ETSI TS 103 286-2 V1.2.1 (2017-08)

TECHNICAL SPECIFICATION

Digital Video Broadcasting (DVB);
Companion Screens and Streams;
Part 2: Content Identification and Media Synchronization

Reference

RTS/JTC-DVB-374-2

Keywords

broadcast, companion screen, DVB,
synchronization, TV

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECtrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel: +41 22 717 21 11
Fax: +41 22 717 24 81

The Digital Video Broadcasting Project (DVB) is an industry-led consortium of broadcasters, manufacturers, network operators, software developers, regulatory bodies, content owners and others committed to designing global standards for the delivery of digital television and data services. DVB fosters market driven solutions that meet the needs and economic circumstances of broadcast industry stakeholders and consumers. DVB standards cover all aspects of digital television from transmission through interfacing, conditional access and interactivity for digital video, audio and data. The consortium came together in 1993 to provide global standardization, interoperability and future proof specifications.

The present document is part 2 of a multi-part deliverable covering the DVB Companion Screens and Streams Specification, as identified below:

Part 1: "Concepts, roles and overall architecture";

Part 2: "Content Identification and Media Synchronization";

Part 3: "Discovery".

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

It is assumed that the reader is familiar with ETSI TS 103 286-1 [i.10] that provides background information on the concepts covered in the present document.

Personal, smart devices like tablet computers and smart phones enable new user experiences for broadcast service consumption. Many of these new experiences require synchronization between the broadcast content presented on the TV Device and the content presented on the personal device.

The present document enables the identification of, and synchronization with timed content and trigger events on TV devices (for example a Connected TV or STB) and related content presented by an application running on a personal device. Example use cases are:

- presenting a question and a choice of possible answers on a personal device, which are related to what is currently happening on a quiz show that is the current TV Programme;
- alternative audio intended to be consumed through the personal device (e.g. via connected headphones) and along with the broadcast video content on the TV device, such as an alternative commentary, an alternative language, clean audio for hearing impaired or audio descriptions for the visually impaired;
- seamlessly switching between different camera views on the personal device synchronously with a football game on the TV Device;
- presenting advertisements on the personal device which are related to the broadcast content, and in synchronization with the broadcast content (for example an advertisement for a product that is currently prominently visible in the broadcast video); and
- presenting a targeted advertisement to the user on the personal device at the time of presenting a generic interstitial in the broadcast content.

To enable such use cases, two functionalities are needed. The first functionality is the identification of broadcast content and finding of associated content for presentation on the personal device. The identification mechanisms defined in the present document are hence designed to take the following limiting factors into account:

- Different broadcast platforms may use different solutions to identify broadcast programmes.
- Synchronized transmission of broadcast and personal content through different transmission infrastructures is difficult.
- Broadcast platforms are bandwidth sensitive, and the amount of data needed for identification and synchronization should hence be kept to a minimum.
- Interactions of the personal device or the TV device with resources over broadband connections may take significant time. Furthermore, in a series of requests, the amount of time spent on each of them can vary largely and unpredictably.
- Applications presenting content on the personal device, and which are controlled by trigger events, need to identify of the content currently being presented by the TV device and determine the associated content for presentation on the personal device. In addition they also will need to subscribe to, and receive notifications of, any trigger event signalling in the broadcast service.

