OpenCable™ Specifications

XML Representation of TV Services Metadata

OC-SP-XTSM-I01-131106

ISSUED

Notice

This OpenCable specification is the result of a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. for the benefit of the cable industry and its customers. This document may contain references to other documents not owned or controlled by CableLabs®. Use and understanding of this document may require access to such other documents. Designing, manufacturing, distributing, using, selling, or servicing products, or providing services, based on this document may require intellectual property licenses from third parties for technology referenced in this document.

Neither CableLabs nor any member company is responsible to any party for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this document, or any document referenced herein. This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein.

DISCLAIMER

This document is published by Cable Television Laboratories, Inc. ("CableLabs®").

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various agencies; technological advances; or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein. CableLabs makes no representation or warranty, express or implied, with respect to the completeness, accuracy, or utility of the document or any information or opinion contained in the report. Any use or reliance on the information or opinion is at the risk of the user, and CableLabs shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, or utility of any information or opinion contained in the document.

This document is not to be construed to suggest that any affiliated company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any cable member to purchase any product whether or not it meets the described characteristics. Nothing contained herein shall be construed to confer any license or right to any intellectual property, whether or not the use of any information herein necessarily utilizes such intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number:	OC-SP-XTSM-I01-131106			
Document Title:	XML Representation of TV Services Metadata			
Revision History:	I01 - Released 11/6/13			
Date:	November 6, 2013			
Status:	~~Work in Progress~~	~~Draft~~	**Issued**	~~Closed~~
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	**Public**

Key to Document Status Codes

Work in Progress An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.

Draft A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.

Issued A generally public document that has undergone Member and Technology Supplier review, cross-vendor interoperability, and is for Certification testing if applicable. Issued Specifications are subject to the Engineering Change Process.

Closed A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks

CableLabs® is a registered trademark of Cable Television Laboratories, Inc. Other CableLabs marks are listed at http://www.cablelabs.com/certqual/trademarks. All other marks are the property of their respective owners.

Contents

1 SCOPE.. 1

1.1 Introduction and Purpose.. 1
1.2 Scope 1
1.3 Requirements.. 1

2 REFERENCES.. 2

2.1 Normative References 2
2.2 Informative References.. 2
2.3 Reference Acquisition 3

3 TERMS AND DEFINITIONS.. 4

4 ABBREVIATIONS AND ACRONYMS 5

5 OVERVIEW 6

5.1 XML for TV Services... 6
5.2 MPEG2 Transport Stream Message Semantics... 7
5.2.1 ETV Application Messaging Protocol 7
5.2.2 SCTE 35 7
5.2.3 Content Advisories.. 7

6 CUE CONTEXT.. 8

6.1 CueContext Type... 8
6.1.1 name.. 9
6.1.2 version.. 9
6.1.3 event 10
6.1.4 targetStartTime 10
6.1.5 uri... 10
6.1.6 contentId 10
6.1.7 number 10
6.1.8 total 10
6.1.9 priority 10
6.1.10 parameter.. 10
6.2 Parameter Type.. 11
6.2.1 name.. 11
6.2.2 type... 11
6.2.3 value.. 11
6.3 applicationEvent Element.. 11
6.3.1 uri Attribute.. 11
6.3.2 event Attribute.. 11
6.4 contentInsertion Element.. 12
6.4.1 uri Attribute.. 12
6.4.2 event Attribute.. 12
6.5 contentDescription Element 12
6.5.1 event Attribute.. 12

7 USE CASES 14

7.1 Content Insertion 14

7.1.1 Representing an SCTE-35 Splice Insert 14

7.2 Interactive Application Event 14

7.2.1 Representing messages from an ETV Integrated Signaling Stream (EISS) 14

7.3 Content Identification 15

7.3.1 Representing Content Advisories from SCTE-65 Content Advisory Descriptor 15

ANNEX A XML SCHEMA (NORMATIVE) 17

APPENDIX I MAPPING (INFORMATIVE) 18

I.1 EISS 18

I.2 SCTE 35 19

APPENDIX II EXAMPLES (INFORMATIVE) 23

II.1 SCTE 35 Splice Insert as contentInsertion 23

II.2 TTML 23

II.3 WebVTT 23

APPENDIX III USE IN HTML5 TIMED TEXT TRACKS (INFORMATIVE) 24

Figures

Figure 1 - Usage of XML payload for TV Services 6

Figure 2 - In-band vs. Out-of-band Usage of Cue Context 7

Figure 3 - CueContext Element Relationships 8

Figure 4 - CueContext Structure 9

Figure 5 - Parameter Structure 11

Figure 6 - Usage of XML with HTML5 TimedTextTracks and Client Library 24

Tables

Table 1 - Content ID Values 10

Table 2 - Event Values for applicationEvent 11

Table 3 - Event Values for contentInsertion 12

Table 4 - Event Values for contentDescription 12

This page left blank intentionally.

1 SCOPE

1.1 Introduction and Purpose

This document is intended for architects, designers and implementers of television services with a need to provide synchronized metadata in a common XML format.

1.2 Scope

This document describes an XML format for representing metadata synchronized with audio/video content. It does not specify a means of transport. Although some informative mappings are provided from other formats, they are not exhaustive or mandatory.

1.3 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"SHALL" This word means that the item is an absolute requirement of this specification.

"SHALL NOT" This phrase means that the item is an absolute prohibition of this specification.

"SHOULD" This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.

"SHOULD NOT" This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

"MAY" This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

2 REFERENCES

2.1 Normative References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

[ADI 1.1] CableLabs Asset Distribution Interface Specification Version 1.1, MD-SP-ADI1.1-C01-120803, August 3, 2012, Cable Television Laboratories, Inc.

[Ad-ID] Ad-ID, Advertising Identification and Management, http://www.ad-id.org/help/structure.cfm.

[EIDR] EIDR: ID Format, Version 1.0.2, January 30, 2012, http://eidr.org/documents/EIDR_ID_Format_v1.02_Jan2012.pdf.

[RFC 4246] IETF RFC 4246, International Standard Audiovisual Number (ISAN) URN Definition, February 2006.

[SCTE 65] ANSI/SCTE 65 2008, Service Information Delivered Out-of-Band for Digital Cable Television.

