|  |
| --- |
| 3GPP TR 26.982 V0.1.1 (2024-01) |
| Technical Report |
| 3rd Generation Partnership Project;Technical Specification Group SA WG4;IP Multimedia Subsystem (IM);Multimedia Telephony;Implementation Guidelines for Multiparty Real-Time Text(Release 18) |
|   |
|  |  |
|  |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. |

|  |
| --- |
|  |
| ***3GPP***Postal address3GPP support office address650 Route des Lucioles - Sophia AntipolisValbonne - FRANCETel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16Internethttps://www.3gpp.org |
| ***Copyright Notification***No part may be reproduced except as authorized by written permission.The copyright and the foregoing restriction extend to reproduction in all media.© 2024, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).All rights reserved.UMTS™ is a Trade Mark of ETSI registered for the benefit of its members3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational PartnersLTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational PartnersGSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 4

1 Scope 6

2 References 6

3 Definitions of terms, symbols and abbreviations 6

3.1 Terms 6

3.2 Symbols 7

3.3 Abbreviations 7

4 Introduction of multiparty real-time text (Multiparty RTT) 7

4.1 General 7

4.2 Study on Multiparty RTT requirements from other standards 7

4.2.1 General requirements 7

4.2.2 Performance requirements 8

5 Possible solutions to enable Multiparty RTT over RTP 8

5.1 Architecture considerations 8

5.2 Possible procedures 9

5.2.1 RTT-mixed SDP negotiation between two parties Procedure 9

5.2.2 RTT-mixed SDP negotiation for Multiparty Procedure 10

5.2.3 Multiparty RTT Processing Procedure 11

6 Possible solutions to enable Multiparty RTT over IMS data channel 11

6.1 Architecture considerations 11

6.2 Possible procedures 12

6.2.1 Multi DC Streams 12

6.2.1.1 UE Aware Mode 13

6.2.1.2 UE Unaware Mode 15

6.2.2 Single DC Stream 17

Annex <A> (informative): Change history 19

# Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# 1 Scope

The present document provides the protocol details for the Multiparty Real-Time Text (Multiparty RTT) service in the IP Multimedia (IM) Core Network (CN) subsystem based on the requirements from 3GPP TS 22.173 [2].

The Multiparty RTT service is an operator specific service by which an operator enables the subscribers in conference to use real-time text during the conference session.

The present document is applicable to User Equipment (UE) and Application Servers (AS) which are intended to support the Multiparty RTT service.

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 22.173: "Multimedia Telephony Service and supplementary services".

[3] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS); Stage 2".

[4] 3GPP TS 24.147: "Conferencing Using IP Multimedia Core Network; Stage 3".

[5] ITU-T T.140 Protocol for multimedia application text conversation, 02/1998, and T.140 Addendum 1, 02/2000

[6] IETF RFC 5194(2008) “Framework for Real-Time Text over IP Using the Session Initiation Protocol (SIP)”

[7] ITU-T F.700 Framework Recommendation for multimedia services, 11/2000

[8] IETF RFC 9071(2021): “RTP-Mixer Formatting of Multiparty Real-Time Text”

[9] IETF RFC 8865 (2021): “T.140 Real-Time Text Conversation over WebRTC Data Channels”

[10] IETF RFC 4103 (2005): "RTP Payload for Text Conversation", G. Hellstrom and P. Jones.

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

## 3.2 Symbols

void

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

AS Application Server

BFCP Binary Floor Control Protocol

CSCF Call Session Control Function

DC Data Channel

IMS IP Multimedia Subsystem

ITU-T International Telecommunication Union-Telecommunication Standardization Sector

MRF Multimedia Resource Function

MTSI Multimedia Telephony Service over IMS

PRACK Provisional Response Acknowledgement

RTC Real-Time Communication

RTP Real-time Transport Protocol

RTT Real-Time Text

SDP Session Description Protocol

SIP Session Initiation Protocol

UE User Equipment

WebRTC Web Real-Time Communication

# 4 Introduction of multiparty real-time text (Multiparty RTT)

## 4.1 General

Real-time text (RTT) may be used during conference sessions so that all call participants are enabled to create and send RTT to the other participants, and the text being presented in readable chunks growing in real time with indication of source. The presentation provides an approximate view of the relative timing of text from different parties.