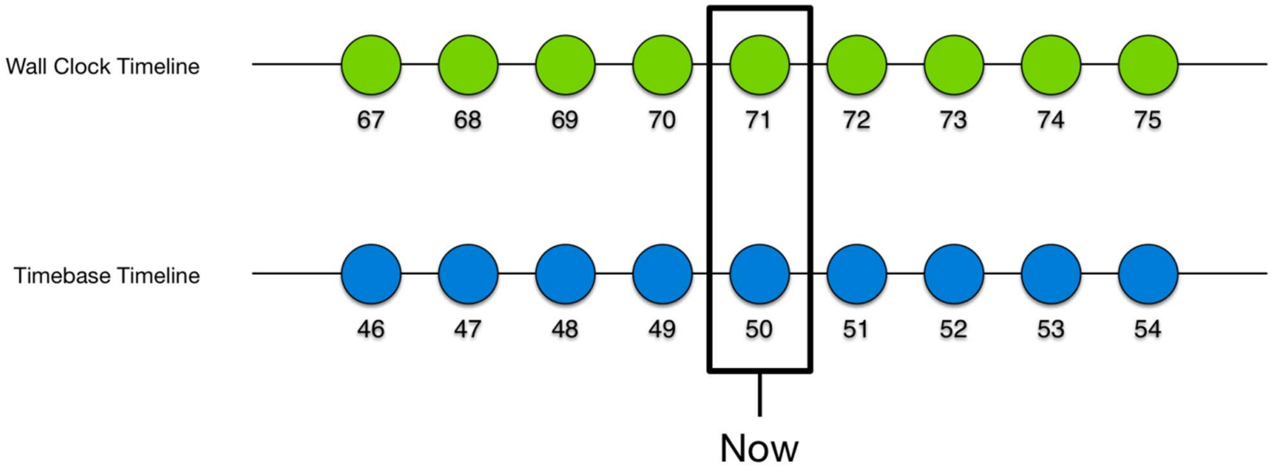
The second functionality is the synchronization of playback on the personal device with the playback on the TV device. Typically, an accuracy of at least 40 ms is required for frame-accurate synchronization between two video streams or lip sync between audio and video. The synchronicity between pieces of content is influenced by several factors:

- Propagation delays are different for different transmission networks and technologies, like terrestrial broadcast, satellite broadcast, IP multicast, and IP unicast; this can lead to arrival time differences of up to ten secs when transmitting through different paths.
- If the media is delivered via a Content Delivery Network (CDN), a significant amount of time (often 30 secs or more) is needed to ingest the content into the CDN before it becomes available for retrieval.
- Media processing function like transcoding can take up to several seconds of time which may limit their applicability to certain use cases.
- Streaming delivery through switched packet networks (for instance typical broadband Internet services) requires buffering for smooth presentation of media content. The size of the buffer depends on the technology used and the presence of any local post-processing for quality enhancement. This may lead to additional delays of up to 500 ms at the receiving device.

The present document provides an architectural framework for timeline synchronization between a presentation on one device and associated Timed Content on another, optionally using trigger events. The present document provides the protocol interfaces to provide this functionality given the limiting and influencing factors described above. These mechanisms are designed to take the following additional limiting factors into account:

- Related pieces of content may have different types of Timelines, with different tick rates and resolutions.
- The clocks of related pieces of content may exhibit different error properties (e.g. drift or jitter) if these clocks are not genlocked ("generator locked", i.e. synchronized at the source).
- As a consequence of processing during content production or distribution, timelines may be offset between different related pieces of content, even if they are of the same type and if clocks are genlocked.
- Timelines embedded into content (e.g. PTS for MPEG TS, or composition time of ISOBMFF) may be modified by the distribution network due to multiplexing, transcoding, and re-origination.
- Timelines transmitted along with content may be removed by distribution networks.
- Timelines can, and will, include discontinuities.
- Depending on the particular encoding of timestamps, some timelines will wrap around during presentation, as only a limited number of bits is available to express a Time Value on such Timeline.
- The system clocks of TV devices and personal devices run independently, and will hence exhibit different error behaviour (e.g. drift).

Figure 0.1 shows a basic, conceptual model for time-controlled playback. A local wall clock advances steadily, and the playback of the media streams is timed accordingly to achieve a smooth presentation. To enable this, the media streams are adorned with their own timebase timeline, which is compared to the wall clock timeline. During playback, whenever the wall clock timeline advances, the media player computes the corresponding point on the media's timebase timeline, and retrieves the associated chunk of media data for playback. To achieve time-controlled and smooth playback the media player will typically apply an offset to the media stream's timebase timeline and also adjust the playback rate of the media stream's timebase timeline in these computations. As the playback proceeds over time, media players will typically make dynamic re-adjustments of the offset and playback rate, to accommodate variations in the wall clock's progress, and in the delivery of the media stream.