[SCTE 67] ANSI/SCTE 67 2010, Recommended Practice for SCTE 35 Digital Program Insertion Cueing Message for Cable.

[SMPTE 330] ST 330:2011, Unique Material Identifier (UMID).

[SMPTE 2029] ST 2029:2009, Uniform Resource Names for SMPTE Resources.

[XML] W3C Extensible Markup Language (XML) 1.0 (Fifth Edition), November 26 2008.

[TMS ID] TMS Unique ID, Tribune Media Services, http://www.tribunemediaservices.com/.

[TTML] W3C Timed Text Markup Language 1 (TTML1) (Second Edition), September 24, 2013.

[XML Schema2] W3C XML Schema Part 2: Datatypes Second Edition.

2.2 Informative References

[ETV AM] OC-SP-ETV-AM1.0.1-120614, Enhanced TV Application Messaging Protocol 1.0, June 14, 2012, Cable Television Laboratories, Inc.

[ISO 13818-1] ISO/IEC 13818-1: 2013, Information technology -- Generic coding of moving pictures and associated audio information: Systems, www.iso.org.

[SCTE 35] SCTE 35 2013, Digital Program Insertion Cueing Message for Cable.

[WebVTT] Web Video Text Tracks Format, W3C Draft Community Group Specification, August 11, 2013

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027;
  Phone +1-303-661-9100; Fax +1-303-661-9199; http://www.cablelabs.com
- Internet Engineering Task Force (IETF) Secretariat, 48377 Fremont Blvd., Suite 117, Fremont, California 94538, USA, Phone: +1-510-492-4080, Fax: +1-510-492-4001, http://www.ietf.org
- SMPTE, www.smpte.org
- Society of Cable Telecommunications Engineers (SCTE), www.scte.org
- World Wide Web Consortium (W3C), www.w3c.org

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Client Application A program that executes on a viewing device, typically synchronized with an audio or video stream.

Chapter A temporal segment within an audio/video stream usually associated with metadata describing the segment.

Cue A specific action to be carried out or information that applies at a specific time.

Timed text track A grouping of cues all having a common function (or kind).

MPEG-2 Transport Stream (TS) A standard format for transmission of video, audio and associated data (ISO/IEC 13818-1).

Enhanced TV Services Services that enhance the experience of watching a television video feed beyond the presentation of audio and video.

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations:

ADI	Asset Distribution Interface
ATSC	Advanced Television Systems Committee
bslbf	Bit-string left bit first
EIDR	Entertainment ID Registry
ETV	Enhanced Television
EISS	ETV Integrated Signaling Stream
HTML	Hypertext Markup Language
ID	Identifier
ISAN	International Standard Audiovisual Number
MPEG	Motion Picture Experts Group
PMT	Program Map Table
PTS	Presentation Time Stamp
rpchof	Remainder polynomial coefficients, highest order first
uimsbf	Unsigned integer most significant bit first
SCTE	Society of Cable Telecommunications Engineers
SMPTE	Society of Motion Picture and Television Engineers
TMS	Tribune Media Services
TS	Transport Stream
TTML	Timed Text Markup Language
TV	Television
UMID	Unique Material Identifier
URI	Uniform Resource Identifier
URN	Uniform Resource Name
WebVTT	Web Video Text Tracks
XML	Extensible Markup Language

5 OVERVIEW

This specification defines an XML format for representing TV Services that can be used in media containers that allow XML messages, HTML5 text track cues, etc., or used in other places. The term TV Services is used to describe supplemental information that may accompany audio/video content to enhance the experience for the user or provide advanced functionality. Some of this information, such as captions, subtitles, video descriptions and chapters, may already be fully expressed using existing specifications. This specification fills in the gaps by providing a common XML format for additional forms of metadata, which include, but are not limited to:

- Synchronization of interactive applications with content
- Client-side content (advertising) insertion
- Content advisories for enabling parental controls
- Other forms of content identification not suitably expressed as chapters

5.1 XML for TV Services

Figure 1 shows how MPEG-2 TS metadata is delivered to an IP client application, for example, an HTML5 user agent, and how the XML format is intended to be used for both in-band and out-of-band delivery with other media formats that employ XML content. Metadata delivered in-band is part of the same file or stream as the video and audio. Out-of-band metadata is delivered as a separate file or stream and is time-indexed to allow synchronization with the video. Section 6 defines normative requirements for the XML content. Definition of the in-band extraction process and out-of-band delivery are out of scope of this specification and will be covered in other application platform specifications.

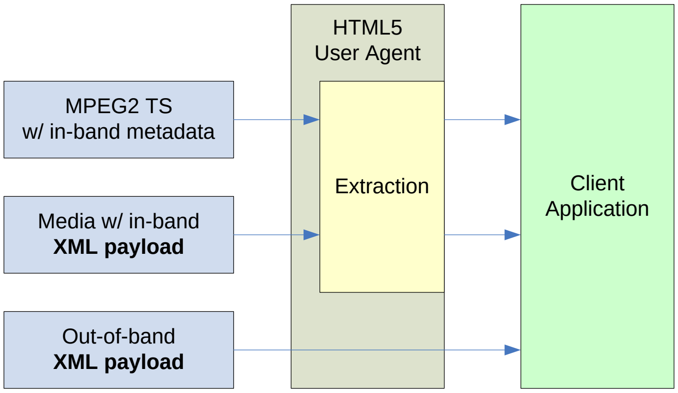

Figure 1 - Usage of XML payload for TV Services

A common XML payload can be used for representing both in-band and out-of-band cues as shown in Figure 2.