The MTSI client in terminal may support multiparty real-time text (Multiparty RTT) as defined in this clause, the Multiparty RTT functionality has two solutions, over RTP and over IMS data channel. For the case of DCMTSI client in terminal, only the IMS data channel based solution is applicable.

The Multiparty RTT functionality for MTSI enables support of real-time text communication between each participant in a conference session. It addresses scenarios in some special groups of people, e.g., deaf person calls emergency service using RTT.

## 4.2 Study on Multiparty RTT requirements from other standards

### 4.2.1 General requirements

The general Multiparty RTT requirements from existing standards are listed as follows:

- A solution should be applicable to IMS as specified in 3GPP TS 23.228 [3]. Additionally, 3GPP TS 24.147 [4] provides the protocol details for conferencing within IMS based on SIP, SIP Events, SDP and the Binary Floor Control BFCP.

- If text loss is detected or suspected, a missing text marker should be inserted in the text stream as defined in ITU-T T.140[5].

- 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 as defined in ITU-T T.140 [5].

- It should be possible to use real-time text in conferences both as a medium of discussion between individual participants (for example, for sidebar discussions in real-time text while listening to the main conference audio) and for central support of the conference with real-time text interpretation of speech. Further session setup and control requirements can be found in RFC5194 [6].

### 4.2.2 Performance requirements

The Multiparty RTT performance requirements from existing standards are listed as follows:

- The mixer performance requirements can be expressed in one number, extracted from the user requirements on real-time text expressed in ITU-T F.700 [7], where it is stated that for "good" usability, text characters should not be delayed more than 1 second from creation to presentation. For "usable" usability the figure is 2 seconds.

- If buffering is provided in the data channel, it should not delay transmission more than 500ms. A buffering time of 300ms is recommended when the application or end-to-end network conditions are not known to require another value as indicated in RFC 4103[10].

# 5 Possible solutions to enable Multiparty RTT over RTP

## 5.1 Architecture considerations

The Multiparty RTT over RTP solution can reuse the existing architecture, which is defined in clause 4 of TS 23.228[3].

According to clause 1.2 of RFC 9071[8], for multiparty considerations, several alternatives were introduced, but only two alternatives were selected when searching for an efficient and easily implemented multiparty method for real-time text:

1. RTP-mixer-based method for multiparty-aware endpoints:

This solution is used when the endpoint supports multiparty-aware identifying by “a=rtt-mixer” in the SDP negotiation procedure. Only one single RTP stream for each participant, the source is indicated in the CSRC element in the RTP packets. Text from one source should be transmitted in the same packet if available for transmission at the same time. Text from different sources should not be transmitted in the same packet.

The main advantage of this solution is that it provides good performance for multiparty RTT communication with real time transmission. But it also creates new requirements on the endpoint.

1. Mixing for multiparty-unaware endpoints:

This solution is used as a fallback solution when the receiving endpoint is not capable of handling the mixed format. This is made possible by having the mixer insert a new line and a text-formatted source label before each switch of text source in the stream. Switching the source can only be done in places in the text where it does not disturb the perception of the contents. Text from only one source at a time can be presented in real time. The delay will therefore vary.

The main advantage of this solution is no need modifications in existing user devices implementing RFC4103[10] for real-time text. But the text

## 5.2 Possible procedures

#### 5.2.1 RTT-mixed SDP negotiation between two parties Procedure



Figure 5.2.1-1 RTT-mixed SDP negotiation between two parties

The main steps for RTT-mixed SDP negotiation between two parties are shown as below:

1. If the caller party supports RTP-mixer-based method, when the caller party initiates an SDP offer, it can add “a=rtt-mixer” in “m=text” line. The SDP example is shown as below:

Table 5.2.1.1: SDP example

|  |
| --- |
| **SDP offer** |
|  |

2-3. If the called party supports RTP-mixer-based method, when the called party receives an SDP offer containing “a=rtt-mixer” in “m=text” line, it would include “a=rtt-mixer” in the corresponding “m=text” line in the SDP answer. The SDP example is shown as below:

Table 5.2.1.2: SDP example

|  |
| --- |
| **SDP answer** |
|  |

4-5. If the called party doesn’t support RTP-mixer-based method, when the called party receives an SDP offer containing “a=rtt-mixer” in “m=text” line, it would remove “a=rtt-mixer” in the corresponding “m=text” line in the SDP answer. The SDP example is shown as below:

Table 5.2.1.3: SDP example

|  |
| --- |
| **SDP answer** |
|  |

#### 5.2.2 RTT-mixed SDP negotiation for Multiparty Procedure



Figure 5.2.2-1 RTT-mixed SDP negotiation for Multiparty

The main steps for RTT-mixed SDP negotiation for Multiparty are shown as below:

1-2. UE-A creates a conference.

