ETSI TS 126 506 V18.4.0 (2024-10)

TECHNICAL SPECIFICATION

5G;
5G Real-time Media Communication Architecture (Stage 2)
(3GPP TS 26.506 version 18.4.0 Release 18)

Reference

RTS/TSGS-0426506vi40

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

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)".

[12] 3GPP TS 23.548: "5G System Enhancements for Edge Computing; Stage 2".

[13] IETF RFC 8825: “Overview: Real-Time Protocols for Browser-Based Applications”.

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].

RTC Application: A Native WebRTC Application or a Web App that is compliant with the profile of a WebRTC-based application defined in the present document.

RTC endpoint: An entity that is capable of participating in an RTC session and exchanging real-time media and data by incorporating an instance of the WebRTC Framework.

NOTE: A UE incorporating an RTC Client (including an RTC Access Function) as well as an RTC Application is an RTC endpoint. An RTC AS is an RTC endpoint by virtue of containing a Media Function and a WebRTC Signalling Function.

RTC Client: UE function comprising an RTC Access Function and an RTC Media Session Handler which interacts with functions in the network and UE applications.

RTC Access Function: A set of functions including an instance of the WebRTC Framework. The RTC Access Function exchanges real-time media with one or more RTC endpoints via reference point RTC-4m or RTC-12, and the RTC Access Function exchanges signalling messages with WebRTC Signalling Function via reference point RTC-4s. Also, the RTC Access Function exposes client APIs defined in the present document to the RTC Application at reference point RTC-7 and to the RTC Media Session Handler at reference point RTC-11.

WebRTC Framework: A well-defined subset of the WebRTC protocol stack for data transport and data framing that supports real-time media communication between an RTC endpoint and its peer(s) within the scope of an RTC session.

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
RTT	Round-Trip Time
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

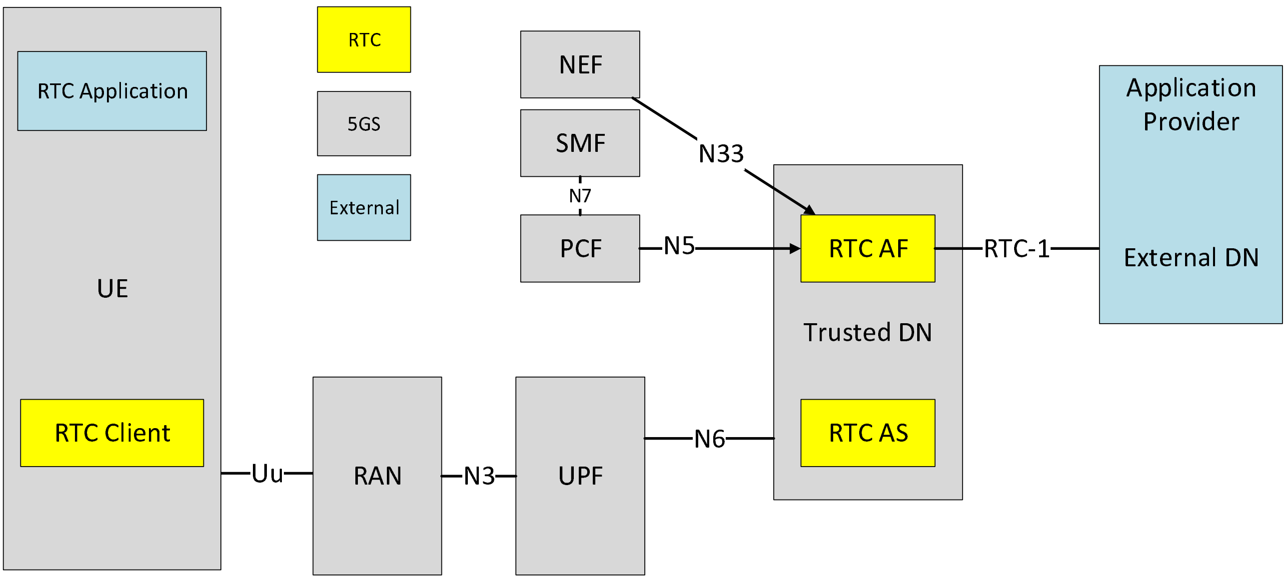
4 Real-Time media Communication Architecture

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 below.



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.

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

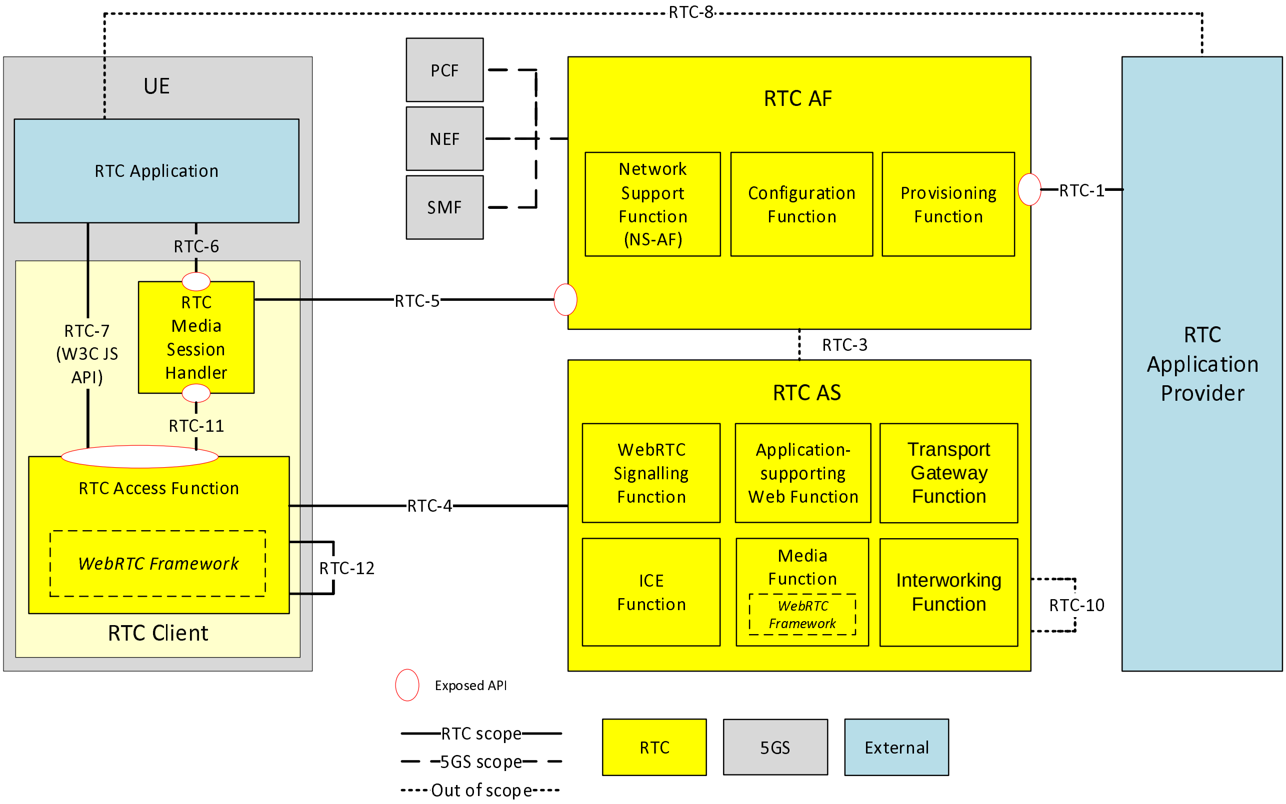
The media data is exchanged between two or more RTC endpoints over a 5G System as defined in TS 23.501 [11]. An RTC endpoint incorporates an instance of the WebRTC Framework configured by the RTC System defined in the present document. An RTC endpoint is typically realised by a UE, but an RTC AS, possibly deployed as an edge computing server as defined in clause 4.4.2, may also play the role of RTC endpoint. The Application Provider provides a RTC Application on the UE to make use of RTC endpoint and network functions using interfaces and APIs. The RTC architecture defines the functions and entities to support WebRTC-based service over a 5G System Two main functions are defined in the Trusted DN.

- RTC AF: An Application Function as defined in TS 26.501 [6] 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 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: Void.
NOTE 3: Red ovals indicate API provider functions.
NOTE 4: The RTC Access Function may be realised by a web browser in deployments of the RTC Client that support Web App through the W3C defined JavaScript APIs including WebRTC API.

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 Client are defined in clause 4.2 and the reference points shown in figure 4.1.1-2 are defined in clause 4.3.

Two types of RTC Application are defined in the present document:

- *Native WebRTC App:* An RTC Application running on the UE that makes use of client APIs at reference points RTC-6 and RTC-7.

- *Web App:* A web application running in a web browser on the UE that makes use of the W3C-defined WebRTC APIs.

NOTE: Detailed deployment architecture for the *Native WebRTC App* and the *Web App* are described in annex B.

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.

