

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: December 11, 2021

S. Murillo  
A. Gouaillard  
CoSMo Software  
June 09, 2021

WebRTC-HTTP ingestion protocol (WHIP)  
draft-murillo-whip-02

Abstract

While WebRTC has been very successful in a wide range of scenarios, its adoption in the broadcasting/streaming industry is lagging behind. Currently there is no standard protocol (like SIP or RTSP) designed for ingesting media in a streaming service, and content providers still rely heavily on protocols like RTMP for it.

These protocols are much older than webrtc and lack by default some important security and resilience features provided by webrtc with minimal delay.

The media codecs used in older protocols do not always match those being used in WebRTC, mandating transcoding on the ingest node, introducing delay and degrading media quality. This transcoding step is always present in traditional streaming to support e.g. ABR, and comes at no cost. However webrtc implements client-side ABR, also called Network-Aware Encoding by e.g. Huavision, by means of simulcast and SVC codecs, which otherwise alleviate the need for server-side transcoding. Content protection and Privacy Enhancement can be achieved with End-to-End Encryption, which preclude any server-side media processing.

This document proposes a simple HTTP based protocol that will allow WebRTC endpoints to ingest content into streaming services and/or CDNs to fill this gap and facilitate deployment.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 11, 2021.

#### Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. Overview . . . . .	4
4. Protocol Operation . . . . .	5
4.1. ICE and NAT support . . . . .	6
4.2. Webrtc constraints . . . . .	7
4.3. Load balancing and redirections . . . . .	7
4.4. STUN/TURN server configuration . . . . .	7
4.5. Authentication and authorization . . . . .	8
4.6. Simulcast and scalable video coding . . . . .	8
4.7. Protocol extensions . . . . .	8
5. Security Considerations . . . . .	9
6. IANA Considerations . . . . .	9
7. Acknowledgements . . . . .	9
8. Normative References . . . . .	9
Authors' Addresses . . . . .	10

#### 1. Introduction

WebRTC intentionally does not specify a signaling transport protocol at application level, while RTCWEB standardized the signalling protocol itself (JSEP, SDP O/A) and everything that was going over the wire (media, codec, encryption, ...). This flexibility has allowed for implementing a wide range of services. However, those services are typically standalone silos which don't require

interoperability with other services or leverage the existence of tools that can communicate with them.

In the broadcasting/streaming world, the usage of hardware encoders that would make it very simple to plug in (SDI) cables carrying raw media, encoding it in place, and pushing it to any streaming service or CDN ingest is ubiquitous. Having to implement a custom signalling transport protocol for each different webrtc services has hindered adoption.

While some standard signalling protocols are available that can be integrated with WebRTC, like SIP or XMPP, they are not designed to be used in broadcasting/streaming services, and there also is no sign of adoption in that industry. RTSP, which is based on RTP and maybe the closest in terms of features to webrtc, is not compatible with WebRTC SDP offer/answer model.

In the specific case of ingest into a platform, some assumption can be made about the server-side which simplifies the webrtc compliance burden, as detailed in webrtc-gateway document [I-D.draft-alvestrand-rtcweb-gateways].

This document proposes a simple protocol for supporting WebRTC as ingest method which is:

- o Easy to implement,
- o As easy to use as current RTMP URIs.
- o Fully compliant with Webrtc and RTCWEB specs.
- o Allow for both ingest in traditional media platforms for extension and ingest in webrtc end-to-end platform for lowest possible latency.
- o Lowers the requirements on both hardware encoders and broadcasting services to support webrtc.
- o Usable both in web browsers and in native encoders.