3. UE-A will finish SDP negotiation with MRF, the RTT-mixed SDP negotiation procedure is the same as Figure 5.2.1-1.

4-6. UE-A invites UE-B to the conference, UE-A sends a REFER message to IMS, IMS will finish SDP negotiation with UE-B, the RTT-mixed SDP negotiation procedure is the same as Figure 5.2.1-1.

7-9. UE-A invites UE-C to the conference, UE-A sends a REFER message to IMS, IMS will finish SDP negotiation with UE-C, the RTT-mixed SDP negotiation procedure is the same as Figure 5.2.1-1.

10-11. UE-D joins the conference, IMS will finish SDP negotiation with UE-D, the RTT-mixed SDP negotiation procedure is the same as Figure 5.2.1-1.

#### 5.2.3 Multiparty RTT Processing Procedure



Figure 5.2.3-1 Multiparty RTT over RTP Processing

Taking into consideration that some UEs might not support RTT, e.g. UE-A and support, while same

As illustrated in figure 5.2.3-1,the main steps for these typical scenarios are shown as below:

1. UE-A sends RTT in the conference, the RTT content in RTP packet would follow RFC 4103[10].

2. MRF acts as a mixer, and MRF will decide how to handle the RTT content based on the SDP negotiation on rtt-mixer with UE-B and UE-C.

3. For UE-B that supports RTT-mixer-based method, MRF will modify the RTP packets, set CC=1, and put UE-A in the CSRC list. An example is shown as below:

|Seq no 101, Time=20400 |

|CC=1 |

|CSRC list A |

|R2: Empty, Offset=600 |

|R1: Empty, Offset=300 |

|P: A1 |

4. For UE-C that does not support RTT-mixer-based method, MRF will treat it as multiparty-unaware endpoint, a presentable label be composed and sent for the source initially in the session and after each source switch. An example is shown as below:

|Seq no 101, Time=20400 |

|CC=0 |

|SSRC |

|R2: Empty, Offset=600 |

|R1: Empty, Offset=300 |

|P: [UE-A]A1 |

# 6 Possible solutions to enable Multiparty RTT over IMS data channel

## 6.1 Architecture considerations

The Multiparty RTT over data channel solution is based on data channel architecture, which is defined in clause AC.2.1 of TS 23.228 [3].

According to clause 5.5 of RFC8865 [9], for multiparty considerations, two alternatives were considered when searching for an efficient and easily implemented multiparty method for real-time text:

1. Multiple DC streams, one per participant:

One DC stream per source would be sent in the same session. UE can identify the source by the “label” attribute in the DC stream ID line when receiving RTT. If a new UE is added to the conference, a new downlink stream ID indicating the new UE would be added to all the existing participants. The conference application needs to manage the mapping relationship between the UE identity and the steam ID of each participant, obtain the corresponding UE identity according to the stream ID when receiving the real-time text of each participant.

The main advantage of this solution is, as a straightforward solution, the load per source is low. But with a high number of participants, the overhead of establishing and maintaining the high number of data channels required may be high, even if the load per channel is low.

1. Single DC stream, each participate use only one DC stream:

Only one DC stream for each participate, no SIP negotiation procedure for each participant when a new UE is added to the conference. The conference server would add a source information in front of the RTT content by identifying the label attribute in the DC stream ID line when receiving RTT from a UE.

The main advantage of this solution is no need negotiation when a new UE is added to the conference. But the conference server would add decode and re-encode the RTT content.