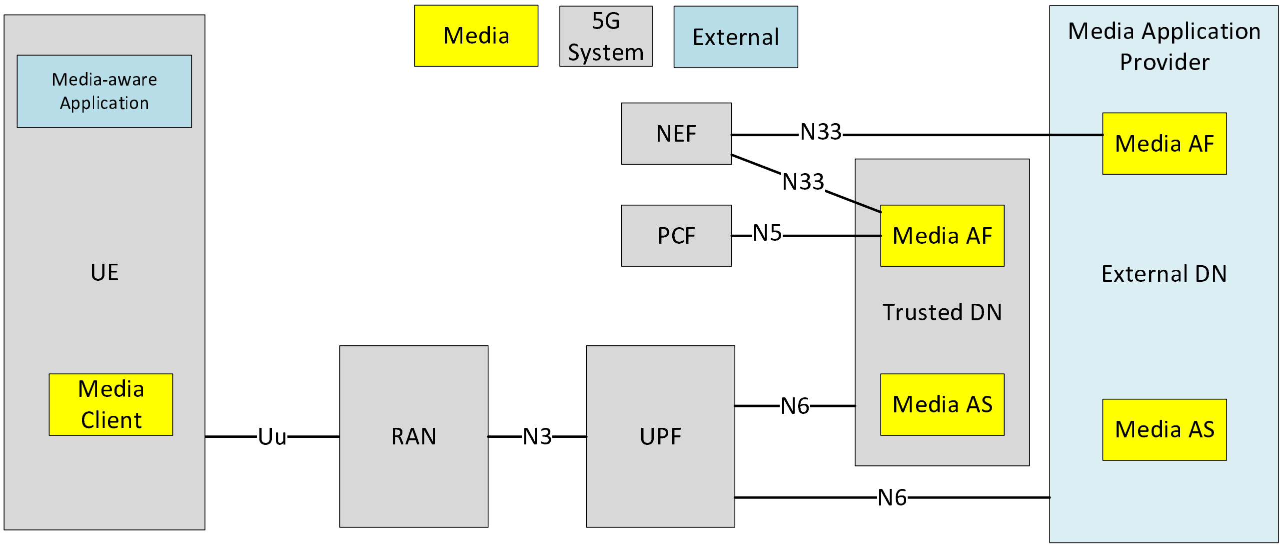


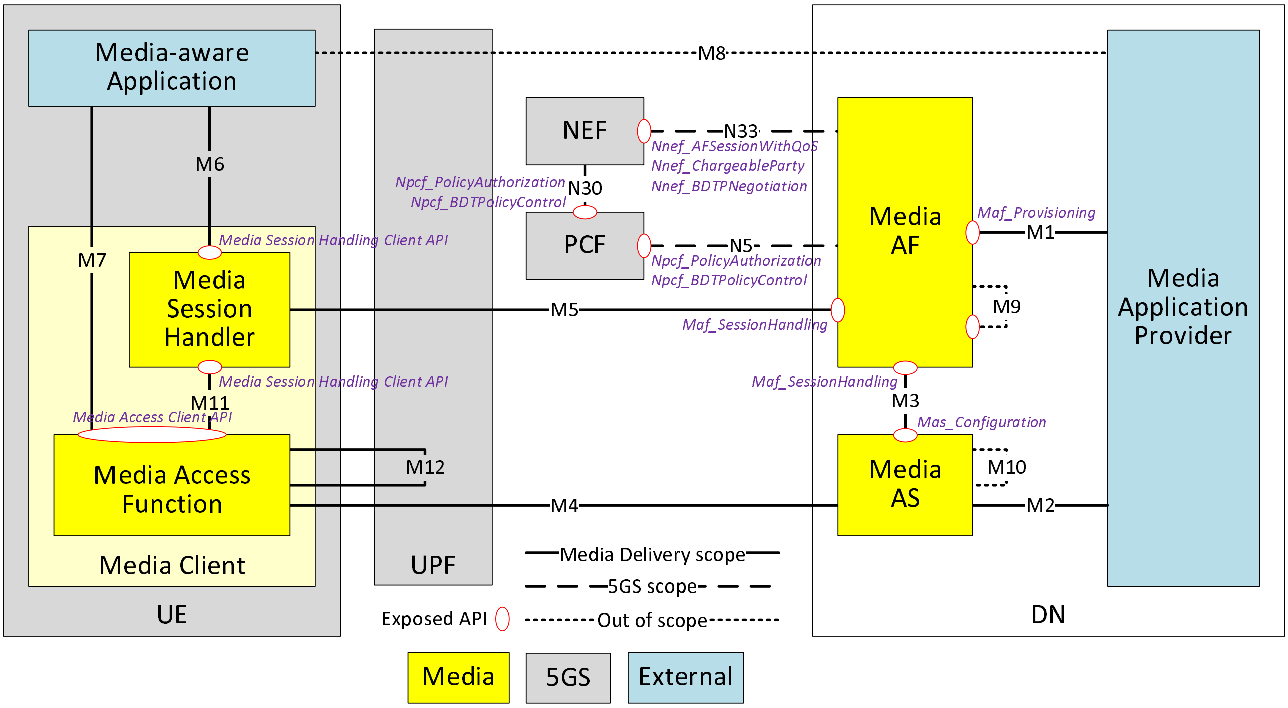
Figure 4.1.2.1-1: Generalized Media Delivery architecture within the 5G System

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 Client 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 Client
Media Session Handler	RTC Media Session Handler
Media Access Function	RTC Access Function
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 in this release.

M10: Reference point between one instance of the Media AS and another for the purpose of distributed service chaining over multiple Media AS instances.

NOTE 6: Reference point M10 is defined by the RTC architecture but is not further specified 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.

M12: Reference point between one RTC Access Function in a UE and another for the purpose of peer-to-peer media transport between different Media Clients when this is permitted by the 5G System.

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	RTC-10
M11	RTC-11
M12	RTC-12

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 Translation (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 Media Session Handler is an entity running on the UE which assists with the integration of the RTC Application. It exchanges, on behalf of the application, information about the RTC sessions with the network.

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

The *metrics reporting* subfunction of the RTC Media Session Handler executes the collection of QoS and QoE metrics measurements from the RTC Access Function, and sends metrics reports to the RTC AF for the purpose of metrics analysis or to enable potential transport optimizations by the network. The metrics to be collected and reported by the RTC Media Session handler are specified in clause 4.5.

The *consumption reporting* subfunction of the RTC Media Session Handler executes the collection of media consumption information from the RTC Access Function and submits consumption reports to the RTC AF for the purpose of RTC session audit. The media consumption information to be collected and reported by the RTC Media Session handler is specified in clause 4.6.

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 RTC 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 RTC session.
- Network Support Function exchanges information about the RTC session with the trusted STUN/TURN/ Signalling function, e.g. to identify a RTC session and associate it with a QoS template.

NOTE: The integration/co-location of this RTC AF and WebRTC signalling function is possible. A co-located WebRTC Signalling Function is able to act as an RTC AF which is accessible to 5GC, and replace some of this RTC AF's interfaces and APIs with the WebRTC Signalling Function. For example, reference point RTC-5 between the RTC AF and the RTC Media Session Handler is replaced to avoid concurrent/redundant requests from UEs.

4.2.6 ICE Function

The MNO may offer trusted ICE functions to the RTC Application to be used during the WebRTC ICE gathering phase. These functions may be STUN and TURN services that facilitate NAT and firewall traversal. In the context of the present document, a TURN service relays WebRTC content at reference point RTC-4m between multiple RTC Access Functions that cannot communicate directly with each other at reference point RTC-12 because the 5G System policy does not permit peer-to-peer communication.

When the ICE functions are provided by the MNO, they may assist with the 5G integration of RTC Applications. This may be achieved by triggering Network Assistance for starting or ongoing RTC sessions.

4.2.7 WebRTC Signalling Function

The WebRTC Signalling Function is used to set up and manage RTC Applications. It offers a standardized signalling protocol for session setup to RTC session participants. 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 RTC session.

The WebRTC Signalling Function manages media flow sessions in both uplink and downlink directions.

4.2.8 Interworking Function

This function provides interworking functionality to enable MNO-facilitated RTC 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 RTC sessions.

4.2.9 Transport Gateway Function

A Transport Gateway Function may be offered by the MNO to support cross-operator RTC 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 Interworking Function.

4.2.10 Media Function

A Media Function may be offered by the MNO to support RTC sessions. It may offer a wide range of functionality such as:

- A *content server* that serves content to the RTC Access Function, e.g. through a data channel
- *Media processing* used to perform tasks such as media transcoding, recording, 3D reconstruction, etc.
- *Scene composition* in which the Media Function composes a 3D scene and distribute it to several point-to-point RTC sessions.
- *Multi-point Control Unit (MCU)*: the Media Function offers multi-party conferencing functionality to merge a number of point-to-point RTC sessions.
- *Selective Forwarding Unit (SFU)*: the Media Function offers the selection, copy, and forwarding functionality of RTC sessions produced by multiple RTC Clients or WebRTC session participants external to the RTC System.
- Maintain uplink and downlink flow context (QoS, remote control and etc.) by interacting with the WebRTC Signalling Function in the RTC AS (see clause 4.2.7).

The Media Function is an RTC endpoint that incorporates an implementation of a WebRTC Framework as the basis of the above functionality.

4.2.11 Application-supporting Web Function

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

4.2.12 RTC Access Function

An RTC Access Function is a set of functions in the RTC Client that offers:

- Access to real-time media exchanged by its WebRTC Framework with that of one or more other RTC endpoints via reference point RTC-4m and/or RTC-12.
- Relaying WebRTC signalling between the RTC Application at reference point RTC-7 and the WebRTC Signalling Function of the RTC AS at reference point RTC-4s.
- Provision of client APIs to the *Native WebRTC App* at reference point RTC-7, as well as exposure of the W3C-defined JavaScript API including WebRTC API [31] to the *Web App* at reference point RTC-7.
- Provision of client APIs to the RTC Media Session Handler at reference point RTC-11.

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 bit rate recommendations.

4.3.3 RTC-4: Media-centric transport interface via RTC AS

Reference point RTC-4 is used to exchange the WebRTC media traffic between the RTC Access Function and the Media Function of the RTC AS (see clause 4.2.10) as well as to exchange signalling information relating to the WebRTC session with the RTC AS such as WebRTC Signalling Function (see clause 4.2.7).

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 subsidiary reference points as follows.

RTC-4s:

The RTC-4s is a Subsidiary reference point between the RTC Access Function and the RTC AS such as WebRTC Signalling Function. This interface is used for the exchange of signalling information relating to the RTC session between two or more RTC endpoints via the RTC AS.

RTC-4m:

This subsidiary reference point is used for transmission of media and other related data between two or more RTC endpoints when at least one of RTC endpoints participating in an RTC session is instantiated in the RTC AS.

The traffic at subsidiary reference point RTC-4m includes:

- Media data transmitted over SRTP
- Application data transmitted using Data channel.
- Media related meta-data transmitted using Data channel.
- Other application data.

NOTE 1: The Media Function of the RTC AS maintains the status for both uplink and downlink traffic.

NOTE 2: An RTC Client supports WebRTC streaming functions for uplink and downlink traffic.

4.3.4 RTC-5: Transport control interface

Reference point RTC-5 is used to convey configuration information from the RTC AF to the RTC Media Session Handler and is used by the RTC Media Session Handler to request media session handling support from the RTC AF for RTC sessions. The configuration information may consist of static information such as the following:

- Recommendations for media configurations.
- Configurations of STUN and TURN server locations.
- Configuration of the QoE metrics reporting feature.
- Configuration of the media consumption reporting feature.
- 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 an RTC session and its state.
- RTC MSH requests QoS allocation for a starting or modified session.
- RTC MSH receives notifications about changes to the QoS allocation for the ongoing RTC session.
- RTC MSH receives updated information about the RTC session with the RTC STUN/TURN/Signalling function, e.g. to identify a RTC session and associate it with a QoS template.
- RTC MSH collates QoE metrics received from the RTC Access Function, and submits metrics reports to the RTC AF.
- RTC MSH collates media consumption information received from the RTC Access Function and submits consumption reports to the RTC AF.

4.3.5 RTC-6: Client API

The RTC Media Session Handler 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 reference point RTC-6, 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 metrics 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

Reference point RTC-7 is used to provide the capability of RTC Access Function as API between the RTC Access Function and the RTC Application. The API includes Client API provided for the *Native WebRTC App* and W3C defined JavaScript APIs including WebRTC API provided for the *Web App* (i.e., JavaScript application) running on a browser.

4.3.7 RTC-8: Application interface

RTC-8 is a proprietary reference point between the RTC Application and the RTC Application Provider, which may be used to exchange configuration information related to the RTC session or the application.

4.3.7A RTC-10: RTC AS to another RTC AS interface

RTC-10 is a reference point between one instance of the Media AS and another for the purpose of distributed service chaining over multiple RTC AS instances. This may be present, for example when different RTC Clients are using the services of different Media Functions on separate RTC AS instances.

4.3.8 RTC-11: RTC Client configuration APIs

Reference point RTC-11 is used by the RTC Access Function to configure media session handling in the RTC Media Session Handler and/or used by the RTC Media Session Handler to configure media access in the RTC Access Function.

Because it is internal to the RTC Client, RTC-11 may not be exposed as an API to application developers but may instead be in the form of an internal API.

The RTC Access Function hides details of QoS allocation and network support from the Native WebRTC Application and Web App. It autonomously and transparently invokes the functions offered by the RTC MSH to provide support for the RTC session.

4.3.9 RTC-12: Peer-to-peer media-centric transport interface

Reference point RTC-12 is used to exchange WebRTC traffic between RTC Access Functions in different UEs when the 5G System permits peer-to-peer media transport. The protocols supported at this reference point shall be a subset of those at reference point RTC-4m.

NOTE: In case of peer-to-peer communication at reference point RTC-12, UPF provides the functionalities as specified in TS 23.501 [11], such as traffic usage reporting for billing and the Lawful Intercept (LI) collector interface.

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 the present document 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.