Wall Clock Timeline		67	68	69	70	71	72	73	74	75
Timebase Timeline	rate 0.0	50	50	50	50	50	50	50	50	50
	rate 1.0	46	47	48	49	50	51	52	53	54
	rate -1.0	54	53	52	51	50	49	48	47	46
	rate 2.0	42	44	46	48	50	52	54	56	58

Figure 0.1: Basic model of time-controlled playback

Extending from this basic model, figure 0.2 shows how the playback of two independent media streams can be controlled on two independent media players in a coordinated fashion. To adapt the playback of the two media streams - for example to make the audio from one suitable for consumption with the video from the other - both the wall clocks and the media stream playback need to be coordinated between the two players. In the context of the present document, this happens by exchanging information between the two players across a home network. All mechanisms and solutions defined in the present document will build on and extend from this basic conceptual model. These solutions are not limited to audio-visual content but cover any type of timed content, for example subtitles, trigger events received in the broadcast and timed content generated locally by an application running on either of the devices (e.g. in the quiz show use case mentioned above).

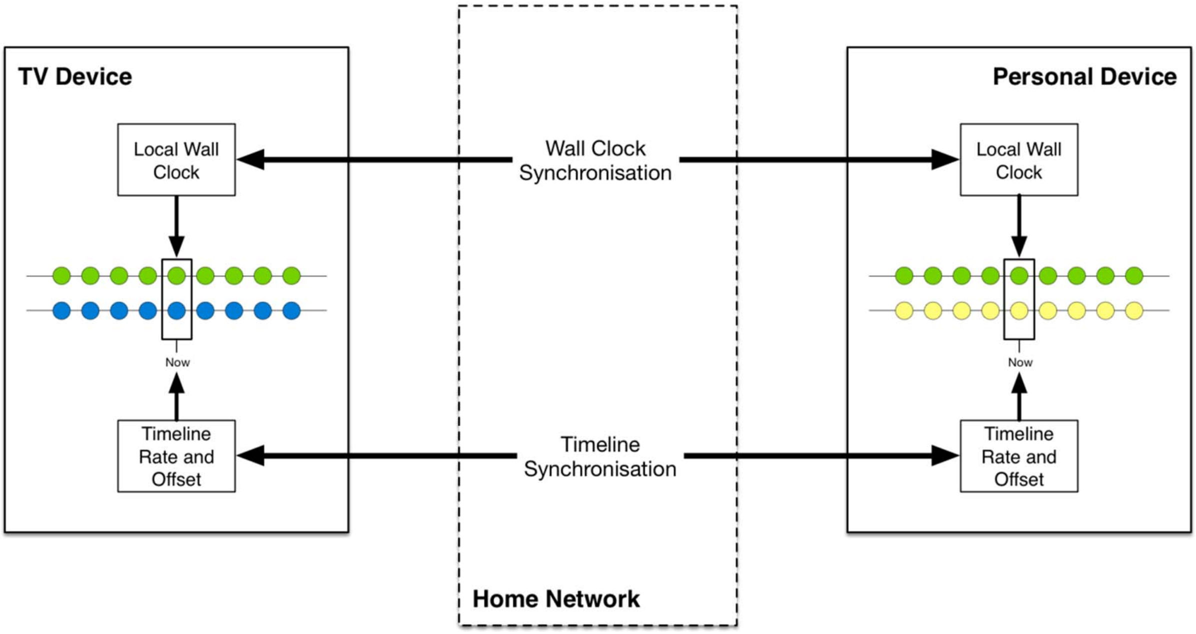


Figure 0.2: Basic model of synchronizing playback between devices

1 Scope

The present document specifies the architecture and protocols for content identification, Timeline Synchronization and Trigger Events for companion screens and streams.

The present document is applicable to:

- the interfaces between the TV Device and the Companion Screen Application:
  - interface for metadata exchange, including content identification;
  - interface for Wall Clock synchronization;
  - interface for Timeline Synchronization;
  - interface for Trigger Events;
- the interface between the Companion Screen Application and the Material Resolution Service (MRS).

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] IETF RFC 3986 (2005): "Uniform Resource Identifier (URI): Generic Syntax".

NOTE: Available at http://www.ietf.org/rfc/rfc3986.txt.

[2] ETSI TS 102 323 (V1.5.1): "Digital Video Broadcasting (DVB); Carriage and signalling of TV-Anytime information in DVB transport streams".

NOTE: Available at http://www.etsi.org/deliver/etsi_ts/102300_102399/102323/01.05.01_60/.

