



**ATSC**

ADVANCED TELEVISION  
SYSTEMS COMMITTEE

# **ATSC Recommended Practice: Haptics for ATSC 3.0**

---

Doc. A/380:2021  
3 February 2021

**Advanced Television Systems Committee**  
1776 K Street, N.W.  
Washington, D.C. 20006  
202-872-9160

The Advanced Television Systems Committee, Inc., is an international, non-profit organization developing voluntary standards and recommended practices for digital television. ATSC member organizations represent the broadcast, broadcast equipment, motion picture, consumer electronics, computer, cable, satellite, and semiconductor industries. ATSC also develops digital television implementation strategies and supports educational activities on ATSC standards. ATSC was formed in 1983 by the member organizations of the Joint Committee on Inter-society Coordination (JCIC): the Electronic Industries Association (EIA), the Institute of Electrical and Electronic Engineers (IEEE), the National Association of Broadcasters (NAB), the National Cable Telecommunications Association (NCTA), and the Society of Motion Picture and Television Engineers (SMPTE). For more information visit [www.atsc.org](http://www.atsc.org).

---

*Note:* The user's attention is called to the possibility that compliance with this document may require use of an invention covered by patent rights. By publication of this document, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. One or more patent holders have, however, filed a statement regarding the terms on which such patent holder(s) may be willing to grant a license under these rights to individuals or entities desiring to obtain such a license. Details may be obtained from the ATSC Secretary and the patent holder.

---

Implementers with feedback, comments, or potential bug reports relating to this document may contact ATSC at <https://www.atsc.org/feedback/>.

### Revision History

Version	Date
Recommended Practice approved	3 February 2021

## Table of Contents

<b>1.</b>	<b>SCOPE .....</b>	<b>5</b>
	1.1 Introduction and Background	5
	1.2 Organization	6
<b>2.</b>	<b>REFERENCES .....</b>	<b>6</b>
	2.1 Informative References	6
<b>3.</b>	<b>DEFINITION OF TERMS .....</b>	<b>7</b>
	3.1 Compliance Notation	7
	3.2 Treatment of Syntactic Elements	7
	3.2.1 Reserved Elements	7
	3.3 Acronyms and Abbreviations	7
	3.4 Terms	8
<b>4.</b>	<b>OVERVIEW OF HAPTICS IN ATSC3.0.....</b>	<b>8</b>
	4.1 High-level Haptics Workflow in ATSC 3.0	8
	4.2 Scenarios Handled in this RP	10
<b>5.</b>	<b>DETAILED WORKFLOW: HAPTIC CONTENT PART OF THE EVENT STREAM.....</b>	<b>12</b>
	5.1 Retrieving Haptic Content from the Cloud Repository	12
	5.2 Broadcast Delivery of Haptic Content	13
	5.3 Broadband Delivery of Haptic Content	13
	5.4 Primary Device to Companion Device Interface	14
	5.5 Synchronization of Haptics with A/V Content	14
<b>6.</b>	<b>DETAILED WORKFLOW: HAPTIC CONTENT IN A SEPARATE FILE.....</b>	<b>14</b>
	6.1 Retrieving the Haptic File URL From the Cloud Repository	14
	6.2 Broadcast Delivery of Haptic File URL and Token	15
	6.3 Broadband Delivery of Haptic File URL and Token	15
	6.4 Primary Device to Companion Device Interface	15
	6.5 Synchronization of Haptics with Audio/Video Content	16
<b>7.</b>	<b>CUSTOMIZATION OF THE HAPTIC EXPERIENCE ON MOBILE DEVICES.....</b>	<b>16</b>
<b>8.</b>	<b>DELIVERY OF HAPTICS-ENABLED ADVERTISEMENTS TO MOBILE DEVICES .....</b>	<b>17</b>
	<b>ANNEX A USAGE SCENARIOS .....</b>	<b>19</b>

## Index of Figures and Tables

<b>Figure 4.1</b> High-level haptics workflow—mobile devices as primary devices.	9
<b>Figure 4.2</b> High-level haptics workflow—mobile devices as companion devices.	9
<b>Figure 4.3:</b> Haptic track is part of the event itself.	11
<b>Figure 4.4:</b> Haptic track is in a separate file.	12
<b>Figure 7.1</b> Portion of Figure 4.4 showing the addition of customization parameters into the content request.	17
<b>Figure 8.1</b> MAGNA-IPG Media Labs immersion study on effectiveness of haptics-enabled mobile ads.	18

## ATSC Recommended Practice: Haptics for ATSC 3.0

### 1. SCOPE

This document provides recommended practices for adding haptics functionality to ATSC 3.0 broadcast (and broadband) content when viewed on mobile devices with built-in haptic actuators, such as smartphones and tablets.

#### 1.1 Introduction and Background

The term “haptics” refers to the generation of touch-related sensations in a device or interface. Haptics is widely used in consumer devices 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 to create a tactile sensation. For mobile phones and game controllers, these actuators are typically small vibrating motors built into the device. 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.

One of the promises of ATSC 3.0 is the ability to view broadcast content on mobile devices (A/300 [2]) when they are used either as primary devices (i.e., have an ATSC 3.0 tuner built-in) or as companion devices (A/338 [4]). At the time of writing, most mobile devices in the market have just one haptic actuator, but new devices with multiple actuators are beginning to show up. The key idea: if these haptic actuators can be utilized in synchronization with the audio-visual content, then the viewing experience of ATSC 3.0 content on these mobile devices would be significantly enhanced and new use cases enabled. Some common examples of the types of content that could benefit from the haptic signals:

- Live sports (major league and professional sports) – viewers can feel the game events (such as hits, catches, etc.), in addition to watching and listening to them, enhancing their overall immersive experience of the game.
- Action films/shows – a haptic track can increase the immediacy and impact of explosions, car chases, etc., drawing the viewer into the action.
- Advertisements – studies have shown that haptics-enabled mobile ads improve brand favorability and are more effective in driving purchase intent among viewers (see Section 8).