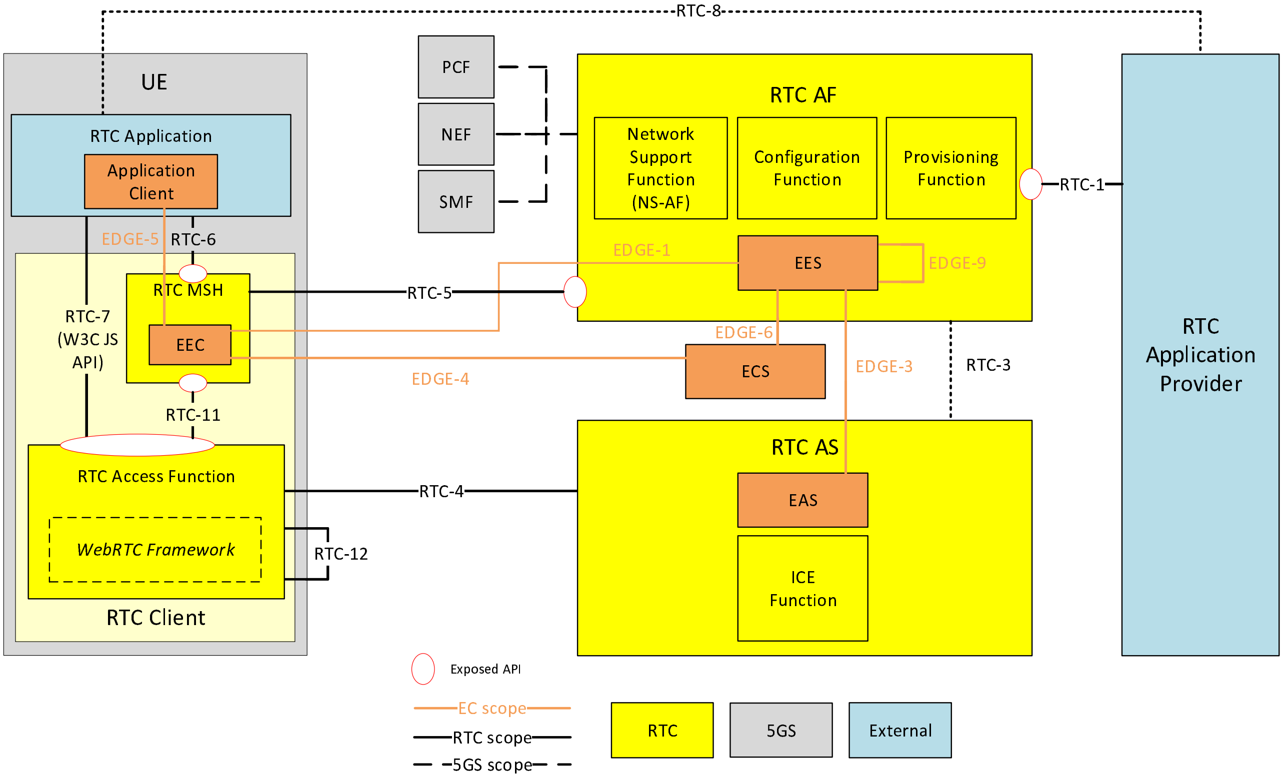
- The RTC Application Provider provisions edge resources through reference point RTC-1, in a similar fashion as defined in TS 26.501 [6], enabling client-driven and/or AF-driven edge configuration.

- In the client-driven approach, the Native WebRTC Application becomes aware of the support of edge processing in the network and takes steps, such as using the APIs at reference point EDGE-5, to discover and locate a suitable RTC AS instance in the Edge DN, similar to the process defined in clause 8.1 of [6].

- In the AF-driven approach, the RTC Application Provider provisions the RTC AF to deploy edge processing for the media sessions of the corresponding Provisioning Session, similar to the process defined in clause 8.2 of [6]. In this case, instantiating edge resources is the responsibility of the RTC AF, based on monitoring the load of the deployed EAS instances. A WebRTC Application may become aware of the deployed EAS through the RTC Application Provider from the RTC MSH through reference point RTC-5 (and possibly RTC-6). This appropriate EAS instance for a particular EAS Client to use is discovered by means of DNS resolution with support from the DNS server (e.g., EASDF/DNS resolver) as defined in TS 23.548 [12].

When the RTC Application is a web application, the implementation of the interface at reference point EDGE-5 to discover the RTC AS/EAS location by accessing the EEC is difficult if the Web browser does not implement interfaces necessary for supporting edge-enabled RTC applications/services.

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



NOTE: This figure corresponds to collaboration scenario 2. Other subfunctions of the RTC AS required to support collaboration scenarios 3 and 4 are not depicted.

Figure 4.4.2-1: Edge-enabled RTC architecture

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.

4.5 QoE metrics reporting for RTC

Per clause 4.3.2:

- An RTC Client shall support the collection and reporting at reference point RTC-5 of the QoE metrics defined in table 4.5-1 for real-time media it receives from reference points RTC-4 and RTC-12.

- An RTC AS shall support the collection and reporting at reference point RTC-3 of the QoE metrics defined in table 4.5-1 about the media that it receives from RTC Clients at reference point RTC-4.

These metrics are relevant for real-time media communication services over 5G System and are valid for speech, video and text media.

Table 4.5-1: QoE metrics for RTC

Metric	Definition
Corruption duration	For a particular component of the RTC session, the gap between the time of the last good media unit received before the corruption event and the time of the first subsequent good media unit. This metric shall report both the total corruption duration within each sampling period and the number of such corruption events that occurred within that sampling period.
Successive loss of RTP packets	The number of RTP packets lost in succession, measured separately for each received media component of the RTC session.
Average frame rate	The mean average media playback frame rate over a metrics sampling period, measured separately for each received media component of the RTC session. The value is calculated as the number of frames displayed during the sampling period divided by the time duration of that sampling period.

Average presentation latency	For a particular component of the RTC session, the mean average of the difference between the expected presentation time of each received media unit in the sample (as described by the media codec) and the actual presentation time of that media unit. This metric shall be reported once when its value exceeds a threshold indicated in the metrics reporting configuration and shall not be reported again until it falls below that threshold and subsequently exceeds it.
Sync loss duration	The time difference between value A and value B, measured separately for each received media component of the RTC session. Value A represents the time difference between the actual presentation time of the last played media unit of a video component and the last played media unit of its corresponding speech/audio component. Value B represents the time difference between the expected presentation time of the last played media unit of the video component and the last played media unit of its corresponding speech/audio component. This metric shall be reported once when its value exceeds a threshold indicated in the metrics reporting configuration and shall not be reported again until it falls below that threshold and subsequently exceeds it.
Average application Round-Trip Time	The mean average total round-trip latency between a pair of participants in an RTC session during the metrics sampling period, measured separately for each received media component of the RTC session. Each application Round-Trip Time sample is calculated as the RTP packet-level network Round-Trip Time (RTT) plus the additional delay due to buffering and other processing in the RTC Client and/or RTC AS.
Average encoded media bit rate	For each received media component of an RTC session, the total number of bits encoded for active media frames divided by the total time over which they were captured.

4.6 Media consumption reporting for RTC

Per clause 4.3.2:

- An RTC Client shall support the collection and reporting at reference point RTC-5 of information about the real-time media it consumes from reference points RTC-4 and RTC-12.
- An RTC AS shall support the collection and reporting at reference point RTC-3 of information about the real-time media it consumes from reference point RTC-4.

Details of the consumption reporting information to be collected and reported by the RTC Client are for further study.

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
- Discovery of ICE candidates
- Session establishment
- QoS request (either client-driven or WebRTC signalling function/server-driven)
- RTC traffic delivery

- QoS updates
- Metrics collection and reporting
- Consumption collection and reporting
- 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.

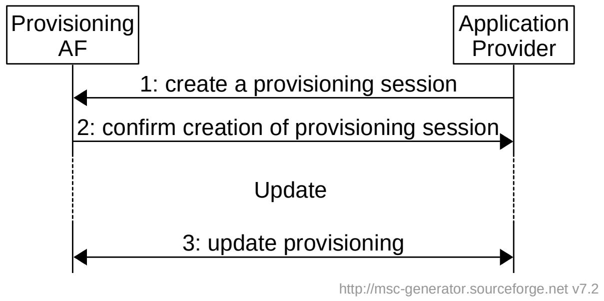


Figure 5.2.1-1: Provisioning procedure

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:

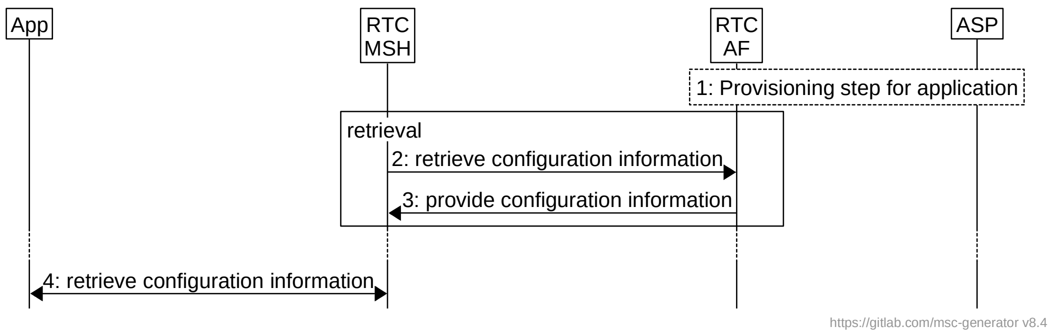


Figure 5.2.2-1: Configuration procedure

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.

5.2.3 Metrics reporting

5.2.3.1 Metrics reporting by RTC Access Function

This procedure is used to provision the QoE metrics reporting feature in the RTC AF and subsequently to configure the RTC Media Session Handler of an RTC Client to collect and report QoE metrics for the real-time media it has received. The RTC Media Session Handler collates the QoE metrics from the RTC Access Function (via reference point RTC-11) and submits metrics reports to the RTC AF via reference point RTC-5. The QoE metrics to be collected and reported are specified in clause 4.5.

The metrics reporting procedure is illustrated in figure 5.2.3.1-1.

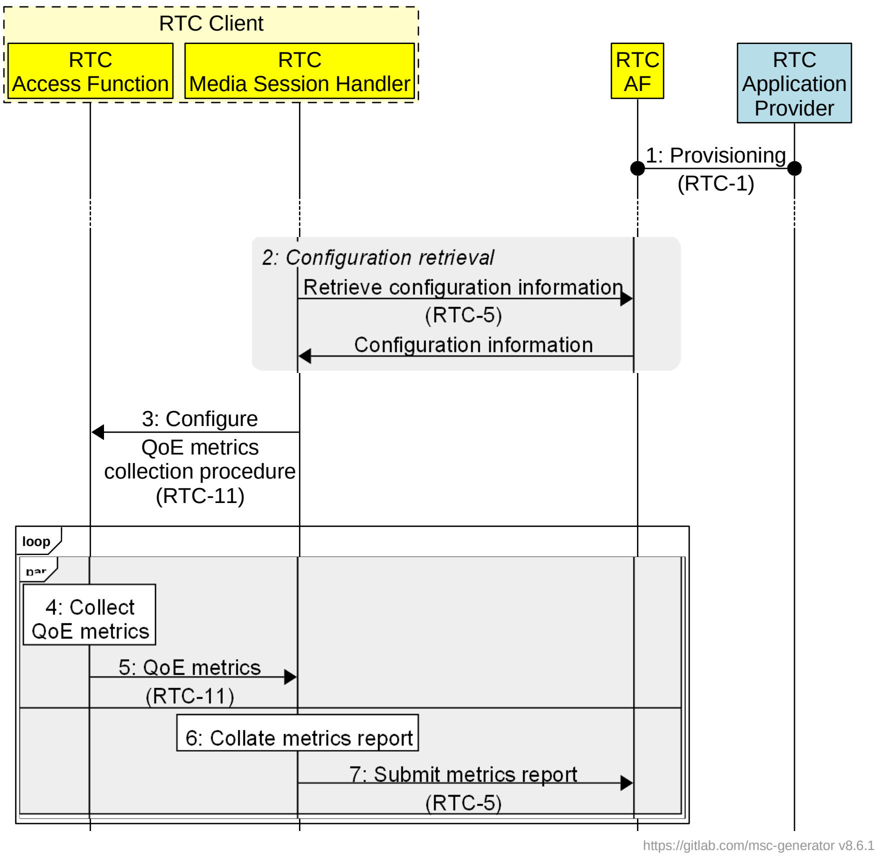


Figure 5.2.3.1-1: Metrics reporting procedure

The call flow is as follows:

1. An RTC Application Provider provisions resources for RTC sessions with metrics collection and reporting support.
2. The RTC Media Session Handler requests configuration information from the RTC AF relating to metrics collection and reporting for RTC sessions and the RTC AF provides the requested configuration information to the RTC Media Session Handler.