[3] ETSI TS 102 851 (V1.3.1): "Digital Video Broadcasting (DVB); Uniform Resource Identifiers (URI) for DVB Systems".

NOTE: Available at http://www.etsi.org/deliver/etsi_ts/102800_102899/102851/01.03.01_60/.

[4] IETF RFC 5234 (2008): "Augmented BNF for Syntax Specifications: ABNF".

NOTE: Available at http://www.ietf.org/rfc/rfc5234.txt.

[5] ETSI TS 102 034: "Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks".

NOTE: Available at http://www.etsi.org/deliver/etsi_ts/102000_102099/102034/.

[6] ISO/IEC 13818-1:2015: "Information technology -- Generic coding of moving pictures and associated audio information -- Part 1: Systems".

[7] ISO/IEC 14496-12:2015: "Information technology -- Coding of audio-visual objects -- Part 12: ISO base media file format".

[8] IETF RFC 6455 (2011): "The WebSocket Protocol".

NOTE: Available at http://www.ietf.org/rfc/rfc6455.txt.

[9] IETF RFC 2616 (1999): "Hypertext Transfer Protocol -- HTTP/1.1".

NOTE: Available at http://www.ietf.org/rfc/rfc2616.txt.

[10] IETF RFC 2818 (2000): "HTTP Over TLS".

NOTE: Available at http://www.ietf.org/rfc/rfc2818.txt.

[11] IETF RFC 4627 (2006): "The application/json Media Type for JavaScript Object Notation (JSON)".

NOTE: Available at http://www.ietf.org/rfc/rfc4627.txt.

[12] IETF RFC 2916 (2000): "E.164 number and DNS".

NOTE: Available at http://www.ietf.org/rfc/rfc2916.txt.

[13] ETSI EN 300 468: "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".

[14] W3C Recommendation: "Cross-Origin Resource Sharing", 16 January 2014.

NOTE: Available at http://www.w3.org/TR/cors/.

[15] IETF RFC 6454: "The Web Origin Concept".

NOTE: Available at http://tools.ietf.org/html/rfc6454.

[16] ETSI TS 103 285: "Digital Video Broadcasting (DVB); MPEG-DASH Profile for Transport of ISO BMFF Based DVB Services over IP Based Networks".

[17] ISO/IEC 23009-1:2014: "Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats".

[18] W3C Recommendation: "Media Fragments URI 1.0 (basic)", 25 September 2012.

NOTE: Available at http://www.w3.org/TR/media-frags/.

[19] W3C Recommendation: "Server-Sent Events", 03 February 2015.

NOTE: Available at http://www.w3.org/TR/eventsource/.

[20] IETF RFC 6902 (2013): "JavaScript Object Notation (JSON) Patch".

[21] ETSI ES 202 184 (V2.3.1): "MHEG-5 Broadcast Profile".

NOTE: Available at http://www.etsi.org/deliver/etsi_es/202100_202199/202184/02.03.01_60/.

[22] ETSI TS 102 727 (V1.1.1): "Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.2.2".

NOTE: Available at http://www.etsi.org/deliver/etsi_ts/102700_102799/102727/01.01.01_60/.

[23] ETSI TS 102 809 (V1.3.1): "Digital Video Broadcasting (DVB); Signalling and carriage of interactive applications and services in Hybrid broadcast/broadband environments".

NOTE: Available at http://www.etsi.org/deliver/etsi_ts/102800_102899/102809/01.03.01_60/.

[24] IETF RFC 4648 (2006): "The Base16, Base32, and Base64 Data Encodings".

NOTE: Available at http://www.ietf.org/rfc/rfc4648.txt.

[25] ISO/IEC 13818-1:2015/Amd 1: 2015: "Amendment 1 - Delivery of timeline for external data".

NOTE: This document has published corrigenda.

[26] IETF RFC 2141 (1997): "URN Syntax".

NOTE: Available at http://www.ietf.org/rfc/rfc2141.txt.

[27] ETSI TS 102 539: "Digital Video Broadcasting (DVB); Carriage of Broadband Content Guide (BCG) information over Internet Protocol (IP)".

[28] ANSI INCITS 4-1986 (R2007): "Information Systems - Coded Character Sets - 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII)".

