ETSI TS 126 506 V18.2.0 (2024-05)

TECHNICAL SPECIFICATION

5G;
5G Real-time Media Communication Architecture (Stage 2)
(3GPP TS 26.506 version 18.2.0 Release 18)

Reference

DTS/TSGS-0426506vi20

Keywords

5G

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

Important notice

The present document can be downloaded from:
https://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

If you find a security vulnerability in the present document, please report it through our
Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

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.

Legal Notice

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under https://webapp.etsi.org/key/queryform.asp.

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.

Contents

Foreword

This Technical Specification 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.

Introduction

1 Scope

The present document specifies an architecture for real-time media communication integrated into the 5G System. To support Mobile Network Operator (MNO) and third-party services for real-time media, essential functionalities and interfaces are specified. The primary scope of this Technical Specification is the documentation of the following aspects:

- The definition of a real-time media communication architecture mapped to the 5GS architecture, with relevant core building blocks, reference point, and interfaces to support modern operator and third-party media services, based on the 5GMS architecture.
- Definition of all relevant reference points and interfaces to support different collaboration scenarios between 5G System operator and third-party media communication service provider, including but not limited to an Augmented Reality (AR) media communication service provider.
- Call flows and procedures for different real-time communication service types.
- Specification to support functionalities relevant to AR such as split-rendering or spatial computing on top of a 5G System based on this architecture.

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 TR 26.998: "Support of 5G glass-type Augmented Reality / Mixed Reality (AR/MR) devices".

[3] 3GPP TS 26.119: "Media Capabilities for Augmented Reality".

[4] 3GPP TS 26.113: "Enabler for Immersive Real-time Communication".

[5] 3GPP TR 26.930: "Study on the enhancement for Immersive Real-Time communication for WebRTC".

[6] 3GPP TS 26.501: "5G Media Streaming (5GMS); General description and architecture".

[7] 3GPP TS 23.558: "Architecture for enabling Edge Applications".

[8] 3GPP TS 38.321: "NR; Medium Access Control (MAC) protocol specification".

[9] 3GPP TS 36.321: "LTE; Medium Access Control (MAC) protocol specification".

[10] 3GPP TS 26.114: "IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction".

[11] 3GPP TS 23.501: " System architecture for the 5G System (5GS)".

3 Definitions of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in 3GPP 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 3GPP TR 21.905 [1].

3.2 Symbols

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

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP 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 3GPP TR 21.905 [1].

AR	Augmented Reality
EAS	Edge Application Server
ECS	Edge Configuration Server
EEC	Edge Enabler Client
EES	Edge Enabler Server
IETF	Internet Engineering Task Force
ICE	Interactive Connectivity Establishment
IMS	IP Multimedia Subsystem
MCU	Multi-point Control Unit
MNO	Mobile Network Operator
MR	Mixed Reality
MSH	Media Session Handler
MTSI	Multimedia Telephony Service for IMS
NAT	Network Address Translation
RTC	Real-Time Media Communication
SDP	Session Description Protocol
SFU	Selective Forwarding Unit
STUN	Session Traversal Utilities for NAT
TURN	Traversal Using Relays around NAT
W3C	World Wide Web Consortium
WebRTC	Web Real-Time Communication

4 Real-Time media Communication Architecture

4.1 Overall architecture for Real-Time media Communication (RTC)

4.1.1 Definition of RTC architecture

Real-Time media Communication (RTC) over 5G system in the context of this specification is defined as the delivery of delay-sensitive media from one peer to another with support of 5G network. AR conversational service described in TR 26.998 [2] is a typical use cases for RTC, which enables end-users to directly communicate real-time media including AR/MR media content as specified in TS 26.119 [3]. As identified in clause 8.4 of TR 26.998, there may be different options to enable such AR conversational service, for example re-use of parts of MTSI as defined in TS 26.114 [10] such as the IMS data channel or 5G Media Streaming for managed services.

The overall RTC architecture is shown in Figure 4.1.1-1 as below.

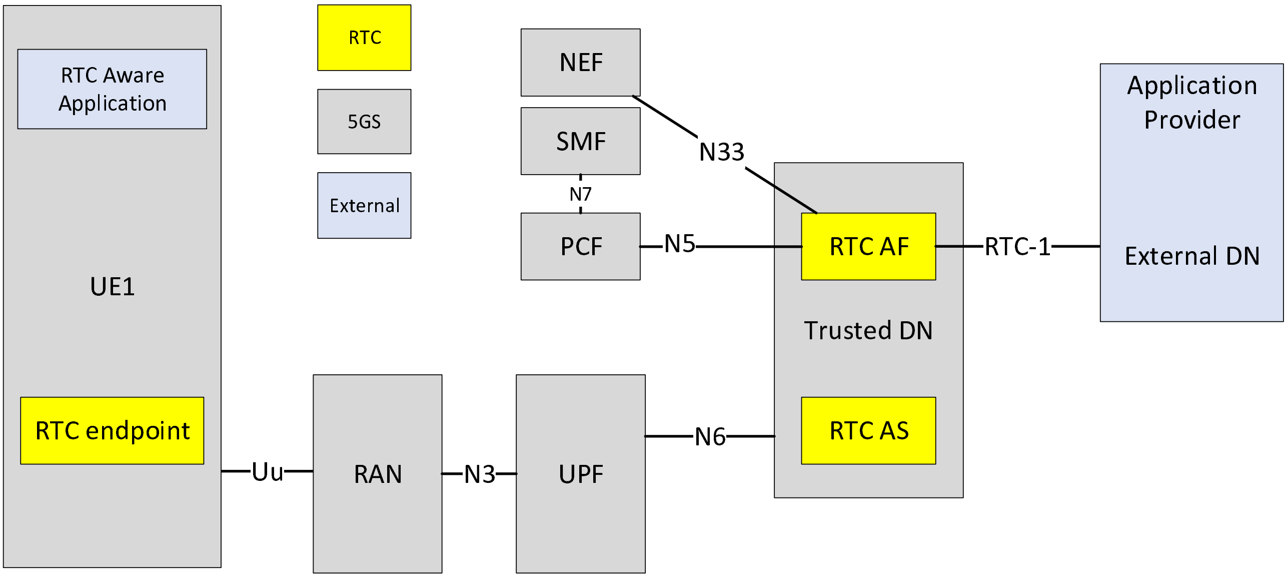


Figure 4.1.1-1: Real-time media communication (RTC) in 5G System

NOTE: The functions indicated by the yellow filled boxes are in scope of the present document for RTC. The functions indicated by the grey boxes are defined in 5G System specifications. The functions indicated by the blue boxes are neither in scope of 5G RTC nor 5G System specifications.

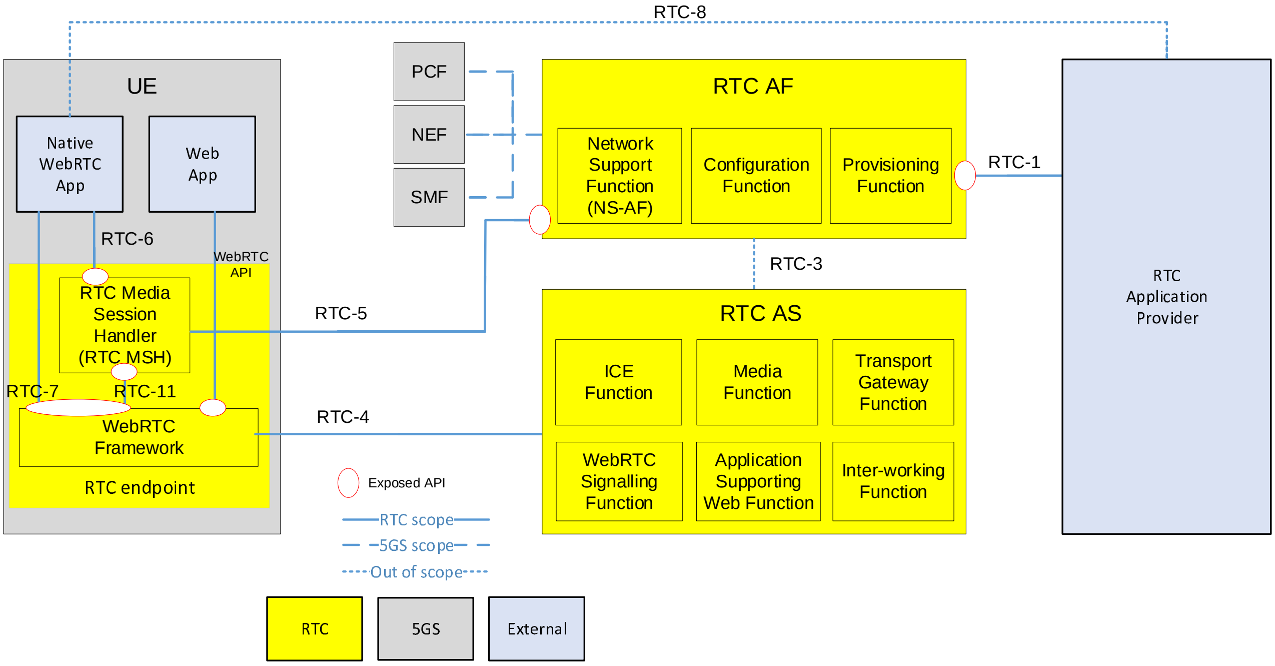
The media data is exchanged between two or more RTC endpoints over a 5G System as defined in TS 23.501 [11]. The RTC endpoint is an endpoint configured by RTC architecture in the present document. It is typically a UE, but an edge computing server can also be the RTC endpoint. The Application Provider provides a RTC Aware-Application on the UE to make use of RTC endpoint and network functions using interfaces and APIs. RTC architecture provides the core functions and entities to support WebRTC-based service over 5G System, two main functions are defined in the trusted DN.

- RTC AF: An Application Function as defined in TS 26.501 [6], but dedicated to real-time media communication.
- RTC AS: An Application Server dedicated to real-time media communication.

NOTE: If both the RTC AF and RTC AS are deployed in an external DN, this is out of scope of the present document.

The detailed RTC architecture mapping to the overall high-level architecture in Figure 4.1.1-1 is shown in Figure 4.1.1-2 below.

NOTE: Figure 4.1.1-2 illustrates only the link from one RTC endpoint to the RTC AF and RTC AS. The link from another RTC endpoint in communication with the first one is symmetric.



NOTE 1: Some subfunctions may not be required depending on the collaboration scenario. Description of collaboration scenario and its architecture variant are specified in annex A.

NOTE 2: The WebRTC Framework subfunction is a WebRTC protocol stack whose implementation is specified by W3C and IETF.

NOTE 3: Red ovals indicate API provider functions.

Figure 4.1.1-2: RTC General Architecture

The WebRTC Signalling Function may be co-located with the RTC AF. In such deployments, the WebRTC Signalling Function acts as an RTC AF with access to the 5G Core, and some of the RTC AF interactions with the WebRTC Signalling Function may be replaced to avoid concurrent/redundant requests from the RTC endpoint in the UE. Specifically, media session handling interactions between the RTC AF and the UE at reference point RTC-5 may be replaced by the equivalent WebRTC signalling interactions defined at reference point RTC-4.

The subfunctions inside the RTC AF, RTC AS and the RTC endpoint are defined in clause 4.2 and the reference pointsshown in Figure 4.1.1-2 are defined in clause 4.3.

4.1.2 Generalized Media Delivery architecture

4.1.2.1 Generalized Media Delivery in the 5G System

This clause and subsequent subclauses of clause 4.1.2 define a generalized Media Delivery architecture of which the architecture for Real-Time Communication (RTC) defined elsewhere in the present document is one possible realisation. In case of any misalignment between the two, the RTC architecture has precedence over this generalised architecture.

Due to the similarity of the 5GMS architecture (as defined in TS 26.501 [6]) to the architecture for Real-Time media Communication (RTC) defined in the present document, the RTC functions and 5GMS functions may share or may make use of many common functionalities for both media session handling and media delivery. A generalized Media Delivery architecture that integrates 5GMS and RTC functionality in the 5G System is defined in figure 4.1.2.1-1.

NOTE: Full integration of 5GMS and RTC is not addressed in the present document.