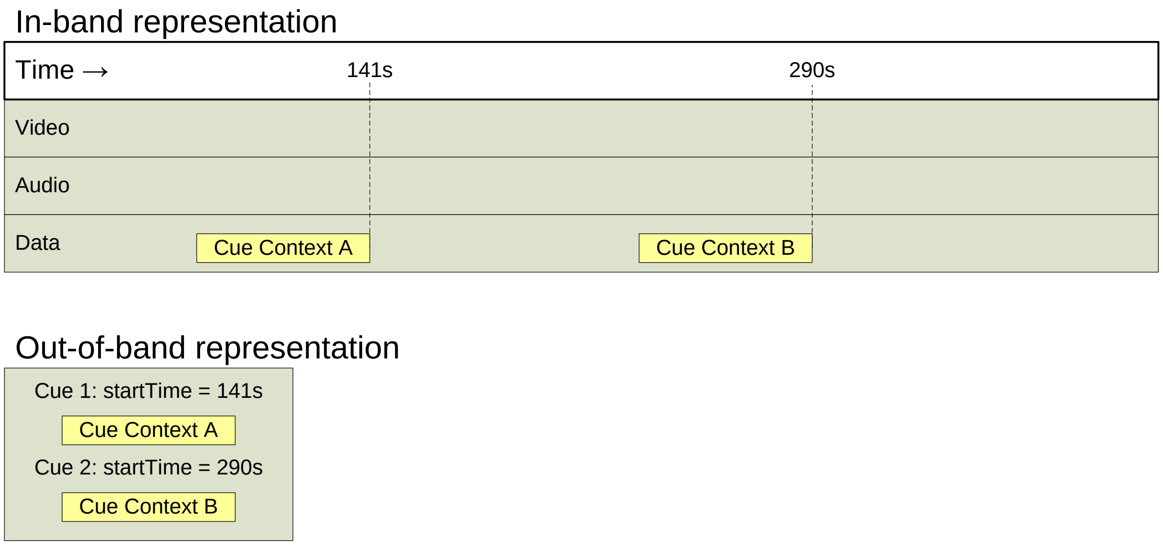

Figure 2 - In-band vs. Out-of-band Usage of Cue Context

5.2 MPEG2 Transport Stream Message Semantics

Traditional television services are commonly distributed using MPEG2 Transport Streams in the form of descriptors or elemental data streams. The data carried is used to identify points or regions of interest within the media that are related to an application. This information may simply describe the content, as in the case of a content advisory, or serve to signal actions to the client, such as insertion of an advertisement or loading of an interactive application. The method of extraction of this information from the stream by the client is outside the scope of this specification.

5.2.1 ETV Application Messaging Protocol

An in-band signaling format for the purpose of triggering interactive events on the client is specified in [ETV AM]. These events may include loading, presenting, and terminating interactive applications, as well as signaling useful information to these applications. The stream of these events is commonly referred to as EISS. A method for mapping EISS to this XML format is specified in Section I.1.

5.2.2 SCTE 35

Insertion of digital content for the purpose of advertising is handled in traditional cable television using [SCTE 35]. In addition to content insertion points, the use of in-band segmentation descriptors also allows for identification of content. A method for mapping from an arbitrary SCTE 35 segmentation descriptor to this XML format is specified in Section I.2.

5.2.3 Content Advisories

Content Advisories are carried in the Content Advisory Descriptor as specified in [SCTE 65] (see Section 6.5). Advisories are segmented into dimensions with each having a rating value. The dimension definitions are provided in the Region Rating Table (also from [SCTE 65]). The U.S. Region Rating Table is assumed if not provided with the video.

6 CUE CONTEXT

This specification defines an XML schema complex type `CueContext` used to instantiate one of the application-specific cue context element types, `applicationEvent`, `contentInsertion`, `contentDescription`, defined in this specification. Figure 3 shows the relationship between the base `CueContext` type and the elements defined in this specification.

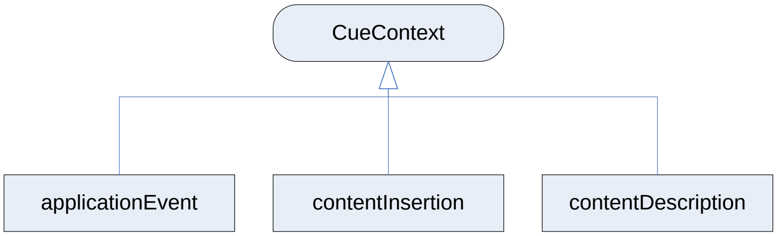

Figure 3 - CueContext Element Relationships

6.1 CueContext Type

The `CueContext` type is shown in Figure 4. The attributes and elements are defined below but MAY be expanded or overridden for each derived element type. All of the attributes in `CueContext` are optional as indicated by the dashed border in the diagram. When an attribute is omitted, no default value is implied unless otherwise stated in this specification. In addition to the elements defined in this specification, additional attributes MAY be included, but these additional attributes SHALL be specified by a namespace outside of this specification.

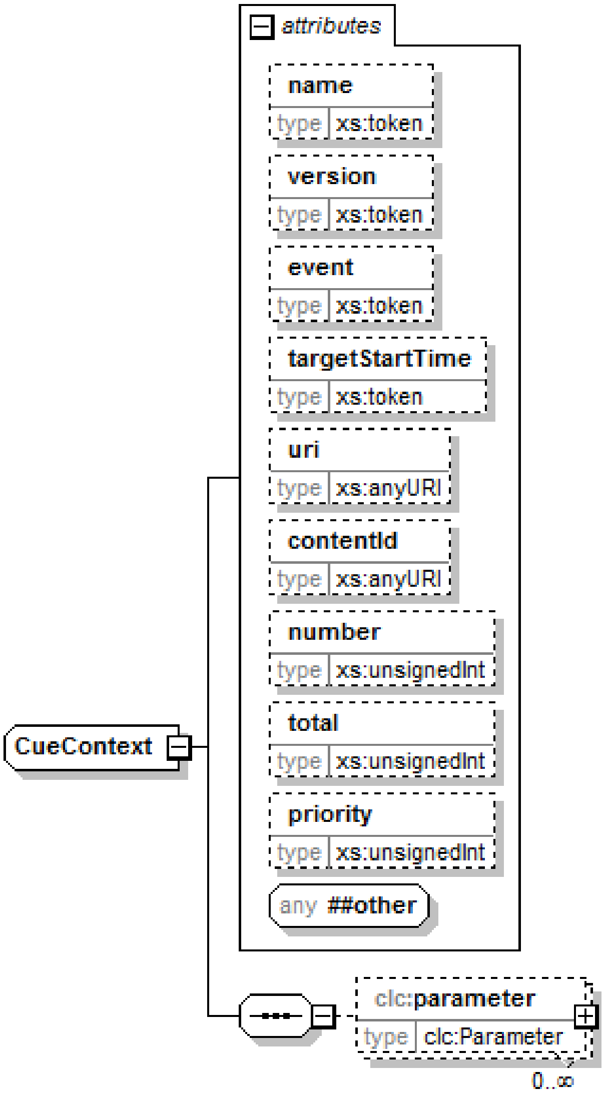

Figure 4 - CueContext Structure

6.1.1 name

The `name` attribute specifies a unique name for this context item that distinguishes it from other context items. It is expected that the `name` MAY be repeated in multiple cues if the intent is to modify a previous context item, such as in the case of changing or cancelling a previous cue. The `name` SHOULD NOT be reused in a subsequent cue that is not the same context item. This `name` value MAY be used as the identifier on the cue itself if it is known to be globally unique.

