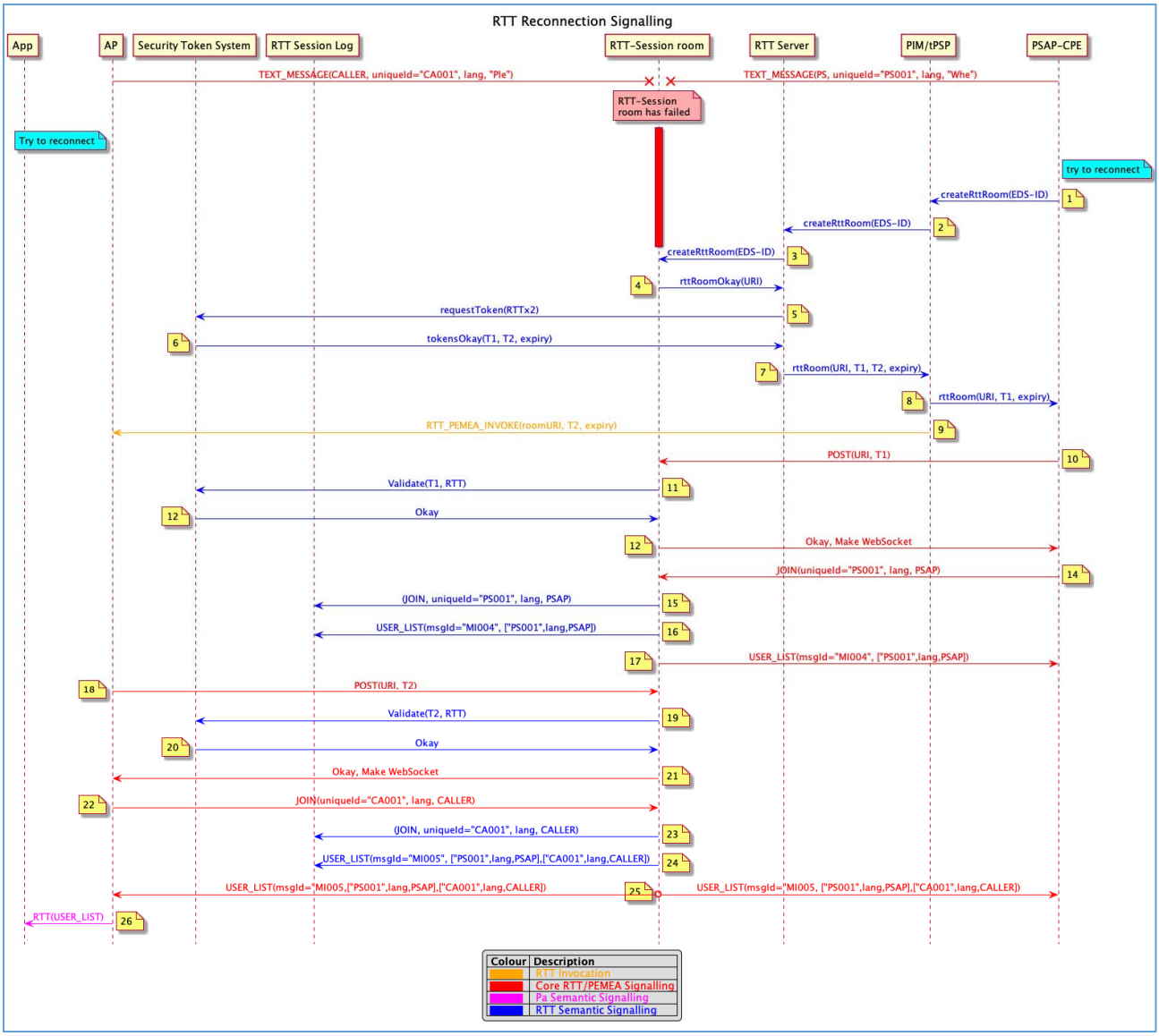

Figure 5: PEMEA RTT reconnection signalling

8 RTT PEMEA message and type definitions

8.1 Overview

The RTT PEMEA protocol messages are defined as a series of JSON documents exchanged between the AP or PEMEA terminating node and an "RTT-session room" established inside the secure emergency network. The RTT-session room is established solely for communications with a single emergency session. Each emergency session requiring the use of the RTT PEMEA service has its own RTT-session 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 and are also maintained in a repository outside of the present document and are available for download from ETSI Forge https://forge.etsi.org/rep/emtel/ts-103-871/pemea-rtt-schema/-/tree/1.2.1.The subsequent sub-clauses in clause 7 of the present document describe each of the RTT PEMEA messages, its function, elements and any key constraints. Messages exchanges and procedures are specified in clause 6.