3. The RTC Media Session Handler configures the metrics collection procedure in the RTC Access Function.

4. The RTC Access Function collects QoE metrics about the real-time media it has received.

5. The RTC Media Session Handler receives collected metrics from the RTC Access Function.

6. The RTC Media Session Handler collates the received QoS metrics into metrics reports.

7. The RTC Media Session Handler periodically submits metrics reports to the RTC AF.

5.2.3.2 Metrics reporting by RTC AS

This procedure is used to provision the QoE metrics reporting feature in the RTC AF and subsequently to configure the RTC AS to collect and report QoE metrics for the real-time media it has received. The RTC AS collates the QoE metrics from its Media Function and submits metrics reports to the RTC AF via reference point RTC-3. The QoE metrics to be collected and reported are specified in clause 4.5.

The metrics reporting procedure is illustrated in figure 5.2.3.2-1.

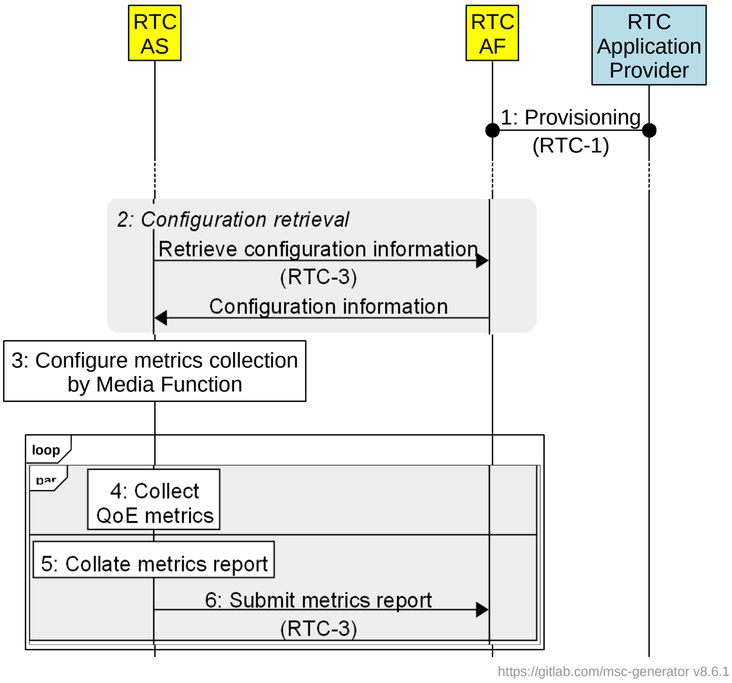


Figure 5.2.3.2-1: Metrics reporting procedure

The call flow is as follows:

1. An RTC Application Provider provisions resources for RTC sessions with metrics collection and reporting support.

2. The RTC AS requests configuration information from the RTC AF relating to metrics collection and reporting for RTC sessions and the RTC AF provides the requested configuration information to the RTC AS.

3. The RTC AS configures the metrics collection procedure in its Media Function.

4. The RTC AS Media Function collects QoE metrics about the real-time media it has received.

5. The RTC AS collates the received QoS metrics into metrics reports.

6. The RTC AS periodically submits metrics reports to the RTC AF.

5.2.4 Consumption reporting

5.2.4.1 Consumption reporting by RTC Access Function

This procedure is used to provision the collection of media consumption information in the RTC AF and subsequently to configure the RTC Media Session Handler of an RTC Client to collect and report consumption information for the real-time media it has received. The RTC Media Session Handler collates the consumption reporting information from the RTC Access Function (via reference point RTC-11) and submits consumption reports to the RTC AF via reference point RTC-5. The media consumption information to be collected and reported is specified in clause 4.5.

The consumption reporting procedure is illustrated in figure 5.2.4.1-1.

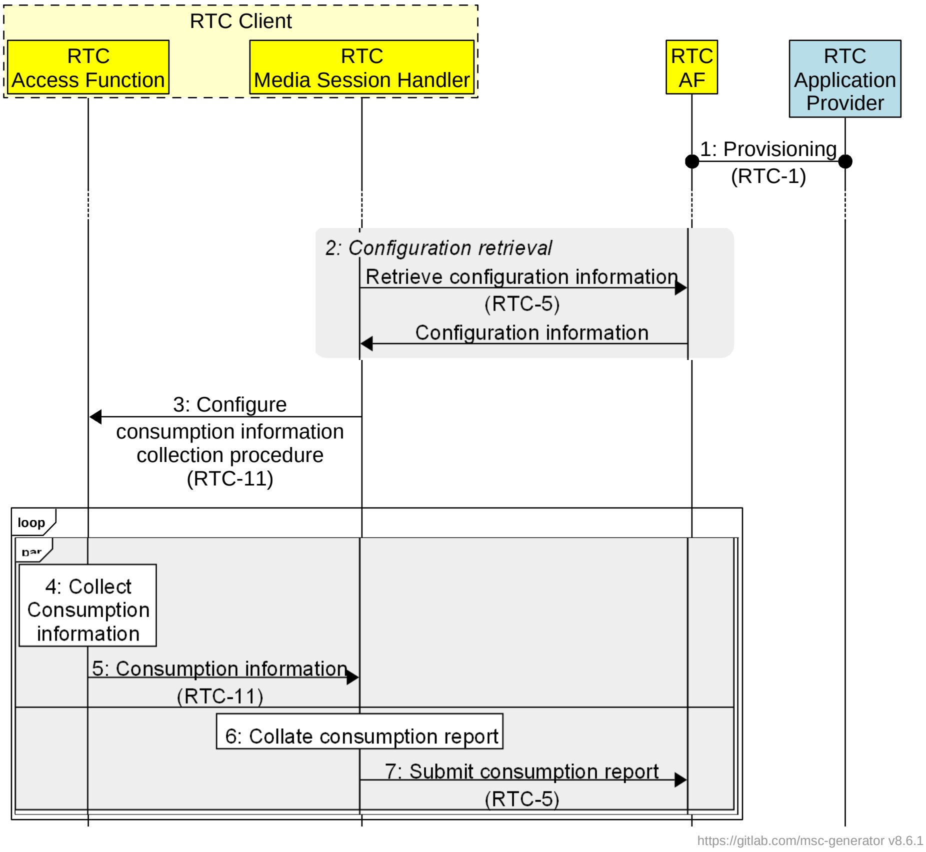


Figure 5.2.4.1-1: Consumption reporting procedure

The call flow is as follows:

1. An RTC Application Provider provisions resources for RTC sessions with media consumption information collection and reporting support.

2. The RTC Media Session Handler requests the configuration information from the RTC AF relating to media consumption collection and reporting for RTC sessions and the RTC AF provides the requested configuration information to the RTC Media Session Handler.

3. The RTC Media Session Handler configures the media consumption information collection procedure in the RTC Access Function.

4. The RTC Access Function collects consumption information about the real-time media it has received.

5. The RTC Media Session Handler receives collected media consumption information from the RTC Access Function.

6. The RTC Media Session Handler collates the received media consumption information into consumption reports.

7. The RTC Media Session Handler periodically submits consumption reports to the RTC AF.

5.2.4.2 Consumption reporting by RTC AS

This procedure is used to provision the collection of media consumption information in the RTC AF and subsequently to configure the RTC AS to collect and report consumption information for the real-time media it has received. The RTC AS collates the consumption reporting information from its Media Function and submits consumption reports to the RTC AF via reference point RTC-3. The media consumption information to be collected and reported is specified in clause 4.5.

The consumption reporting procedure is illustrated in figure 5.2.4.2-1.

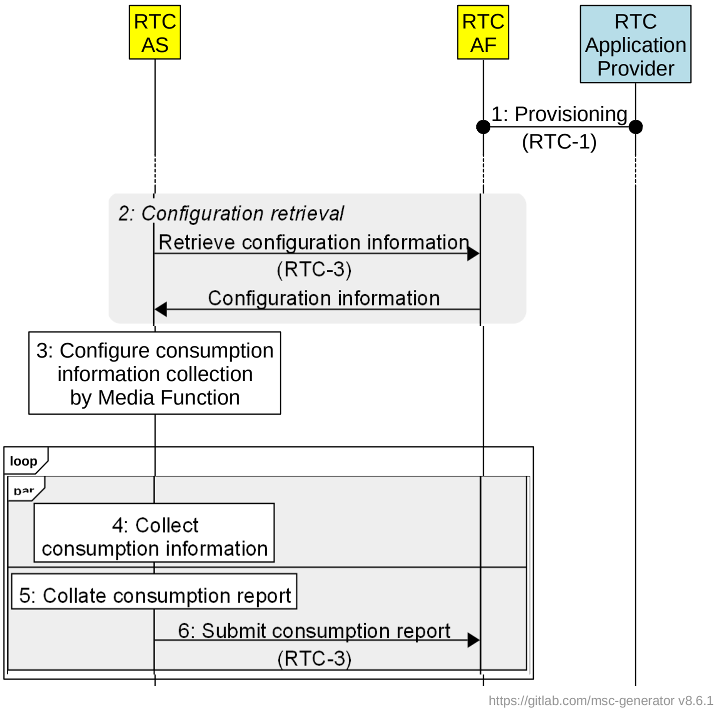


Figure 5.2.4.2-1: Consumption reporting procedure

The call flow is as follows:

1. An RTC Application Provider provisions resources for RTC sessions with media consumption information collection and reporting support.
2. The RTC AS requests the configuration information from the RTC AF relating to media consumption collection and reporting for RTC sessions and the RTC AF provides the requested configuration information to the RTC AS.
3. The RTC AS configures the media consumption information collection procedure in its Media Function.
4. The RTC AS Media Function collects consumption information about the real-time media it has received.
5. The RTC AS collates the received media consumption information into consumption reports.
6. The RTC AS periodically submits consumption reports to the RTC AF.

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.

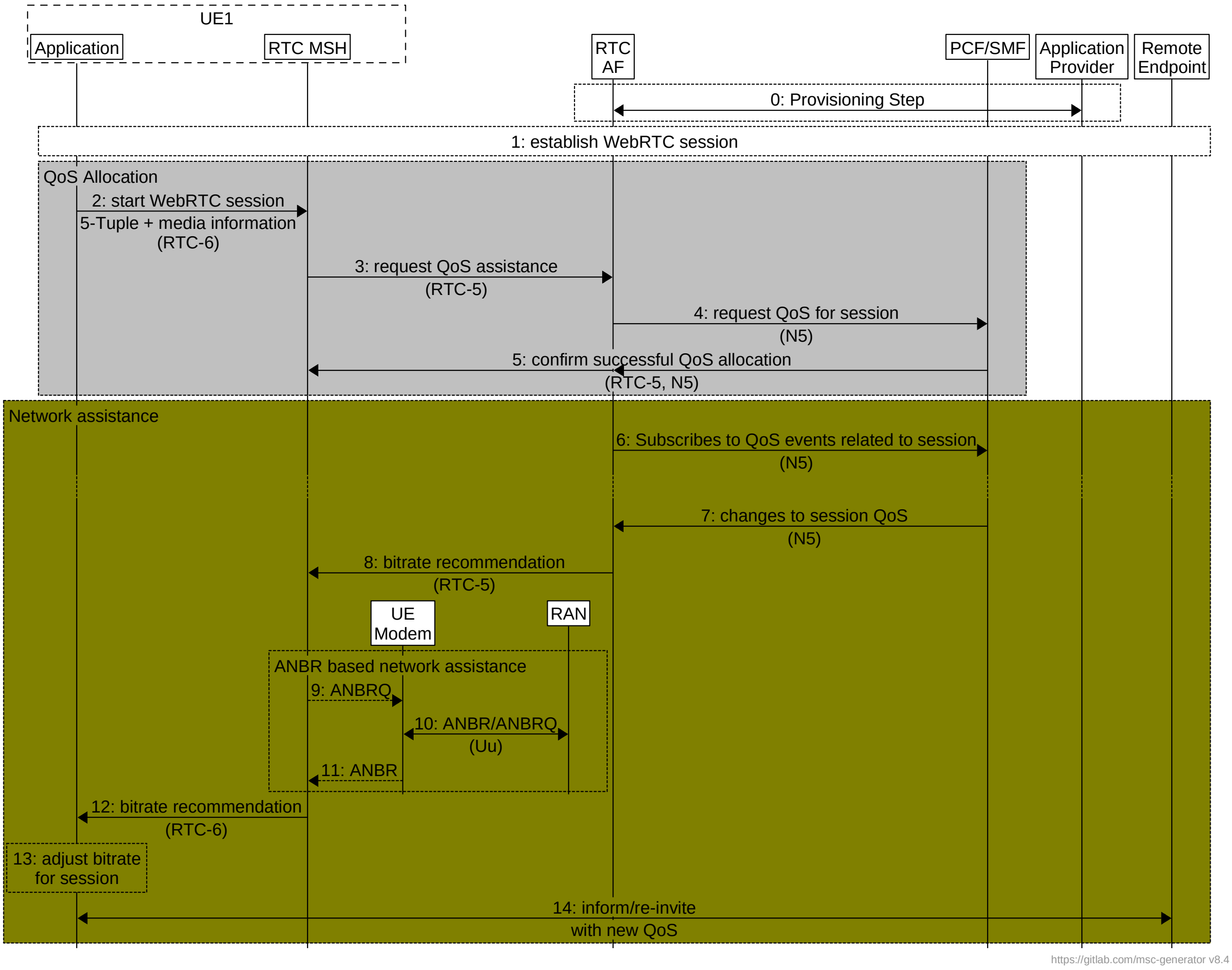


Figure 5.3-1: Call flow for Over-the-top (OTT) RTC sessions (collaboration scenario 1)

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 bit rate recommendations.
9. Alternatively, the MSH may interact with the UE Modem to trigger to query the recommended bit rate 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 bit rate 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 bit rate 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 bit rate adaptation have not been specified. The UE may determine the corresponding IP layer bit rate 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 bit rate adaptation have not been specified.