## 2. Terminology

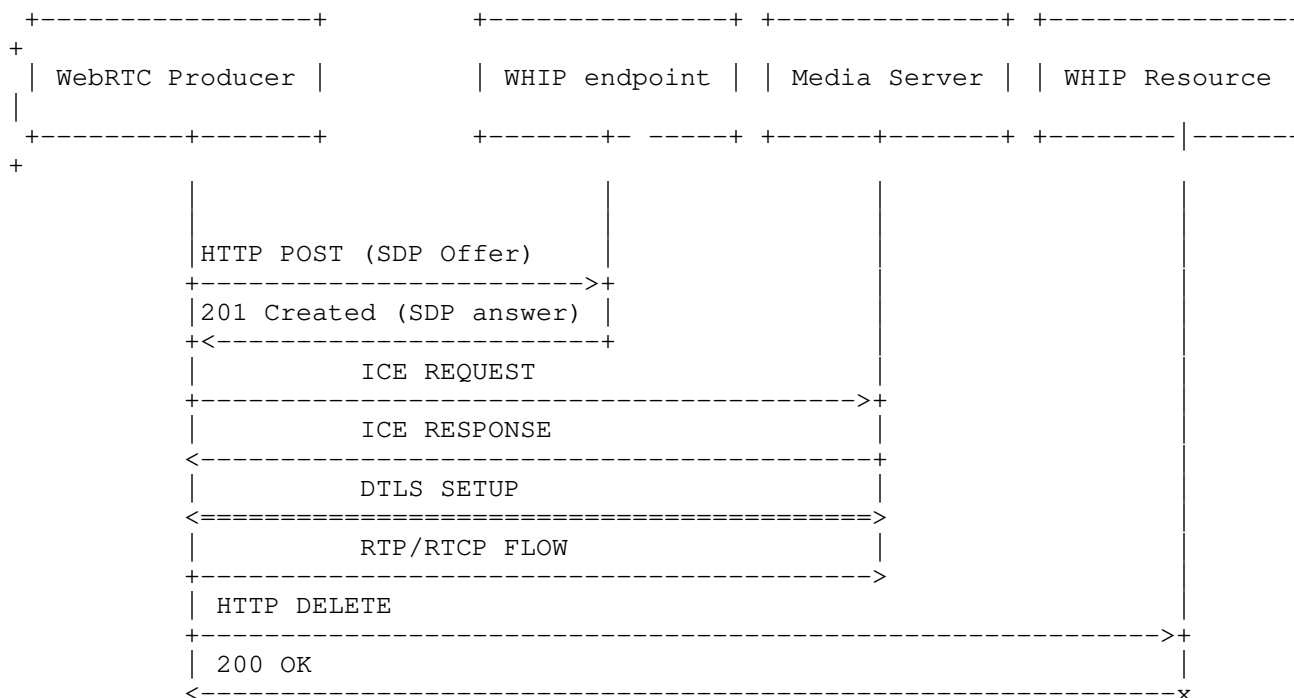
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

- o WHIP client: WebRTC Media encoder or producer that acts as client on the WHIP protocol and encodes and delivers the media to a remote media server.
- o WHIP endpoint: Ingest server receiving the initial WHIP request.
- o WHIP endpoint URL: URL of the WHIP endpoint that will create the WHIP resource
- o Media Server: WebRTC media server that establishes the media session with the WHIP client and receives the media produced by it.
- o WHIP resource: Allocated resource by the WHIP endpoint for an ongoing ingest session that the WHIP client can send request for altering the session (ICE operations or termination, for example).
- o WHIP resource URL: URL allocated to a specific media session by the WHIP endpoint which can be used to perform operations such terminating the session or ICE restarts.

### 3. Overview

The WebRTC-HTTP ingest protocol (WHIP) uses an HTTP POST request to perform a single shot SDP offer/answer so an ICE/DTLS session can be established between the encoder/media producer and the broadcasting ingestion endpoint.

Once the ICE/DTLS session is set up, the media will flow unidirectionally from the encoder/media producer to the broadcasting ingestion endpoint. In order to reduce complexity, no SDP renegotiation is supported, so no tracks or streams can be added or removed once the initial SDP O/A over HTTP is completed.



#### WHIP session setup and teardown

#### 4. Protocol Operation

In order to setup an ingestion session, the WHIP client will generate an SDP offer according to the JSEP rules and do an HTTP POST request to the WHIP endpoint configured URL.

The HTTP POST request will have a content type of application/sdp and contain the SDP offer as body. The WHIP endpoint will generate an SDP answer and return it on a 201 Accepted response with content type of application/sdp and the SDP answer as body and a Location header pointing to the newly created resource.

SDP offer SHOULD use the sendonly attribute and the SDP answer MUST use the recvonly attribute.

Once a session is setup ICE consent freshness [RFC7675] will be used to detect abrupt disconnection and DTLS teardown for session termination by either side.