Ideally, haptics would be part of the ATSC 3.0 broadcast stream. However, this requires standardization of haptic tracks in the DASH/ROUTE and MPEG MMT standards that are currently used as ATSC 3.0 transport protocols – a multi-year process. Until that standardization is complete, a more expedient option (that is the subject of this RP) would be to either include the haptic content as part of the event stream or retrieve a separate content-specific and device-specific haptic file from a cloud (or other) repository and play it in synchronization with the audio/video content in the broadcast stream.

The rest of this document will be devoted to the following details pertaining to this expedient option. Specifically:

- Details of handshaking between the mobile device and the haptics repository

- How to customize the haptics experience by user preference and device type
- Delivery of haptics-enabled advertisements to mobile devices

## 1.2 Organization

The document is organized as follows:

- Section 1 – Outlines the scope of this document and provides a general introduction
- Section 2 – Lists references and applicable documents
- Section 3 – Provides definitions of terms, acronyms, and abbreviations for this document
- Section 4 – Overview of haptics in ATSC 3.0
- Section 5 – Addresses the detailed workflow when haptic content is part of the event stream
- Section 6 – Addresses the detailed workflow when haptic content is in separate files
- Section 7 – Addresses customization of haptics experience by user and device type
- Section 8 – Addresses delivery of haptics-enabled advertisements to mobile devices
- Annex A – Describes three usage scenarios

## 2. REFERENCES

All referenced documents are subject to revision. Users of this Standard are cautioned that newer editions might or might not be compatible.

### 2.1 Informative References

The following documents contain information that may be helpful in applying this Standard.

- [1] IEEE: “Use of the International Systems of Units (SI): The Modern Metric System,” Doc. SI 10, Institute of Electrical and Electronics Engineers, New York, N.Y.
- [2] ATSC: “ATSC Standard: ATSC 3.0 System,” Doc. A/300:2020, Advanced Television Systems Committee, Washington, DC, 15 May 2020.
- [3] ATSC: “ATSC Standard: Application Signaling,” Doc. A/337:2019, Advanced Television Systems Committee, Washington, DC, 30 April 2019.
- [4] ATSC: “ATSC Standard: Companion Device, with Amendment No. 1 (A/338),” Doc. A/338:2019, Advanced Television Systems Committee, Washington, DC, 21 January 2020.
- [5] ATSC: “ATSC Standard: ATSC 3.0 Interactive Content, with Corrigendum No. 1 and Amendment No. 1”, Doc. A/344:2020, Advanced Television Systems Committee, Washington, DC, 12 October 2020.
- [6] DASH IF: “Guidelines for Implementation: DASH-IF Interoperability Points for ATSC 3.0, Version 1.1,” DASH Interoperability Forum, June 12, 2018. <https://dashif.org/docs/DASH-IF-IOP-for-ATSC3-0-v1.1.pdf>
- [7] ISO/IEC: “Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats,” Doc. ISO/IEC FDIS 23009-1:2019 (E), International Organization for Standardization/International Electrotechnical Commission, Geneva, Switzerland.
- [8] Apple: Apple Haptic and Audio Pattern (AHAP) file format: [https://developer.apple.com/documentation/corehaptics/representing\\_haptic\\_patterns\\_in\\_aha\\_p\\_files](https://developer.apple.com/documentation/corehaptics/representing_haptic_patterns_in_aha_p_files)

[9] Weddle, Amaya & Yu, Hua. (2013). “How does audio-haptic enhancement influence emotional response to mobile media?” 2013 5th International Workshop on Quality of Multimedia Experience, QoMEX 2013 - Proceedings. 158-163. 10.1109/QoMEX.2013.

### 3. DEFINITION OF TERMS

With respect to definition of terms, abbreviations, and units, the practice of the Institute of Electrical and Electronics Engineers (IEEE) as outlined in the Institute’s published standards [1] should be used. Where an abbreviation is not covered by IEEE practice or industry practice differs from IEEE practice, the abbreviation in question will be described in Section 3.3 of this document.

#### 3.1 Compliance Notation

This section defines compliance terms for use by this document:

**should** – This word indicates that a certain course of action is preferred but not necessarily required.

**should not** – This phrase means a certain possibility or course of action is undesirable but not prohibited.

#### 3.2 Treatment of Syntactic Elements

This document contains symbolic references to syntactic elements used in the audio, video, and transport coding subsystems. These references are typographically distinguished by the use of a different font (e.g., *restricted*), may contain the underscore character (e.g., *sequence\_end\_code*) and may consist of character strings that are not English words (e.g., *dynrng*).

##### 3.2.1 Reserved Elements

One or more reserved bits, symbols, fields, or ranges of values (i.e., elements) may be present in this document. These are used primarily to enable adding new values to a syntactical structure without altering its syntax or causing a problem with backwards compatibility, but they also can be used for other reasons.

The ATSC default value for reserved bits is '1'. There is no default value for other reserved elements. Use of reserved elements except as defined in ATSC Standards or by an industry standards-setting body is not permitted. See individual element semantics for mandatory settings and any additional use constraints. As currently reserved elements may be assigned values and meanings in future versions of this Standard, receiving devices built to this version are expected to ignore all values appearing in currently reserved elements to avoid possible future failure to function as intended.

#### 3.3 Acronyms and Abbreviations

The following acronyms and abbreviations are used within this document.

<b>AHAP</b>	Apple Haptic and Audio Pattern
<b>AMP</b>	Application Media Player
<b>API</b>	Application Programming Interface
<b>ATSC</b>	Advanced Television Systems Committee
<b>A/V</b>	Audio/Video
<b>CD</b>	Companion Device
<b>DASH</b>	Dynamic Adaptive Streaming over HTTP
<b>ERM</b>	Eccentric Rotating Mass

<b>HD</b>	High Definition
<b>HTTP</b>	Hypertext Transfer Protocol
<b>ID</b>	Identification
<b>iOS</b>	Apple Operating System
<b>ISO BMFF</b>	International Organization for Standardization Base Media File Format
<b>JS</b>	JavaScript
<b>JSON</b>	JavaScript Object Notation
<b>LRA</b>	Linear Resonant Actuator
<b>MMT</b>	Multimedia Transport
<b>MMTP</b>	Multimedia Transport Protocol
<b>MPD</b>	Media Presentation Description
<b>MPEG</b>	Motion Pictures Expert Group
<b>PD</b>	Primary Device
<b>REST</b>	Representational State Transfer
<b>RMP</b>	Receiver Media Player
<b>ROUTE</b>	Real-time Object Delivery over Unidirectional Transport
<b>RP</b>	Recommended Practice
<b>SD</b>	Standard Definition
<b>TV</b>	Television
<b>URL</b>	Uniform Resource Locator
<b>VCM</b>	Voice Coil Motor

### 3.4 Terms

The following terms are used within this document.