NOTE 2: The eNodeB may determine the corresponding IP layer bit rate 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 bit rate adaptation have not been specified.

NOTE 3: The recommended/queried bit rate as signalled over the LTE and NR access is defined to be in kbps at the physical layer. The uplink/downlink bit rate 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 bit rate recommendation to the RTC application.
13. The application may act on the bit rate recommendation, e.g. by reducing the uplink media bit rate.
14. The application may request remote endpoint to adjust the bit rate 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.

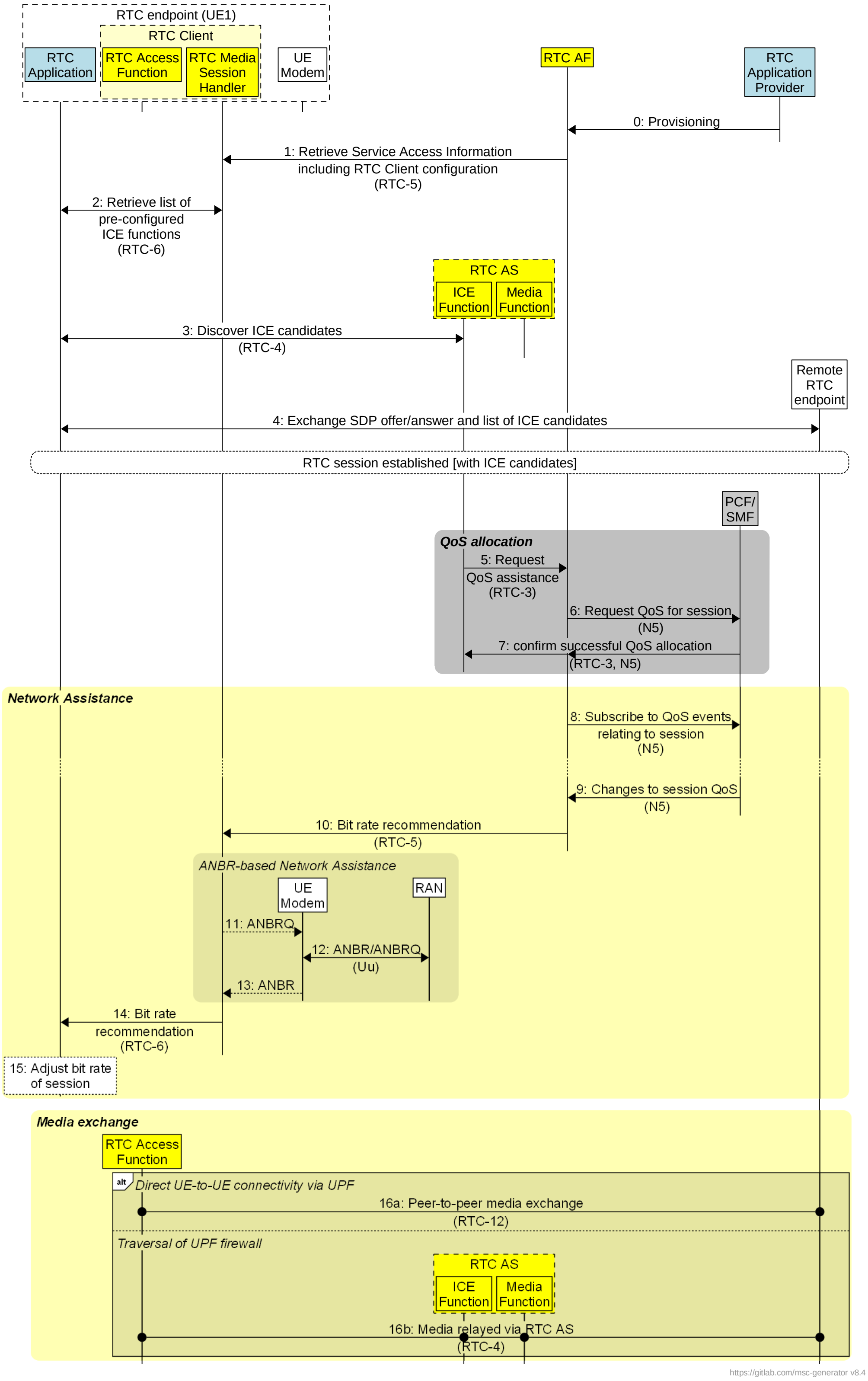


Figure 5.4-1: Call flow for network-supported RTC session (collaboration scenario 2)

The working assumptions are:

- The Application on UE1 and the remote RTC endpoint use an external WebRTC Signalling Function to establish the RTC session.

Initially:

0. A Provisioning Session is created by the RTC Application Provider at the RTC AF. This may include a list of trusted STUN/TURN servers to be used by the RTC Client for establishing RTC sessions.

The steps are as follows:

1. The RTC Media Session Handler retrieves Service Access Information from the RTC AF via reference point RTC-5. This includes configuration for the RTC Client, including a list of trusted STUN/TURN servers that it may use for establishing RTC sessions.

2. The RTC Application on UE1 queries the RTC MSH for the list of trusted ICE servers.

3. The RTC Application on UE1 discovers and tests the ICE candidates to validate that they are suitable for the connection to the remote RTC endpoint.

4. The RTC Application on UE1 and the remote RTC endpoint use an external RTC Signalling Function to exchange information about ICE candidates and to exchange the SDP offer/answer.

When an RTC 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 RTC Access Function) 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 RTC AF interacts with the PCF at reference point N5 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 RTC AF and the RTC AS.

8. The RTC AF also subscribes to events relating to the QoS flows of the RTC session with the PCF and SMF.

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

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

11. Alternatively, the RTC Media Session Handler may interact with the UE Modem to trigger to query the recommended bit rate 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.

    The RAN, based on the network status, returns the recommended bit rate 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 bit rate 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 bit rate adaptation have not been specified. The UE may determine the corresponding IP layer bit rate 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 bit rate adaptation have not been specified.

NOTE 2: The eNodeB may determine the corresponding IP layer bit rate 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 bit rate adaptation have not been specified.

NOTE 3: The recommended/queried bit rate as signalled over the LTE and NR access is defined to be in kbps at the physical layer. The uplink/downlink bit rate 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.

13. The UE modem notifies the recommended bit rate to the RTC Media Session Handler.

14. The RTC Media Session Handler notifies the recommended bit rate to the RTC Application.

15. The RTC Application may act on the bit rate recommendation, e.g. by reducing the uplink media bit rate.

16. Media traffic is exchanged with the remote RTC endpoint. If a TURN server is present in the RTC Client configuration obtained in step 1, media is relayed to the other RTC endpoint by the ICE Function of the RTC AS via reference point RTC-4, possibly with additional media processing by the Media Function of the RTC AS, as required by the RTC session.

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

In collaboration scenario 3, MNO hosts the RTC 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: When the RTC Application is a web-based application (i.e., a Web App), the RTC MSH function is not supported.

The call flow for this scenario when RTC MSH is involved are as shown in figure 5.5-1.

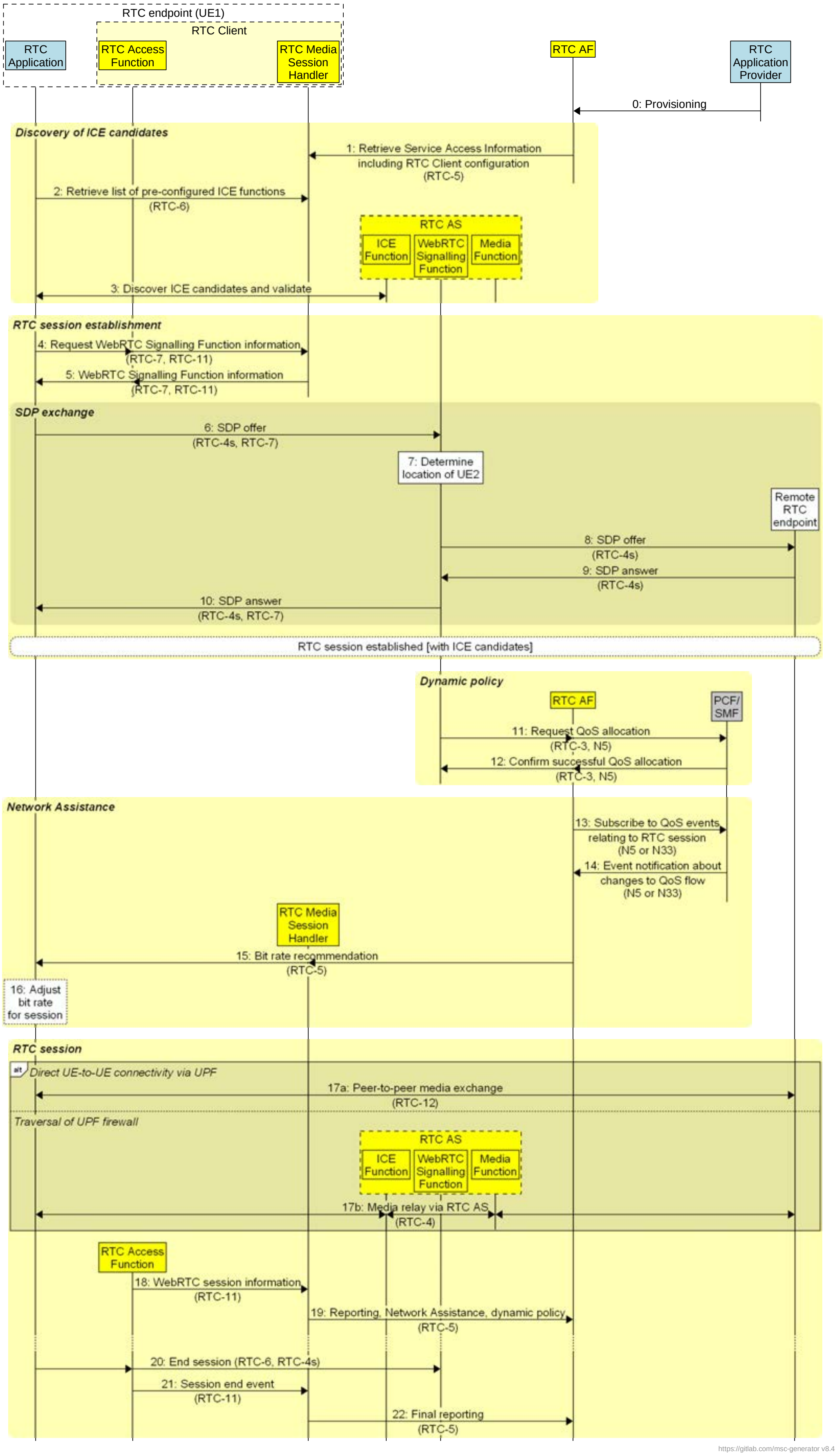


Figure 5.5-1: Call flow for MNO-facilitated RTC session (collaboration scenario 3)

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 RTC 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. *Retrieve Service Access Information:* The RTC MSH retrieves Service Access Information from the RTC AF that includes RTC Client configuration: a list of trusted STUN/TURN servers, trusted WebRTC Signalling Function and data channel service endpoint addresses and their capabilities. The RTC Client may use this configuration information for establishing RTC sessions.