6.1.2 version

The `version` attribute allows for context items with the same `name` to be distinguished for the purposes of determining which representation takes precedence. In cases where two context items have the same `name` attribute, the one with the higher value order takes precedence. The version may be any token value, but the ordering SHALL be deterministic. The version SHOULD be a numeric value (either integer or decimal) to make this as clear as possible.

6.1.3 event

The event attribute specifies an intended action associated with the cue. The allowed values for this attribute vary for each element that derives from CueContext. An enumeration is not provided in the XML Schema of the CueContext; however, the values SHALL adhere to the values listed in this specification for the derived elements.

6.1.4 targetStartTime

The targetStartTime attribute allows the cue to reference an event in the future. An example of how this might be used is to provide advanced warning of a content insertion cue that will follow shortly and give the application time to load the content to be inserted. The value of this attribute is specified as the time in milliseconds relative to start of media timeline.

6.1.5 uri

The uri attribute provides either the identity or location of the resource associated with this CueContext. Further elaboration of this attribute's value is provided for each derived element.

6.1.6 contentId

The contentId attribute represents the identifier of the content currently playing in the video. The contentId value SHALL take one of the forms listed below if it is expected to be correctly interpreted by the client.

Table 1 - Content ID Values

Type	Format	Example
Ad-ID	[Ad-ID]	http://www.ad-id.org/help/structure.cfm
ADI	[ADI 1.1]	urn:cablelabs:md:paid:provider.com:ABCD1234567890123456
EIDR	[EIDR]	http://eidr.org/documents/EIDR_ID_Format_v1.02_Jan2012.pdf
ISAN	[RFC 4246]	URN:ISAN:0123-1230-3210-2310-1 URN:ISAN:1881-66C7-3420-6541-9-9F3A-0245-U
UMID	[SMPTE 330] and [SMPTE 2029]	urn:smpte:umid:0123456789ABCDEF….
TMS ID	[TMS ID]	www.tribunemediaservices.com

6.1.7 number

The number attribute provides the position of the current cue within a sequence of cues for the program (as identified by the corresponding contentId). This value SHALL be a positive integer with 1 representing the first item and so on.

6.1.8 total

The total attribute provides the total number of expected cues within the program (as identified by the programId). This value SHALL be a positive integer.

6.1.9 priority

The priority attribute specifies the relative priority for the cue event, where higher values indicate higher priority. Priority is used when multiple applications are signaled at the same time.

6.1.10 parameter

The parameter element is defined in Section 6.2 and allows for zero or more values of additional payload to be provided along with the cue. This payload is specific to the application and not defined in this specification. The client application SHALL either understand how to interpret all parameters or disregard the cue entirely.

6.2 Parameter Type

A Parameter is simply an arbitrary piece of data that may accompany a CueContext. The Parameter SHALL have a value and may optionally have an id and type to further describe the value.

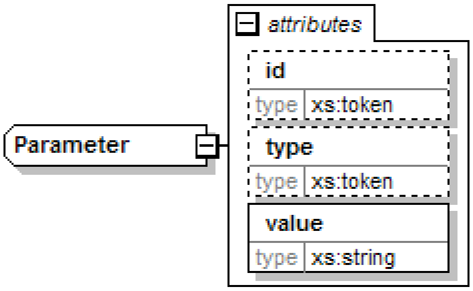

Figure 5 - Parameter Structure

6.2.1 name

The optional name attribute MAY be provided if the parameter order is not known by the application or there may be a variable set of parameters.

6.2.2 type

The optional type attribute MAY be provided to indicate the type of the corresponding value attribute. The value of this attribute SHALL correspond to one of the types defined in [XML Schema2] and derived from anySimpleType.

6.2.3 value

The value attribute contains the actual value of the parameter. This may be anySimpleType as defined by [XML Schema2] including hexBinary or base64Binary if appropriate.

6.3 applicationEvent Element

The applicationEvent element derives from CueContext type and is used to control synchronization of applications with video. The methods by which the events are created and executed on the client are outside the scope of this specification. It is expected that the client will understand how to handle applicationEvents or disregard them. The following variations are expected from the base CueContext type.

6.3.1 uri Attribute

The uri attribute SHOULD indicate the application that is being referenced.

6.3.2 event Attribute

The event attribute SHALL contain one of the following values and is not case-sensitive.

Table 2 - Event Values for applicationEvent

Value	Description
LOAD	Fetch the application from its source and prepare for execution
START	Present the application to the user
SUSPEND	Pause the application and disable presentation with the intention that it may be resumed at a later time
TERMINATE	Stop the application and free any resources used

Value	Description
DATA	Application data is provided in one or more Parameter elements – no further action is implied

6.4 contentInsertion Element

The `contentInsertion` element derives from `CueContext` type and is used to specify the insertion of alternate content into the currently running video. The method by which this is performed is outside the scope of this specification. It is expected that the client will understand how to handle `contentInsertion`s or disregard them. The following variations are expected from the base `CueContext` type.

6.4.1 uri Attribute

The `uri` attribute can provide additional information about the content to be inserted. For example, it could be a reference to the content to be inserted or it could provide context to the client application.

6.4.2 event Attribute

The `event` attribute SHALL contain one of the following values and is not case-sensitive.