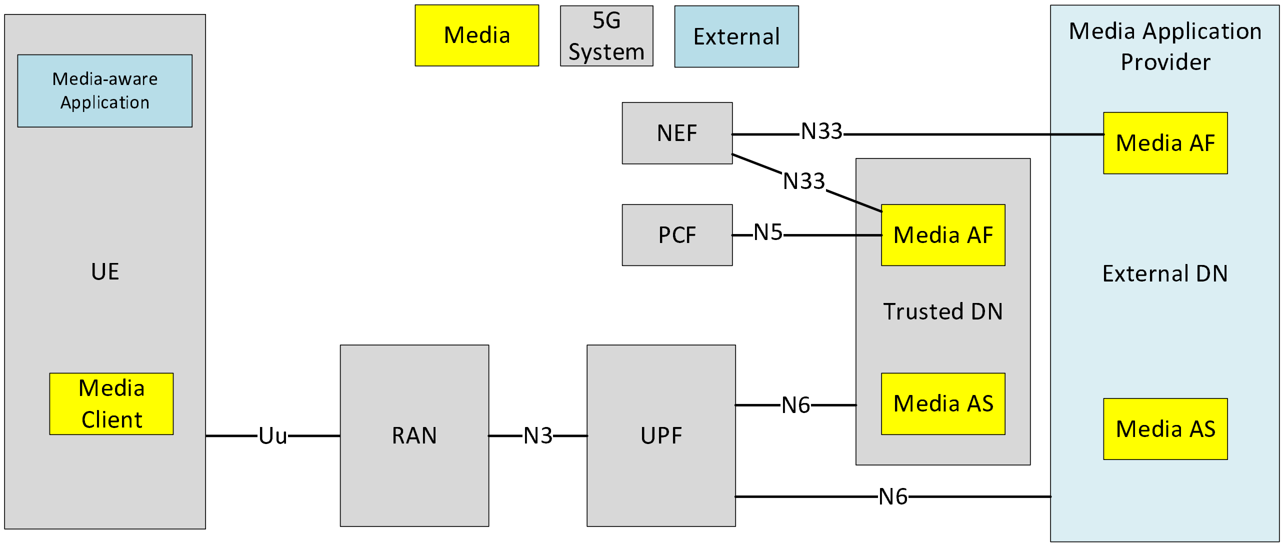


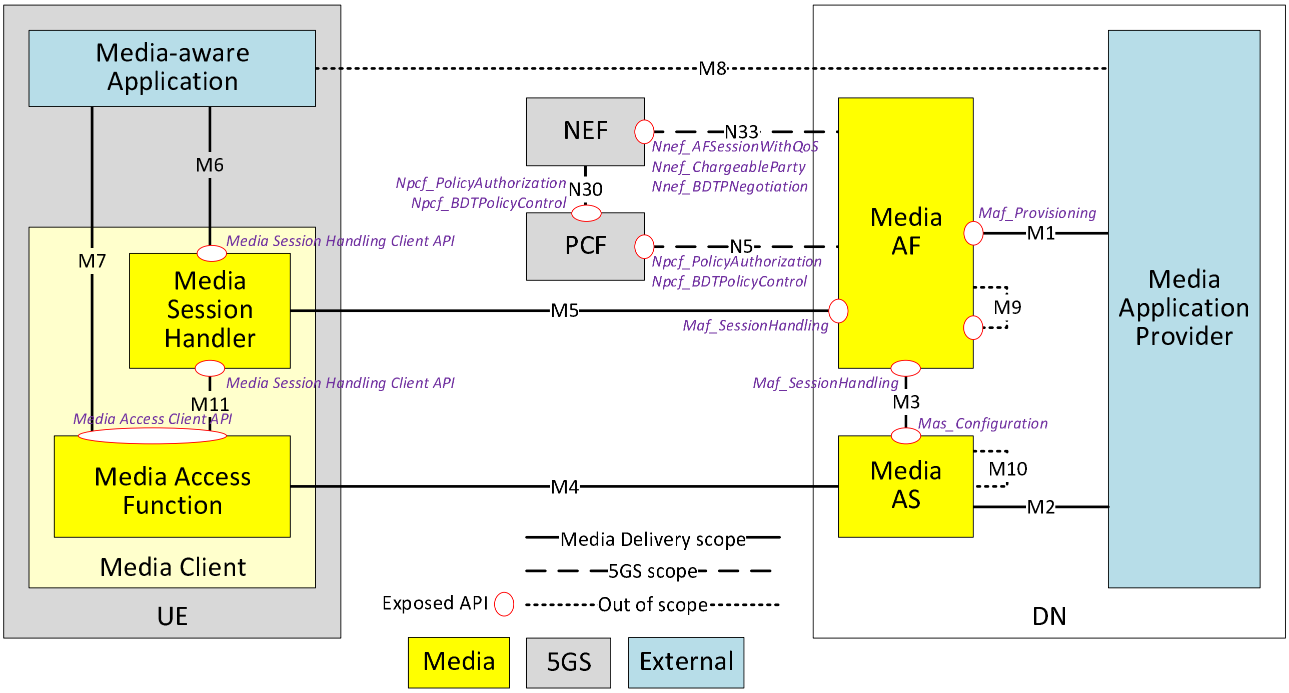
Figure 4.1.2.1-1: Generalized Media Delivery architecture within the 5G System

In this representation:

- The *Media Application Provider* plays the role of the RTC Application Provider.
- The *Media-aware Application* plays the role of the Native WebRTC App.
- The RTC AF is one possible realisation of the general *Media AF*.
- The RTC AS is one possible realisation of the general *Media AS*.
- The RTC endpoint is part of the general *Media Client*.

4.1.2.2 Reference architecture for Media Delivery

A functional description with additional details as well as reference points is provided below, as illustrated in figure 4.1.2.2-1.



NOTE 1: Exposed APIs are named in *italics*.

NOTE 2: If the Media Client is deployed as a monolithic functional block, it may choose not to expose interfaces externally at reference point M11.

Figure 4.1.2.2-1: Generalized Media Delivery architecture

4.1.2.3 Network Functions and UE entities

Functional definitions may be generalized as follows:

- **Media AF:** An Application Function as defined in clause 6.2.10 of TS 23.501 [11] dedicated to Media Delivery.

- **Media AS:** An Application Server dedicated to Media Delivery.

- **Media Client:** A UE internal function dedicated to Media Delivery comprising:

    - **Media Session Handler:** An entity on the UE that communicates with the Media AF in order to establish, control and support the delivery of a media session.

    - **Media Access Function:** An entity on the UE that communicates with the Media AS in order to access and deliver media content. The media access function for example may be further sub-divided into content delivery protocols, codecs, media types and metadata representation.

- **Media-aware Application:** An application entity on the UE that makes use of 3GPP-defined APIs to invoke the Media Session Handler and/or the Media Access Function in order to support Media Delivery.

NOTE: An application (e.g., a web browser application) that does not invoke either the Media Session Handler or the Media Access Function using 3GPP-defined APIs is not considered a Media-aware Application and is not mapped into the generalized Media Delivery reference architecture.

Table 4.1.2.3-1: Mapping of RTC functions to generalized Media Delivery architecture

Generalized media architecture function	RTC function
Media AF	RTC AF
Media AS	RTC AS
Media Client	RTC endpoint
Media Session Handler	RTC Media Session Handler
Media Access Function	WebRTC Framework
Media Application Provider	RTC Application Provider
Media-aware Application	Native WebRTC App

4.1.2.4 Reference points

The following reference points are defined for Media Delivery:

M1: Reference point between the Media Application Provider and the Media AF for the provisioning of Media Delivery.

M2: Reference point between the Media Application Provider and the Media AS for the purposes of ingesting media into the Media AS or egesting media from the Media AS.

NOTE 1: Reference point M2 is not defined by the RTC architecture in this release.

M3: Reference point between the Media AF and the Media AS for the purposes of Media AS configuration and/or for media session handling in relation to Media Delivery.

NOTE 2: Reference point M3 is defined by the RTC architecture in this release but specification is for future study.

M4: Reference point between the Media AS and the Media Access Function in the UE for the purpose of downlink transport of media from the Media AS to the Media Access Function ("content distribution") or uplink transport of media from the Media Access Function to the Media AS ("content contribution").

NOTE 3: Session setup signalling at reference point RTC-4 lies outside the scope of reference point M4.

M5: Reference point between the Media AF and the Media Session Handler in the Media Client for the purpose of media session handling in relation to Media Delivery.

M6: Reference point between the Media-aware Application and the Media Session Handler for the purpose of configuring the Media Session Handler.

M7: Reference point between the Media-aware Application and the Media Access Function for the purpose of media access control.

M8: Reference point between the Media-aware Application and the Media Application Provider.

NOTE 4: Reference point M8 is private and therefore beyond the scope of standardisation.

M9: Reference point between one instance of the Media AF and another for the purpose of Media AF instance chaining.

NOTE 5: Reference point M9 is not defined by the RTC architecture.

M10: Reference point between one instance of the Media AS and another for the purpose of peer-to-peer media transport between different Media Clients.

NOTE 6: Reference point M10 is not defined by the RTC architecture in this release.

M11: Reference point between the Media Session Handler and the Media Access Function (both in the Media Client) for the purpose of configuring the Media Session Handler and/or media access control.

Table 4.1.2.4-1: Mapping of RTC reference points to generalized Media Delivery architecture

Generalized Media Delivery architecture reference point	RTC reference point
M1	RTC-1
M2	Not defined
M3	RTC-3
M4	RTC-4
M5	RTC-5
M6	RTC-6
M7	RTC-7
M8	RTC-8
M9	Not defined
M10	Not defined
M11	RTC-11

4.1.2.5 Interfaces and APIs

4.1.2.5.1 Interfaces and APIs supporting media session handling

The Media AF exposes the following network service interfaces for media session handling:

- *Provisioning API* (*Maf_Provisioning*): External API, exposed to the Media Application Provider by the Media AF at reference point M1 to provision the usage of the Media Delivery and to obtain feedback.

- *Media Session Handling API* (*Maf_SessionHandling*) exposed by a Media AF to the Media Session Handler at reference point M5 and/or to the Media AS at reference point M3 for media session handling, control, reporting and assistance that also include appropriate security mechanisms, e.g. authorization and authentication.

The Media Session Handler exposes the following UE APIs for media session handling:

- *Media Session Handling Client API*: exposed by the Media Session Handler to the Media-aware Application at reference point M6 and to the Media Access Function at reference point M11, for configuring media session handling, including service launch.

4.1.2.5.2 Interfaces and APIs supporting media transport

The Media AS exposes the following network service interfaces to support media transport:

- *Media Application Server Configuration API* (*Mas_Configuration*) used by the Media AF at reference point M3 to configure the Media AS.

The Media AS exposes the following media transport interfaces:

- *Application Provider media transport interface* between the Media AS and the Media Application Provider, used to exchange media data using a media transport protocol at reference point M2.

- *Client-facing media transport interface* between the Media Access Function and the Media AS, used to exchange media data using a media transport protocol at reference point M4.

The Media Access Client exposes the following UE APIs for media access control:

- *Media Access Control API* exposed by the Media Access Function to the Media-aware Application at reference point M7 and to the Media Session Handler at reference point M11, in order to configure and communicate with the Media Access Function.

4.1.2.5.3 Interfaces and APIs supporting application functionality

The Media Application Provider exposes the following network service interfaces to support application functionality:

- *Application-private API* used for information exchange between the Media-aware Application and the Media Application Provider at reference point M8.

4.2 Functions and entities

4.2.1 General

This clause defines minimal and essential functions as well as extra functions and entities that may appear in certain deployment or collaboration scenarios.

4.2.2 Provisioning Function

The Provisioning Function of the RTC AF enables an RTC Application Provider to provision the following functionalities:

- QoS support for WebRTC sessions.
- Charging for WebRTC sessions.
- Collection of consumption and QoE metrics data related to WebRTC sessions.
- Offering Interactive Connectivity Establishment (ICE) functionality to support Network Address Traversal /NAT) such as Session Traversal Utilities for NAT (STUN) and Traversal Using Relays around NAT (TURN) servers.
- The WebRTC Signalling Function in the RTC AS, potentially offering interoperability with other compatible signalling servers.

The Provisioning Function may not be relevant to all collaboration scenarios and some of the 5G support functionality may be offered without RTC Application Provider provisioning.

4.2.3 Configuration Function

The Configuration Function stores WebRTC-related configuration information and makes them accessible to the UE. It stores information and recommendations to operate network-assisted WebRTC sessions over 5G system.

The configuration information may consist of static information such as the following:

- Recommendations for media configurations
- Configurations of STUN and TURN server locations
- Configuration about consumption and QoE reporting
- Discovery information for WebRTC signalling and data channel servers and their capabilities in static and/or dynamic way.

NOTE: The integration/co-location of this RTC AF and WebRTC signalling function is possible. Co-located WebRTC signalling function is able to act as a RTC AF which is accessible to 5GC, and replace some of this RTC AF's interfaces and APIs with WebRTC signalling function. For example, interfaces and APIs between this RTC AF and UE will be replaced to avoid concurrent/redundant requests from UE.

