**3GPP TSG-SA WG4 Meeting #129-eS4-241441**

**e-Meeting, 19 – 23 August 2024**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.3* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **26.506** | **CR** | **0006** | **rev** | **-** | **Current version:** | **18.3.0** |  |
|  | | | | | | | | |
| *For* [*HE**LP*](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **X** | Radio Access Network |  | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | Terminology correction | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | NTT | | | | | | | | | |
| ***Source to TSG:*** | S4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | GA4RTAR | | | | |  | ***Date:*** | | | 2024-08-09 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **F** |  | | | | | ***Release:*** | | | Rel-18 |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)  Rel-20 (Release 20)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | Terminologies are defined for RTC specification. However, there are some ambiguous and inaccurate definition and inconsistent description in the specification. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | The definition of RTC endpoint and WebRTC Framework are modified.  Usage of the terminology is modified for the alighment with the definition.  Added descriptions clarifying the position of WebRTC non-browser and WebRTC browser by representing the correspondence and difference between RTC and IETF-defined functional blocks. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | There are possible misunderstanding and incorrect implementations. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 3.1, 4.1.1, 4.2.12, B.1, B.2, B.3 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **X** | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  | **X** | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  | **X** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

\* \* \* First Change \* \* \* \*

## 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 exchanging real-time media and data by incorporating an instance of the WebRTC Framework.

NOTE: By the above definition, a UE incorporating an RTC Client is an RTC endpoint, and an RTC AS is also an RTC endpoint by virtue of incorporating a Media 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.

\* \* \* Next Change \* \* \* \*

### 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.



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.

\* \* \* Next Change \* \* \* \*

### 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.

\* \* \* Next Change \* \* \* \*

# 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.

\* \* \* Next Change \* \* \* \*

# 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.

\* \* \* End of Changes \* \* \* \*