Table 3 - Event Values for contentInsertion

Value	Description
LOAD	Pre-load the content, but do not display
INSERT	Insert the new content while allowing the base content to continue to run at the same speed
SWITCH	Pause the base content before inserting the new content
RESUME	Terminate the inserted content and resume play of the base content
CANCEL	Cancel a previously cued pending insertion with the same id

6.5 contentDescription Element

The `contentDescription` element derives from `CueContext` type and is used to specify descriptive metadata about the currently running video. Client behavior associated with this element is outside the scope of this specification. The following variations are expected from the base `CueContext` type.

6.5.1 event Attribute

The allowable values map very closely to those in [SCTE 35].

Table 4 - Event Values for contentDescription

Value	Description
CONTENT_ID	Carries an identifier for the currently running content
ADVISORY	Carries an advisory or rating for the currently running content
PROGRAM_START	Indicates the start of a program
PROGRAM_END	Indicates the end of a program

Value	Description
PROGRAM_EARLY_TERMINATION	Indicates the program has ended early
PROGRAM_BREAKAWAY	Indicates a temporary departure from the currently running program
PROGRAM_RESUMPTION	Indicates the previously running program has been resumed
PROGRAM_RUNOVER_PLANNED	Indicates the program is running beyond the scheduled end time
PROGRAM_RUNOVER_UNPLANNED	Indicates the program is running beyond the scheduled end time
UNSCHEDULED_EVENT_START	Indicates an unscheduled event start
UNSCHEDULED_EVENT_END	Indicates an unscheduled event end

7 USE CASES

This section outlines some of the common use cases for generating XML Content using this specification. Multiple strategies may be provided for each use case in situations where the current practices differ by region or industry.

7.1 Content Insertion

7.1.1 Representing an SCTE-35 Splice Insert

Preconditions:

1. An SCTE-35 cue message is received in an MPEG-2 program stream.
2. The SCTE-35 cue message contains a splice_insert_command.
3. The target start time of the insertion is either immediate or some defined amount of time (usually seconds) after the SCTE-35 cue message is received.

Basic Flow:

1. The media time in which the SCTE-35 cue message is received is also the cue time.
2. The generated cue may be identified using the splice_event_id. This is also the value placed in the `id` attribute of the `CueContext`.
3. The `event` attribute is set to "LOAD" to indicate inserted content should be loaded or pre-staged. A second cue should be generated to signal the start of the insertion and will be described further below.
4. The `targetStartTime` attribute is set by calculating the actual time for insertion based on pts_time and pts_adjustment (if applicable). The method for computing the actual `targetStartTime` SHOULD be based on SCTE-35. When the splice_immediate_flag is set to 1, the `targetStartTime` will equal the current media time.
5. The `contentId` attribute is set to the unique_program_id.
6. The `number` attribute is set to the avail_num.
7. The `total` attribute is set to the avails_expected.
8. All other attributes and additional parameter elements are undefined for representing a splice_insert command.
9. Unless the splice_immediate_flag is set to 1, a second `CueContext` SHOULD be created with the `event` attribute set to "START" to trigger the actual start of the insertion. All other attributes should be set as above.

Alternative Flow(s):

1. When the splice_event_cancel_indicator is set to 1, the `event` attribute SHOULD be set to "CANCEL" to indicate a previously initiated insertion is being cancelled.
2. When the out_of_network_indicator is set to 0, the `event` attribute SHOULD be set to "RESUME" to indicate a previously initiated insertion is scheduled to be completed. Note that a signal will not always be provided to resume the base media content, in which case the inserted content SHOULD be played to completion before returning to the original content.

7.2 Interactive Application Event

7.2.1 Representing messages from an ETV Integrated Signaling Stream (EISS)

Preconditions:

1. EISS messages are present in the program stream.

Basic Flow:

1. The cue time is set to the current media time.
2. The `CueContext` id attribute is set to the value in the application_identifier.
3. The `event` attribute is set based on the descriptor type. For either etv_stream_event_descriptor or etv_application_metadata_descriptor, the `event` attribute will be set to “DATA”. For the etv_application_information_descriptor:
    a. Set the `event` attribute to “START” if application_control_code is “AUTOSTART”.
    b. Set the `event` attribute to “LOAD” if application_control_code is “PRESENT”.
    c. Set the `event` attribute to “TERMINATE” if application_control_code is “DESTROY”.
    d. Set the `event` attribute to “SUSPEND” if application_control_code is “SUSPEND”.
4. Set the `number` attribute to section_number.
5. Set the `total` attribute to last_section_number.
6. For etv_application_information_descriptor only:
    a. Set the `priority` attribute to the application_priority.
    b. Set the `uri` attribute to the initial_resource_locator.
7. Additional payload / private data is represented using the parameter elements.
    a. For etv_application_information_descriptor, the private_data byte array is converted to a parameter with an `id` attribute of private.
    b. For etv_stream_event_descriptor, the header_type, payload_type and payload_byte values are each converted to parameters with `id` attributes to indicate the field names.
    c. For etv_application_metadata_descriptor, each metadata item may be converted to a parameter element.
        i. Set the `id` attribute to the metadata_item_id.
        ii. Set the `type` attribute to correspond with the metadata_item_type.
        iii. Set the `value` attribute to the metadata_item_value_byte.

7.3 Content Identification

7.3.1 Representing Content Advisories from SCTE-65 Content Advisory Descriptor

Preconditions:

1. A Content Advisory Descriptor is discovered as described in SCTE-65.
2. The Rating Region Table is available or implied.

Basic Flow:

1. The cue time should always match the current media time where the descriptor is originally discovered in the program stream.
2. The `CueContext` id attribute should be set to the id of the Rating Region Table or a value of “1” if not provided.
3. The `event` attribute is set to “ADVISORY”.
4. The `id` of the `contentDescription` element SHALL be populated with the rating region as defined in [SCTE 65] that can be used to identify the appropriate region rating table.

5. A `parameter` child element SHALL be specified for each rating dimension value. The dimension index MAY be provided as the `id` attribute of the `parameter` or the dimension index could be omitted if all the defined dimensions are provided in order of index.
6. The `parameter` element's `value` attribute SHALL be populated with the dimension value corresponding to the dimension.
7. The `uri` attribute MAY be used to provide a reference to a region rating table.

