Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 13, 2021

S. Murillo
A. Gouaillard
CoSMo Software
September 09, 2020

WebRTC-HTTP ingestion protocol (WHIP)
draft-murillo-whip-00

Abstract

While WebRTC has been very sucessfull in a wide range of scenarios, its adption 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 traditionnal 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 achieve with End-to-End Encryption, which preclude any server-side media processing.

This document proposes a simple HTTP based protocol that will allow WebRTC endpoings to ingest content into streaming servics 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 March 13, 2021.

Table of Contents

1. Introduction

WebRTC intentionaly 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 detailled in webrtc-gateway document. https://tools.ietf.org/html/draft-ietf-rtcweb-gateways-02

This document proposse a simple protocol for supporting WebRTC as ingest method which is: - Easy to implement, - As easy to use as current RTMP URI. - Fully compliant with Webrtc and RTCWEB specs. - Allow for both ingest in traditionnal media platforms for extention and ingest in webrtc end-to-end platform for lowest possible latency. - Lowers the requirements on both hardware encoders and broadcasting services to support webrtc. - Usable both in web browsers and in native encoders.

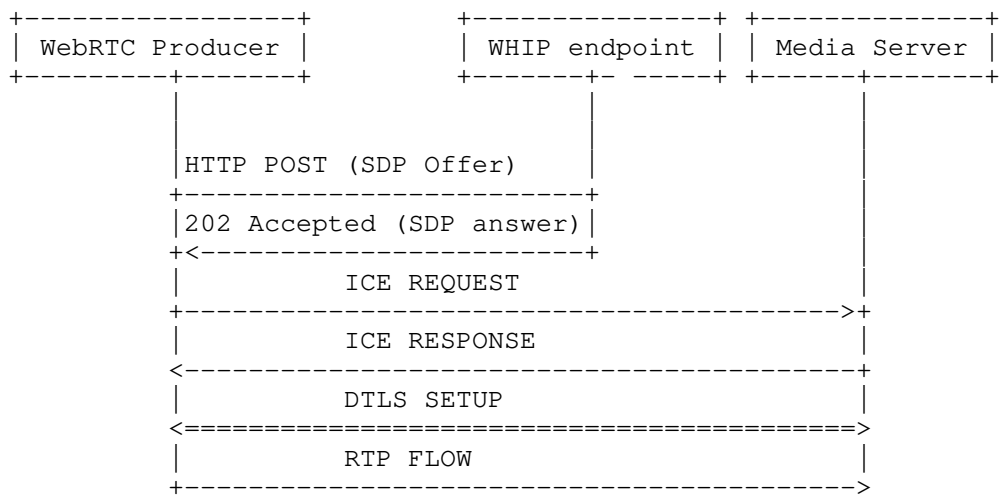
2. Terminology

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].

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 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

4. Protocol Operation

In order to setup an ingestion session, the WebRTC encoder/media producer will generate an SDP offer according the 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 ingestion endpoint will generate an SDP answer and return it on a 202 Accepted response with content type of application/sdp and the SDP answer as body.

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

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

4.1. ICE and NAT support

In order to simplify the protocol, there is no support of exchanging gathered tickle ICE candidates one the SDP offer or answer is sent. So in order to support encoders/media producers behind NAT, the WHIP media server MUST be publicly accessible.

The initial offer by the encoder/media producer 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 encoder/media producer MUST implement full ICE.

If the Encoder/Media producer gathers additional candidates (via STUN/TURN) after the SDP offer is sent, it will send directly a STUN request to the ICE candidates received from the media server as per [I-D.draft-ietf-ice-trickle-21].

4.2. Webrtc contrains

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

SDP bundle SHALL be used by both the encoder/media producer and the media server. The SDP offer created by the encoder/media producer MUST include the bundle-only attribute in all m-lines as per [I-D.draft-ietf-mmusic-sdp-bundle-negotiation-54]. Also, RTCP muxing SHALL be supported by the both the encoder/media producer and the media server.

4.3. Load balancing and redirections

Encoders/media MAY not be colocated on the same server so it is possible to load balance incoming request to different media server. Encoders/media producers 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 server 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. Authentication and authorization

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

4.5. Simulcast and scalable video coding

Both simulcast and scalable video coding (including K-SVC modes) MAY be supported by both media servers and encoders/media producers.

5. Security Considerations

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

6. IANA Considerations

7. Acknowledgements

8. Normative References

[I-D.draft-ietf-ice-trickle-21]
Ivov, E., Rescorla, E., Uberti, J., and P. Saint-Andre, "Trickle ICE: Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol", draft-ietf-ice-trickle-21 (work in progress), April 2018.

[I-D.draft-ietf-mmusic-sdp-bundle-negotiation-54]
Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-54 (work in progress), December 2018.

[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>.

[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>.

Authors' Addresses

Sergio Garcia Murillo
CoSMo Software

Email: sergio.garcia.murillo@cosmosoftware.io

Alexandre Gouaillard
CoSMo Software

Email: alex.gouaillard@cosmosoftware.io

DISPATCH
Internet-Draft
Intended status: Standards Track
Expires: 19 May 2021

Y. K. Muthusamy
C. Ullrich
Immersion Corporation

15 November 2020

The 'haptics' Top-level Media Type
draft-muthusamy-dispatch-haptics-01

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 19 May 2021.

Table of Contents

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 ISOBMFF (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 ISOBMFF

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 ISOBMFF. 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 [ISOBMFF-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 (ISOBMFF) 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 ISOBMFF, 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)

- 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)

* 'havc': 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 13 "Encoder Input Format for Haptics" being developed by ISO/IEC JTC1/SC29 WG 2.

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>.

[MPEG-Haptics-CfP]
"Draft Call for Proposals on the Coded Representation of Haptics",
<https://www.mpegstandards.org/meetings/mpeg-132/>.

[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>.

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

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