## 6.2 Possible procedures

#### 6.2.1 Multi DC Streams



Figure 6.2.1-1 Multi DC Streams Example

An example for three participants in a conference as shown in figure 6.2.1-1, each UE has one uplink stream ID and two downlink stream IDs, if a new UE is added to the conference, a new downlink stream ID indicating the new UE would be added to all the existing participants.

The conference server (DCMF/MRF or DC AS) needs to manage the mapping relationship between the UE identity and the steam ID of each participant, obtain the corresponding UE identity according to the stream ID when receiving the real-time text of each participant, and add the UE identity before the real-time text to correctly display the source.

The call flows for different UE modes are shown below separately:

1) UE Aware Mode: The conference creator needs to be aware of who are in the conference, and carry each participant’s user name when initiates ADC establishment.

2) UE Unaware Mode: The conference creator needs to be aware of who are in the conference, only carry itself user name when initiates ADC establishment.

#### 6.2.1.1 UE Aware Mode



Figure 6.2.1.1-1 Multi DC Streams with UE Aware Mode Call Flow

The steps are shown as below:

**Case 1: UE-A create a conference and join UE-B and UE-C into the conference, then run the RTT application.**

1. UE-A, UE-B and UE-C enter an audio/video conference and download the RTT application on each participant.

2. The UE-A runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including 3 DC stream IDs, one ‘sendonly’ for UE-A sending RTT to other participants, one ‘recvonly’ for receiving UE-B’s RTT, and the last one ‘recvonly’ for receiving UE-C’s RTT, the label attribute in each ‘a=dcmap’ can be get from the conference information, which can identify each DC stream belongs to whom. The SDP offer example is shown as below:

Table 6.2.1.1: SDP example

|  |
| --- |
| **SDP offer** |
|  |

3. DCSF establishes corresponding DC stream IDs for UE-A.

4-5. IMS-A sends an REINVITE message with three stream IDs to UE-B and establish corresponding DC stream IDs for UE-B. The stream IDs are similar to step2.

6-7. IMS-A sends an REINVITE message with three stream IDs to UE-C and establish corresponding DC stream IDs for UE-C. The stream IDs are similar to step2.

**Case 2: UE-D call into the conference and run the RTT application.**

8. UE-D calls into the conference created by UE-A, and runs the RTT application.

9. UE-D runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including one DC stream ID for UE-D sending RTT to other participants.

10. IMS-A establishes the DC stream for UE-D.

11. The IMS-A identifies that there are three participants in the conference, so IMS-A decides to add a new downlink DC stream for each participant, and finally add three downlink streams for UE-D.

12-14. The IMS-A adds a new downlink DC stream for UE-A/UE-B/UE-C simultaneously, for receiving UE-D’s RTT.

15. The IMS-A adds three downlink DC streams for UE-D, for receiving UE-A/UE-B/UE-C’s RTT.

When UE-A sends RTT over the uplink stream ID, DCMF/MRF will simultaneously send the RTT to UE-B, UE-C and UE-D through the dedicated stream ID channel, UE-B, UE-C and UE-D can identify the source by the corresponding “label” attribute that included in the ‘a=dcmap’ line.

#### 6.2.1.2 UE Unaware Mode



Figure 6.2.1.2-1 Multi DC Streams with UE Unaware Mode Call Flow

The steps are shown as below:

**Case 1: UE-A creates a conference and joins UE-B and UE-C into the conference, then runs the RTT application.**

1. UE-A, UE-B and UE-C enter an audio/video conference and download the RTT application on each participant.

2. The UE-A runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including one uplink DC stream ID with ‘sendonly’ for UE-A sending RTT to other participants, the label attribute in ‘a=dcmap’ can be get from UE-A’s identity. The SDP offer example is shown as below:

Table 6.2.1.2.1: SDP example

|  |
| --- |
| **SDP offer** |
|  |

3. DCSF establishes corresponding DC stream ID for UE-A.

4. The IMS-A identifies that there are three participants in the conference, so IMS-A decides to add another two new downlink DC streams for UE-A, and three DC streams including one uplink DC streams and two downlink DC streams for the other participants.