An example representation using rating region 1 [US (50 states + possessions)]

```
<contentDescription id="1" event="ADVISORY">
        <parameter id="0" value="TV-PG"/>
        <parameter id="2" value="L"/>
        <parameter id="4" value="V"/>
</contentDescription>
```

Annex A XML SCHEMA (Normative)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:clc="urn:cablelabs:webvideo:cues"
targetNamespace="urn:cablelabs:webvideo:cues" elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:element name="contentAdvisory" type="clc:CueContext"/>
    <xs:element name="applicationEvent" type="clc:CueContext"/>
    <xs:element name="contentInsertion" type="clc:CueContext"/>
    <xs:element name="contentDescription" type="clc:CueContext"/>
    <xs:complexType name="CueContext">
        <xs:sequence>
            <xs:element name="parameter" type="clc:Parameter" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="name" type="xs:token"/>
        <xs:attribute name="version" type="xs:token"/>
        <xs:attribute name="event" type="xs:token"/>
        <xs:attribute name="targetStartTime" type="xs:integer"/>
        <xs:attribute name="uri" type="xs:anyURI"/>
        <xs:attribute name="contentId" type="xs:anyURI"/>
        <xs:attribute name="number" type="xs:unsignedInt"/>
        <xs:attribute name="total" type="xs:unsignedInt"/>
        <xs:attribute name="priority" type="xs:unsignedInt"/>
        <xs:anyAttribute namespace="##other"/>
    </xs:complexType>
    <xs:complexType name="Parameter">
        <xs:attribute name="name" type="xs:token"/>
        <xs:attribute name="type" type="xs:token"/>
        <xs:attribute name="value" type="xs:anySimpleType" use="required"/>
    </xs:complexType>
</xs:schema>
```

Appendix I MAPPING (Informative)

There are several types of existing in-band cue representations that may be desirable to translate to the representation in this specification. The mapping guidelines for several in-band representations are detailed in this section.

I.1 EISS

eiss_section()	Card.	Size	Type	Map?	Target
table_id	1	8	uimsbf	No	EISS-specific
section_syntax_indicator	1	1	bslbf	No	EISS-specific
reserved1	1	3	bslbf	No	EISS-specific
section_length	1	12	uimsbf	No	Length value
reserved2	1	8	uimsbf	No	EISS-specific
section_number	1	8	uimsbf	Yes	applicationEvent/@number
last_section_number	1	8	uimsbf	Yes	applicationEvent/@total
protocol_version_major	1	8	uimsbf	No	EISS-specific
protocol_version_minor	1	8	uimsbf	No	EISS-specific
application_type	1	16	uimsbf	No	0x0008 only
application_identifier()	1	48		Yes	applicationEvent/@id
application_instance_identifier_length	1	8	uimsbf	No	Length value
application_instance_identifier_data[]	0..n	8	uimsbf	Yes	applicationEvent/parameter[@id=instance]
platform_id_length	1	8	uimsbf	No	Length value
etv_bif_platform_id()	0..n			No	EISS-specific
eiss_descriptor()	0..n			Yes	(see below)
CRC_32	1	32	rpchof	No	Checked by application
etv_application_information_descriptor()					
descriptor_tag	1	8	uimsbf	No	EISS-specific
descriptor_length	1	8	uimsbf	No	Length value
application_control_code	1	8	uimsbf	Yes	applicationEvent/@event
application_version()	-	-	-	-	-
version_major	1	8	uimsbf	Yes	applicationEvent/@version
version_minor	1	8	uimsbf	Yes	applicationEvent/@version
max_protocol_version_major	1	8	uimsbf	No	EISS-specific
max_protocol_version_minor	1	8	uimsbf	No	EISS-specific
application_flags()	-	-	-	-	-
test_flag	1	8	bslbf	Yes	applicationEvent/parameter[@id=test_flag]
resource_update_flags	1	4	bslbf	No	EISS-specific
reserved	1	20	bslbf	No	EISS-specific
application_priority	1	8	uimsbf	Yes	applicationEvent/@priority
initial_resource_locator()	1			Yes	applicationEvent/@uri
private_data	0..n	8	bslbf	Yes	applicationEvent/parameter[@id=private]
etv_media_time_descriptor()					

eiss_section()	Card.	Size	Type	Map?	Target
descriptor_tag	1	8	uimsbf	No	EISS-specific
descriptor_length	1	8	uimsbf	No	Length value
time_value	1	32	uimsbf	No	Interpreted by application
etv_stream_event_descriptor()					
descriptor_tag	1	8	uimsbf	No	EISS-specific
event_counter	1	4	bslbf	No	applicationEvent/parameter
descriptor_length	1	12	uimsbf	No	Length value
time_value	1	32	uimsbf	Yes	cue/startTime
header_type	1	3	uimsbf	Yes	applicationEvent/parameter(s)
payload_type	1	5	uimsbf	Yes	applicationEvent/parameter(s)
payload_byte	0..n	8	bslbf	Yes	applicationEvent/parameter(s)
etv_application_metadata_descriptor()					
descriptor_tag	1	8	uimsbf	No	EISS-specific
reserved1	1	4	bslbf	No	EISS-specific
descriptor_length	1	12	uimsbf	No	Length value
count	1	8	uimsbf	No	Count value
metadata_item_id	0..n	24	uimsbf	Yes	applicationEvent/parameter/@id
metadata_item_type	0..n	4	bslbf	Yes	applicationEvent/parameter/@type
metadata_item_size_in_bytes	0..n	12	uimsbf	No	Length value
metadata_item_value_byte	0..n	8	bslbf	Yes	applicationEvent/parameter/@value

I.2 SCTE 35

splice_info_section()	Card.	Size	Type	Map?	Target
table_id	1	8	uimsbf	No	SCTE35-specific
section_syntax_indicator	1	1	bslbf	No	SCTE35-specific
private_indicator	1	1	bslbf	No	SCTE35-specific
reserved	1	2	bslbf	No	SCTE35-specific
section_length	1	12	uimsbf	No	Length value
protocol_version	1	8	uimsbf	No	SCTE35-specific
encrypted_packet	1	1	bslbf	No	SCTE35-specific
encryption_algorithm	1	6	uimsbf	No	SCTE35-specific