To explicitly terminate the session, the WHIP client MUST perform an HTTP DELETE request to the resource url returned on the Location header of the initial HTTP POST. Upon receiving the HTTP DELETE

request, the WHIP resource will be removed and the resources freed on the media server, terminating the ICE and DTLS sessions.

A media server terminating a session MUST follow the procedures in [RFC7675] section 5.2 for immediate revocation of consent.

The WHIP endpoints MUST return an HTTP 405 response for any HTTP GET, HEAD or PUT requests on the resource URL in order to reserve its usage for future versions of this protocol specification.

The WHIP resources MUST return an HTTP 405 response for any HTTP GET, HEAD, POST or PUT requests on the resource URL in order to reserve its usage for future versions of this protocol specification.

#### 4.1. ICE and NAT support

In order to simplify the protocol, there is no support for exchanging gathered trickle candidates from media server ICE candidates once the SDP answer is sent. So in order to support the WHIP client behind NAT, the WHIP media server SHOULD be publicly accessible.

The initial offer by the WHIP client MAY be sent after the full ICE gathering is complete containing the full list of ICE candidates, or only contain local candidates or even an empty list of candidates.

The WHIP endpoint SDP answer SHALL contain the full list of ICE candidates publicly accessible of the media server. The media server MAY use ICE lite, while the WHIP client MUST implement full ICE.

The WHIP client MAY perform trickle ICE or an ICE restarts [RFC8863] by sending a HTTP PATCH request to the WHIP resource URL with a body containing a SDP fragment with mime type "application/trickle-ice-sdpfrag" as specified in [RFC8840] with the new ice candidate or ice ufrag/pwd for ice restarts. A WHIP resource MAY not support either trickle ICE (i.e. ICE lite media servers) or ICE restart, and it MUST return a 405 Method Not Allowed for any HTTP PATCH request.

A WHIP client receiving a 405 response for an HTTP PATCH request SHALL not send further request for ICE trickle or restart. If the WHIP client gathers additional candidates (via STUN/TURN) after the SDP offer is sent, it MUST send STUN request to the ICE candidates received from the media server as per [RFC8838] regardless if the HTTP PATCH is supported by either the WHIP client or the WHIP resource.

#### 4.2. Webrtc constraints

In order to reduce the complexity of implementing WHIP in both clients and media servers, some restrictions regarding WebRTC usage are made.

SDP bundle SHALL be used by both the WHIP client and the media server. The SDP offer created by the WHIP client MUST include the bundle-only attribute in all m-lines as per [RFC8843]. Also, RTCP muxing SHALL be supported by both the WHIP client and the media server.

Unlike [RFC5763] a WHIP client MAY use a setup attribute value of setup:active in the SDP offer, in which case the WHIP endpoint MUST use a setup attribute value of setup:passive in the SDP answer.

#### 4.3. Load balancing and redirections

WHIP endpoints and media servers MAY not be colocated on the same server so it is possible to load balance incoming requests to different media servers. WHIP clients SHALL support HTTP redirection via 307 Temporary Redirect response code.

In case of high load, the WHIP endpoints may return a 503 (Service Unavailable) status code indicating that the server is currently unable to handle the request due to a temporary overload or scheduled maintenance, which will likely be alleviated after some delay.

The WHIP endpoint MAY send a Retry-After header field indicating the minimum time that the user agent is asked to wait before issuing the redirected request.

#### 4.4. STUN/TURN server configuration

Configuration of the TURN or STUN servers used by the WHIP client is out of the scope of this document.

It is RECOMMENDED that broadcasting server provides an HTTP interface for provisioning the TURN/STUN servers url and short term credentials as in [I-D.draft-uberti-behave-turn-rest-00]. Note that the authentication information or the url of this API are not related to the WHIP endpoint URLs or authentication.

It could also be possible to configure the STUN/TURN server URLs and long term credentials provided by the either broadcasting service or an external TURN provider.

#### 4.5. Authentication and authorization

Authentication and authorization is supported by the Authorization HTTP header with a bearer token as per [RFC6750].

#### 4.6. Simulcast and scalable video coding

Both simulcast and scalable video coding (including K-SVC modes) MAY be supported by both media servers and WHIP clients and negotiated in the SDP O/A.