**reserved** – Set aside for future use by a Standard.

**Actuator** – vibration motor that is driven by the haptic signal.

**Haptics** – the science of touch. From the Greek 'haptikos' - able to touch or grasp.

**CoreHaptics API** – iOS device-specific API that allows developers to incorporate custom haptic feedback in their applications.

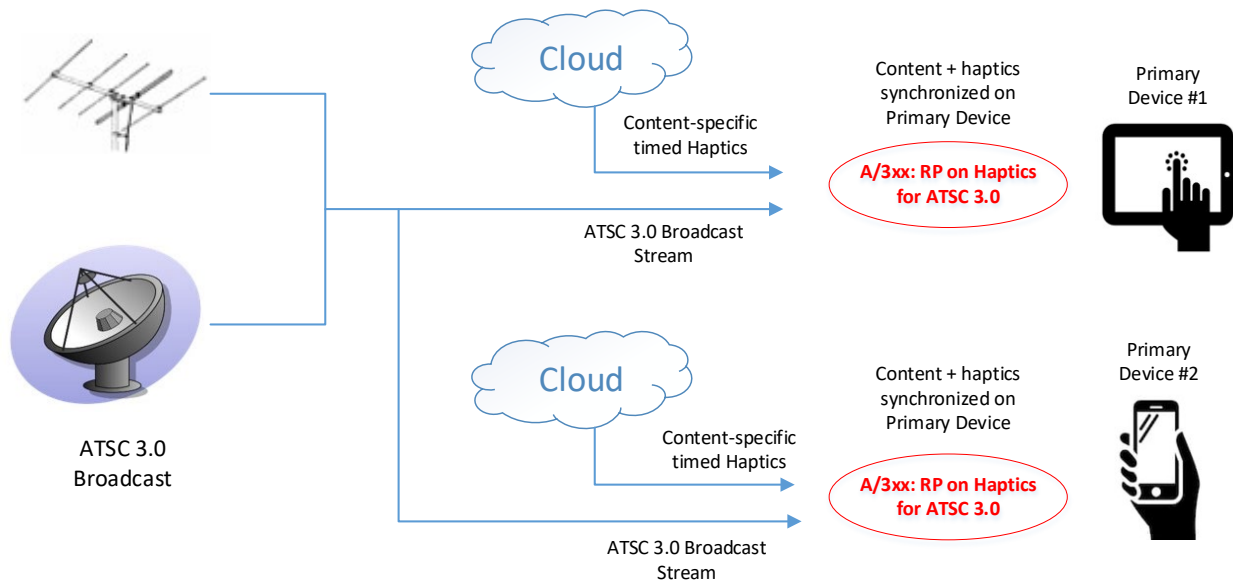
**JSPlayer Haptic Code** – Android-specific JavaScript player code that allows developers to incorporate custom haptic feedback in their applications.

## 4. OVERVIEW OF HAPTICS IN ATSC3.0

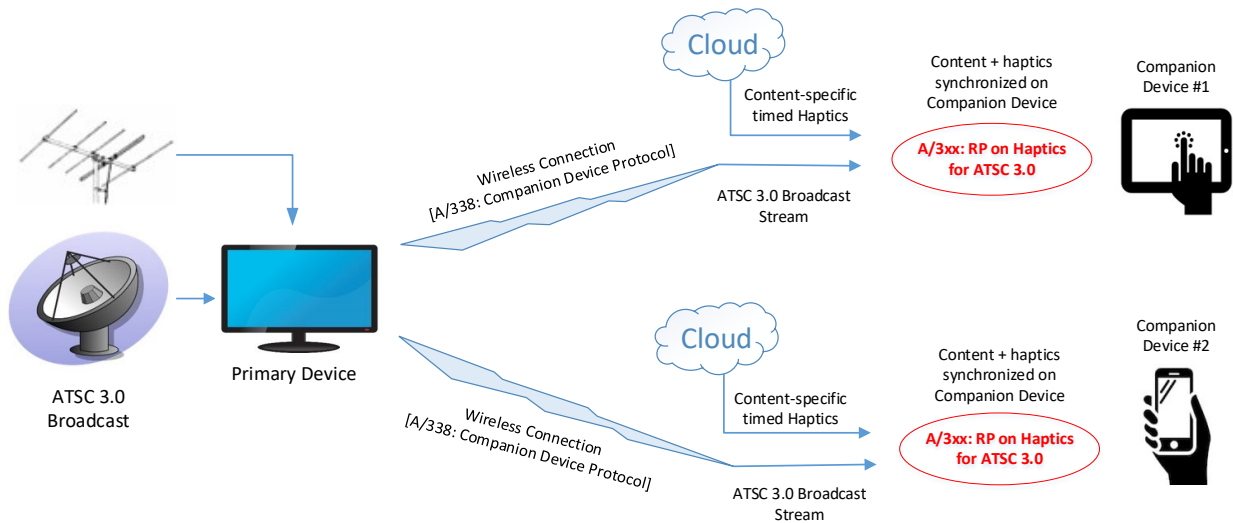
### 4.1 High-level Haptics Workflow in ATSC 3.0

Figure 4.1 and Figure 4.2 below show the high-level workflow of haptics in ATSC 3.0, when mobile devices are used as primary and companion devices, respectively.