4.2.4 RTC Media Session Handler (MSH)

The RTC MSH is an entity running on the UE, which assists with the 5G integration of the WebRTC application. It exchanges, on behalf of the application, information about the WebRTC sessions with the network.

The RTC MSH receives information about a new WebRTC session from the application. It relays the information to the Network Support Function. It also receives events and other network information about the WebRTC session from the Network Support Function, which it may relay to the application.

In addition, one of subfunction in RTC MSH is the metric collection and reporting. It executes the collection of QoS and QoE metrics measurements from the WebRTC Framework and the WebRTC application and sends metrics reports to the RTC AF for the purpose of metrics analysis or to enable potential transport optimizations by the network.

4.2.5 Network support function

The support functionality includes the following:

- Network Support Function receives information from the UE and/or other ASs about a WebRTC session and its state
- Network Support Function requests the network that QoS should be allocated (or satisfied) for a starting or modified session
- Network Support Function receives notification from the network about changes to the QoS allocation for the ongoing WebRTC session
- Network Support Function exchanges information about the WebRTC session with the trusted STUN/TURN/Signalling function, e.g. to identify a WebRTC session and associate it with a QoS template.

NOTE: The integration/collocation of this RTC AF and WebRTC signalling function is possible. Co-located WebRTC signalling function is able to act as a RTC AF which is accessible to 5GC, and replace some of this RTC AF's interfaces and APIs with WebRTC signalling function. For example, interfaces and APIs between this RTC AF and UE will be replaced to avoid concurrent/redundant requests from UE.

4.2.6 Trusted ICE functions

The MNO may offer trusted ICE functions to the WebRTC application to be used during the WebRTC ICE gathering phase. These functions may be STUN and TURN servers that facilitate NAT and firewall traversal.

The MNO-operated trusted ICE functions may assist with the 5G integration of the WebRTC application. This could be done by triggering network assistance to starting or ongoing WebRTC sessions.

4.2.7 Trusted WebRTC signalling function

The trusted WebRTC signalling function is used to setup and manage MNO-operated WebRTC applications. They offer a standardized signalling protocol for the session setup to both parties of the WebRTC session. The WebRTC signalling function handles the offer/answer exchange and has access to the SDP in both directions.

The WebRTC signalling function may use that knowledge to offer network assistance and other 5G features to the endpoints of the WebRTC session.

The WebRTC signalling function manages media flow sessions in both uplink and downlink directions.

4.2.8 Trusted inter-working function

This function provides inter-working functionality to enable MNO-facilitated WebRTC sessions that involve endpoints across different MNOs. They may for example provide cross-network signalling functionality to allow WebRTC signalling server that are hosted in different networks to communicate, in order to establish and manage the WebRTC sessions.

4.2.9 Trusted transport gateway function

A transport gateway function may be offered by the MNO to support cross-operator WebRTC sessions. It may offer the border control function for user plane (e.g., topology hiding, IPv4-IPv6 translation) as a gateway, which is located at the network boundary where different operators or third-party network connects. It works under the control of the trusted inter-working function.

Note: Detailed functionality is specified in TR 26.930 [5].

4.2.10 Trusted media function

A media server may be offered by the MNO to support WebRTC sessions. It may offer a wide range of functionality such as:

- a content server that serves content to the WebRTC application, e.g. through a data channel
- media processing functionality: may be used by the WebRTC application as a relay that performs some media processing function such as transcoding, recording, 3D reconstruction, etc.
- scene composition functionality: the server may compose a 3D scene and distribute it to several point-to-point WebRTC sessions
- Multi-point Control Unit (MCU) functionality: the server may offer multi-party conferencing functionality to merge a number of point-to-point WebRTC sessions
- Selective Forwarding Unit (SFU) functionality: the server may offer the selection, copy, and forwarding functionality of IP steams produced by multiple RTC endpoints (i.e., participants).
- Maintain uplink and downlink flow context (QoS, remote control and etc.) by interacting with the WebRTC signalling function.

4.2.11 Trusted application supporting web function

A web server may be offered by the MNO to support applications by providing web service entry point, authorization/authentication, sharing files, or scheduling conferencing sessions.

4.3 Interfaces

4.3.1 RTC-1: Provisioning interface

The RTC-1 interface allows the Application Provider to provision support for RTC sessions that are offered by it. The provisioning may cover the following aspects:

- QoS support for WebRTC sessions
- Charging provisioning for WebRTC sessions
- Collection of consumption and QoE metrics data related to WebRTC sessions
- Offering ICE functionality such as STUN and TURN servers
- Offering WebRTC signalling function, potentially with interoperability to other signalling servers

The provisioning interface is not relevant to all collaboration scenarios and some of the 5G support functionality may be offered without application provider provisioning.

4.3.2 RTC-3: RTC AS to RTC AF interface

The RTC AS may exchange information regarding the RTC session with the RTC AF. This information may cover QoS flow information and QoS allocation as well as QoE and consumption reports. The RTC AF may subscribe to information about the status of the QoS flow, which it may share with the RTC AS, e.g. in form of bitrate recommendations.

4.3.3 RTC-4: Media-centric transport interface

This interface is used to exchange the WebRTC traffic with the other endpoint as well as to exchange signalling information related to the WebRTC session with the trusted application servers.

The traffic includes:

- Media streams sent over RTP
- Application data sent over data channel
- WebRTC Signalling data along with STUN and TURN servers
- Other application data

RTC-4 may further be grouped into two sub-interfaces as follows.

RTC-4s:

The RTC-4s interface is an interface between the WebRTC framework and the RTC AS such as WebRTC Signalling function. This interface is used for the exchange of signalling information related to the WebRTC session between two or more WebRTC endpoints using trusted application servers. In some cases where the signalling is not handled by WebRTC framework, the RTC-4s interface is an interface between the native WebRTC applications and the WebRTC Signalling server.

RTC-4m:

This interface is used for transmission of media and other related data between two or more RTC endpoints.

The traffic includes

- Media data transmitted over RTP
- Application data transmitted using Data channel
- Media related meta-data transmitted using Data channel

NOTE 1: The Media Server should maintain the status for both uplink and downlink traffic and a separate interface for supporting downlink and uplink is expected to be defined in this specification.

NOTE 2: WebRTC-enabled UE should support streaming functions for uplink and downlink traffic. Therefore a new entity in UE may be defined.

4.3.4 RTC-5: Control transport interface

The RTC-5 interface is an interface between the RTC MSH and the RTC AF. It is used to convey configuration information from the RTC AF to the RTC MSH and to request support for a starting/ongoing WebRTC session. The configuration information may consist of static information such as the following:

- Recommendations for media configurations
- Configurations of STUN and TURN server locations
- Configuration about consumption and QoE reporting
- Discovery information for WebRTC signalling and data channel servers and their capabilities

The support functionality includes the following:

- RTC MSH receives the configuration information
- RTC MSH informs the RTC AF about a WebRTC session and its state
- RTC MSH requests QoS allocation for a starting or modified session
- RTC MSH receives notification about changes to the QoS allocation for the ongoing WebRTC session
- RTC MSH receives the updated information about the WebRTC session with the RTC STUN/TURN/Signalling function, e.g. to identify a WebRTC session and associate it with a QoS template

The RTC functionality that offer application functions to the WebRTC application may equally be provided by Application Servers (RTC AS) instead of RTC AF. These then use a dedicated interface RTC-3 to request configurations and network support for the ongoing WebRTC sessions from the RTC AF.

4.3.5 RTC-6: Client API

The RTC MSH is a function in the UE that provides access to RTC support functions to the native WebRTC applications. These functions may be offered on request, i.e., through the RTC-6 interface, or transparently without direct involvement of the application. The RTC MSH may assist indirectly in the ICE negotiation by providing a list of STUN and TURN server candidates that offer RTC functionality. The RTC MSH also collects QoE metric reports and submits consumption reports. It may also offer media configuration recommendations to the application through RTC-6.

4.3.6 RTC-7: Client interface

This is an interface between WebRTC framework and the native WebRTC Application to directly communicate media-specific information.

4.3.7 RTC-8: Application interface

This is a proprietary interface between the application and the application provider, which may be used to exchange configuration information related to the RTC session or the application.

4.3.8 RTC-11: UE configuration interface

The RTC-11 is an interface between the RTC MSH and the WebRTC framework, both in the RTC endpoint, to configure media session handling and/or media access. It may not be exposed as an API to application developers but may be in form of a direct communication. The WebRTC framework hides away all details of the QoS allocation and network support from the application. It autonomously and transparently invokes the functions offered by the RTC MSH to provide support for the RTC session.

4.4 RTC Architecture extension

4.4.1 Introduction

This clause defines an architecture that enables a RTC Application Provider to provision resources in the Edge Data Network (EDN) for an application through the RTC-1 interface.

Media processing in the edge may be achieved in one of two different ways at the application layer:

1. Client-driven management. RTC Applications that are aware of the edge processing can directly request an edge resource and discover the Edge Application Server (EAS) that is best suited to serve the application.

2. Application Function-driven management. The RTC AF automatically allocates edge resources for new streaming sessions on behalf of the application using information in the RTC provisioning session.

An Edge-enabled RTC Client leverages the Edge Computing capabilities as defined in TS 23.558 [7].

4.4.2 Extended RTC architecture for Edge Computing

4.4.2.1 General

The RTC architecture can be extended to add support for media processing in the edge. The extended architecture is an integration of the RTC architecture defined in TS 26.506 with the architecture for enabling Edge Applications defined in TS 23.558 [7] and TS 26.501 [6].

The extended RTC architecture supports both client-driven as well as Application Function-driven management of the edge processing session.

The RTC Application Provider may request the deployment of edge resources as part of the Provisioning Session.

- The RTC Application Provider provisions the edge provisioning through RTC-1, a similar fashion as defined in TS 26.512 clause 7.10, enabling client-driven and/or Application Function driven edge configuration.

- In the client-driven approach, the WebRTC Application becomes aware of the support of edge processing in the network and takes steps, such as using the EDGE-5 APIs, to discover and locate a suitable RTC AS instance in the Edge DN, similar to the process defined in TS 26.501 clause 8.1.

- In the Application Function driven approach, the RTC Application Provider requests RTC AF to deploy edge processing for the media sessions of the corresponding Provisioning Session, similar to the process defined in TS 26.501 clause 8.2. The WebRTC Application may get aware of the deployed EAS through the Application Service Provider through RTC-8 or through the RTC MSH through RTC-5 (and possibly RTC-6). The EAS is provided together such that the associated can be made by UE between two set of data. Additionally, the EAS may also be discovered through other means, such as DNS resolution with support from the DNS server (e.g., EASDF/DNS resolver) as specified in 3GPP TS 23.548 .

When the WebRTC application is a web application, the implementation of the EDGE-5 interface to discover the RTC AS/EAS location by accessing the EEC is difficult as the Web browser providers may not implement interfaces necessary for supporting edge enabled RTC applications/services. Also, in the Application Function-driven approach the Application Client (AC) and EEC are not used to discover the RTC AS/EAS location.

To resolve the above EAS discovery issue in the Application Function-driven approach and when the WebRTC application is a web application, the EAS information can be shared with the RTC MSH by the RTC AF using RTC-5 interface.

NOTE: Other methods that can be used for sharing EAS information (e.g., sharing EAS hostname to the WebRTC application by RTC-8 or by other means and then using DNS resolution) are FFS.