[29] IETF RFC 6120: "Extensible Messaging and Presence Protocol (XMPP): Core".

NOTE: Available at http://www.ietf.org/rfc/rfc6120.txt.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TS 182 027 (V3.5.1): "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); IPTV Architecture; IPTV functions supported by the IMS subsystem".

NOTE: Available at http://www.etsi.org/deliver/etsi_ts/182000_182099/182027/03.05.01_60/.

[i.2] IETF RFC 5905 (June 2010): "Network Time Protocol Version 4: Protocol and Algorithm Specification", D. Mills, U. Delaware, J. Martin, J. Burbank and W. Kasch.

NOTE: Available at http://www.ietf.org/rfc/rfc5905.txt.

[i.3] IETF RFC 3339 (July 2002): "Date and Time on the Internet: Timestamp", G. Klyne and C. Newman.

NOTE: Available at http://www.ietf.org/rfc/rfc3339.txt.

[i.4] Mark Kalman, Eckehard Steinbach, and Bernd Girod: "Adaptive Media Playout for Low-Delay Video Streaming Over Error-Prone Channels", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No. 6, June 2004.

NOTE: Available at http://www.stanford.edu/~bgirod/pdfs/KalmanCSVT2004.pdf.

[i.5] ETSI TS 182 019 (V3.1.2): "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Content Delivery Network (CDN) Architecture".

NOTE: Available at http://www.etsi.org/deliver/etsi_ts/182000_182099/182019/03.01.02_60/.

[i.6] Recommendation ITU-T H.264 (04/2013): "Advanced video coding for generic audiovisual services".

NOTE: Available at http://www.itu.int/rec/T-REC-H.264-201304-S/en.

[i.7] IETF draft-zyp-json-schema-04 (January 2013): "JSON Schema: core definitions and terminology", F. Galiegue, K. Zyp and G. Court.

NOTE: Available at http://tools.ietf.org/id/draft-zyp-json-schema-04.txt.

[i.8] IETF draft-fge-json-schema-validation-00 (February 2013): "JSON Schema: interactive and non interactive validation", Fge. Galiegue, K. Zyp and G. Court.

NOTE: Available at http://tools.ietf.org/html/draft-fge-json-schema-validation-00.

[i.9] HDMI 2.0 specification.

NOTE: An introduction to the HDMI 2.0 specification is provided at http://www.hdmi.org/manufacturer/hdmi_2_0/. The specification itself is not publicly available.

[i.10] ETSI TS 103 286-1: "Digital Video Broadcasting (DVB); Companion Screens and Streams; Part 1: Concepts, roles and overall architecture".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

active material: material that is considered to represent an aspect of the editorial content currently showing at this point in time on the TV Device

NOTE: Zero, one or more than one Material could be considered to be an Active Material at a given point on a Broadcast Timeline.

actual presentation timestamp: timestamp representing the actual time (in relation to the Wall Clock) at which a Synchronization Client is presenting a particular point in time (in relation to the Synchronization Timeline) of its Timed Content

NOTE: This is represented as a pair of values, consisting of a Time Value on a Synchronization Timeline and a Time Value from a Wall Clock.

broadcast timeline: timeline derived from signalling in a broadcast stream

companion device: IP-connected device such as a mobile phone, tablet, laptop or bespoke accessibility hardware (e.g. a single switch interface)

companion screen application: application that runs on a Companion Device and provides an experience related to the Timed Content that the user is watching on a TV Device

companion screen device: See companion device.

NOTE: This is a synonym.

content identifier: composite identifier that unambiguously identifies the content being viewed on the TV Device

NOTE: Typically a Content Identifier will identify content to the granularity of an individual programme or presentation.

control timestamp: timestamp representing the time (in relation to the Wall Clock) at which a Synchronization Client is recommended to present a particular point in time (in relation to the Synchronization Timeline) of its Timed Content

NOTE: This consists of a pair of values of a Time Value on a Synchronization Timeline and a Time Value from a Wall Clock and other information describing the speed of presentation.

correlation information server: service that provides Correlation Timestamps

correlation timestamp: Time Values on 2 Timelines that correlate to each other

EXAMPLE: A Correlation Timestamp could represent a correlation between a point on the Synchronization Timeline and a point on the Material Timeline.