2. *ICE Servers request:* The RTC 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 ***RTC session establishment*** phase is performed with the following steps in an MNO-facilitated RTC system:

4. *Query configuration information:* In response to a request from the RTC Application, the RTC Access Function queries the RTC MSH for the WebRTC signalling function information. In some cases where the signalling is not handled by RTC Access Function, the Native WebRTC application may directly query 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 the RTC Access Function or, in some cases, to the Native WebRTC application.

In the ***RTC session establishment*** phase, two or more RTC endpoints exchange signalling information relating to the RTC session, such as ICE candidates, SDP offer/answer using the WebRTC Signalling Function in the RTC AS.

NOTE: Figure 5.5-1 illustrates that SDP offer is generated by the RTC Application. However, in SFU/MCU mode, SDP offer is generated by the Media Function in the RTC AS.

6. *SDP offer:* The RTC Application creates a request with an 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, an RTC 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 in the RTC AS sends a request to the RTC AF for the allocation of QoS for the RTC session. The RTC AF sends a request to the PCF to allocate QoS for the media streams of the RTC session

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

If the Network Support Function is supported in the RTC AF, the Network Support Function AF (NS-AF) offers the bit rate 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 RTC session with the PCF and SMF.

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

15. *QoS notifications or bit rate 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 bit rate recommendation to the application.

16. *Adjust media bit rate:* An RTC Application may act on adjusting the bit rate recommendation, e.g., by reducing the uplink media bit rate.

After successful establishment of an RTC session and the bit rate negotiations, the actual ***RTC session*** over 5G may start:

17. *Media transfer:*

    a) In some cases, peer-to-peer communication between two RTC Clients is possible, and the media is exchanged directly with the remote RTC endpoint via reference point RTC-12.

    b) Otherwise, the RTC Application connects to the selected TURN server and/or Media Function in the RTC AS through reference point RTC-4, and the media is relayed to the remote endpoint by the ICE Function of the RTC AS, possibly with additional media processing by the Media Function of the RTC AS, as required by the RTC session.

18. *Method calls and notifications:* Supporting information about the RTC session is passed from the RTC Access Function to the RTC MSH.

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

20. *End session*: The RTC Application informs the RTC Access Function (via RTC-6) and WebRTC Signalling Function (via RTC-4s) that the RTC session has ended.

21. *Session ending event:* The RTC Access Function 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.

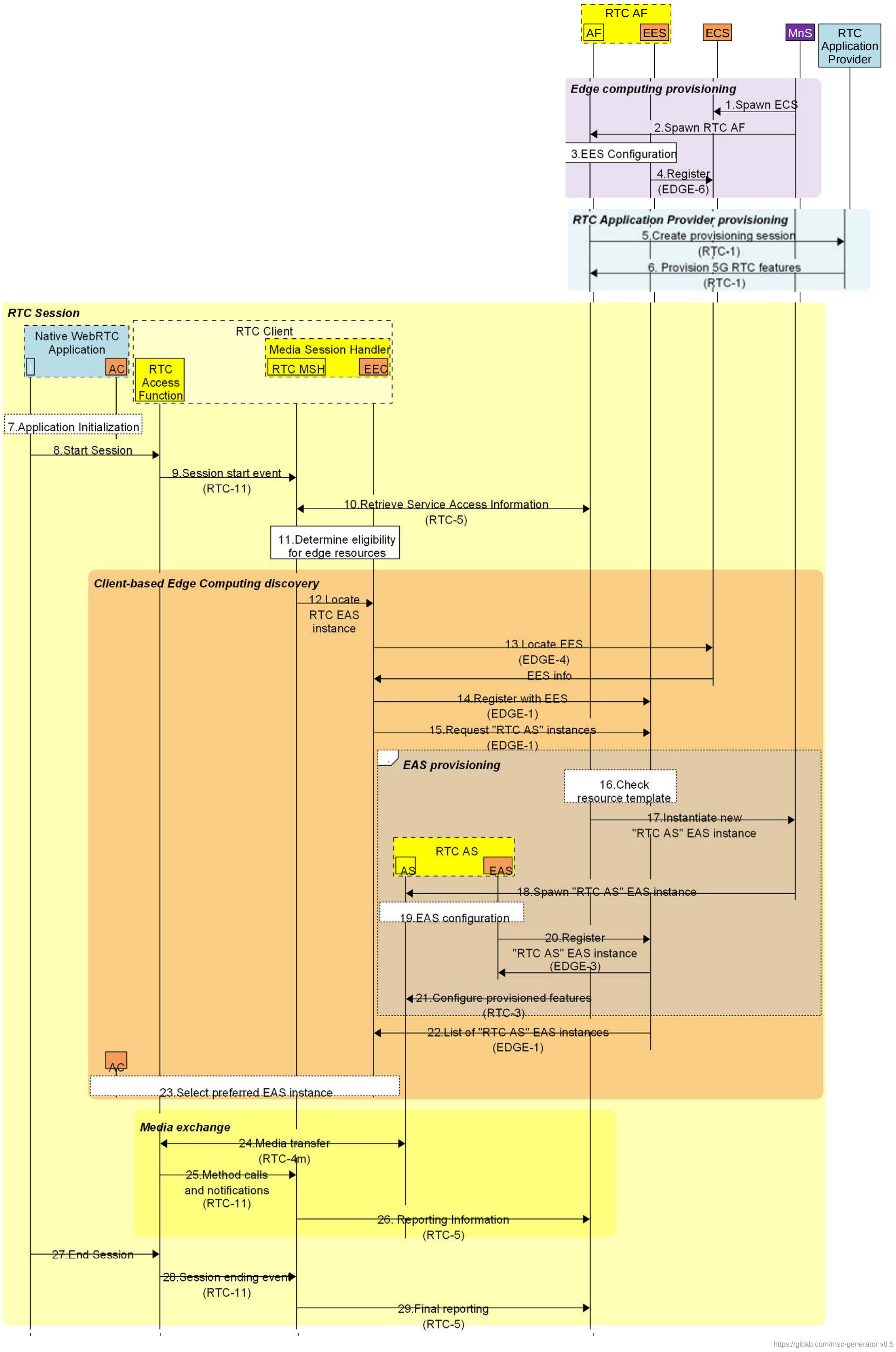


Figure 6.1-1. Client-driven management of RTC edge processing

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:* A new ECS instance is instantiated to manage new or increased demand for edge processing.
2. *Spawn RTC AF:* A new RTC AF that is edge-enabled is instantiated to handle new or increased demand for RTC 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 RTC sessions by the RTC Application Provider. Subsequent updates to the provisioning session are possible.

5. *Create Provisioning Session:* The RTC Application Provider creates a new provisioning session in the RTC AF.
6. *Provision RTC features:* 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 Native WebRTC Application initiates a new RTC session:

7. *Application initialization:* The user launches the Native WebRTC Application. The Native WebRTC Application performs any required initialization steps.
8. *Start RTC session:* The Native WebRTC Application invokes the RTC Access Function with appropriate streaming access parameters.
9. *Session starting event:* The Native WebRTC Application informs the RTC MSH about the start of a new RTC session.
10. *Retrieve Service Access Information:* The RTC MSH retrieves Service Access Information from the RTC AF, including RTC Client configuration information provisioned in step 6.
11. *Determine eligibility for requesting edge resources:* Using information from the Service Access Information, the RTC MSH determines whether the RTC 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 "RTC AS" EAS instance:* 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" EAS instance:* The MnS creates a new instance of the EAS offering RTC AS capability with the requested placement and resources.

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

20. *Register EAS instance 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 ***RTC session*** over 5G may start:

24. *Media transfer:* The Native WebRTC Application connects to the selected "RTC AS" EAS instance and the exchange of WebRTC media begins.

25. *Method calls and notifications:* Supporting information about the RTC session is passed from the RTC Access Function to the RTC MSH.

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

27. *End session:* The Native WebRTC Application informs the RTC Access Function that the RTC session has ended.

28. *Session ending event:* The RTC Access Function 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 Media Session Handler is shown in figure 6.2-1.

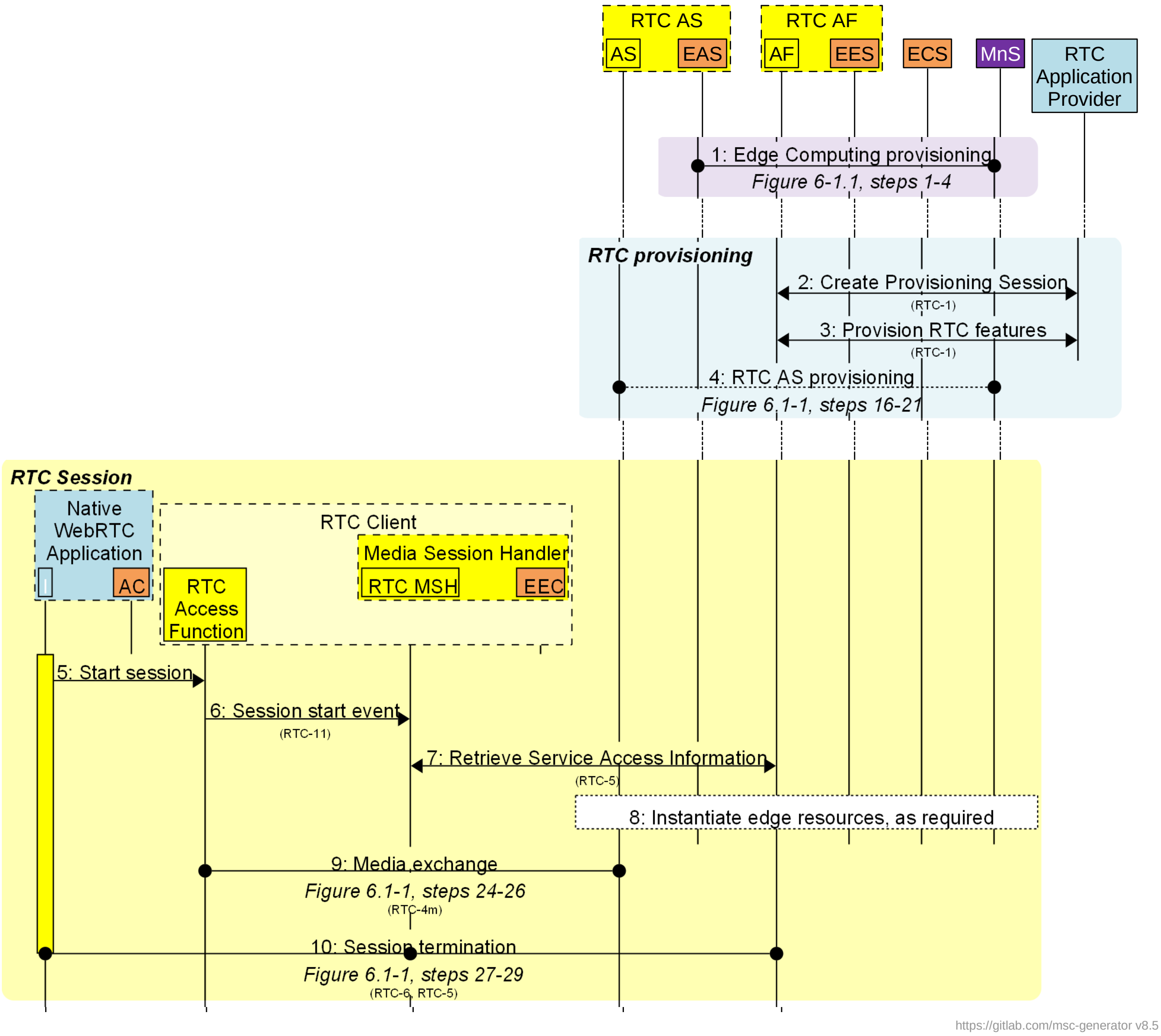


Figure 6.2-1. AF-driven management of RTC edge processing

The steps are:

1. Steps 1-4 as described in clause 6.1.
2. *Create Provisioning Session:* In this step, the RTC Application Provider creates a new provisioning session.
3. *Provision RTC features:* 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 needed), as described in figure 6.1-1, steps 16–21.