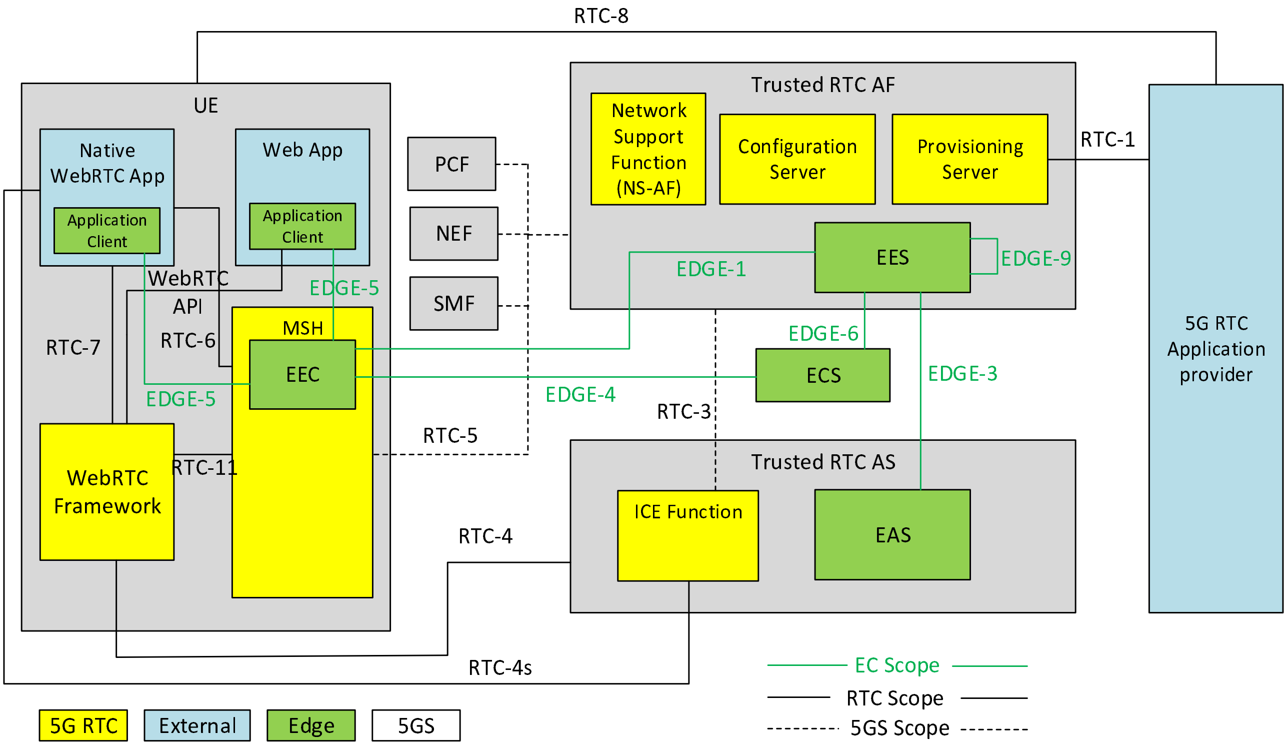


Figure 4.4.2-1: Edge-enabled RTC architecture

NOTE: This architecture diagram is an example for CS-2 scenario.

4.4.2.2 Edge Application Server (EAS)

EAS is the application server resident in the EDN, performing edge-based processing for AR functionalities such as split rendering and spatial computing. The Application Client (AC) connects to the EAS in order to avail the services of the application with the benefits of Edge Computing.

It is possible that the server functions of an application are available only as an EAS.

However, it is also possible that certain server functions are available both at the edge and in the cloud as an EAS and an Application Server resident in the cloud.

The EAS can use the 3GPP Core Network capabilities in the following ways, all of which are optional to support:

a) invoking 3GPP Core Network capabilities via the edge enabler layer through the Edge Enabler Server (EES)

b) invoking 3GPP Core Network function (e.g., PCF) APIs directly, if it is an entity trusted by the 3GPP Core Network; and

c) invoking the 3GPP Core Network capabilities through the capability exposure functions, i.e., SCEF/NEF/SCEF+NEF.

The functions of Edge enabler Client (EEC), Edge Enabler Server (EES), Edge Configuration Server (ECS) are as defined in TS 23.558 [7].

4.4.2.3 Edge Interfaces

Based on the extended architecture, the following interfaces are defined for performing edge-based processing for AR functionalities such as split rendering and spatial computing:

1. A RTC AF that is edge-enabled shall support EES functionality including:
   - EDGE-1 API for supporting registration and provisioning of EEC functions, and discovery by them of EAS instances.
   - EDGE-3 API towards the EAS function of RTC AS instances.
   - EDGE-6 API for registering with an ECS function.
   - EDGE-9 API for media session relocation.
2. A RTC AS that is edge-enabled shall support EAS functionality including the EDGE-3 API for registration with the EES.
3. A RTC MSH that is edge-enabled should support EEC functionality including:
   - Invoking the EES function using the EDGE-1 API.
   - Invoking the ECS function using the EDGE-4 API.
   - EDGE-5 API exposed to the Application Client.
4. A WebRTC Application that is edge-enabled shall support Application Client functionality and should invoke the ECS function using the EDGE-5 API.

5 Procedures for basic RTC architecture

5.1 General

The RTC procedures that are defined in this clause are classified based on the collaboration scenarios that are described in Annex A. Depending on the scenario, only a subset of the functions that are defined in clause 4.2 may be involved.

In general, the RTC call flow may consist of the following procedures.

- Provisioning
- Configuration
- ICE candidates discovery
- Session establishment
- QoS request (either client-driven or WebRTC signalling function/server-driven)
- WebRTC traffic delivery
- QoS updates
- Session termination

5.2 Common Procedure

5.2.1 Provisioning

An application provider may use the RTC-1 interface to provision network assistance and other resources for its RTC sessions.

This procedure is common to the different collaboration scenarios.

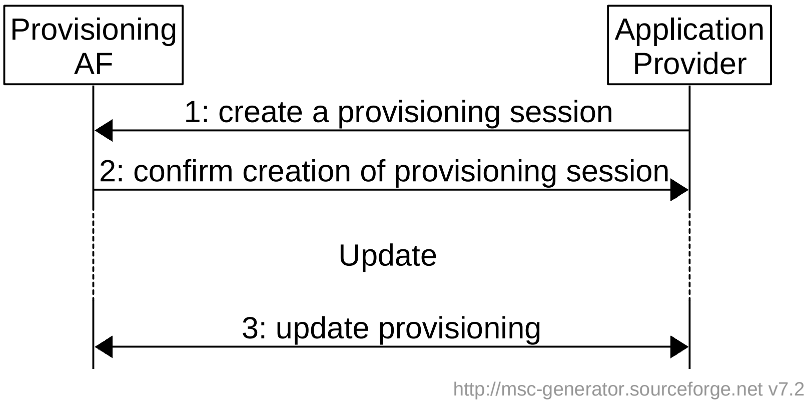

Provisioning AF
Application Provider
1: create a provisioning session
2: confirm creation of provisioning session
Update
3: update provisioning
http://msc-generator.sourceforge.net v7.2

Figure 5.2.1-1: Provisioning procedure

5.2.2 Configuration

The Configuration procedure is used to pre-configure the RTC MSH with information that it makes available to RTC applications through the RTC-6 interface.

This information includes the following:

- The location and capabilities of trusted ICE functions
- The location and capabilities of trusted WebRTC Signaling functions
- The edge configuration as defined in clause 6.

The RTC MSH retrieves the configuration information from the RTC AF.

The configuration procedure is illustrated in Figure 5.2.2-1:

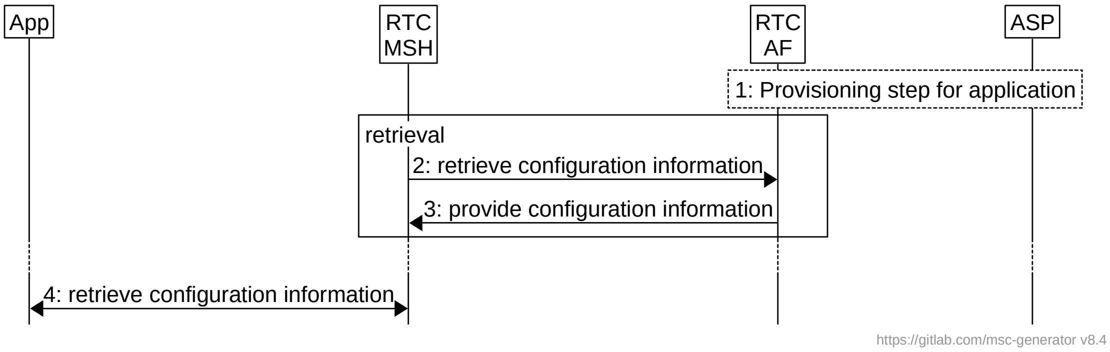

App
RTC MSH
RTC AF
ASP
1: Provisioning step for application
retrieval
2: retrieve configuration information
3: provide configuration information
4: retrieve configuration information
https://gitlab.com/msc-generator v8.4

Figure 5.2.2-1: Configuration procedure

The steps are as follows:

1. An ASP provisions resources for its RTC sessions
2. A single retrieval of the configuration information is done:
    a. The RTC MSH requests the configuration information for RTC sessions. It may provide the application identifier to retrieve configuration information specific to that application.

    b. The RTC AF provides the requested configuration information.

The RTC MSH makes the information available to the Application through the RTC-6 interface.

5.3 Call flow for Over-the-top (OTT) RTC sessions (CS#1)

The RTC session is established between two RTC endpoints using external signalling mechanisms. Each endpoint of the connection that is using the 5G system may benefit from 5G network support for the network path within that 5G network.

The following call flow applies to this scenario.

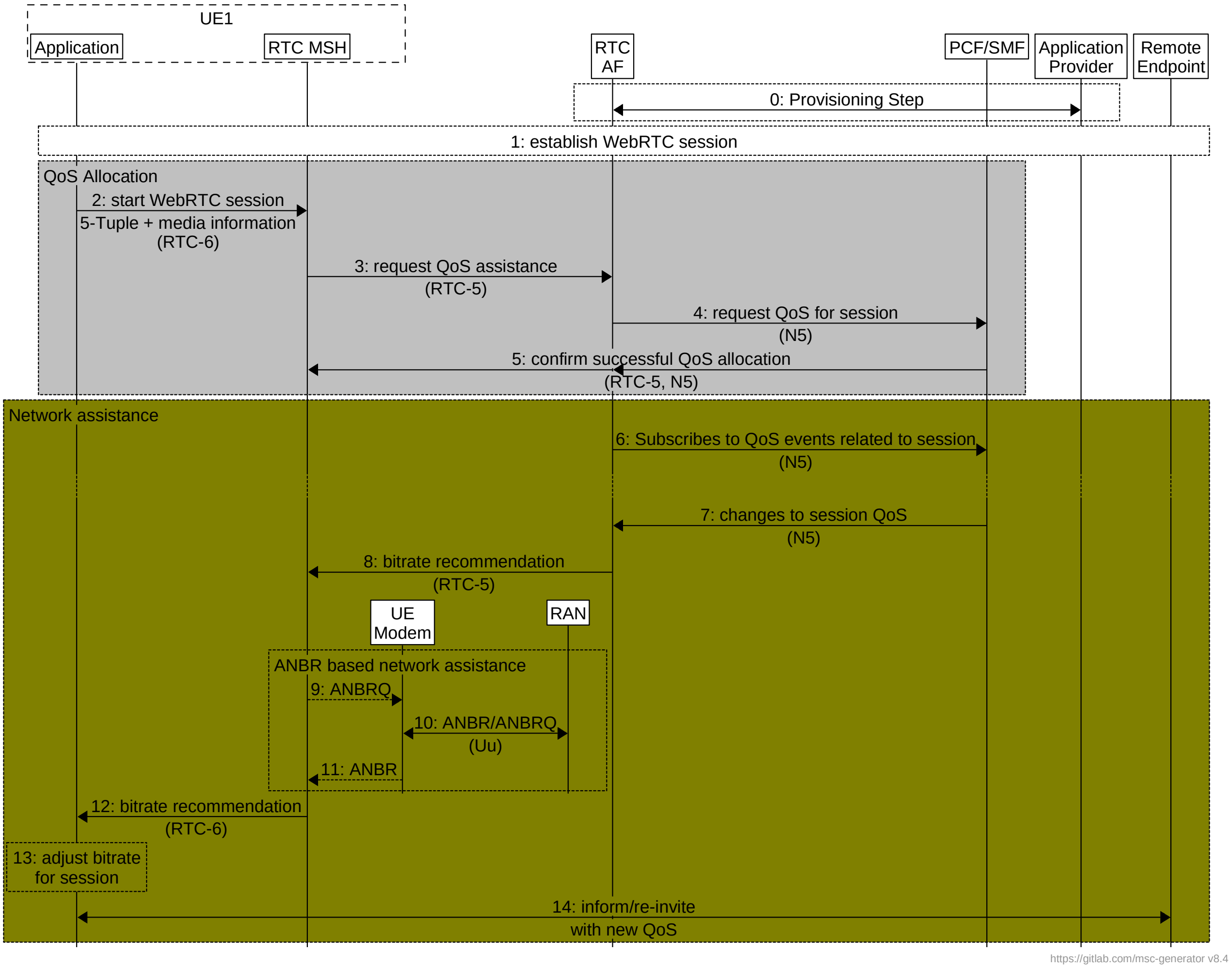

UE1
Application
RTC MSH
RTC AF
PCF/SMF
Application Provider
Remote Endpoint
0: Provisioning Step
1: establish WebRTC session
QoS Allocation
2: start WebRTC session
5-Tuple + media information (RTC-6)
3: request QoS assistance (RTC-5)
4: request QoS for session (N5)
5: confirm successful QoS allocation (RTC-5, N5)
Network assistance
6: Subscribes to QoS events related to session (N5)
7: changes to session QoS (N5)
8: bitrate recommendation (RTC-5)
UE Modem
RAN
ANBR based network assistance
9: ANBRQ
10: ANBR/ANBRQ (Uu)
11: ANBR
12: bitrate recommendation (RTC-6)
13: adjust bitrate for session
14: inform/re-invite with new QoS
https://gitlab.com/msc-generator v8.4

Figure 5.3-1: Call flow for Over-the-top (OTT) RTC sessions (collaboration scenario 1)

The working assumptions are:

- The application on UE1 and the remote endpoint (e.g., UE2 or server for edge computing) use an external WebRTC signalling function to establish the WebRTC session.