**Figure 4.1** High-level haptics workflow—mobile devices as primary devices.



**Figure 4.2** High-level haptics workflow—mobile devices as companion devices.

In Figure 4.1, the mobile devices are the primary devices (i.e., they have ATSC 3.0 tuners built-in). In Figure 4.2, a conventional TV is the primary device that receives the ATSC 3.0 broadcast. The Companion Device Protocol (described in A/338 [4]) is used to transmit the content to companion devices. In both cases, content-specific and device-specific haptic files are downloaded from the cloud and played in sync with the audio/video content in the broadcast stream.

Since Figure 4.2 involves more components in the workflow (the primary device, the communication and signaling required between the PD and CD, etc.) and subsumes the workflow shown in Figure 4.1, the rest of the document will focus on the workflow shown in Figure 4.2. Wherever appropriate, aspects of this RP specific to the workflow in Figure 4.1 will be called out. It is worth reiterating, however, that this RP applies to mobile devices when used as both primary devices and companion devices.

## 4.2 Scenarios Handled in this RP

Depending on the size of the haptic track (that varies by content), the haptic track can either be included in the ATSC 3.0 event stream itself or it can be retrieved from a separate URL along with an authentication token. The workflows for the two cases are shown in Figure 4.3 and Figure 4.4, respectively. In either case, the ‘PD A/V Content’ may or may not be the same as the ‘CD A/V Content’, depending on whether the CD application is playing the same content as the PD or supplemental content. Further, the ‘PD A/V Content’ might be played back by either the Application Media Player (AMP) or the Receiver Media Player (RMP), as described in Section 7 of A/344 [5].

Figure 4.3 shows the case of the haptic track being sparse enough to be included as an event in the ATSC 3.0 broadcast stream. Since the broadcaster knows the content, it uses the content ID to request the haptic file pertaining to that content from the cloud (or other) repository. For live sports events, the output from multiple sensors on the field of play are part of the ‘Sensor Feed’ that is uploaded to the Cloud Repository and used to generate the haptic events that become part of the haptic content JSON. The number and types of sensors used to generate these live haptic events and the format of the Sensor Feed may be proprietary to each broadcaster. As such, the mechanism used for upload of the Sensor Feed to the Cloud Repository is outside the scope of this RP. The haptic content JSON is part of the event stream to the primary device and onto the ATSC 3.0 Companion Device (CD) app via the WebSocket connection. The CD app passes the haptic signal to the Device Haptic API for rendering, via either the CoreHaptics API (on iOS devices) or the JS Player Haptic code (on Android devices) or other appropriate API on the receiving device (e.g., PlayStation). The haptics signal is then rendered by the companion device in sync with the audio/video content.

Figure 4.4 shows the more involved case of the haptic track being too large to fit in the event stream. Since the broadcaster knows the content, it uses the content ID to request the URL and authentication token for the haptic file pertaining to that content from the cloud (or other) repository. For live sports events, the output from multiple sensors on the field of play are part of the ‘Sensor Feed’ that is uploaded to the Cloud Repository and used to generate the haptic events that become part of the haptic content file. The URL for the haptic content and an authentication token are inserted into the ATSC 3.0 event stream by the broadcaster. When this event stream reaches the CD app on the companion device, the CD app uses the URL and token to retrieve the haptic content from the same repository. From this point on, the steps are the same as in Figure 4.3, resulting in the haptic track being rendered in sync with the audio/video content.

The end-to-end detailed workflow of each of these scenarios is addressed in the following two sections.

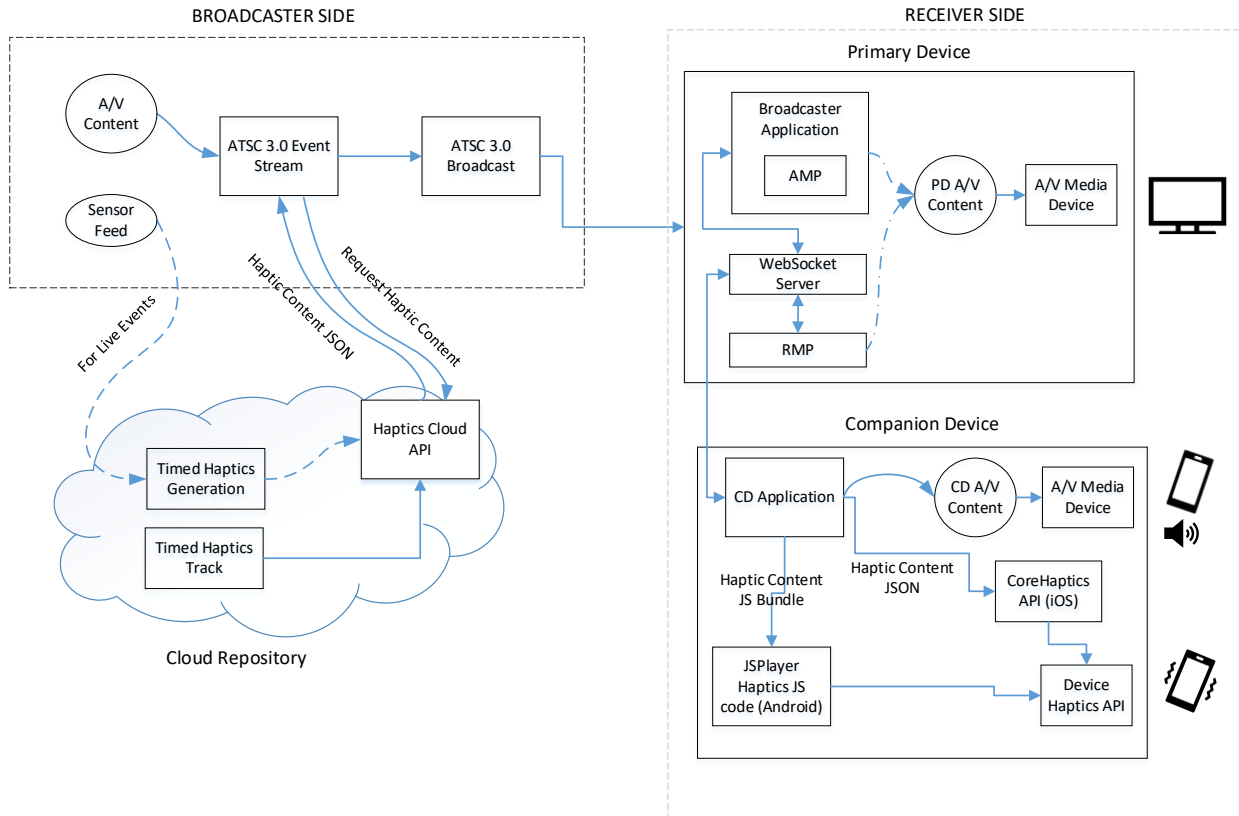


Figure 4.3: Haptic track is part of the event itself.