pts_adjustment	1	33	uimsbf	Yes	add to cue/startTime
cw_index	1	8	uimsbf	No	SCTE35-specific
reserved	1	12	bslbf	No	SCTE35-specific
splice_command_length	1	12	uimsbf	No	Length value
splice_command_type	1	8	uimsbf	Yes	interpreted by application
splice_null()	0..1			No	
splice_schedule()	0..1			No	Out of scope
splice_insert()	0..1			Yes	(see below)
time_signal()	0..1			Yes	(see below)

splice_info_section()	Card.	Size	Type	Map?	Target
bandwidth_reservation()	0..1			Yes	
private_command()	0..1			No	
descriptor_loop_length	1	16	uimsbf	No	Length value
splice_descriptor()	0..n			Yes	(see below)
alignment_stuffing	0..n	8	bslbf	No	SCTE35-specific
E_CRC_32	0..1	32	rpchof	No	SCTE35-specific
CRC_32	1	32	rpchof	Yes	checked by application
splice_schedule()				No	Out of scope
splice_count	1	8	uimsbf	No	Out of scope
splice_event_id	0..n	32	uimsbf	Yes	Out of scope
splice_event_cancel_indicator	0..n	1	bslbf	Yes	Out of scope
reserved	0..n	7	bslbf	No	Out of scope
out_of_network_indicator	0..n	1	bslbf	Yes	Out of scope
program_splice_flag	0..n	1	bslbf		Out of scope
duration_flag	0..n	1	bslbf		Out of scope
reserved	0..n	5	bslbf		Out of scope
utc_splice_time	0..n	32	uimsbf	Yes	Out of scope
component_count	0..n	8	uimsbf		Out of scope
component_tag	0..m	8	uimsbf		Out of scope
utc_splice_time	0..m	32	uimsbf		Out of scope
break_duration()	0..n				Out of scope
auto_return	0..n	1	bslbf		Out of scope
reserved	0..n	6	bslbf		Out of scope
duration	0..n	33	uimsbf	Yes	Out of scope
unique_program_id	0..n	16	uimsbf	Yes	Out of scope
avail_num	0..n	8	uimsbf	Yes	Out of scope
avails_expected	0..n	8	uimsbf	Yes	Out of scope
splice_insert()					
splice_event_id	1	32	uimsbf	Yes	contentInsertion/@id
splice_event_cancel_indicator	1	1	bslbf	Yes	contentInsertion/@event[CANCEL]
reserved	1	7	bslbf	No	SCTE35-specific
out_of_network_indicator	0..1	1	bslbf		contentInsertion/event[varies]
program_splice_flag	0..1	1	bslbf	No	SCTE35-specific
duration_flag	0..1	1	bslbf	No	SCTE35-specific
splice_immediate_flag	0..1	1	bslbf	Yes	set cue/startTime to current time
reserved	0..1	4	bslbf	No	SCTE35-specific
splice_time()					
time_specified_flag	0..1	1	bslbf	No	SCTE35-specific
reserved	0..1	6	bslbf	No	SCTE35-specific
pts_time	0..1	33	uimsbf	Yes	cue/startTime
component_count	0..1	8	uimsbf	No	Count value

splice_info_section()			Card.	Size	Type	Map?	Target
	component_tag		0..n	8	uimsbf	No	Out of scope
	splice_time()		0..n			No	Out of scope
		time_specified_flag	0..n	1	bslbf	No	Out of scope
		reserved	0..n	6	bslbf	No	Out of scope
		pts_time	0..n	33	uimsbf	No	Out of scope
	break_duration()		0..1				
		auto_return	0..1	1	bslbf	No	
		reserved	0..1	6	bslbf	No	SCTE35-specific
		duration	0..1	33	uimsbf	Yes	Use to calculate cue/endTime
	unique_program_id		0..1	16	uimsbf	Yes	contentInsertion/@contentId
	avail_num		0..1	8	uimsbf	Yes	contentInsertion/@number
	avails_expected		0..1	8	uimsbf	Yes	contentInsertion/@total
time_signal()							
	splice_time()						
		time_specified_flag		1	bslbf	No	SCTE35-specific
		reserved		6	bslbf	No	SCTE35-specific
		pts_time		33	uimsbf	Yes	cue/startTime
private_command()							
	identifier			32	uimsbf	No	Out of scope
	private_byte		0..n	8	uimsbf	No	Out of scope
avail_descriptor()							
	splice_descriptor_tag		1	8	uimsbf	No	SCTE35-specific
	descriptor_length		1	8	uimsbf	No	Length value
	identifier		1	32	uimsbf	No	SCTE35-specific
	provider_avail_id		1	32	uimsbf	No	Out of scope
DTMF_descriptor()							
	splice_descriptor_tag		1	8	uimsbf	No	SCTE35-specific
	descriptor_length		1	8	uimsbf	No	Length value
	identifier		1	32	uimsbf	No	SCTE35-specific
	preroll		1	8	uimsbf	No	Out of scope
	dtmf_count		1	3	uimsbf	No	Count value
	reserved		1	5	bslbf	No	SCTE35-specific
	DTMF_char		0..n	8	uimsbf	No	Out of scope
segmentation_descriptor()							
	splice_descriptor_tag		1	8	uimsbf	No	SCTE35-specific
	descriptor_length		1	8	uimsbf	No	Length value
	identifier		1	32	uimsbf	No	SCTE35-specific
	segmentation_event_id		1	32	uimsbf	Yes	contentInsertion/@id

splice_info_section()		Card.	Size	Type	Map?	Target
	segmentation_event_cancel_indicator	1	1	bslbf	Yes	contentInsertion/@event[CANCEL]
	reserved	1	7	bslbf	No	SCTE35-specific
	program_segmentation_flag	1	1	bslbf	No	SCTE35-specific
	segmentation_duration_flag	1	1	bslbf	No	SCTE35-specific
	reserved	1	6	bslbf	No	SCTE35-specific
	component_count	1	8	uimsbf	No	Count value
	component_tag	0..n	8	uimsbf	No	Out of scope
	reserved	0..n	7	bslbf	No	Out of scope
	pts_offset	0..n	33	uimsbf	No	Out of scope
	segmentation_duration	0..1	40	uimsbf	Yes	used to calculate cue/endTime
	segmentation_upid_type	1	8	uimsbf	Yes	contentInsertion/@contentId (prefix)
	segmentation_upid_length	1	8	uimsbf	No	Length value
	segmentation_upid()	1	variable		Yes	contentInsertion/@contentId
	segmentation_type_id	1	8	uimsbf	Yes	contentInsertion/@event[Varies]
	segment_num	1	8	uimsbf	Yes	contentInsertion/@number
	segments_expected	1	8	uimsbf	Yes	contentInsertion/@total

Appendix II Examples (Informative)

II.1 SCTE 35 Splice Insert as contentInsertion

This example shows a simple SCTE 35 splice insert as converted to a `contentInsertion` element. Since this element only contains the context, the base timing information is represented in the enclosing cue representation (not shown).