0. A provisioning session may have been created by the AP with the MNO.

Network assistance for the RTC session is achieved through the following steps:

1. The application on UE1 uses application-specific signalling functions to establish a WebRTC session with remote endpoint.

2. The application informs the RTC MSH about the new RTC session and shares information about the media streams and their associated 5-Tuples.

3. The RTC MSH requests network assistance for the RTC session and provides the transport and bandwidth information to the Network Support AF.

4. The Network Support AF uses the N5 or N33 interface to request QoS allocation. It may request differential charging based on pre-existing provisioning for these sessions.

5. Confirmation of QoS allocation is notified to the Network Support AF and the RTC MSH.

6. The Network Support AF will also subscribe to events related to the QoS flows of the RTC session with the PCF and SMF.

7. The Network Support AF receives notifications about any changes to the QoS flows of the RTC session from the PCF or the SMF.

8. The Network Support AF sends notifications to the RTC MSH about changes to the session. This information may contain for example be bitrate recommendations.

9. Alternatively, the MSH may interact with the UE Modem to trigger to query the recommended bitrate on the uplink or downlink direction.

10. The UE Modem then sends the ANBRQ (Access Network Bit Rate Query) signalling to the RAN as defined in TS 38.321 [8] for NR access and TS 36.321[9] for LTE access.

11. The RAN, based on the network status, returns the recommended bitrate to the UE modem as requested. The recommended bit rate is in kbps at the physical layer at the time when the decision is made.

NOTE 1: The UE may determine the corresponding IP layer bitrate based on the long-term average of the IP packet sizes, L2 header sizes, and ROHC header sizes, but the translation methodologies and the estimation error levels required to implement accurate media bitrate adaptation have not been specified. The UE may determine the corresponding IP layer bitrate based on the long-term average of the IP packet sizes, L2 header sizes, and ROHC header sizes, but the translation methodologies and the estimation error levels required to implement accurate media bitrate adaptation have not been specified.

NOTE 2: The eNodeB may determine the corresponding IP layer bitrate based on the long-term average of the IP packet sizes, L2 header sizes, and ROHC header sizes, but the translation methodologies and the estimation error levels required to implement accurate media bitrate adaptation have not been specified.

NOTE 3: The recommended/queried bitrate as signalled over the LTE and NR access is defined to be in kbps at the physical layer. The uplink/downlink bitrate at the physical layer is $r_{UL/DL} = \frac{\sum_k L_k}{T}$, where $L_k$is the bit-length of the k-th successfully transmitted/received TB by the UE within the window *T*. In TS 36.321[9] and 38.321[8], a window length of 2000 ms is applied.

12. The RTC MSH forwards the bitrate recommendation to the RTC application.

13. The application may act on the bitrate recommendation, e.g. by reducing the uplink media bitrate.

14. The application may request remote endpoint to adjust the bitrate of the downlink media.

5.4 Call flow for Network-supported RTC sessions (CS#2)

The MNO offers access to trusted ICE functionality to UEs that wish to participate in RTC sessions. The session establishment takes into account the configured trusted ICE functions.

The call flow is as follows.

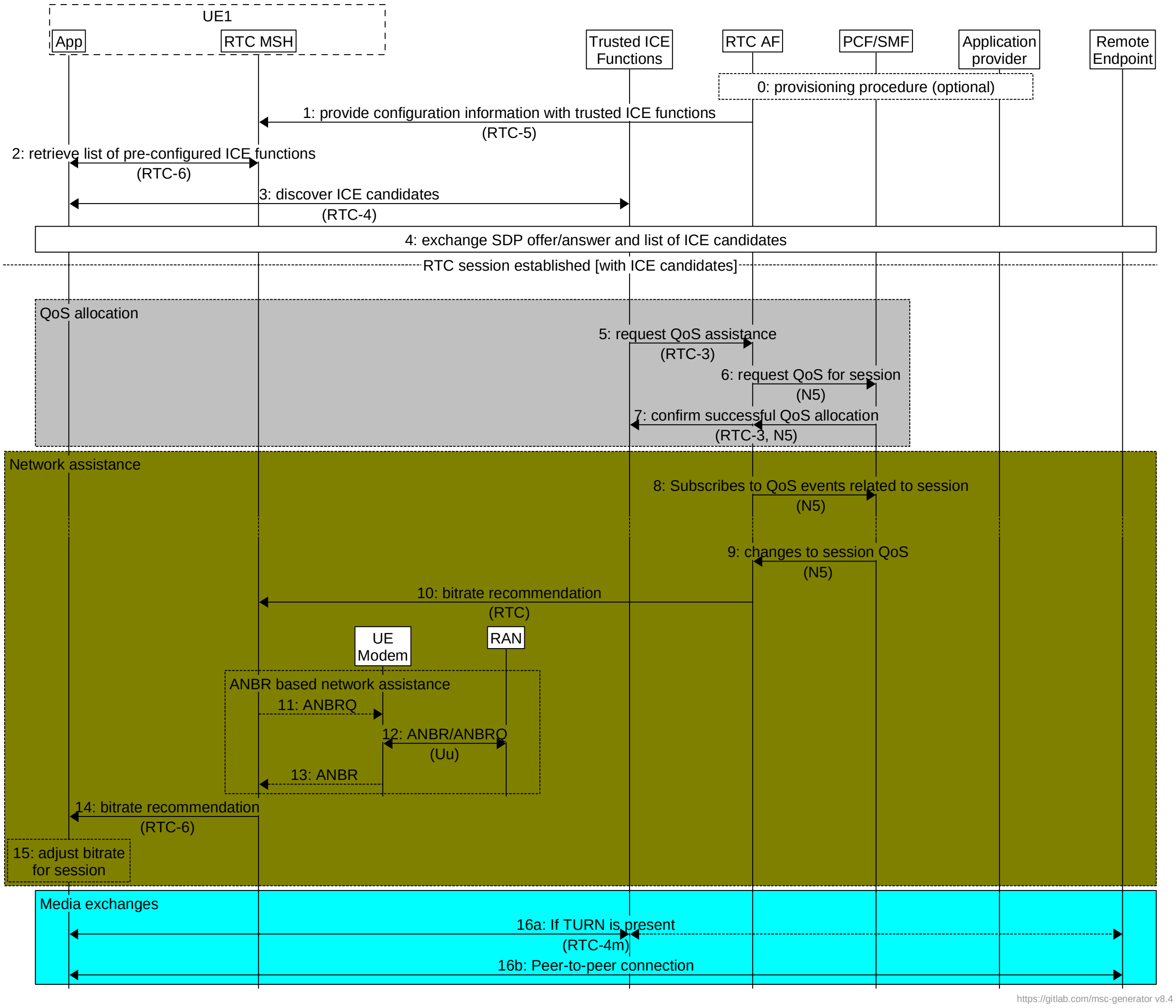

UE1
App
RTC MSH
Trusted ICE Functions
RTC AF
PCF/SMF
Application provider
Remote Endpoint
0: provisioning procedure (optional)
1: provide configuration information with trusted ICE functions
(RTC-5)
2: retrieve list of pre-configured ICE functions
(RTC-6)
3: discover ICE candidates
(RTC-4)
4: exchange SDP offer/answer and list of ICE candidates
RTC session established [with ICE candidates]
QoS allocation
5: request QoS assistance
(RTC-3)
6: request QoS for session
(N5)
7: confirm successful QoS allocation
(RTC-3, N5)
Network assistance
8: Subscribes to QoS events related to session
(N5)
9: changes to session QoS
(N5)
10: bitrate recommendation
(RTC)
UE Modem
RAN
ANBR based network assistance
11: ANBRQ
12: ANBR/ANBRQ
(Uu)
13: ANBR
14: bitrate recommendation
(RTC-6)
15: adjust bitrate for session
Media exchanges
16a: If TURN is present
(RTC-4m)
16b: Peer-to-peer connection
https://gitlab.com/msc-generator v8.4

Figure 5.4-1: Call flow for Network-supported RTC sessions (collaboration scenario 2)

The working assumptions are:

- The application on UE1 and remote endpoint use an external WebRTC signalling function to establish the WebRTC session.

0. A provisioning session may have been created by the AP with the MNO.

Call flow using network-supported RTC session is achieved through the following steps:

1. The RTC AF uses the RTC-5 interface to provide the RTC MSH with a list of trusted STUN/TURN servers that the UE may use for establishing RTC sessions.

2. The application queries the RTC MSH for the list of trusted ICE servers.

3. The UE discovers and tests the ICE candidates to validate that they are suitable for the connection.

4. The application on UE1 and the remote endpoint use an external RTC signalling function to exchange information about ICE candidates and to exchange the SDP offer/answer.

Then, the WebRTC session is established using the most suitable ICE candidate.

5. The STUN or TURN server in ICE function, upon reception of the allocation request by the application (or WebRTC framework) may extract the 5-Tuple information for each of the media sessions and convey the information to the Network Support AF in RTC AF for requesting QoS assistance.

6. The Network Support AF uses the N5 interface to request QoS allocation. It may request differential charging based on pre-existing provisioning for these sessions.

7. Confirmation of QoS allocation is notified to the Network Support AF and the RTC MSH.

8. The Network Support AF will also subscribe to events related to the QoS flows of the WebRTC session with the PCF and SMF.

9. The Network Support AF receives notifications about any changes to the QoS flows of the WebRTC session from the PCF or the SMF. Then, the Network Support AF sends notifications to the ICE function (STUN/TURN server).

10. The STUN/TURN server may forward the bitrate recommendation to the RTC MSH, if the allocation session is still active.

11. Alternatively, the MSH may interact with the UE Modem to trigger to query the recommended bitrate on the uplink or downlink direction.

12. The UE Modem then sends the ANBRQ (Access Network Bit Rate Query) signalling to the RAN as defined in TS 38.321 [8] for NR access and TS 36.321 [9] for LTE access.

13. The RAN, based on the network status, returns the recommended bitrate to the UE modem as requested. The recommended bit rate is in kbps at the physical layer at the time when the decision is made.

NOTE 1: The UE may determine the corresponding IP layer bitrate based on the long-term average of the IP packet sizes, L2 header sizes, and ROHC header sizes, but the translation methodologies and the estimation error levels required to implement accurate media bitrate adaptation have not been specified. The UE may determine the corresponding IP layer bitrate based on the long-term average of the IP packet sizes, L2 header sizes, and ROHC header sizes, but the translation methodologies and the estimation error levels required to implement accurate media bitrate adaptation have not been specified.

NOTE 2: The eNodeB may determine the corresponding IP layer bitrate based on the long-term average of the IP packet sizes, L2 header sizes, and ROHC header sizes, but the translation methodologies and the estimation error levels required to implement accurate media bitrate adaptation have not been specified.

NOTE 3: The recommended/queried bitrate as signalled over the LTE and NR access is defined to be in kbps at the physical layer. The uplink/downlink bitrate at the physical layer is $r_{UL/DL} = \frac{\sum_k L_k}{T}$, where $L_k$is the bit-length of the k-th successfully transmitted/received TB by the UE within the window *T*. In TS 36.321[9] and 38.321[8], a window length of 2000 ms is applied.

14. The application may act on the bitrate recommendation, e.g. by reducing the uplink media bitrate.

15. Media traffic is delivered to remote endpoint. If TURN server is present in the configuration, RTC-4m interface is involved.

5.5 Call flow for MNO-Facilitated RTC sessions (CS#3)

In collaboration scenario 3, MNO hosts the WebRTC sessions by providing a trusted WebRTC signalling function in the RTC AS. In addition, a trusted media server is also present in RTC AS to support SFU and MCU functionality. Note that, when the WebRTC application is a web-based application, the RTC MSH function is not supported.

The call flows for this scenario when RTC MSH is involved are as shown in Figure 5.5.1.