Table 4: RTT PEMEA messages

Message type	Description
JOIN	Message sent to the RTT-session room when the user wants to join the RTT-session.
ERROR	Sent by the RTT-session room in the event that a JOIN request is received containing a uniqueId already in use by an active/ONLINE user.
USER_LIST	Message sent from the RTT-session room to all participants containing all users whenever a user enters or leaves the RTT-session.
TEXT_MESSAGE	Message sent either from a participant to the RTT-session room, or from the RTT-session room to all participants containing a user's characters. Message history is transferred as text messages. History is sent when a user joins the RTT-session room.

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

8.2 Data types

8.2.1 language

Is the language that the user will be communicating in through the RTT session. The language may be any of the pertinent languages from the IANA language subtag registry [4].

8.2.2 room

Is a unique string providing a name for the RTT-session room. This is usually the URI used to specify the room attachment provided when the RTT service is invoked by the PSAP.

8.2.3 timestamp

All messages are sent with a timestamp and to avoid offsets, time zones, daylight savings changes etc, the time is always absolute. It is specified as an integer in milliseconds since the UTC epoch of 00:00:00 1st January 1970.

8.2.4 user

Defines a user in the RTT-session room. It consists of:

- name;
- role;
- uniqueId.

The name is a string that identifies a handle to which the user relates, this may be their name "George" for example, or their telephone number, tel: +34666554433 for example.

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

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

The uniqueId shall be generated by a participant and is used to uniquely identify the message stream in the event that more than one user in the session uses the same name and role. The uniqueId needs to have little chance of collisions with other generated identities and so should not be based purely on static data, such as the name and role, and should be large enough so that collisions are avoided. Attempts to join an RTT session with a uniqueId already in use shall result in a rejection of the join request via an ERROR message.

8.2.5 userInfo

Is used to combine information about the user:

- user: defined in clause 8.2.4;
- language: defined in clause 8.2.1;
- status.

The status field is used to describe what the user is doing.

Table 6: UserInfo status values

Status	Description
ONLINE	The user is in the RTT-session. This may be a new user joining the RTT-session, or maybe an existing user connected to the RTT-session room.
OFFLINE	The user was, but is no longer, in the RTT session. The RTT-session room may only use this status as an indicator that a user has left the RTT session and then delete knowledge of the connection, or it may maintain a list of all users that have ever joined the RTT session.

8.2.6 Void

8.3 JOIN message

8.3.1 Message overview

The JOIN message is the message sent from the participant to the RTT-session room when the user wants to join the session. This may be the AP, the PSAP-CPE or another trusted user. The JOIN message is resent if for some reason the connection between the entity and the RTT-session room is lost but the RTT session is not concluded.

The JOIN message consists of the following required fields.

Table 7: JOIN message fields and description

Element Name	Description	
type	"JOIN". The message being sent by the end-point to the RTT-session room.	
user	name	Name and role of entity joining the RTT session. The Name may be the user's name or their telephone number.
	role	The role will depend on the type of user joining the RTT session. In the case of the user initiating the emergency communication this will be "CALLER".
	uniqueId	uniqueId for this user in the RTT session.
language	Is the language that the user will be communicating in over RTT. It is a language from the IANA language subtag registry [4].	
since	Send all messages after this time. The time is specified as milliseconds since epoch. A value of zero means send all messages. When a participant is connecting to the RTT-session room for the first time then it will send a "since" value of zero, indicating that it wants all messages. For example, an AP may do this in case the PSAP call-taker joined the RTT-session before the caller did. This ensures that when the history is sent from the RTT-session room to the AP that the AP receives all messages in chronological order.	

8.3.2 Examples

The JOIN message is also used to reconnect to the RTT-session room in the case that the connection terminated, AP or RTT server restarted. In this case, the AP will set the since value to be the time that the AP knew it last had a connection to the RTT-session room, often this will be last received message from the RTT-session room. On a successful connection, the RTT-session room will send all messages that have occurred "since" the specified time.

```
{
    "type":"JOIN",
    "user":{
        "name":"PSAP-IXHJh219",
        "role":"PSAP",
        "uniqueId":"jgh204nq9md"
    },
    "language":"es",
    "since":0
}
```

8.4 ERROR message

8.4.1 Message overview

The ERROR message is sent by the RTT-session room in response to a JOIN request containing a uniqueId already in use by an active/ONLINE user in the session. The message is not intended to be visible to the end-user since either the AP or call-taker CPE will generate the uniqueId automatically. Refer to clause 7.3.4 for procedures on message rejection.