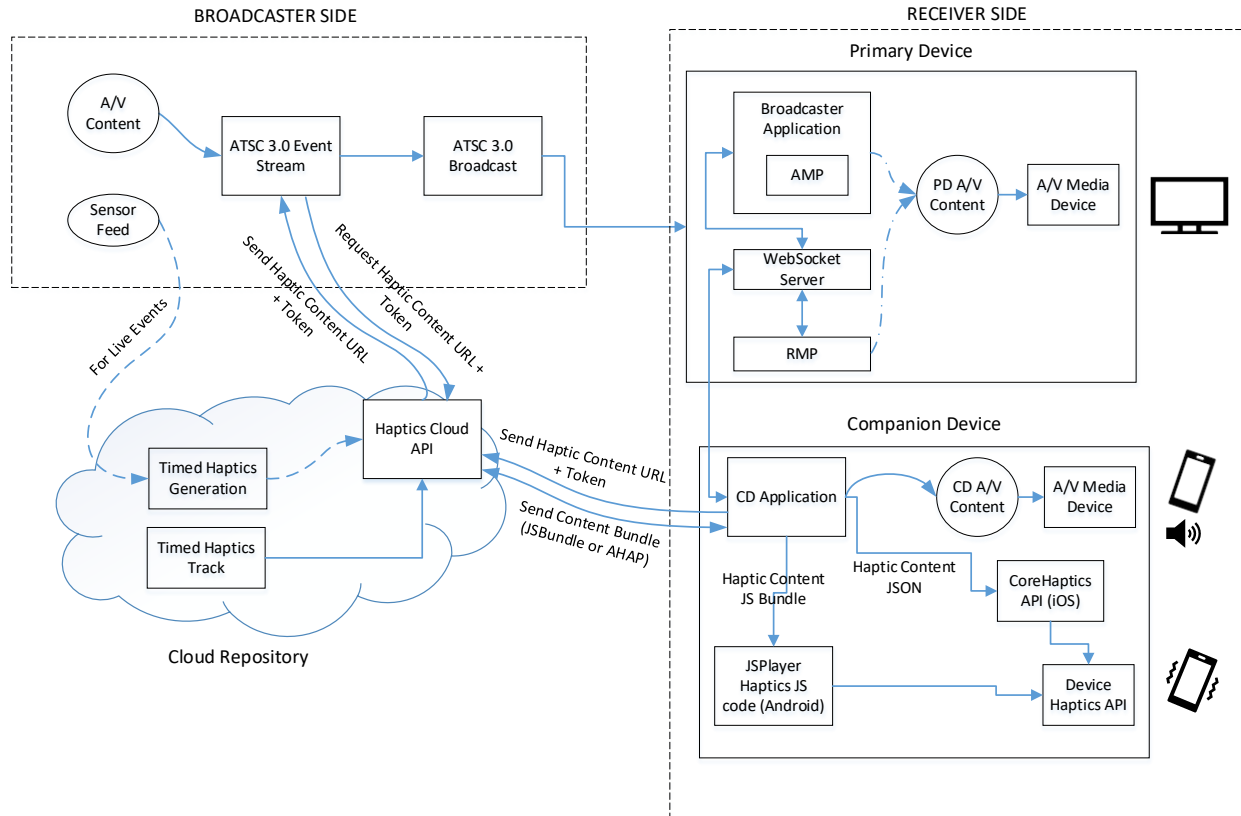


Figure 4.4: Haptic track is in a separate file.

## 5. DETAILED WORKFLOW: HAPTIC CONTENT PART OF THE EVENT STREAM

In the scenario diagrammed in Figure 4.3, there could be multiple short haptic events (like clicks, ticks, buzzes, etc.) throughout the media content, but they are sparse enough that it is most efficient to include these events in the ATSC 3.0 event stream itself.

### 5.1 Retrieving Haptic Content from the Cloud Repository

The broadcaster would access the haptics cloud repository using a simple RESTful web interface. We envision broadcasters using just the GET interface to retrieve the haptic content JSON. POST and PUT interfaces to add to, or update existing haptic content, respectively, are reserved for haptic content creators, unlikely to be used by ATSC 3.0 broadcasters, and therefore out of scope for this RP.

```
GET https://<user>:<password>@<server>:<contentport>/query?contentID=<content_id>
```

where:

- <user> – broadcaster ID known to the repository
- <password> – broadcaster password
- <server> – URL of the haptics cloud repository
- <contentport> – port number of the REST server that serves haptic content
- <contentID> – ID of the haptic content

The call returns the haptic content JSON. An Apple AHAP [8] sample is provided below:

```
{
  "Pattern" : [
    {
      "Event" : {
        "Time": 0.5,
        "EventType": "HapticContinuous"
        "EventDuration" : 0.25,
        "EventParameters": [
          { "ParameterID": "HapticIntensity", "ParameterValue":0.8 }
        ]
      }
    }
  ]
}
```

This example would render a short 250 ms vibration at 80% of maximum intensity and with an offset of 500 ms from the master (typically audio) timeline.

There could be multiple such short haptic events throughout the media content, resulting in multiple haptic-specific insertions into the ATSC Event Stream.

## 5.2 Broadcast Delivery of Haptic Content

In DASH-based broadcasts, the haptic content is best carried in an inband event stream, signaled by an **InbandEventStream** element, as specified in Section 5.6.3 of the DASH-IF IOP [6]. The `@schemeIdUri` of the **InbandEventStream** element shall be of the form `"tag:atsc.org,2016:event"` with `@value` set to `"hpe"`. Each media segment within the Representation containing an **InbandEventStream** element will have an associated `emsg` box, as described in Section 4.1.1 of A/337 [3]. The `timescale`, `presentation_time_delta`, `event_duration`, `id`, and `message_data` fields of the `emsg` box will contain the reference time scale, start time, duration, instance identification, and the control messages, respectively, of the haptic content, extracted from the haptic content JSON, as shown in the example in Section 5.1.