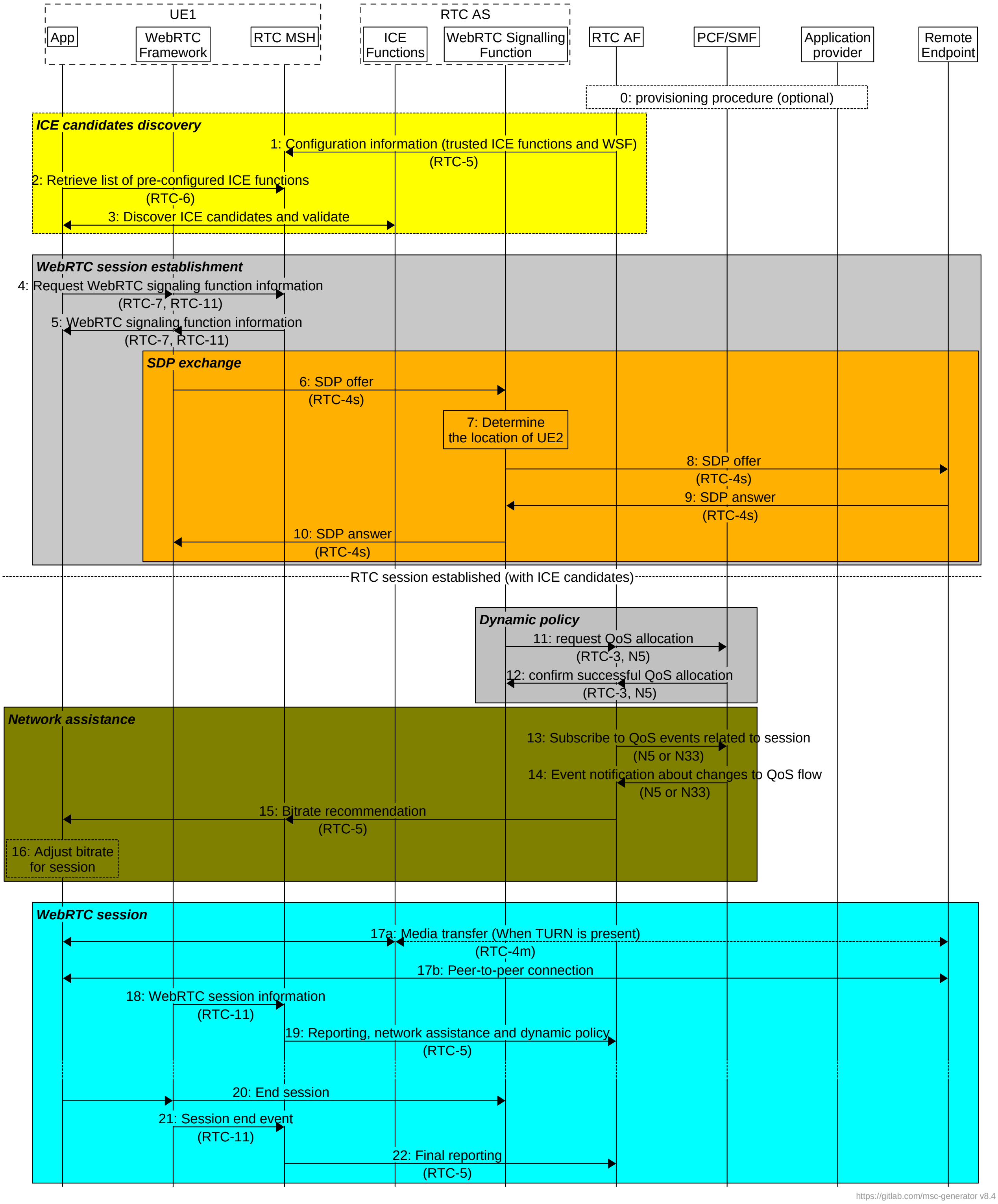

UE1
App
WebRTC Framework
RTC MSH
RTC AS
ICE Functions
WebRTC Signalling Function
RTC AF
PCF/SMF
Application provider
Remote Endpoint
0: provisioning procedure (optional)
ICE candidates discovery
1: Configuration information (trusted ICE functions and WSF)
(RTC-5)
2: Retrieve list of pre-configured ICE functions
(RTC-6)
3: Discover ICE candidates and validate
WebRTC session establishment
4: Request WebRTC signaling function information
(RTC-7, RTC-11)
5: WebRTC signaling function information
(RTC-7, RTC-11)
SDP exchange
6: SDP offer
(RTC-4s)
7: Determine the location of UE2
8: SDP offer
(RTC-4s)
9: SDP answer
(RTC-4s)
10: SDP answer
(RTC-4s)
RTC session established (with ICE candidates)
Dynamic policy
11: request QoS allocation
(RTC-3, N5)
12: confirm successful QoS allocation
(RTC-3, N5)
Network assistance
13: Subscribe to QoS events related to session
(N5 or N33)
14: Event notification about changes to QoS flow
(N5 or N33)
15: Bitrate recommendation
(RTC-5)
16: Adjust bitrate for session
WebRTC session
17a: Media transfer (When TURN is present)
(RTC-4m)
17b: Peer-to-peer connection
18: WebRTC session information
(RTC-11)
19: Reporting, network assistance and dynamic policy
(RTC-5)
20: End session
21: Session end event
(RTC-11)
22: Final reporting
(RTC-5)
https://gitlab.com/msc-generator v8.4

Figure 5.5.1: Call flows for MNO facilitated RTC sessions (collaboration scenario 3)

The RTC Application Provider may create a ***Provisioning Session*** with the RTC AF and starts provisioning the usage of the RTC Streaming session between two endpoints. During the establishment phase, the used features such as content consumption measurement, logging, collection and reporting; QoE metrics measurement, logging, collection and reporting; dynamic policy; network assistance; are negotiated and detailed configurations are exchanged.

The RTC Application Provider ***Provisioning Session*** phase is optionally performed prior to the establishment of any related WebRTC sessions by the RTC Application Provider. Detailed procedure is addressed in clause 5.2.1.

The ***ICE candidate discovery*** phase is performed with the following steps in an MNO-facilitated RTC system:

1. Configuration information: The RTC AF uses the RTC-5 interface to provide the RTC MSH with a list of trusted STUN/TURN servers, trusted WebRTC signalling function and data channel servers and their capabilities. The UE may use this configuration information for establishing RTC sessions.

2. ICE Servers request: The application queries the RTC MSH for the list of trusted ICE servers.

3. ICE candidate validation: The UE discovers and tests the ICE candidates to validate that they are suitable for the connection.

The ***WebRTC session establishment*** phase is performed with the following steps in an MNO-facilitated RTC system:

4. Query configuration information: The WebRTC framework queries the RTC MSH for the WebRTC signalling function information. In some cases where the signalling is not handled by WebRTC framework, the native WebRTC application queries the RTC MSH for the WebRTC Signalling server information.

5. Configuration information: RTC MSH sends the WebRTC signalling function and data channel servers and their capabilities information to WebRTC framework or in some cases with native WebRTC application.

In ***SDP exchange*** phase, two or more WebRTC endpoints exchange signalling information related to the WebRTC session such as ICE candidates, SDP offer/answer using the trusted WebRTC signalling function.

NOTE: Figure 5.5.1 illustrates that SDP offer is generated by the WebRTC Framework or native WebRTC Application. However, in SFU/MCU mode, SDP offer is generated by Media Function in RTC AS.

6. SDP offer: The WebRTC Framework or native WebRTC Application creates a request with SDP offer which includes the ICE candidates and sends it to the WebRTC signalling function.

7. Determine remote endpoint location: The WebRTC signalling function uses the registration information to locate the remote endpoint

8. SDP offer: The WebRTC signalling function forwards the request to remote endpoint.

9. SDP answer: Upon accepting the offer, remote endpoint responds to signalling function with SDP answer.

10. SDP answer: WebRTC signalling function sends the SDP answer to the UE1.

With this, a WebRTC session is established between RTC endpoints using the most suitable ICE candidate and the WebRTC signalling function.

The ***Dynamic policy*** phase is then performed to allocate QoS for the media streams of the RTC session with the following steps:

11. QoS request: The WebRTC signalling function sends a request to RTC AF for the allocation of QoS for the session. The RTC AF sends a request to the PCF to allocate QoS for the media streams of the RTC session

12. Confirmation: PCF or SMF confirms the successful allocation of network support or QoS allocation.

If the Network support function feature is supported in the RTC AF, then the Network Support Function AF (NS-AF) offers the bitrate recommendation request API based on existing policy templates, through the usage of either the *Npcf_PolicyAuthorization* API over N5 interface, or the *Nnef_AFSessionWithQoS* over N33 interface to the PCF. If no corresponding AF application session context already exists, the NS-AF uses the *Npcf_PolicyAuthorization_Create* method over N5 interface with the appropriate service information to create and provision an application session context. The ***Network assistance*** phase is performed with the following steps in an MNO-facilitated RTC system.

13. Subscribe to QoS events: The NS-AF also subscribes to events related to the QoS flows of the WebRTC session with the PCF and SMF.

14. QoS events: The NS-AF receives notifications about any changes to the QoS flows of the WebRTC session from the PCF or the SMF.

15. QoS notifications/ Bitrate recommendations: The NS-AF may send notifications to the RTC MSH about the changes in QoS flow. When the allocated session is active, the RTC MSH forwards the bitrate recommendation to the application.

16. Adjust media bitrate: The WebRTC application may act on adjusting the bitrate recommendation, e.g., by reducing the uplink media bitrate.

After successful creation of a WebRTC session and the bitrate negotiations, the actual ***WebRTC session*** over 5G may start:

17. Media transfer:
    a) The WebRTC Application may connect to the selected TURN server and/or Media Function in the RTC AS through the RTC-4m interface and real-time communication starts, and the media is delivered to the remote endpoint.

    b) In some cases, a peer-to-peer connection is also possible and the media is delivered directly to the remote endpoint.

18. Method calls and notifications: Supporting information about the WebRTC session is passed from the WebRTC framework to the RTC MSH.

19. Reporting, network assistance, and dynamic policy: The RTC MSH exchanges supporting information about the WebRTC session with the RTC AF.

20. End session: The WebRTC Application informs the WebRTC framework that the RTC session has ended. It is also sent to the WebRTC Signalling Function to terminate the session.

21. Session ending event: The WebRTC framework informs the RTC MSH about the end of the RTC session.

22. Final reporting: The RTC MSH performs any final reporting to the RTC AF.

6 Procedures for Edge Processing

6.1 Client-driven Management of RTC Edge Processing

The detailed call flow for client-driven management of edge processing session is shown in Figure 6.1-1.