If the client supports simulcast and wants to enable it for publishing, it MUST negotiate the support in the SDP offer according to the procedures in [RFC8853] section 5.3. A server accepting a simulcast offer MUST create an answer according to the procedures [RFC8853] section 5.3.2.

#### 4.7. Protocol extensions

In order to support future extensions to be defined for the WHIP protocol, a common procedure for registering and announcing the new extensions is defined.

Protocol extensions supported by the WHIP server MUST be advertised to the WHIP client on the 201 created response to initial HTTP POST request to the WHIP endpoint by inserting one Link header for each extension with the extension "rel" type attribute and the uri for the HTTP resource that will be available for receiving request related to that extension.

Protocol extensions are optional for both WHIP clients and servers. WHIP clients MUST ignore any Link attribute with an unknown "rel" attribute value and WHIP servers MUST not require the usage of any of the extensions.

Each protocol extension MUST register a unique "rel" attribute values at IANA starting with the prefix: "urn:ietf:params:whip:".

For example, taking a potential extension of server to client communication using server sent events as specified in <https://html.spec.whatwg.org/multipage/server-sent-events.html#server-sent-events>, the url for connecting to the server side event resource for the published stream will be returned in the initial HTTP "201 Created" response with a "Link" header and a "rel" attribute of "urn:ietf:params:whip:server-sent-events".

The HTTP 201 response to the HTTP POST request would look like:



"HTTP/1.1 201 Created Content-Type: application/sdp Location:  
https://whip.ietf.org/publications/213786HF Link:  
<https://whip.ietf.org/publications/213786HF/  
sse>;rel="urn:ietf:params:whip:server-side-events " "

## 5. Security Considerations

HTTPS SHALL be used in order to preserve the WebRTC security model.

## 6. IANA Considerations

## 7. Acknowledgements

## 8. Normative References

[I-D.draft-alvestrand-rtcweb-gateways]

Alvestrand, H. and U. Rauschenbach, "WebRTC Gateways",  
draft-alvestrand-rtcweb-gateways-02 (work in progress),  
March 2015.

[I-D.draft-uberti-behave-turn-rest-00]

Uberti, J., "A REST API For Access To TURN Services",  
draft-uberti-behave-turn-rest-00 (work in progress), July  
2013.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119,  
DOI 10.17487/RFC2119, March 1997,  
<<https://www.rfc-editor.org/info/rfc2119>>.

[RFC5763] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework  
for Establishing a Secure Real-time Transport Protocol  
(SRTP) Security Context Using Datagram Transport Layer  
Security (DTLS)", RFC 5763, DOI 10.17487/RFC5763, May  
2010, <<https://www.rfc-editor.org/info/rfc5763>>.

[RFC6750] Jones, M. and D. Hardt, "The OAuth 2.0 Authorization  
Framework: Bearer Token Usage", RFC 6750,  
DOI 10.17487/RFC6750, October 2012,  
<<https://www.rfc-editor.org/info/rfc6750>>.

[RFC7675] Perumal, M., Wing, D., Ravindranath, R., Reddy, T., and M.  
Thomson, "Session Traversal Utilities for NAT (STUN) Usage  
for Consent Freshness", RFC 7675, DOI 10.17487/RFC7675,  
October 2015, <<https://www.rfc-editor.org/info/rfc7675>>.

- [RFC8838] Ivov, E., Uberti, J., and P. Saint-Andre, "Trickle ICE: Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol", RFC 8838, DOI 10.17487/RFC8838, January 2021, <<https://www.rfc-editor.org/info/rfc8838>>.
- [RFC8840] Ivov, E., Stach, T., Marocco, E., and C. Holmberg, "A Session Initiation Protocol (SIP) Usage for Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (Trickle ICE)", RFC 8840, DOI 10.17487/RFC8840, January 2021, <<https://www.rfc-editor.org/info/rfc8840>>.
- [RFC8843] Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", RFC 8843, DOI 10.17487/RFC8843, January 2021, <<https://www.rfc-editor.org/info/rfc8843>>.
- [RFC8853] Burman, B., Westerlund, M., Nandakumar, S., and M. Zanaty, "Using Simulcast in Session Description Protocol (SDP) and RTP Sessions", RFC 8853, DOI 10.17487/RFC8853, January 2021, <<https://www.rfc-editor.org/info/rfc8853>>.
- [RFC8863] Holmberg, C. and J. Uberti, "Interactive Connectivity Establishment Patiently Awaiting Connectivity (ICE PAC)", RFC 8863, DOI 10.17487/RFC8863, January 2021, <<https://www.rfc-editor.org/info/rfc8863>>.