It is also possible to carry the haptic content by `EventStream` elements appearing in the `Period` element of the MPD (as described in Section 4.1.1 of A/337 [3]). However, the dynamic nature of haptics events (especially for live sports content) makes delivery using **InbandEventStream** elements the more appropriate choice. It is recommended to use **InbandEventStream** for all haptic content, for the sake of consistency.

In MMT-based broadcasts, the haptic content is best carried in `evti` boxes in MPUs, as this method is well-suited for dynamic events. The syntax of an `evti` box is described in Section 4.1.2 of A/337 [3], with the fields being populated from the haptic content JSON, as shown in the example in Section 5.1. The haptic control messages would in the `event_data` field of the `evti` box.

## 5.3 Broadband Delivery of Haptic Content

Broadband delivery of events as described in Section 4.2 of A/337 [3], holds for haptic events as well. Dynamic events delivered incrementally over broadband will be delivered as `emsg` boxes in DASH segments for ROUTE-based services or as `evti` boxes for MMTP-based services. In either case, the same format is employed for each box, as described in Section 5.2 above and in Section 4.2 of A/337 [3].

## 5.4 Primary Device to Companion Device Interface

The Broadcaster Application on the PD can subscribe to the event streams from the broadcaster using the ATSC 3.0 Web Socket interface, as described in Section 8 of A/344 [5]. A Broadcaster Application wishing to be notified of a haptic event will register for that type of event using the `@schemeIdUri` and `@value` parameters of the event, using the Event Subscribe API, as described in Section 9.6.1 of A/344 [5]. Through the WebSocket interface supplied to a CD, the CD app receives notifications from the Broadcaster Application operating on the PD, as described in Section 5.6 of A/338 [4], and can subscribe to the Event Subscribe API as well. As the content playback progresses, haptic events contained within the content are sent to the CD as notifications. The CD app extracts the haptic content from the `emsg` or `evti` boxes and passes them, as JSON objects, on to the CoreHaptics API on iOS devices or the JSPlayer Haptic JS API on Android devices (see Figure 4.3). These APIs interface with the low-level haptic rendering API to render the haptic signals.

## 5.5 Synchronization of Haptics with A/V Content

For DASH-based services (broadcast or broadband), `emsg` boxes will carry the haptic event data. Specifically, the `presentation_time_delta` field (in version 1; see Section 5.10.3.3.3 in ISO/IEC 23009-1 [7]) or the `presentation_time` field (in version 2; see Section 5.10.3.3.3 in ISO/IEC 23009-1 [7]) will contain the presentation time offset or the presentation time itself, respectively, of the haptic event. This time will be used by the CD application to synchronize playback of the haptic signals and the audio/video content.

For MMT-based services, `evti` boxes will carry the haptic event data. Specifically, the `event_presentation_time_delta` field (see Section 4.1.2 in A/337 [3]) carries the presentation time offset for the haptic event. This timing information is used by the CD application to synchronize the playback of the haptic signals and the audio/video content.

## 6. DETAILED WORKFLOW: HAPTIC CONTENT IN A SEPARATE FILE

In this scenario (Figure 4.4), the haptic track is typically too long to be incorporated into the event stream and is best stored as a separate haptic file that is retrieved from the cloud and played in sync with the audio/video media.

### 6.1 Retrieving the Haptic File URL From the Cloud Repository

The broadcaster would access the haptics cloud repository using a simple RESTful web interface. We envision broadcasters using just the GET interface to retrieve the URL to the haptic content and an authentication token. POST and PUT interfaces to add to, or update existing haptic content, respectively, are reserved for haptic content creators, unlikely to be used by ATSC 3.0 broadcasters, and therefore out of scope for this RP.

```
GET https://<user>:<password>@<server>:<URLport>/query?contentID=<content_id>
```

where:

`<user>` – broadcaster ID known to the repository

`<password>` – broadcaster password

`<server>` – URL of the haptics cloud repository

`<URLport>` – port number of the REST server that serves haptic content URLs

`<content_id>` – ID of the haptic content

This call returns (at least) the haptic content URL <url> and authentication token <authtoken>. Various haptic cloud repositories may include other information in the JSON object returned by this call (repository ID, etc.), but these two elements are the minimum required.

```
{ "hapticData": {  
  "url": <url>,  
  "authtoken": <authtoken>  
}
```

This URL and token are carried in the ATSC event stream to the CD application that can use it to retrieve the haptic content, as described below.

## 6.2 Broadcast Delivery of Haptic File URL and Token

In DASH-based broadcasts, the haptic file URL and token are best carried in an inband event stream, signaled by an **InbandEventStream** element, as specified in Section 5.6.3 of the DASH-IF IOP [6]. The @schemeIdUri of the **InbandEventStream** element shall be of the form "tag:atsc.org,2016:event" with @value set to "hpf". Each media segment within the Representation containing an **InbandEventStream** will have an associated msg box, as described in Section 4.1.1 of A/337 [3]. The timescale, presentation\_time\_delta, event\_duration, and id fields of the msg box will be zero, while the message\_data field contains the haptic file URL and the authentication token. The prefetch time for a haptic track to be retrieved and downloaded in this manner is typically 50 ms plus any channel latency.

It is also possible to carry the haptic file URL and token by **EventStream** elements appearing in the Period element of the MPD (as described in Section 4.1.1 of A/337 [3]). However, the dynamic nature of haptics events (especially for live sports content) makes delivery using **InbandEventStream** elements the more appropriate choice. It is recommended to use **InbandEventStream** for all haptic content, for the sake of consistency.

In MMT-based broadcasts, the haptic file URL and authentication token are best carried in evt i boxes in MPUs, as this method is well-suited for dynamic events. The syntax of an evt i box is described in Section 4.1.2 of A/337 [3]. The haptic file URL and authentication token would be in the event\_data field of the evt i box.

