**Source: NTT**

**Title: FS\_eiRTCW Permanent Document**

**Version: 0.1.1**

**Document for: Agreement**

**Agenda Item: 10.6**

# Scope

The present document extends immersive Real-time Communication for WebRTC (iRTCW) and introduces a new concept called native WebRTC signalling.

In 3GPP, the use of WebRTC technology has been investigated since Rel-12 (around 2014). They are a network-based architecture for WebRTC access to IMS specified in Annex U to TS 23.228 and its stage 3 protocols specified in TS 24.371. They define functional entities including WIC (WebRTC IMS Client) and eP-CSCF (P-CSCF enhanced for WebRTC). The eP-CSCF is assumed to be located in the Home IMS domain and communicates with other IMS entities using the existing interfaces. For the C-plane signalling between WIC and eP-CSCF, those specifications specify an option to use SIP over WebSocket, whose information model can be used for options other than SIP over WebSocket. Although SIP satisfies almost all conversational applications, it is somewhat over-engineered or too strict to extend. Another method which is flexible, extensible, and can be optimized for new XR conversational applications, therefore, should be investigated. These requirements remind us of the original design principle of WebRTC. WebRTC, by its inherent characteristics, does not regulate C-plane signalling and allow a wide range of C-plane signalling. This study looks over this design principle again and investigates a new SIP-decoupled C-plane signalling, called native WebRTC.

Regarding the level of signalling details, TS 24.371 specifies a signalling transport mechanism using SIP over WebSocket, but it is not a mandatory mechanism for eP-SCSF. Even though there are other options such as XMPP or other application protocols over WebSocket, a RESTful based interface, etc., TS 24.371 does not specify any details of C-plane signalling using other options. Each service provider (e.g., operator) develops its own application by following the guidelines in TS 24.371. Its subscriber downloads the application and connects to the service and other subscribers only within the same service. Detailed C-plane signalling is left open to each operator’s design. In contrast, this study tries to identify a new C-plane signalling in detail (as an interface specification) to the extent that client implementations based on it have enough interoperability. This realizes connectivity to any operators or roaming services for new XR real-time communications. Operators can provide the interface common to them according to well-defined C-plane signalling specifications. Clients can connect to any operators via the interface (see Figure1).

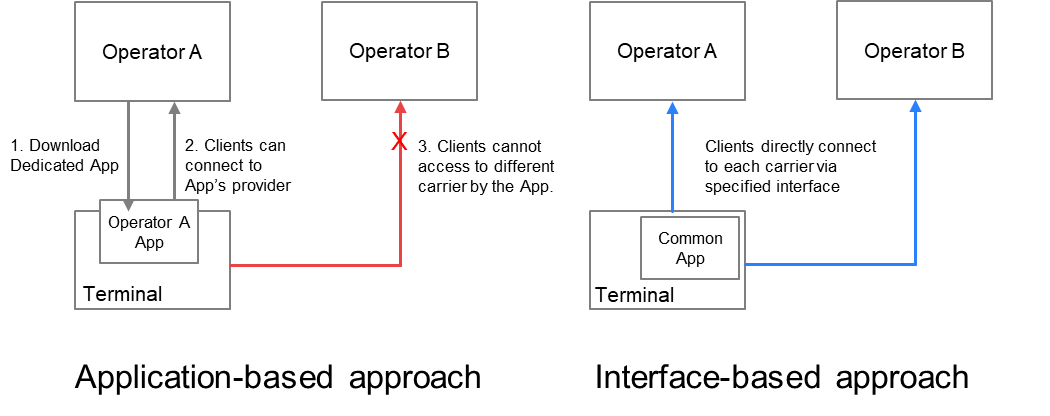


Figure 1. Two approaches for defining specifications and their application connectivity

# 2 Motivations for Native WebRTC Signalling

## 2.1 C-plane Signalling comparison

The C-plane signalling can be expressed as follows. Now, there are roughly four possible methods, classified in terms of their protocol stacks (see Figure 2).

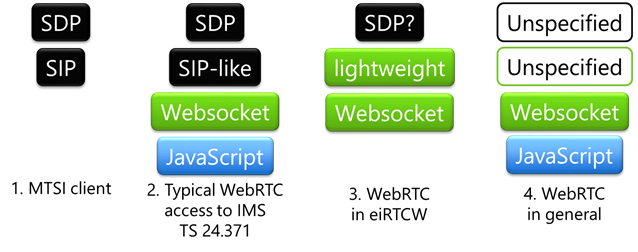


Figure 2. Comparison of protocol stacks

The first method is MTSI-based, using SIP and SDP. General C-plane signalling requirements for conversational services can be covered by SIP. Interoperability is fine with the existing 5G core network. It is to be treated in IMS-based AR Conversational Services (IBACS).