The Native WebRTC Application initiates a new RTC session:

5. *Start session:* The WebRTC Application invokes the RTC Access Function with appropriate real-time streaming access parameters.
6. *Session starting event:* The application informs the RTC MSH about the start of a new RTC session.

7. *Retrieve Service Access Information:* The RTC MSH retrieves Service Access Information from the RTC AF, including RTC Client configuration information.

8. *Instantiate edge resources:* The RTC AF determines whether edge resources need to be instantiated to support the RTC session, for example by requesting that the MnS instantiates a new "RTC AS" EAS.

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 table A.1-1.

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

- Over the top WebRTC: the RTC session runs completely over the top. However, the MNO may offer support in form of QoS allocation, bit rate recommendations, and QoE report collection based on request by the UE.
- MNO-provided WebRTC functions: On top of over-the-top scenario, the MNO additionally offers ICE Function in RTC AS.
- MNO-facilitated WebRTC services: the MNO hosts and facilitates RTC sessions by providing a WebRTC Signalling Function, which may also offer 5G network assistance.
- Interoperable WebRTC services: collaboration scenario 3 is extended with functions to support MNO to MNO interoperability.

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, is FFS.

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 Fignalling Function)
RTC Media Session Handler	Required	Optional	Optional	Optional
Network Support Function	Required	Required	Optional (maybe fulfilled by WebRTC Signalling Function)	Optional (maybe fulfilled by WebRTC Signalling Function)
ICE Function	N/A	Required	Optional	Optional
WebRTC Signalling Function	N/A	N/A	Required	Required
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 RTC 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 RTC AF to support QoS allocation, bit rate recommendations, and QoE report collection.

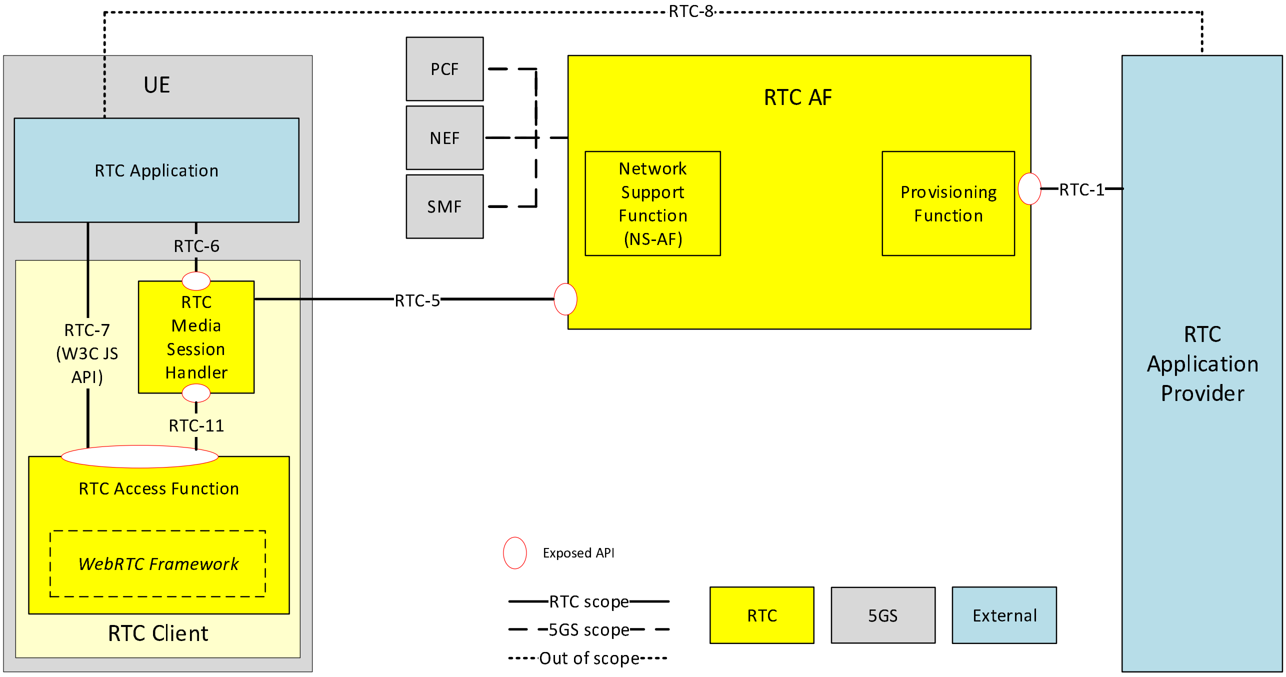


Figure A.2-1: Derivative RTC architecture for collaboration scenario 1

A.3 Collaboration scenario 2

Figure A.3-1 shows the architecture variant for the collaboration scenario 2 when MNO provides ICE Function in the RTC AS. It also instantiates the Configuration Function in the RTC AF to support the network-assisted RTC sessions over 5G System.

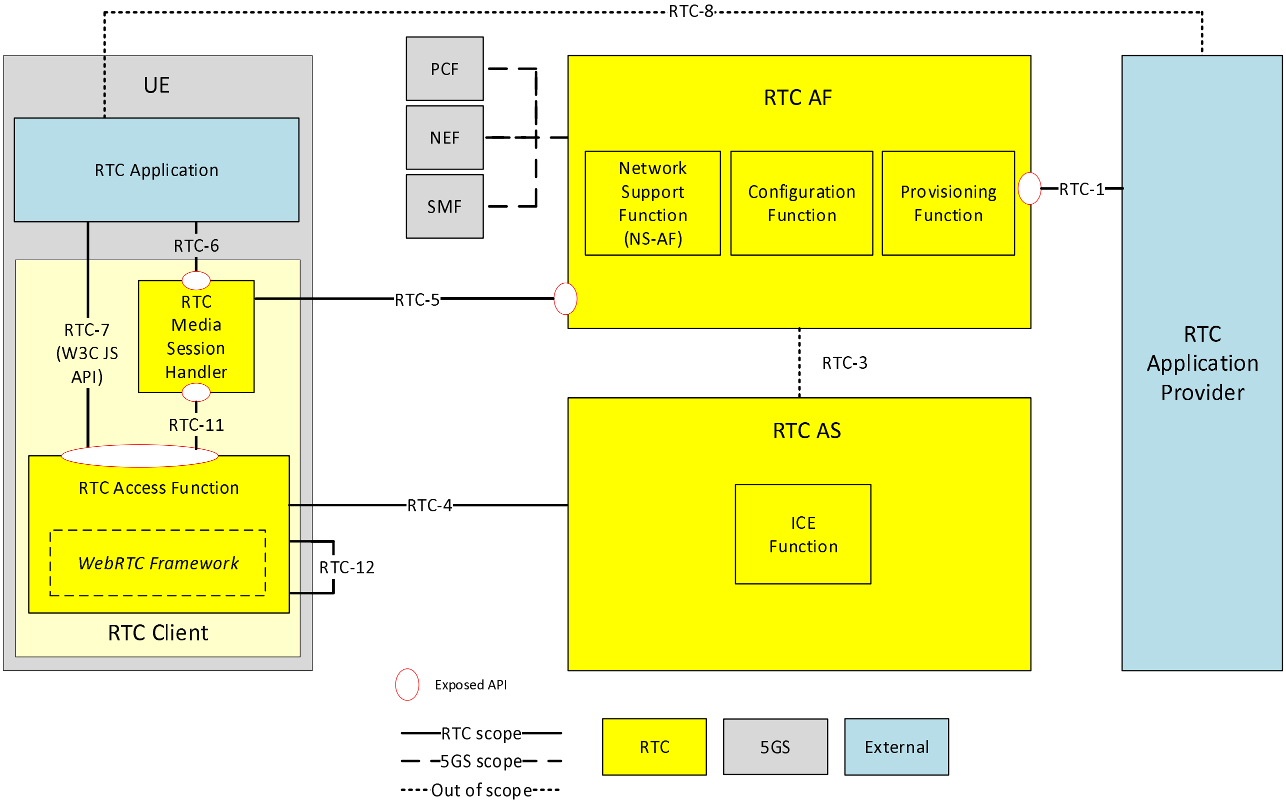


Figure A.3-1: Derivative RTC architecture for collaboration scenario 2

NOTE: RTC-4 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 RTC sessions by providing the WebRTC Signalling Function in the RTC AS. In addition, the Media Function is present in the RTC AS to provide the role of RTC endpoint.

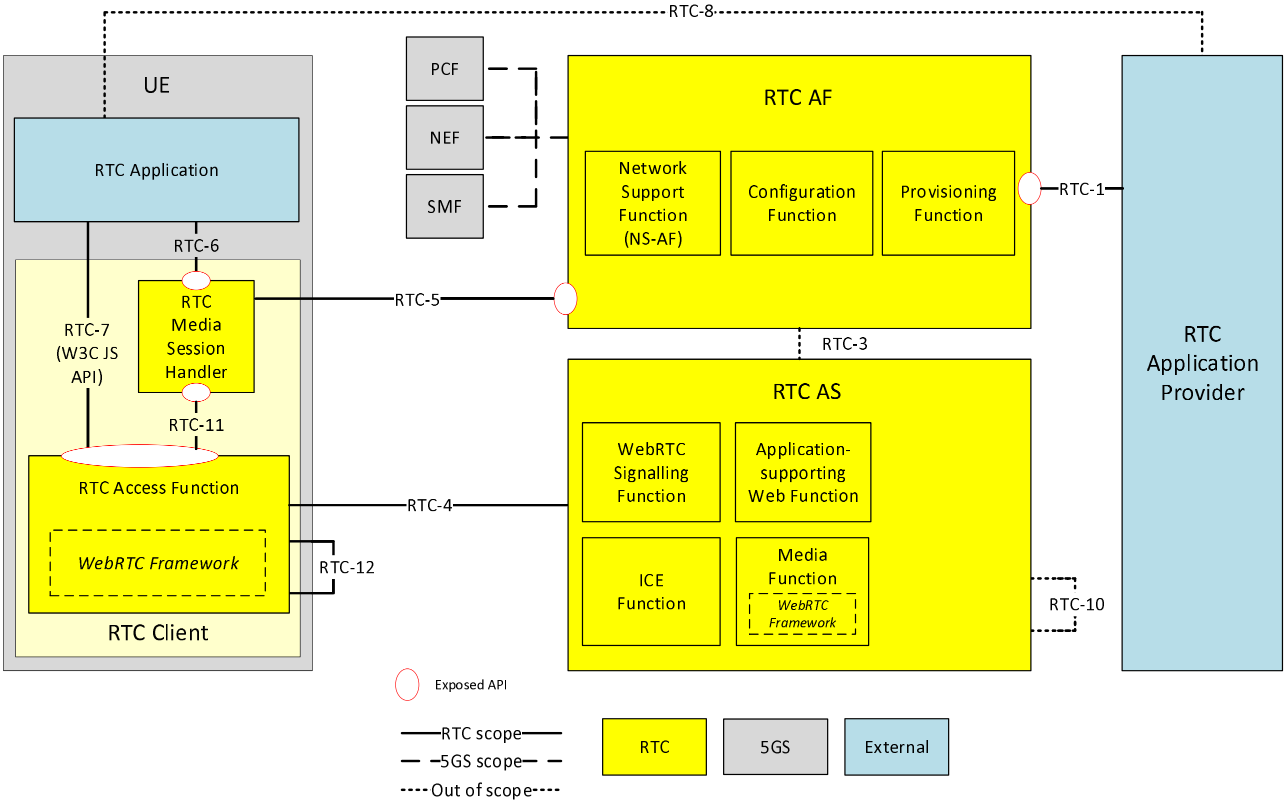


Figure A.4-1: Derivative RTC architecture for collaboration scenario 3

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 (normative): Architecture variants for RTC Application

B.1 General

This annex describes variants of the RTC reference architecture (see figure 4.1.1-2) for different kinds of RTC Application and describes the correspondence to the terminology defined in IETF RFC 8825 [13].