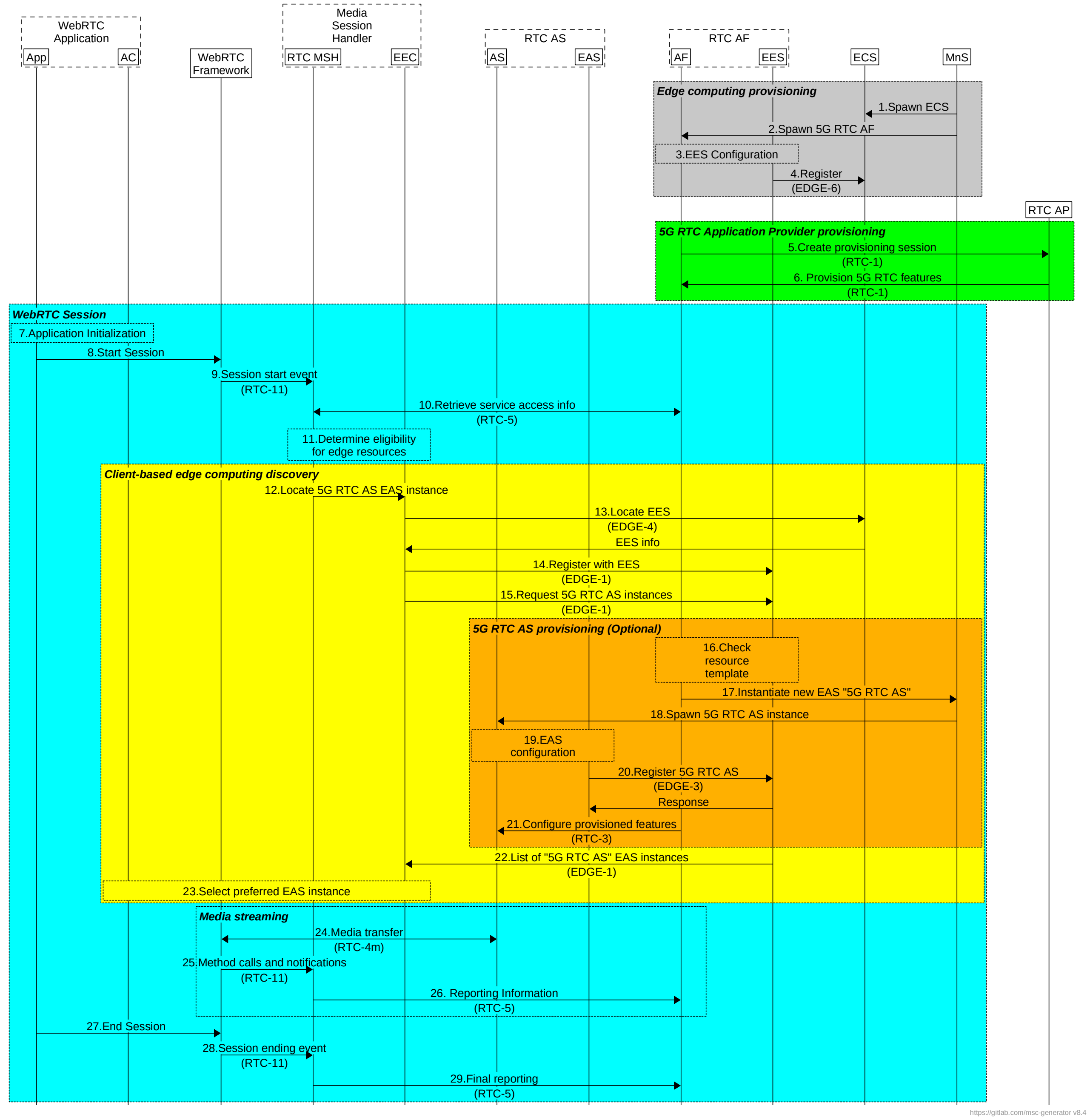

WebRTC Application
App
AC
WebRTC Framework
Media Session Handler
RTC MSH
EEC
RTC AS
AS
EAS
RTC AF
AF
EES
ECS
MnS
RTC AP
Edge computing provisioning
1.Spawn ECS
2.Spawn 5G RTC AF
3.EES Configuration
4.Register (EDGE-6)
5G RTC Application Provider provisioning
5.Create provisioning session (RTC-1)
6. Provision 5G RTC features (RTC-1)
WebRTC Session
7.Application Initialization
8.Start Session
9.Session start event (RTC-11)
10.Retrieve service access info (RTC-5)
11.Determine eligibility for edge resources
Client-based edge computing discovery
12.Locate 5G RTC AS EAS instance
13.Locate EES (EDGE-4)
EES info
14.Register with EES (EDGE-1)
15.Request 5G RTC AS instances (EDGE-1)
5G RTC AS provisioning (Optional)
16.Check resource template
17.Instantiate new EAS "5G RTC AS"
18.Spawn 5G RTC AS instance
19.EAS configuration
20.Register 5G RTC AS (EDGE-3)
Response
21.Configure provisioned features (RTC-3)
22.List of "5G RTC AS" EAS instances (EDGE-1)
23.Select preferred EAS instance
Media streaming
24.Media transfer (RTC-4m)
25.Method calls and notifications (RTC-11)
26. Reporting Information (RTC-5)
27.End Session
28.Session ending event (RTC-11)
29.Final reporting (RTC-5)
https://gitlab.com/msc-generator v8.4

Figure 6.1-1. Client-driven management of RTC edge processing

The ***Edge Computing Provisioning*** phase is a provisioning phase, that may be repeated several times (e.g., to extend edge processing coverage to new geographical areas or to increase the capacity of an already provisioned area). All steps in this phase are optional and performed on need basis. The steps are:

1. Spawn ECS: In this step, a new ECS instance is instantiated to manage new or increased demand for edge processing.
2. Spawn RTC AF: In this step, a new RTC AF that is edge-enabled is instantiated to handle new or increased demand for WebRTC sessions with edge processing.
3. EES Configuration: The EES is configured for a specific Edge Data Network (EDN).
4. EES Registration with ECS: The EES registers with the ECS that is in authority over the target EDN.

The ***RTC Application Provider Provisioning*** phase is performed prior to the establishment of any related WebRTC sessions by the RTC Application Provider. Subsequent updates to the provisioning session are possible.

5. Create Provisioning Session: In this step, the RTC Application Provider creates a new provisioning session.

6. Provision RTC features: In this step, the RTC Application Provider may create different configurations such as QoS support, charging, collection of consumption, offering STUN/TURN servers, WebRTC signalling function, Edge Processing, etc.

The WebRTC Application initiates a new RTC session:

7. Application Initialization: The user launches the WebRTC Application. The WebRTC application performs any required initialization steps.

8. Start session: The WebRTC Application invokes the WebRTC framework with appropriate real-time streaming access parameters.

9. Session starting event: The application informs the RTC MSH about the start of a new WebRTC session over 5G.

10. Retrieve service access information: The RTC MSH retrieves Service Access Information from the RTC AF appropriate to the WebRTC session.

11. Determine eligibility for requesting edge resources: Using information from the Service Access Information, the RTC MSH determines whether the WebRTC session is eligible for requesting edge resources.

If the eligibility criteria are met in the previous step, the UE discovers an EAS instance offering RTC AS functionality in the ***Client-based Edge Computing Discovery*** phase:

12. Locate EAS instances: The RTC MSH asks the EEC to discover the location of one or more suitable EAS instances offering the RTC AS capability that can serve the application.

13. Locate local EES: The EEC queries the ECS for a suitable EES (EDGE-4 API).

14. Register with EES: The EEC registers with the selected EES (EDGE-1 API).

15. Request list of “RTC AS” EAS instances: The EEC queries the EES for one or more EAS instances offering the “RTC AS” capability that can serve the session, using EAS discovery filters (see Table 8.5.3.2-2 in [2]) provided by the Application Client, e.g. “RTC AS” for EAS type, appropriate values for service feature(s), and other EAS characteristics.

The optional sub-flow ***RTC AS Provisioning*** is for provisioning an additional RTC AS instance if a suitable EAS instance offering the "RTC AS" capability cannot be located. The steps are:

16. Check resource template: The RTC AF checks the provisioned edge processing resource template for the related application to determine the requirements of the application.

17. Instantiate new EAS/RTC AS: The RTC AF requests the MnS to instantiate a new RTC AS EAS instance with the specified requirements and considering parameters provided in the query by the EEC.

18. Spawn RTC AS instance: The MnS creates a new instance of the EAS offering RTC AS capability with the requested placement and resources.

19. EAS configuration: The newly instantiated RTC AS EAS instance is configured.

20. Register EAS with EES: The newly instantiated EAS instance registers itself with the triggering EES.

21. Configure provisioned features: This may include configuring and launching the server-side application in the RTC AS.

Completion of UE Edge Computing Discovery phase:

22. List of suitable “-RTC AS” EAS instances: The EES/RTC AF responds to the EEC with a list of “RTC AS” EAS instances and their characteristics in an EAS discovery response (see Table 8.5.3.3-1 in [16]).

23. Select preferred “RTC AS” EAS instance: The AC and/or EEC select(s) a “RTC AS” EAS instance from the provided list, based on the AC’s desired criteria.

After successful discovery of a “RTC AS” EAS instance, the actual ***WebRTC session*** over 5G may start:

24. Media transfer: The WebRTC Application connects to the selected EAS "RTC AS" and the real-time streaming starts.

25. Method calls and notifications: Supporting information about the WebRTC session is passed from the WebRTC framework to the RTC MSH.

26. Reporting, network assistance, and dynamic policy: The RTC MSH exchanges supporting information about the WebRTC session with the RTC AF.

27. End session: The WebRTC Application informs the WebRTC framework that the RTC session has ended.

28. Session ending event: The WebRTC framework informs the RTC MSH about the end of the RTC session.

29. Final reporting: The RTC MSH performs any final reporting to the RTC AF.

6.2 AF-driven Management of RTC Edge Processing

The detailed call flow for AF-driven management of edge processing session by using the RTC MSH is shown in Figure 6.2-1.

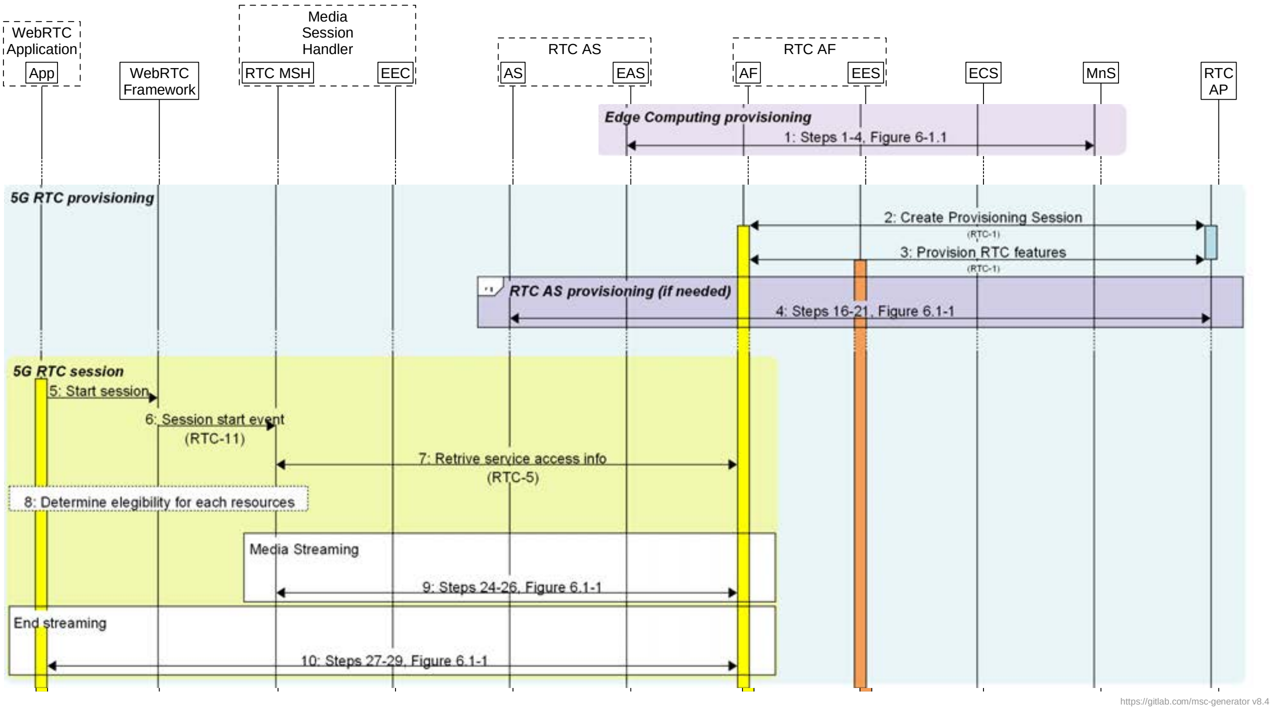

WebRTC Application
App
WebRTC Framework
Media Session Handler
RTC MSH
EEC
RTC AS
AS
EAS
RTC AF
AF
EES
ECS
MnS
RTC AP
Edge Computing provisioning
1: Steps 1-4, Figure 6-1.1
5G RTC provisioning
2: Create Provisioning Session
(RTC-1)
3: Provision RTC features
(RTC-1)
RTC AS provisioning (if needed)
4: Steps 16-21, Figure 6.1-1
5G RTC session
5: Start session
6: Session start event
(RTC-11)
7: Retrive service access info
(RTC-5)
8: Determine elegibility for each resources
Media Streaming
9: Steps 24-26, Figure 6.1-1
End streaming
10: Steps 27-29, Figure 6.1-1
https://gitlab.com/msc-generator v8.4

Figure 6.2-1. AF-driven management of RTC edge processing

The steps are:

1. Steps 1-4 as described in TS 26.501 clause 8.1.

2. Create Provisioning Session: In this step, the RTC Application Provider creates a new provisioning session.

3. Provision RTC features: In this step, the RTC Application Provider may create different configurations such as QoS support, charging, collection of consumption, offering STUN/TURN servers, WebRTC signalling function, edge processing, etc.

4. RTC AS provisioning if need, as described in Figure 6.1-1, steps 16-21.

The WebRTC Application initiates a new RTC session:

5. Start session: The WebRTC Application invokes the WebRTC framework with appropriate real-time streaming access parameters.

6. Session starting event: The application informs the RTC MSH about the start of a new WebRTC session over 5G.

7. Retrieve Service Access Information: The RTC MSH retrieves Service Access Information from the RTC AF appropriate to the WebRTC session.

8. Determine eligibility for requesting edge resources: Using information from the Service Access Information, the RTC MSH determines whether the WebRTC session is eligible for requesting edge resources.