## Authors' Addresses

Sergio Garcia Murillo  
CoSMo Software

Email: [sergio.garcia.murillo@cosmosoftware.io](mailto:sergio.garcia.murillo@cosmosoftware.io)

Alexandre Gouaillard  
CoSMo Software

Email: [alex.gouaillard@cosmosoftware.io](mailto:alex.gouaillard@cosmosoftware.io)

DISPATCH  
Internet-Draft  
Intended status: Standards Track  
Expires: 15 November 2021

Y. K. Muthusamy  
C. Ullrich  
Immersion Corporation  
14 May 2021

The 'haptics' Top-level Media Type  
draft-muthusamy-dispatch-haptics-02

Abstract

This memo serves to register and document the 'haptics' top-level media type, under which subtypes for representation formats for haptics may be registered. This document also serves as a registration application for a set of intended subtypes, which are representative of some existing subtypes already in use.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 15 November 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Terminology . . . . .	3
2. Background and Justification . . . . .	3
2.1. MPEG ISOBMFF . . . . .	3
2.2. Haptic Sub-modalities . . . . .	4
2.3. Another Human Sense . . . . .	4
2.4. Commercial Uptake . . . . .	4
2.5. Haptic Data Formats in Use . . . . .	5
2.6. Haptic Subtypes (envisioned standards) . . . . .	6
2.7. 'application' top-level type not suitable . . . . .	6
3. Security Considerations . . . . .	7
4. IANA Considerations . . . . .	7
4.1. Definition and Encoding . . . . .	8
4.2. Registration Procedure . . . . .	8
4.3. Subtype Registrations . . . . .	8
4.3.1. IVS Haptics Type . . . . .	8
4.3.2. HAPT Haptics Type . . . . .	9
5. Normative References . . . . .	10
6. Informative References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

The term 'haptics' refers to the generation of touch-related sensations in a device or interface. Haptics is widely used in consumer devices in order to provide touch-based feedback to users. The most common use of haptics is in mobile devices, where it is used to provide feedback to users interacting with the touchscreen, e.g., typing on a virtual keyboard. Haptic technologies are unlike audio and visual enabling technologies in the sense that they require some form of actuation in order to create a tactile sensation. For mobile phones and game controllers, these actuators are typically small vibrating motors. For large touchscreens in vehicles, these actuators can be specialized piezoelectric materials. Haptic capabilities are found in nearly every modern smartphone and game and virtual reality controller, making these devices an ideal target for enhanced media experiences.

Internet Media Types [RFC6838] are used to label content carried over Internet protocols. This document defines a new top-level type 'haptics' according to Section 4.2.7 of [RFC6838]. This top-level type indicates that the content specifies haptic data. Under this top-level type, different representation formats of haptics may be registered.

## 1.1. Terminology

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

## 2. Background and Justification

Haptic signals provide an additional layer of entertainment and sensory immersion for the user. Haptic tracks, in separate files, can be combined with audio/video files and played back in sync to provide an overall immersive media experience (audio, visual, tactile) for the user. More recently, haptic tracks embedded in standard file formats such as ISO BMFF (ISO Base Media File Format), enable playback of the haptic signals over one or more actuators, simultaneously with audio and video playback.

### 2.1. MPEG ISO BMFF

Historically, there has not been a registration of formats for haptics. However, haptics has been proposed as a first-order media type (at the same level as audio and video) in ISO BMFF. This proposal was made to the MPEG Systems File Format sub-group in April 2020. The proposal was accepted and has progressed to DIS (Draft International Standard) in October 2020 [ISO BMFF-AMD1][AMD1-Status]. Once it completes its progression through the MPEG standardization stages (expected in October 2021), haptics will become part of the ISO/IEC 14496-12 (ISO BMFF) standard. Given this development, a strong case can be made for haptics to be added to the list of top-level media type recognized by the IETF.