B.2 RTC Application is a Native WebRTC App

The *Native WebRTC App* implements reference point RTC-6 for invoking media session handling functionality and reference point RTC-7 for establishing and controlling RTC session by using API provided by the RTC Access Function. The *Native WebRTC App* also uses reference point RTC-7 and RTC-4s to exchange RTC session signalling with the WebRTC Signalling Function in the RTC AS.

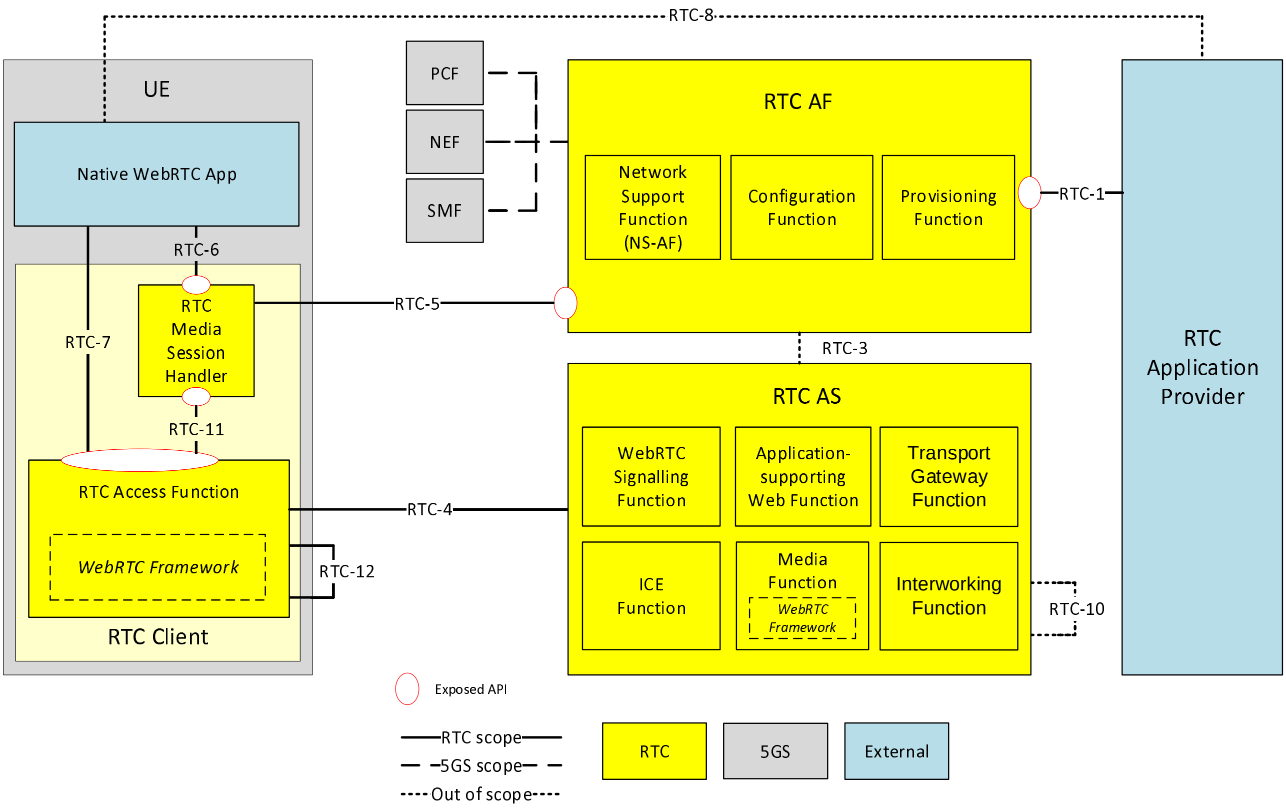


Figure B.2-1. RTC architecture variants for Native WebRTC App

This variant corresponds to the case defined in IETF RFC 8825 [13] where the WebRTC Endpoint is a *WebRTC non-browser*. The terminology correspondence is as follows:

WebRTC non-browser: An entity consisting of a Native WebRTC App and an RTC Access Function, as depicted in figure B.2-1, which has following characteristics.

- The WebRTC non-browser is a native application typically implemented by third-party application developers other than the library or Operating System developer, typically using programming languages other than JavaScript.
- The WebRTC non-browser is typically linked with a library corresponding to the RTC Access Function that provides the media plane protocol stack of the WebRTC Framework. Reference point RTC-7 is realised as the public interface of this library.

- The RTC Access Function library linked with the WebRTC non-browser has access to system APIs provided by the Operating System (e.g., its Socket API).
- The WebRTC non-browser linked with the RTC Access Function library performs WebRTC signalling.
- The WebRTC non-browser linked with the RTC Access Function library including WebRTC Framework terminates audio/video media and data over a WebRTC data channel.
- The RTC Access Function library linked with the WebRTC non-browser may assist the Native WebRTC App's control plane signalling.

B.3 RTC architecture for Web App

The *Web App* does not instantiate reference points RTC-6. Instead, it uses the W3C defined JavaScript APIs (including WebRTC API) which is exposed by RTC Access Function via RTC-7 to exchange RTC session signalling with the WebRTC Signalling Function in the RTC AS.

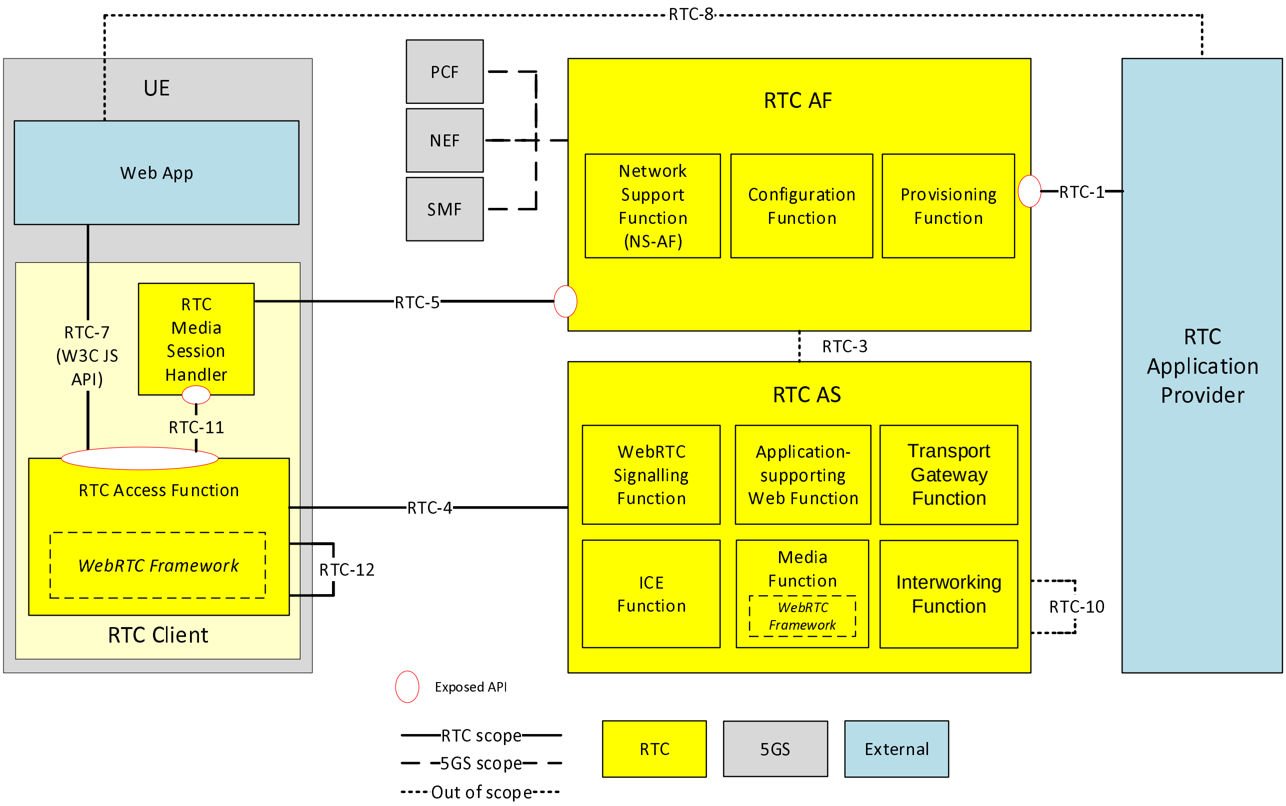


Figure B.3-1. RTC architecture variants for Web App

This variant corresponds to the case defined in IETF RFC 8825 [13] where the WebRTC Endpoint is a *WebRTC browser*. The terminology correspondence is as follows.

WebRTC browser: An entity corresponding to RTC Access Function depicted in figure B.3-1, which has the following characteristics.

- The WebRTC browser is typically a web browser not modified by third-party application developers.
- The WebRTC browser hosts JavaScript applications (per section 3 of IETF RFC 8825 [13]) which provide the actual services.
- The WebRTC browser implements the media plane protocol stack of the WebRTC Framework for the purpose of terminating audio/video media at reference point RTC-4m or RTC-12, and provides control of those media components by exposing the WebRTC API and related other APIs to JavaScript applications at reference point RTC-7.

- The WebRTC browser implements the media plane protocol stack of the WebRTC Framework for the purpose of exchanging data over a WebRTC data channel at reference point RTC-4m or RTC-12, and provides the transported data to JavaScript applications by exposing the WebRTC API to them at reference point RTC-7.
- The WebRTC browser exposes a WebSocket API to the JavaScript application for the purpose of transporting signalling messages to other RTC endpoints via reference point RTC-7 and then RTC-4s.

JavaScript API: As defined in section 2.2 of IETF RFC 8825 [13]. This corresponds to APIs exposed by the RTC Access Function at reference point RTC-7, which has the following characteristics.

- Those APIs are W3C-defined JavaScript APIs implemented and exposed by the WebRTC browser and which can be utilized by the JavaScript applications including WebRTC API, and WebSocket API.

JavaScript application: Per section 3 of IETF RFC 8825 [13]. This corresponds to Web App depicted in figure B.3-1, which has following characteristics.

- The JavaScript application is a Web application running on the WebRTC browser (and typically implemented by third-party application developers rather than by a library or Operating System developer) using JavaScript.
- The JavaScript application has the capability to control of audio/video media by using the WebRTC API and related other APIs provided by the WebRTC browser at reference point RTC-7.
- The JavaScript application has the capability of exchanging data over a WebRTC data channel by using the WebRTC API provided by the WebRTC browser at reference point RTC-7.
- The JavaScript application terminates WebRTC signalling using the WebSocket API provided by the WebRTC browser at reference point RTC-7 and uses this to drive the transport of signalling messages via reference point RTC-4s.

Annex C (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
2024-06	SA#104	SP-240692	0005	2	F	Terminology alignment in RTC architecture	18.3.0
2024-09	SA#105	SP-241111	0007	1		[GA4RTAR] Clarification on metrics and consumption collection and reporting procedures	18.4.0
2024-09	SA#105	SP-241111	0006	2		Terminology correction	18.4.0

History

Document history		
V18.2.0	May 2024	Publication
V18.3.0	July 2024	Publication
V18.4.0	October 2024	Publication