COS device: See companion device.

NOTE: This is a synonym.

earliest presentation timestamp: timestamp representing the earliest time (in relation to the Wall Clock) at which a Synchronization Client believes it can present a particular point in time (in relation to the Synchronization Timeline) of its Timed Content

NOTE: This is represented as a pair of values consisting of a Time Value on a Synchronization Timeline and a Time Value from a Wall Clock.

generator locked: See genlocked.

NOTE: This is a synonym.

genlocked: synchronized with respect to their clock ticks

home network: collection of one or more subnets of an IP network to which devices are connected (wired or wirelessly) within the typical home environment

JSON document: textual document consisting of data represented in JSON format rooted at a JSON object or array

latest presentation timestamp: timestamp representing the latest time (in relation to the Wall Clock) at which a Synchronization Client believes it can present a particular point in time (in relation to the Synchronization Timeline) of its Timed Content

NOTE: This is represented as a pair of values consisting of a Time Value on a Synchronization Timeline and a Time Value from a Wall Clock.

link proxy: service, accessed via the Internet, that proxy's communication between the TV Device and a Companion Screen Application

NOTE: Intended for scenarios where the Companion Screen Application is not connected to the same Home Network as the TV Device.

material: unique piece or segment of final editorial content

NOTE: This can include (but is not limited to): a one-off TV Programme, an episode of a TV series, an advertisement, trailer or some other Interstitial.

material identifier: identifier, that has type and value, for Material

material information: collection of information describing Materials and their relationship to the timeline of a broadcast

Material Resolution Service (MSR): service providing Material Information relevant to a particular CI

media synchronization application server: elementary function that coordinates the process of obtaining a shared agreement on the progress of Timelines among all Synchronization Clients

NOTE: This is for the purpose of enabling the Synchronization Clients to present Timed Content simultaneously with respect to each other.

synchronization client: elementary function that wishes to align its presentation of Timed Content with other Synchronization Clients by communicating with a Media Synchronization Application Server

synchronization timeline: timeline used in communication between a Synchronization Client and the MSAS to give the Synchronization Client an understanding of the progress of time along that timeline

tick rate: speed of a clock

NOTE: Usually measured in ticks/seconds.

time value: time position expressed as a numeric value

timed content: audio, video or any other type of streamed or file based media or application generated content whose presentation is aligned with a Timeline

timeline: reference frame for describing time, represented as a linear scale against which time can be measured for a particular system

NOTE: This could manifest in various ways, such as: as a local oscillator, the progress of a broadcast or the time position within an item of media content.

timeline correlation: correlation between two Timelines

timeline mapping: correlation that is bounded to an interval of time

timeline selector: string describing the type and location of timeline signalling

timeline synchronization: achieving a shared understanding of the progress of time along a Synchronization Timeline

NOTE: Typically a sharing of understanding between an MSAS and an SC, or between an MSAS and the collection of SCs; coordinated by the MSAS.

timestamp: pair of two values each of which represents a Time Value on a timeline such that the two Time Values correspond to the same moment in time

trigger event: timestamped notification of a point in the broadcast

TV device: television or set-top-box device that receives and renders DVB broadcast or IPTV television services, or other Timed Content and is connected to a Home Network

NOTE: It may receive broadcast content from the DVB network (including IPTV) and may receive on-demand content and may be capable of recording and playing back DVB services. The TV Device can be a horizontal (e.g. TV set) or vertical market device (e.g. IPTV STB).

wall clock: linear monotonic clock that is not assumed to represent real date and time

NOTE: The Wall Clock is intended for sharing between two or more entities for the purposes of having a common synchronized time reference frame.

wall clock synchronization: achieving a shared agreement on the progress of time of a Wall Clock

NOTE: This is a process that takes place between devices.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

4CC	4 Character Code
ABNF	Augmented BNF
AIT	Application Information Table
AMP	Adaptive Media Playout
API	Application Programming Interface
APT	Actual Presentation Timestamp
ASCII	American Standard Code for Information Interchange
AV	Audio and Video
BAT	Bouquet Association Table
BCG	Broadband Content Guide
BNF	Backus Naur Form
CDN	Content Delivery Network
CI	Content Identifier
CII	Content Identification and other Information
CIS	Correlation Information Server
CORS	Cross Origin Resource Sharing
COS	Companion Screen
CRID	Content Reference IDentifier
CSA	Companion Screen Application
CSS	Companion Screens and Streams
CSS-CII	Interface for Content Identification and other Information
CSS-MRS	Interface for Material Resolution Service