We envision the following designations for haptics in mp4 files, once the top-level type 'haptics' is registered:

1. 'haptics/mp4' - mp4 files with just haptic tracks in them (e.g., streaming games, haptics files for haptic vests, belts, gloves, etc.)
2. 'video/mp4' - mp4 files with video, audio, and haptics (to ensure consistency with existing mp4 files with video content)
3. 'audio/mp4' - mp4 files with audio and haptics (to ensure consistency with existing mp4 files with audio content without any video)

## 2.2. Haptic Sub-modalities

There are multiple sub-modalities of haptics:

- \* Vibrotactile (touch, vibration)
- \* Kinesthetic (force feedback)
- \* Surface (surface friction)
- \* Spatial, non-contact (ultrasound)
- \* Thermal (temperature)

Therefore, designating 'haptics' as a top-level media type would enable the definition of data formats pertaining to these sub-modalities in a more streamlined manner. This would not be possible if 'haptics' were to be placed under other top-level types like 'audio', 'video', or 'application'.

## 2.3. Another Human Sense

The top-level media type 'audio' pertains to the human sense of hearing, the top-level media type 'video' pertains to the human sense of seeing, so it only makes sense for the (equally important) human sense of touch to be represented by another top-level media type 'haptics'. Placing 'haptics' under 'audio' or 'video' is not reflective of the kinds of files or use cases that would need haptics but have nothing whatsoever to do with audio or video.

## 2.4. Commercial Uptake

Haptics is rapidly becoming a standard feature of consumer electronic devices. For example:

- \* iPhone (191+ million units sold in 2019): native support for haptic encoded data
- \* Android (1.18+ billion units sold in 2019): API support of haptic buffers
- \* W3C (HTML vibration API): Optionally supported in mobile web browsers
- \* Game consoles (39+ million units sold in 2019): MS Xbox, Sony PlayStation, Nintendo Switch, etc.
- \* XR devices (9+ million units sold in 2019): OpenXR haptic API

Haptic media is expected to be commonly exchanged between these devices. Since they represent the majority of CE devices, a strong case can be made for 'haptics' as a top-level media type.

## 2.5. Haptic Data Formats in Use

There are multiple instances of existing haptic data formats that would live as sub-types under the proposed 'haptics' top-level media type. While these subtypes have *not* been registered with IANA or standardized (yet), the prevalence of these haptic data formats in a large number of devices around the world, pre-dating the standardization of haptic tracks in ISO/BMFF, provides a compelling argument for 'haptics' to be designated as a top-level media type:

- \* 'ahap': The AHAP haptic data format [AHAP] is currently the standard encoding on all iOS devices + iOS connected game peripherals. The format has seen usage and adoption beyond Apple devices as well, with decoders available for Android and other XR systems.
- \* 'ogg': Google has introduced a proprietary extension to the OGG format in the latest version of Android 11. This encoding enables haptic media to be stored in OGG files.
- \* 'ivs': The IVS haptic data format is currently a vendor-specific format that is in use:
  - In mobile phones from LG Electronics (specifically, the models V30, V40, and the newest V50) that are sold worldwide
  - In gaming phones from ASUS (specifically, models ROG, ROG Phone II, ROG Phone 3) that are sold worldwide
- \* 'hapt': The HAPT haptic data format is currently a vendor-specific format that is in use:
  - In mobile haptic advertising (for W3C devices)
  - The following Japanese game developers use the HAPT format as part of Immersion's TouchSense SDK:
    - o KLAB: <https://www.klab.com/en/> (<https://www.klab.com/en/>)
    - o Craft&Meister: [http://www.crafts-meister.co.jp/pc/company\\_en.html](http://www.crafts-meister.co.jp/pc/company_en.html) ([http://www.crafts-meister.co.jp/pc/company\\_en.html](http://www.crafts-meister.co.jp/pc/company_en.html))

- Tencent is using the TouchSense SDK for their popular social media application QQ and live streaming application NOW: Immersion-Announces-Tencent-Licenses-TouchSense-Technology-Deliver (<https://www.businesswire.com/news/home/20171026006443/en/Immersion-Announces-Tencent-Licenses-TouchSense%C2%AE-Technology-Deliver>)

Given the widespread use of these subtypes, it makes sense for 'haptics' to be a top-level media type.