5-6. IMS-A sends an REINVITE message adding two downlink stream IDs to UE-A and establish corresponding DC stream IDs for UE-A. The SDP offer example is shown as below:

Table 6.2.1.2.2: SDP example

|  |
| --- |
| **SDP offer** |
| m=application 911 UDP/DTLS/SCTP webrtc-datachannelc=IN IP6 2001:db8::3a=max-message-size:1000a=sctp-port 5000a=setup:actpassa=dcmap:200 label="A-Identity";subprotocol="t140" a=dcsa:200 fmtp:t140 cps=20 recvonlya=dcsa:200 hlang-send:es eoa=dcmap:201 label="B-Identity";subprotocol="t140" a=dcsa:201 fmtp:t140 cps=20 sendonlya=dcsa:201 hlang-recv:es eoa=dcmap:202 label="C-Identity";subprotocol="t140" a=dcsa:202 fmtp:t140 cps=20 sendonlya=dcsa:202 hlang-recv:es eo |

7-8. IMS-A sends an REINVITE message with three stream IDs to UE-B and establish corresponding DC stream IDs for UE-B. The stream IDs are similar to step4.

9-10. IMS-A sends an REINVITE message with three stream IDs to UE-C and establish corresponding DC stream IDs for UE-C. The stream IDs are similar to step4.

**Case 2: UE-D calls into the conference and run the RTT application.**

11. UE-D calls into the conference created by UE-A, and runs the RTT application.

12. UE-D runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including one DC stream ID for UE-D sending RTT to other participants.

13. IMS-A establishes the DC stream for UE-D.

14. The IMS-A identifies that there are three participants in the conference, so IMS-A decides to add a new downlink DC stream for each participant, and finally add three downlink streams for UE-D.

15-17. The IMS-A adds a new downlink DC stream for UE-A/UE-B/UE-C simultaneously, for receiving UE-D’s RTT.

18. The IMS-A adds three downlink DC streams for UE-D, for receiving UE-A/UE-B/UE-C’s RTT.

When UE-A sends RTT over the uplink stream ID, DCMF/MRF will simultaneously send the RTT to UE-B, UE-C and UE-D through the dedicated stream ID channel, UE-B, UE-C and UE-D can identify the source by the corresponding “label” attribute that included in the ‘a=dcmap’ line.

#### 6.2.2 Single DC Stream



Figure 6.2.2-1 Single DC Stream Example

Figure 6.2.2-1 illustrates the single DC stream example. T140 protocol is too old to be extended to support adding the source label, so the conference server (DCMF/MRF or DC AS) can add a source label getting from the “label” attribute of ‘a=dcmap’ line in front of the text content when receiving the real-time text from a UE, and the terminal can display it directly without modification.



Figure 6.2.2-2 Single DC Stream Call Flow

The steps are shown as below:

**Case 1: UE-A create a conference and join UE-B and UE-C into the conference, then run the RTT application:**

1. UE-A, UE-B and UE-C enter an audio/video conference and download the RTT application on each participant.

2. The UE-A runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including only one DC stream ID, the SDP offer example:

Table 6.2.2.1: SDP example

|  |
| --- |
| **SDP offer** |
|  |

3. DCSF establishes corresponding DC stream ID for UE-A.

4-5. IMS-A sends an REINVITE message with only one stream ID to UE-B and establishes corresponding DC stream ID for UE-B. The stream ID is similar to step2.

6-7. IMS-A sends an REINVITE message with only one stream ID to UE-C and establishes corresponding DC stream ID for UE-C. The stream ID is similar to step2.

**Case 2: UE-D call into the conference and run the RTT application.**

8. UE-D calls into the conference created by UE-A, and runs the RTT application.

9. UE-D runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including one DC stream ID for UE-D sending and receiving RTT.

10. IMS-A establishes the DC stream for UE-D.

When UE-A sends RTT over the uplink stream ID, DCMF/MRF will identify the source by the application data channel established is between UE-A and DCMF/MRF, and then add the UE-A’s identity as source to the RTT content. DCMF simultaneously send the RTT to UE-B, UE-C and UE-D through the dedicated stream ID channel, UE-B, UE-C and UE-D directly display the RTT content.

Annex <A> (informative):
Change history

|  |
| --- |
| Change history |
| Date | Meeting | TDoc | CR | Rev | Cat | Subject/Comment | New version |
| 2023-11 | SA4#126 | S4-231625 |  |  |  | Initial draft | 0.1.0 |