```
<contentInsertion id="6543" version="1" event="INSERT" targetStartTime="4000" uri="http://ads.mso.com"
number="1" total="3" xmlns="urn:cablelabs:webvideo:cues"></contentInsertion>
```

II.2 TTML

This example shows how an `applicationEvent` element might be represented as part of a TTML document. Note that the timing information is provided as part of the TTML div tag.

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:tt xml:space="preserve" xml:lang="en-us" ttp:timeBase="media" xsi:schemaLocation="http://www.w3.org/ns/ttml
ttaf1-dfxp-document.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:tt="http://www.w3.org/ns/ttml" xmlns:ttm="http://www.w3.org/ns/ttml#metadata"
xmlns:ttp="http://www.w3.org/ns/ttml#parameter" xmlns:tts="http://www.w3.org/ns/ttml#styling">
        <tt:head/>
        <tt:body>
                <tt:div xml:id="X1" begin="5.0s" end="10.0s">
                        <tt:metadata>
                                <applicationEvent name="xyz" version="1.0" event="LOAD"
uri="http://app.com/myApp" priority="0" number="1" total="3" xmlns="urn:cablelabs:webvideo:cues">
                                        <parameter type="string" value="something" name="1"/>
                                        <parameter type="unsignedInt" value="42" name="2"/>
                                </applicationEvent>
                        </tt:metadata>
                </tt:div>
        </tt:body>
</tt:tt>
```

II.3 WebVTT

This example shows how an `applicationEvent` element might be embedded within a WebVTT format. Note that the timing information is provided by using the WebVTT format.

```
WEBVTT FILE

1
00:00:05.000 --> 00:00:10.000
<applicationEvent name="xyz" version="1.0" event="LOAD" uri="http://app.com/myApp" priority="0" number="1"
total="3" xmlns="urn:cablelabs:webvideo:cues">
        <parameter type="string" value="something" name="1"/>
        <parameter type="unsignedInt" value="42" name="2"/>
</applicationEvent>
```

Appendix III Use in HTML5 Timed text tracks (Informative)

The XML types defined by this specification can be used as the content of a `TextTrackCue` in HTML5. Additionally, this same format can be carried in common text track expression languages (e.g., TTML, WebVTT) as well as in-band with video formats that allow XML or arbitrary text to be transported. Figure 1 shows opportunities for expressing TV Services as XML. The primary use case is as an XML representation of the message in the media resource. An additional use case is that the same XML representation can be used as the content of a `TextTrackCue`, which can be exposed to the client application. A library is envisioned that will convert non-XML formats to this representation and re-interpret the metadata if appropriate. For example, an in-band signal for advertising insertion may imply an additional cue to pre-load the ad content prior to another cue required to play the ad content. Definition of the conversion library is out of scope of this document.

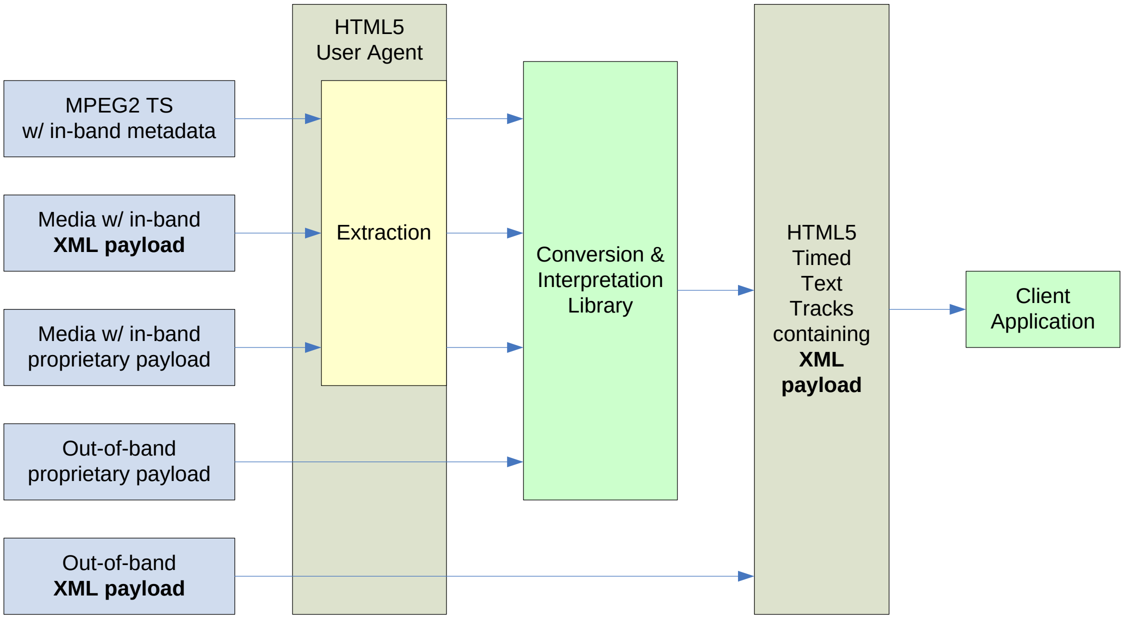

Figure 6 - Usage of XML with HTML5 TimedTextTracks and Client Library