Table 8: Error message fields and description

Element Name	Description
type	"ERROR" The message being sent by the RTT-session room in response to the JOIN request.
room	String identifying the RTT-session room.
reasonCode	This is a token indicating why the JOIN was refused by the RTT-session room. See Table 10 for valid reasonCodes.
reason	Optional field containing text describing the problem.
timestamp	Integer Number of milliseconds from epoch (00:00:00:00 1st January 1970). See clause 8.2.3.

Table 9: ERROR reasonCode values

reasonCode	Description
idInUse	Sent when a JOIN request is received by the RTT-session room that contains a uniqueId already in use by an active/online participant.
badMessage	The message sent by the participant to the RTT-session room was malformed.

8.4.2 Error message example

```
{
    "type":"ERROR",
    "room":"ttRRkzORz",
    "reasonCode":"idInUse",
    "reason":"uniqueId sh4786793384881h32jh already in use",
    "timestamp":1574092280231
}
```

8.5 USER_LIST message

The USER_LIST message is sent to all participants in the RTT session whenever a user enters and leaves the RTT session.

Table 10: USER_LIST message fields and description

Element Name	Description
type	"USER_LIST" The message being sent by the RTT-session room to the participants.
room	String identifying the RTT-session room.
timestamp	Integer Number of milliseconds from epoch (00:00:00:00 1st January 1970). See clause 8.2.3.
users	An array of userInfo containing the name, role and status of each user in the RTT-session room. See clause 8.2.4.

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

```
{
    "type":"USER_LIST",
    "room":"ttRRkzORz",
    "timestamp":1574092280231,
    "users":[
        {
            "language":"es",
            "user":{
                "name":"George",
                "role":"CALLER",
                "uniqueId":"jgh204nq9md"
            },
            "status":"OFFLINE"
        },
        {
            "language":"es",
            "user":{
                "name":"PSAP-IXHJh219",
                "role":"PSAP",
               "uniqueId":"ljfvgtsy26540"
            },
            "status":"ONLINE"
        }
    ]
}
```

8.6 TEXT_MESSAGE message

The text message is used by a RTT session participant to contribute to send real-time written text to other participants in the RTT session. Every message has an identifier associated with RTT-session room so that it can be uniquely identified.

After the RTT-session room server receives a TEXT_MESSAGE message from a user, it will send it to all the participants in the RTT-session, including the sender.

Text message fields are different depending on who the sender is. When the sender of text message is a participant, then Table 12 represents the required fields.

Table 11: TEXT_MESSAGE message fields and description from participant

Element Name	Description
type	"TEXT_MESSAGE" refer to Table 4.
message	The details of the text being sent and the language in which the message is composed.

```
{
    "type":"TEXT_MESSAGE",
    "message":"hola"
}
```

The RTT-session room is responsible for relaying text messages to all the participants and the text messages it sends consists of all of the fields in Table 13.

Table 12: TEXT_MESSAGE message fields and description

Element Name	Description
id	Unique identifier for this message within the RTT-session room.
type	"TEXT_MESSAGE" refer to Table 4.
room	The identifier for the RTT-session room.
timestamp	The time that the message was sent. Refer to clause 8.2.3.
user	The user sending or that sent the text message. Refer to clause 8.2.4.
message	The details of the text being sent and the language in which the message is composed.

```
{
    "id":"5dd2bd8ba5568000079fa11c",
    "type":"TEXT_MESSAGE",
    "room":"ttRRkzORz",
    "timestamp":1574092171988,
    "user":{
        "name":"PSAP-XqwFbQ-A",
        "role":"PSAP",
        "uniqueId":"jgh204nq9md"
    },
    "message":"holajd\b\b"
}
```

9 RTT PEMEA message and type definitions

The RTT-session room is responsible for all session logging. It shall log all messages into and out of the room. This ensures that in the event of an audit there is a trail showing that the messages were sent to a specific room participant.

Message logging shall include all characters sent by a user, including backspace characters so that in the event of an incident, the entire history of the channel can be viewed.