The second is the method specified in TS 24.371. It enables the WebRTC clients to communicate over an IMS-based core network; only the interfaces for downloading dedicated applications and the signalling path using WebSocket are specified for C-plane signalling. Ordinary implementations adopt SIP-like protocols over WebSocket. In most cases, it is partially SIP-compliant or tightly coupled with SIP to adapt WebRTC clients in IMS domain.

The third method is an alternative to the second method that uses SIP-like protocol over WebSocket. The third method uses another signalling protocol over WebSocket, but SIP-decoupled approaches are investigated. It can be more lightweight, omitting features that is not used in XR conversational. Some constraints on SDP are necessary for interoperability. Non-browser based implementations are also in the scope. This method is the main subject of this study, FS\_eiRTCW.

The other is a general WebRTC protocol stack that is not specified and left open to the users (i.e., service providers). C-plane may be SIP, XMPP, http, etc. A general WebRTC application uses SDP syntax compliant to RFC 4566 for its internal representation, when setting the local and remote descriptions. The application can transform the SDP into its on-the-wire format, which can be constructed from SDP and be serialized out to SDP.

Editor’s Note:   
- The reason why WebRTC signalling is necessary  
- Comparison interworking between WebRTC signalling and existing SIP

# 3 Use Cases

Editor’s Note:   
- Use cases and communication methods (P2P, SFU, MCU)

# 4 Gap Analysis for WebRTC Architecture

Editor’s Note:   
Analyze gaps and identify required enhancements of terminal device and network architectures including additional functional entities (e.g., WebRTC Signalling Server, ICE-STUN Server, IMS Interworking Gateway, NNI Gateway).  
  
- Stage2 work requirements  
- Necessary functional blocks  
- Architectural comparison (details are in annex)

# 5 Requirements for C-Plane Signalling

Editor’s Note:   
Identify signalling protocol details (e.g., based on JSON) for the common WebRTC-based immersive RTC session management.  
Identify information elements in the C/U-Plane signal (including NNI) to enhance connectivity of media sessions with carrier assistance for WebRTC-based applications (including OTT applications).  
Identify the minimal functional capabilities needed to support the enhancements identified in above.  
  
- Stage3 requirements  
- C-plane signalling requirements  
- Signalling methods comparison (details are in annex)

# 6 Requirements for U-plane Signalling

# 7 Interworking with IMS Network

# 8 Tethered Cases

Editor’s Note: SmarTAR-related clause;  
Identify enhancements for E2E QoS realizations over 5G systems for communications between MNOs and WebRTC clients operating over non-5G links (e.g., Wi-Fi) using WebRTC-based transport. This also includes communication between WebRTC clients operating on tethering/tethered devices.

# 9 Security Considerations

Editor’s Note:   
Considerations that the third-party access to the operator network need to be controlled with SLAs and with secure access to protect the underlying network resources.  
- Rate limiting  
- Abuse protection  
- Security measures

# 10 Related Groups Considerations

Editor’s Note:   
Identify collaboration formation with other WGs in 3GPP and SDOs including IETF and W3C.

# 11 Conclusions and Recommendations

# 12 Open Issues

## 12.1 Materials for further study

a) The C-plane signalling protocol should support basic WebRTC service operations such as client registration, authentication and authorization; call control; and data channel management that are relevant to the new architecture.

c) Security considerations for interoperable WebRTC services such as authentication, authorization, and key management

d) Deployment options of traditional WebRTC functions in 5G network, and mapping of those functions to 5G media architecture

NOTE: Mapping of WebRTC functions to 5GMS functions to be confirmed in 5GAREA study

e) Feasibility to use existing 5GMS architecture enablers for betterment of WebRTC services.

Annex <A>:  
Architectural WebRTC Entity Examples

Editor’s Note:   
Architectural example of integration of WebRTC with 5G network

Annex <B>:  
Protocol Stack Examples

Editor’s Note:   
Definite example of C-plane protocol stack  
Reference of U-plane (other TS/TRs) and supplemental explanation

Annex <C>:  
WebRTC Signalling Protocol Examples

Editor’s Note:   
Expected signalling regulation examples (Async API)

Annex <D>:  
WebRTC Signalling Flow Examples

Editor’s Note:   
Sequence and message examples using Annex C

Annex <E>:  
Conference Management Protocol Examples

Editor’s Note:   
Examples of conference session management (OpenAPI)

Annex <F>:  
Conference Management Flow Examples

Editor’s Note:   
Sequence and message examples using Annex E