## 6.3 Broadband Delivery of Haptic File URL and Token

Broadband delivery of events as described in Section 4.2 of A/337 [3], holds for haptic events as well. Dynamic events delivered incrementally over broadband will be delivered as msg boxes in DASH segments for ROUTE-based services or as evt i boxes for MMTP-based services. In either case, the same format is employed for each box, as described in Section 6.2 above and in Section 4.2 of A/337 [3].

## 6.4 Primary Device to Companion Device Interface

The Broadcaster Application on the PD can subscribe to the event streams from the broadcaster using the ATSC 3.0 Web Socket interface, as described in Section 8 of A/344 [5]. A Broadcaster Application wishing to be notified of a haptic event will register for that type of event using the @schemeIdUri and @value parameters of the event, using the Event Subscribe API, as described in Section 9.6.1 of A/344 [5]. Through the WebSocket interface supplied to a CD, the CD app receives notifications from the Broadcaster Application operating on the PD, as described in Section 5.6 of A/338 [4] and can subscribe to the Event Subscribe API as well. As the content

playback progresses, haptic events contained within the content are sent to the CD as notifications. The CD app extracts the haptic file URL and authentication token from the `emsg` or `evti` boxes, accesses the haptics cloud repository to retrieve the haptic content JSON and passes it onto the CoreHaptics API on iOS devices or the JSPlayer Haptic JS API on Android devices (see Figure 4.4). These APIs interface with the low-level haptic rendering API to render the haptic signals.

The CD application accesses the haptics cloud repository using a simple RESTful web interface, like that described in Sections 5.1 and 6.1 above.

```
GET https://<server>:<contentport>/query?fileID=<fileURL>&auth=<authtoken>
```

where:

<server> – URL of the haptics cloud repository

<contentport> – port number of the REST server that serves haptic content

<fileURL> – URL of the haptic file (see Section 6.1)

<authtoken> – authentication token (see Section 6.1)

## 6.5 Synchronization of Haptics with Audio/Video Content

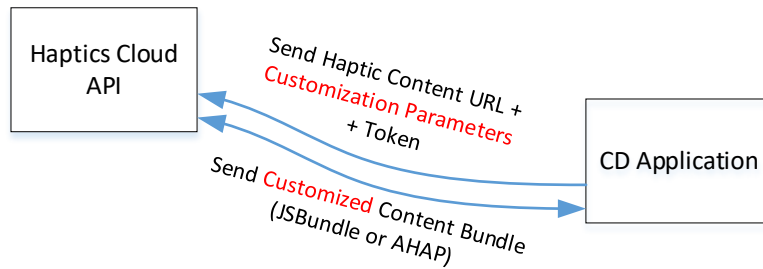
In the case where the haptic content is in a separate file, the presentation time of the haptic signals are part of the haptics data structure. For example, in Section 5.1, the AHAP sample has a Time field that represents the offset of the haptic signal with respect to the audio timeline. The time fields in the `emsg` and `evti` boxes are ignored.

## 7. CUSTOMIZATION OF THE HAPTIC EXPERIENCE ON MOBILE DEVICES

Studies have shown that user preferences for haptic experiences vary widely [9]. For example, while watching live sports like American football on mobile devices, some viewers prefer to “feel” every action in the game (e.g., every hit on the quarterback, linemen bumping into each other, every fall to the ground, etc.) while others would like to “feel” only the major events in the game (e.g., a touchdown score, an interception, a sack, etc.). Further, not all mobile devices have the same quality or number of actuators – most have a single SD (standard definition) actuator, while many of the newer devices have one or more HD (high-definition) actuators. For these reasons, it is necessary to customize the haptic experience based on user preferences and device capabilities.

Such customization will not be possible for the case of the haptic content being part of the event stream (Figure 4.3 and Section 5), since it is not possible to determine the user preference or device type before the content starts playing on the CD. However, it is possible to customize the haptic experience for the case of the haptic content being in a separate file (Figure 4.4 and Section 6), since the content is retrieved by the CD application, using the file URL and authentication token. Including the device capabilities and user preference information in that content request will ensure that the correct haptic file pertaining to that user preference and device type is retrieved. This is shown in Figure 7.1, that illustrates how a portion of Figure 4.4 can be modified to include customization parameters in the content request from the CD application.





**Figure 7.1** Portion of Figure 4.4 showing the addition of customization parameters into the content request.

Considering these customization parameters, the API call to the haptics cloud repository in Section 6.4 would be modified as follows:

```

GET
https://<server>:<contentport>/query?fileID=<fileURL>&auth=<authtoken>&userpref=<userpref>&acttype=<actuator_type>
  
```

where:

- <server> – URL of the haptics cloud repository
- <contentport> – port number of the REST server that serves haptic content
- <fileURL> – URL of the haptic file (see Section 6.1)
- <authtoken> – authentication token (see Section 6.1)
- <userpref> – (optional) user preference value; is one of the following:

User Preference Value	Description
minimum	Used for live sports; haptic events only for major events in the game. This is the default setting for all live sports if no user preference is specified.
typical	Used for non-sports content; haptic events follow the action in the short-form or long-form video content
maximum	Used for live sports; haptic events for all actions in the game

<actuator\_type> – (optional) type of actuator; is one of the following:

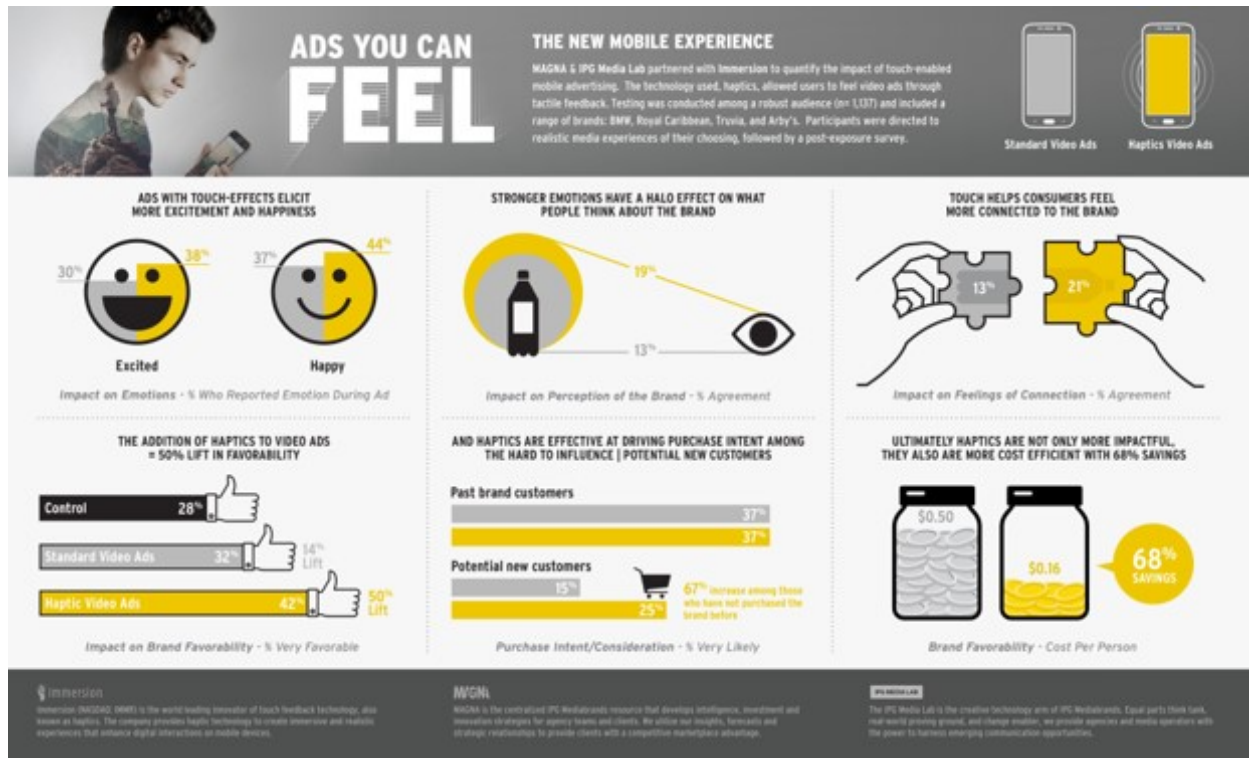
Actuator Type	Description
SD	Refers to standard definition actuators (e.g., ERMs, Narrowband LRAs). Default setting, if no actuator type is specified, to reflect the large number of SD actuator devices still in use.
HD	Refers to high-definition actuators (e.g., wideband LRAs, piezoelectric, VCMs)

## 8. DELIVERY OF HAPTICS-ENABLED ADVERTISEMENTS TO MOBILE DEVICES

It is common knowledge that advertisements are big drivers of revenue for broadcasters. Specifically, mobile ads are an excellent way to reach the coveted demographics that spend a lot of time on their mobile devices. Given that ATSC 3.0 broadcasts are designed to be consumed on

mobile devices, mobile ads will continue to be significant sources of revenue for ATSC 3.0 broadcasters as well.

Haptics-enabled mobile advertisements have been shown to elicit more excitement and happiness, evoke stronger positive emotions in people about the brand, improve brand favorability, are more cost-efficient, and most importantly, are effective in driving purchase intent among the hard-to-influence and potential new customers. Figure 8.1 below summarizes the key findings of the joint MAGNA-IPG Media Lab-Immersion study on the effectiveness of haptics-enabled mobile ads.



**Figure 8.1** MAGNA-IPG Media Labs immersion study on effectiveness of haptics-enabled mobile ads.

The ‘Haptic Content in a Separate File’ scenario, described in Section 6, can be used for haptics-enabled mobile ad delivery to CDs. The ad content URL and authentication token are retrieved and inserted into the ATSC 3.0 Event stream, just like regular media content, as described in Section 6.1. It is the responsibility of the broadcaster to insert ads at appropriate locations within the content and the corresponding events, signaling the presence of haptic file URLs and tokens, into the ATSC 3.0 event stream. The rest of the workflow is identical to that described in Sections 6.2 through 6.5.

## Annex A Usage Scenarios

Three example usage scenarios are described below.

**Scenario A:** Julio is watching a broadcast concert of his preferred rock & roll band on the TV screen (primary device). A notification pop-up on the TV informs him that alternative camera views of the concert presenting each musician, along with a synchronized haptic track, are available through a dedicated application on his CD. Julio launches that application which informs Julio that close-ups of the guitarist, bassist, singer, and drummer are available. Julio selects the guitarist during the guitar solo and switches to the drummer later in the song. Media content on the TV screen and the companion screen are synchronously rendered, and Julio is able to actually feel the guitar strings and the drums on his CD application.

**Scenario B:** Mark and his wife Jane are both huge fans of American football. At the start of the live broadcast of a game on the TV screen (primary device), they get a notification that the game is also available through a dedicated application on their companion devices with customizable haptics. Mark and Jane both launch the application on their respective smartphones and proceed to customize their haptics experience, using the pop-up menu in the CD application. Mark is a big fan of feeling every game action (clash of the offensive and defensive linemen, hits on the quarterback, sacks, tackles, interceptions, etc.), while Jane just likes haptics for major game events (touchdown scores, field goals, interceptions). Mark's selection represents the *maximum* user preference value in Section 7, while Jane's selection represents the *minimum* user preference value in Section 7. The CD application on each device automatically determines the actuator type and includes it in the customization parameters list, to ensure that the downloaded haptic file can be played properly on that device. The game on the TV screen and the companion screens are synchronously rendered, and Mark and Jane enjoy the customized haptics experiences on their respective companion devices.

**Scenario C:** While watching the football game in Scenario B, the broadcaster runs a set of 30-second ads mid-roll. For Mark, these ads are rendered on the companion device and each ad unit renders a coordinated haptic track that keeps his attention during the break. Since Jane has selected 'minimum' haptics, she does not receive haptic effects during the ad units, and she spends the ad break doing something else.

– End of Document –