Annex A (normative): RTT/PEMEA JSON schema

A.1 General

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

A.2 RTT invocation schema

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

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

A.3 JOIN schema

This schema provides the schema for the JOIN message.

```
{
    "$schema": "http://json-schema.org/draft-07/schema",
    "type": "object",
    "title": "RTT JOIN message Schema",
    "properties": {
        "type": {
            "const": "JOIN"
        },
        "language": {
            "type": "string"
        },
        "since": {
            "type": "number"
        },
        "user": {
            "$ref": "#/definitions/user"
        }
    },
    "definitions": {
        "user": {
            "type": "object",
            "properties": {
                "name": {
                    "type": "string"
                },
                "role": {
                    "$ref": "#/definitions/role"
                },
                "uniqueId": {
                    "type": "string"
                }
            },
```

```
            "required": ["name", "role", "uniqueId"]
        },
        "role": {
            "type": "string"
        }
    },
    "required": ["language","since","user", "type"]
}
```

A.4 USER_LIST schema

This schema specifies the USER_LIST message.

```
{
    "$schema": "http://json-schema.org/draft-07/schema",
    "type": "object",
    "title": "RTT USER_LIST message Schema",
    "properties": {
        "type": {
            "const": "USER_LIST"
        },
        "room": {
          "type": "string"
        },
        "users": {
            "type": "array",
             "items": {
               "$ref": "#/definitions/userStatus"
             }
        },
        "timestamp": {
            "type": "number"
        }
    },
    "definitions": {
        "userStatus": {
          "type": "object",
          "required": ["language", "user", "status"],
          "properties": {
            "language": { "type": "string"},
            "user": {"$ref": "#/definitions/user"},
            "status": {
              "type": "string",
              "enum": ["OFFLINE", "ONLINE"]
            }
          }
        },
        "user": {
            "type": "object",
            "properties": {
                "name": {
                    "type": "string"
                },
                "role": {
                    "type": "string"
                },
                "uniqueId": {
                    "type": "string"
                }
            },
            "required": ["name", "role", "uniqueId"]
        }
    },
    "required": ["type", "room", "users", "timestamp"]
}
```

A.5 TEXT_MESSAGE from participant schema

This schema defines the TEXT_MESSAGE sent by participants to the RTT-session room.

```
{
    "$schema": "http://json-schema.org/draft-07/schema",
    "type": "object",
    "title": "RTT TextMessage message Schema for participant",
    "properties": {
        "type": {
            "const": "TEXT_MESSAGE"
        },
        "message": {
            "type": "string"
        },
    },
    "required": ["message", "type"]
}
```

A.6 TEXT_MESSAGE from RTT-session room schema

This schema defines the TEXT_MESSAGE schema for TEXT_MESSAGEs sent by the RTT-session room to participants.

```
{
    "$schema": "http://json-schema.org/draft-07/schema",
    "type": "object",
    "title": "RTT TextMessage message Schema for rtt-server",
    "properties": {
        "id": {
            "type": "string"
        },
        "type": {
            "const": "TEXT_MESSAGE"
        },
        "room": {
            "type": "string"
        },
        "message": {
            "type": "string"
        },
        "user": {
            "$ref": "#/definitions/user"
        },
        "timestamp": {
            "type": "number"
        }
    },
    "definitions": {
        "user": {
            "type": "object",
            "properties": {
                "name": {
                    "type": "string"
                },
                "role": {
                    "type": "string"
                },
                "uniqueId": {
                    "type": "string"
                }
            },
            "required": ["name", "role", "uniqueId"]
        }
    },
    "required": ["id", "message", "type", "room", "user", "timestamp"]
}
```

A.7 RTT ERROR message schema

This schema defines the RTT ERROR message.

```
{
    "$schema": "http://json-schema.org/draft-07/schema",
    "type": "object",
    "title": "RTT Error  Schema",
    "properties": {
        "type": {
            "const": "ERROR"
        },
        "room": {
          "type": "string"
        },
        "reasonCode": {
          "type": "string"
        },
        "reason": {
          "type": "string"
        },
        "timestamp": {
            "type": "number"
        }
    },
    "required": ["type", "room", "reasonCode", "reason", "timestamp"]
}
```

Annex B (informative): Recommended TLS cipler suits

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

Table B.1: 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 B.2: 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		
V1.1.1	December 2022	Publication
V1.2.1	August 2024	Publication