## 2.6. Haptic Subtypes (envisioned standards)

The MPEG ISOBMFF proposal included an informative annex of known haptic coding formats with proposed FourCC codes for them. These codes are not registered yet, but the plan is indeed to standardize these haptic coding formats in the near future. Once standardized, they will also live as subtypes under the proposed 'haptics' top-level media type:

- \* 'hmpg': the selected coding format from the MPEG Call for Proposals on the Coded Representation of Haptics [MPEG-Haptics-CfP]
- \* 'hiee': IEEE P1918.1.1 vibrotactile coding standard being developed under the IEEE Tactile Internet initiative as part of the 5G URLL profile.
- \* 'henm': enumerated effects haptic coding format (based on MIDI)
- \* 'havic': audio-to-vibe haptic coding format (automatic audio to vibration conversion algorithms)

## 2.7. 'application' top-level type not suitable

From the above arguments, it is clear that haptics does not really belong under any other media type. To reiterate, there are three main reasons why the 'haptics' media type does not fit under the 'application' top-level type:

- \* haptics connects to a sensory system, touch/motion, directly, and is more specific than the abstract 'application' type, and
- \* 'application' has historically been used for applications, i.e., code, which means it is viewed and treated with great care for security. 'haptics' is not code, just as 'audio' and 'video' are not code either.



- \* haptics is a property of a media stream, it is not an application under any normal definition. As such, it should be its own type.

### 3. Security Considerations

Haptics are interpreted data structures that represent collections of different media rendering instructions intended to be decoded and rendered on target device hardware. Haptic data can be represented as collections of signal data and/or descriptive text in XML/JSON or similar format. Signal data is typically not executed by endpoint processors and represents minimal security risk. Descriptive text is typically parsed and represented in memory using standard XML data structures. This data is utilized to construct one or more signals that are sent to the endpoint device hardware.

Because of the media/rendering nature of the data path for haptic coded data the security profile of haptic data is expected to be largely consistent with the security profile of visual and audio media data.

As with any synthesized media data (audio, video, and haptics), there is a security risk associated with execution of commands based on the descriptive encoding either through its inherent extensibility or through the insertion of arbitrary executable data in the descriptive format itself. Indeed, media rendering systems are normally implemented with a mix of user and kernel space execution since these media must ultimately make their way to a hardware system. In theory, malicious instructions present in descriptive haptic media have the potential to execute arbitrary code in kernel space, effectively bypassing system permissions structures and/or execution sandboxes.

Haptics, audio, and video media have widespread use and careful attention should be paid by operating system and device driver implementors to ensure that synthesis and rendering signal paths do not provide attack surfaces for malicious payloads. Ultimately, any coded representation of haptic media is insufficient to implicitly provide sufficient security and this protection should be enforced by the operating system implementor.

### 4. IANA Considerations

This specification registers a new top-level type, 'haptics', in the standards tree, adds it as an alternative value of "Type Name" in the media types registration form [Media-Type-Registration], and registers several subtypes for it.

#### 4.1. Definition and Encoding

'haptics' as the primary media content type indicates that the content identified by it requires a certain haptics subsystem such as low-level haptics APIs, which in turn will require hardware capabilities such as one or more actuators to render the haptics media. The 'haptics' media type does not provide any specific information about the underlying data format and how the haptics information should be interpreted -- the subtypes defined within a 'haptics' tree name the specific haptic formats. Unrecognized subtypes of 'haptics' should be treated as 'application/octet-stream'. Implementations may still pass unrecognized subtypes to the haptics subsystem and associated rendering hardware.

#### 4.2. Registration Procedure

New haptics formats should be registered using the online form [Media-Type-Registration]. [RFC6838] should be consulted on registration procedures. In particular, the haptics specification should preferably be freely available.

Note that new parameter sub-values may be defined in the future. If an implementation does not recognize a sub-value in the comma-separated list, it should ignore the sub-value and continue processing the other sub-values in the list.

#### 4.3. Subtype Registrations

In this section, the initial entries under the top-level 'haptics' media type are specified. They also serve as examples for future registrations.