9. Start the media streaming as defined in Figure 6.1-1, steps 24-26.

10. Continue the final steps as defined in Figure 6.1-1, steps 27-29.

Annex A (normative): Architecture variants for collaboration scenarios

A.1 General

This clause addresses the derivative architecture for each of the collaboration scenarios. The four collaboration scenarios are summarized below and further details is specified in Annex A.

The four collaboration scenarios are specified based on the location of required functional entities in trusted domain as defined as follows.

- 5G support for OTT WebRTC: in this scenario the WebRTC session runs completely over the top. However, the MNO may offer support in form of QoS allocation, bitrate recommendations, and QoE report collection based on request by the UE.
- MNO-provided trusted WebRTC functions: in this scenario the MNO offers trusted support functions such as ICE servers to the WebRTC application on the UE.
- MNO-facilitated WebRTC services: the MNO hosts and facilitates WebRTC sessions by providing a trusted WebRTC signalling function, which may also offer 5G network assistance.
- Inter-operable WebRTC services: collaboration scenario 3 is extended with functions to support MNO to MNO inter-operability.

NOTE: Collaboration scenario 4 is in the scope of this specification. Some of its details, which are not specified in the current version of the document, will be specified, after the relevant works are finished.

The list of key functional entities in trusted domain differs from collaboration scenarios as described in Table A.1-1.

Table A.1-1: Mapping of key functions to each collaboration scenarios

Functions	Collaboration scenario 1	Collaboration scenario 2	Collaboration scenario 3	Collaboration scenario 4
Provisioning function	Optional	Optional	Optional	Optional
Configuration function	Optional	Required	Optional (maybe fulfilled by WebRTC signalling function)	Optional (maybe fulfilled by WebRTC signalling function)
RTC MSH	Required	Optional	Optional	Optional
Network support function	Required	Required	Optional (maybe fulfilled by WebRTC signalling function)	Optional (maybe fulfilled by WebRTC signalling function)
Trusted ICE function	N/A	Required	Optional	Optional
Trusted WebRTC signalling function	N/A	N/A	Required	Required
Trusted media function	N/A	Optional	Optional	Optional
NOTE: The collaboration scenario 3 may further split depending on the role of MNO, as addressed in TR 26.930.				

A.2 Collaboration scenario 1:

Figure A.2-1 shows the architecture variant for the collaboration scenario 1 when the WebRTC session is completely running over the top. For this case, many of WebRTC-related entities are not the scope of this specification. However, Network Support Function is present in the trusted domain to support QoS allocation, bitrate recommendations, and QoE report collection.

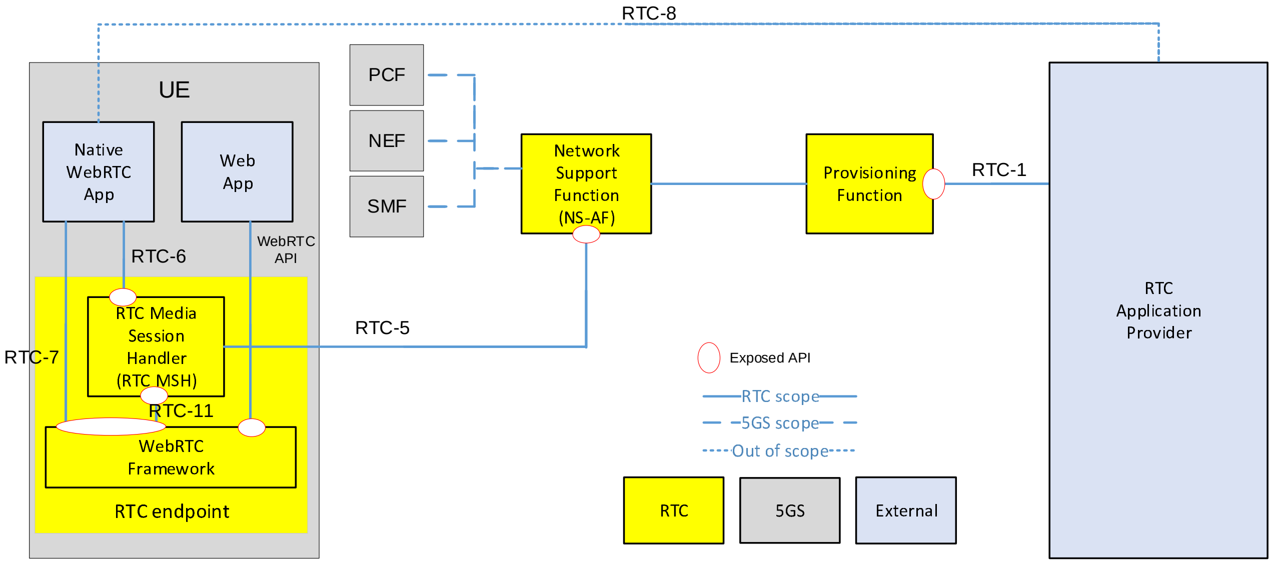


Figure A.2-1: Derivative RTC architecture for collaboration scenario 1

A.3 Collaboration scenario 2:

Figure A.3-1 shows the architecture variant for the collaboration scenario 2 when MNO provides the trusted WebRTC functions such as ICE function. It also contains the configuration function to support the network-assisted WebRTC sessions over 5G system.

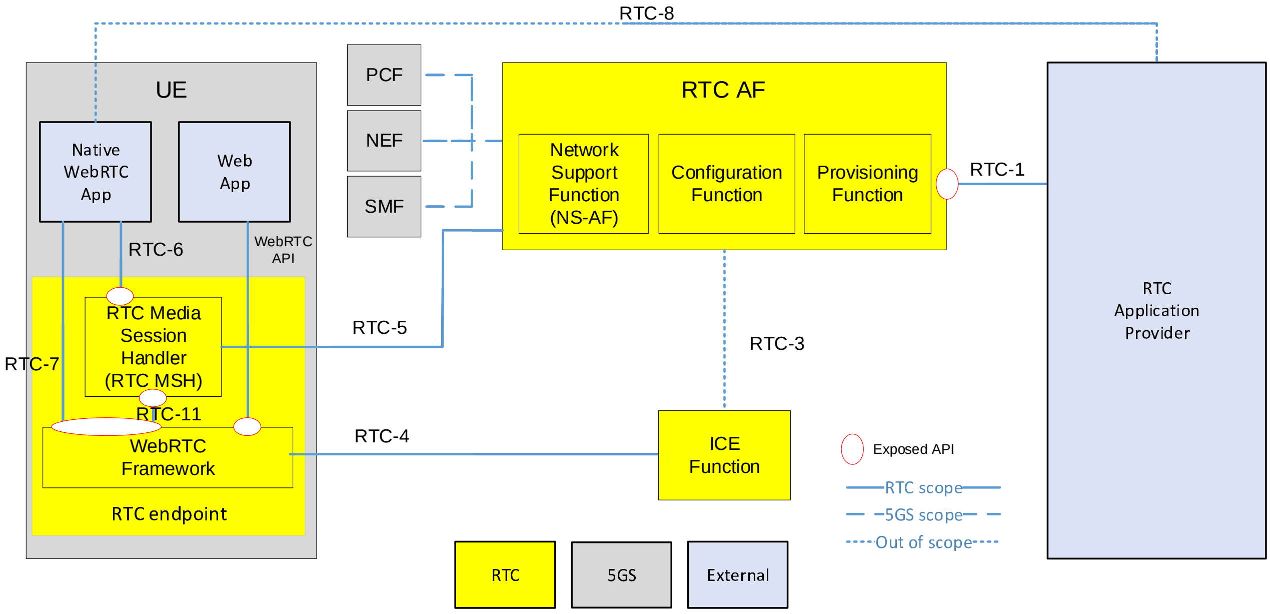


Figure A.3-1: Derivative RTC architecture for collaboration scenario 2

NOTE: RTC-4m interface is present only when the ICE function contains the TURN server in this scenario.

A.4 Collaboration scenario 3:

Figure A.4-1 shows the architecture variant for the collaboration scenario 3 when MNO hosts the WebRTC sessions by providing the trusted WebRTC signalling server in RTC AS. In addition, trusted media server is present in RTC AS to support SFU and MCU functionality.

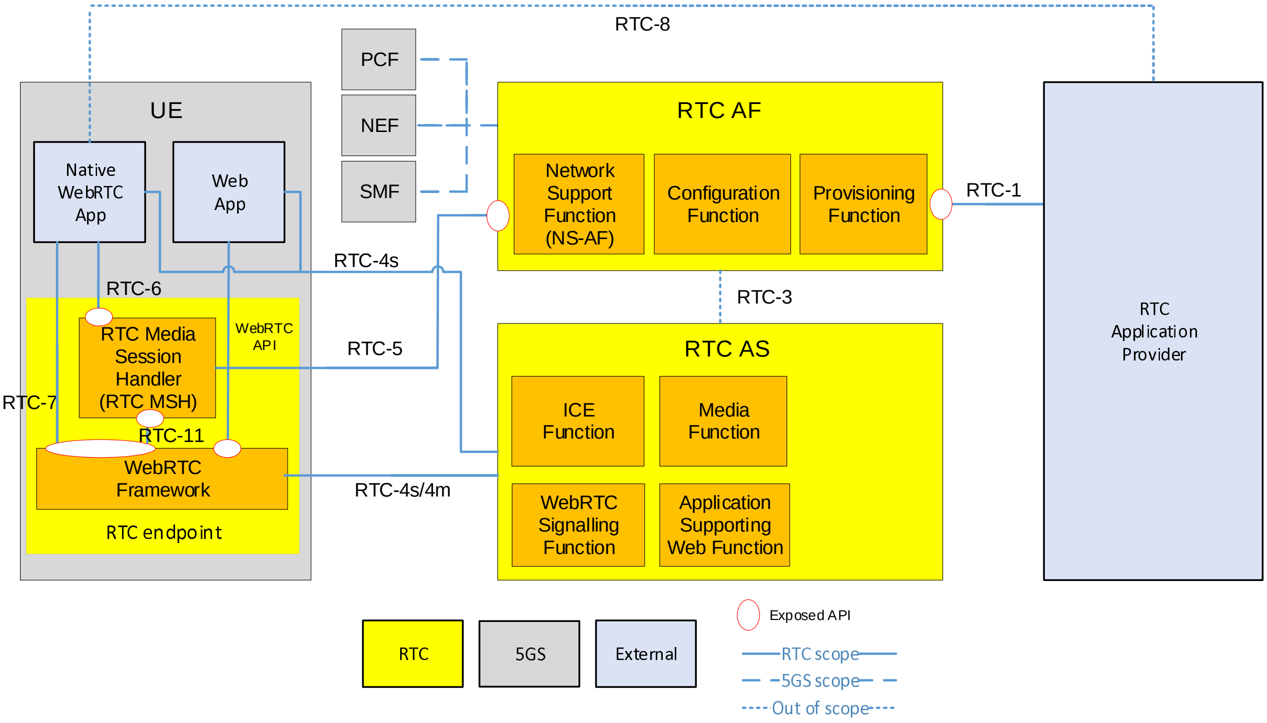


Figure A.4-1: Derivative RTC architecture for collaboration scenario 3

A.5 Collaboration scenario 4:

NOTE: This scenario is extended from collaboration scenario 3 by supporting interoperability between multiple MNOs. The details are FFS.

Annex B (informative):
Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2022-08	SA4#120					Initial draft	0.1.0
2022-11	SA4#121	S4-221543				SA4#121 Agreements: S4-221344, S4-221542, S4-221544, S4-221545, S4-221510, S4-221509, S4-221371, S4-221508	0.2.0
2022-11	SA4#121	S4-221610				Minor update in Scope: word “generic” removed	0.2.1
2022-12						Created by MCC to be presented to TSG for information	1.0.0
2023-02	SA4#122	S4-230343				SA4#122 Agreements: S4-230214, S4-230299, S4-230318, S4-230371	1.1.0
2023-04	SA4#123	S4-230661				SA4#123e Agreements: S4-230488, S4-230499, S4-230657, S4-230709, S4-230710	1.2.0
2023-05	SA4#124	S4-230838				SA4#124 Agreements:S4-231047, S4-231036, S4-230995, S4-230997, S4-230993, S4-231059	1.3.0
2023-06	SA#100	SP-230536				Presented for approval	2.0.0
2023-06	SA#100					TS approved, v 18.0.0 created by MCC	18.0.0
2023-12	SA#102	SP-231373	0003	1	F	New reference point RTC-11 in UE	18.1.0
2024-03	SA#103	SP-240044	0001	3	F	RTC Functions are general Media Functions	18.2.0

History

Document history		
V18.2.0	May 2024	Publication