##### 4.3.1. IVS Haptics Type

Type name: haptics

Subtype name: ivs

Required parameters: None

Optional parameters: None

Encoding considerations: Text/binary

Interoperability considerations: The IVS format is a device-independent haptic effect coding. It is designed to enable interoperability between distinct physical endpoints. Not all devices may be able to render all effects present in an IVS file.

Published specification: ISO/IEC JTC 1/SC 29/WG 2 N 72 "Encoder Input Format for Haptics" being developed by ISO/IEC JTC1/SC29 WG 2 [MPEG-Haptics-Encoder-Format].

Applications that use this media type: All applications that are able to create, edit, or display haptic media content.

Additional information:

- \* File extension(s): Haptic file extensions used for IVS files: .ivs (xml) and .ivt (binary)
- \* Macintosh file type code(s): (no code specified)
- \* Macintosh Universal Type Identifier code: None
- \* Fragment Identifier: None
- \* Deprecated Alias: None

Person & email address to contact for further information: Yeshwant Muthusamy (ymuthusamy@immersion.com)

Change controller: Immersion Corporation

#### 4.3.2. HAPT Haptics Type

Type name: haptics

Subtype name: hapt

Required parameters: None

Optional parameters: None

Encoding considerations: Text/binary

Interoperability considerations: The HAPT format is a device-dependent haptic effect coding based on the RIFF coding standard. It is designed to enable efficient coding of a device-specific haptic effect.

Published specification: HAPT is a logical extension of the RIFF standard [RIFF]

Applications that use this media type: All applications that are able to create, edit, or display haptic media content.

## Additional information:

- \* File extension(s): Haptic file extensions used for HAPT files:  
.hapt
- \* Macintosh file type code(s): (no code specified)
- \* Macintosh Universal Type Identifier code: None
- \* Fragment Identifier: None
- \* Deprecated Alias: None

Person & email address to contact for further information: Yeshwant Muthusamy (ymuthusamy@immersion.com)

Change controller: Immersion Corporation

## 5. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/info/rfc6838>>.

## 6. Informative References

- [ISOBMFF-AMD1] "ISO/IEC 14496-12/CD AMD 1 Information technology - Coding of audio-visual objects - Part 12: ISO base media file format - Amendment 1: Support for new media types (haptics, volumetric visual) and other improvements", <<https://www.iso.org/standard/81604.html>>.
- [AMD1-Status] "Timeline of ISO/IEC 14496-12/CD Amd 1, Edition 6 - Project ID: 81604, ISO/IEC JTC 1/SC 29/WG 3", <<https://sd.iso.org/projects/project/81604/overview>>.
- [AHAP] "Apple Haptic Audio Pattern", <[https://developer.apple.com/documentation/corehaptics/representing\\_haptic\\_patterns\\_in\\_ahap\\_files](https://developer.apple.com/documentation/corehaptics/representing_haptic_patterns_in_ahap_files)>.

[MPEG-Haptics-CfP]  
"Call for Proposals on the Coded Representation of Haptics  
- Phase 1", <[https://www.mpegstandards.org/standards/  
Explorations/40/](https://www.mpegstandards.org/standards/Explorations/40/)>.

[MPEG-Haptics-Encoder-Format]  
"Encoder Input Format for Haptics",  
<[https://www.mpegstandards.org/standards/  
Explorations/40/](https://www.mpegstandards.org/standards/Explorations/40/)>.

[Media-Type-Registration]  
"IANA, Application for a Media Type",  
<<http://www.iana.org/form/media-types>>.

[RIFF] "Resource Interchange File Format",  
<[https://www.loc.gov/preservation/digital/formats/fdd/  
fdd000025.shtml](https://www.loc.gov/preservation/digital/formats/fdd/fdd000025.shtml)>.

## Authors' Addresses

Yeshwant K. Muthusamy  
Immersion Corporation  
330 Townsend St. Suite 234  
San Francisco, CA 94107  
United States of America

Phone: +1 469-583-2171  
Email: [ymuthusamy@immersion.com](mailto:ymuthusamy@immersion.com)

Chris Ullrich  
Immersion Corporation  
330 Townsend St. Suite 234  
San Francisco, CA 94107  
United States of America

Phone: +1 805-320-0774  
Email: [cullrich@immersion.com](mailto:cullrich@immersion.com)