CSS-TE	Interface for Trigger Events
CSS-TS	Interface for Timeline Synchronization
CSS-WC	Interface for Wall Clock
CT	Composition Time
DAM	Draft AMmendment
DASH	Dynamic Adaptive Streaming over HTTP
DSM-CC	Digital Storage Media - Command and Control
EIT	Event Information Table
EPG	Electronic Program Guide
EPT	Earliest Presentation Timestamp
FCC	Fast Channel Change
GPS	Global Positioning System
HDMI	High-Definition Multimedia Interface
HTML	Hyper Text Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HTTP Secure
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPTV	Internet Protocol based Television
ISO	International Standards Organization
ISOBMFF	ISO Base Media File Format
ITU-T	International Telecommunications Union - Telecommunication
JSON	JavaScript Object Notation
LPT	Latest Presentation Timestamp
MI	Material Information
MIME	Multipurpose Internet Mail Extensions
MPD	Media Presentation Description
MPEG	Multimedia Pictures Expert Group
MRS	Material Resolution Service
MSAS	Media Synchronization Application Server
NIT	Network Information Table
NSS	Name Specific String
NTP	Network Time Protocol
OTT	Over The Top
PCR	Program Reference Clock
PES	Program Elementary Stream
PMT	Program Map Table
PTS	Presentation Timestamp
PVR	Personal Video Recorder
REST	REpresentational State Transfer
RET	RETransmission
RFC	Request For Comments
SC	Synchronization Client
SD&S	Service Discovery and Selection
SDT	Service Description Table
SEI	Supplemental Enhancement Information
SI	Service Information
SM	Stream Monitor
SSE	Server Sent Events
STB	Set Top Box
TCP	Terminal Control Protocol
TE	Trigger Event
TEMI	Timed External Media Information
TEN	Trigger Event Notification
TESM	Trigger Event Subscription Management
TESS	Trigger Event Session Setup
TLS	Transport Layer Security
TS	MPEG-2 Transport Stream
TSAP	Transport Stream Adaptation Private data
TV	Television
UDP	User Datagram Protocol
UML	Universal Modelling Language

URI	Uniform Resource Identifier
URL	Universal Resource Locator
URN	Uniform Resource Name
UTC	Coordinated Universal Time
VITC	Vertical Interval Timecode
WC	Wall Clock
XML	eXtensible Markup Language
XMPP	eXtensible Messaging and Presence Protocol

4 Overview

4.1 Interfaces

ETSI TS 103 286-1 [i.10] describes the functional roles, relationships and general architecture within which the specific functionality of the present document operates.

The present document defines the following interfaces from the general architecture:

- Content Identification and other Information (CSS-CII).
- Material Resolution (CSS-MRS).
- Wall Clock (CSS-WC).
- Timeline Synchronization (CSS-TS).
- Trigger Events (CSS-TE).

Support by the TV Device for the CSS-CII, CSS-WC, CSS-TS and CSS-TE interfaces is optional, though the TV Device should support at least the CSS-CII interface. Use of the CSS-MRS interface by the CSA and provision of them by the broadcaster, or a third party, is also optional, as indicated by the presence of the signalling in the content that conveys the location of the relevant server (as described in clause 5.6.2).

If a TV Device implements CSS-TS or CSS-TE, it should implement CSS-WC as the CSS-TS and CSS-TE interfaces refer to the wall clock provided by the CSS-WC.

4.2 Architecture for identification and companion synchronization

4.2.1 General

For a Companion Screen Application (CSA) to be able to present Timed Content synchronized with a TV Device, information needs to flow between the TV Device and the CSA and between the CSA and supporting services on the Internet. The present document assumes that the TV Device only presents a single piece of content at any one time.

This clause recaps the architectural description of interfaces and their functions provided in ETSI TS 103 286-1 [i.10].

The main data flows for the architecture is illustrated in figure 4.2.1.1.

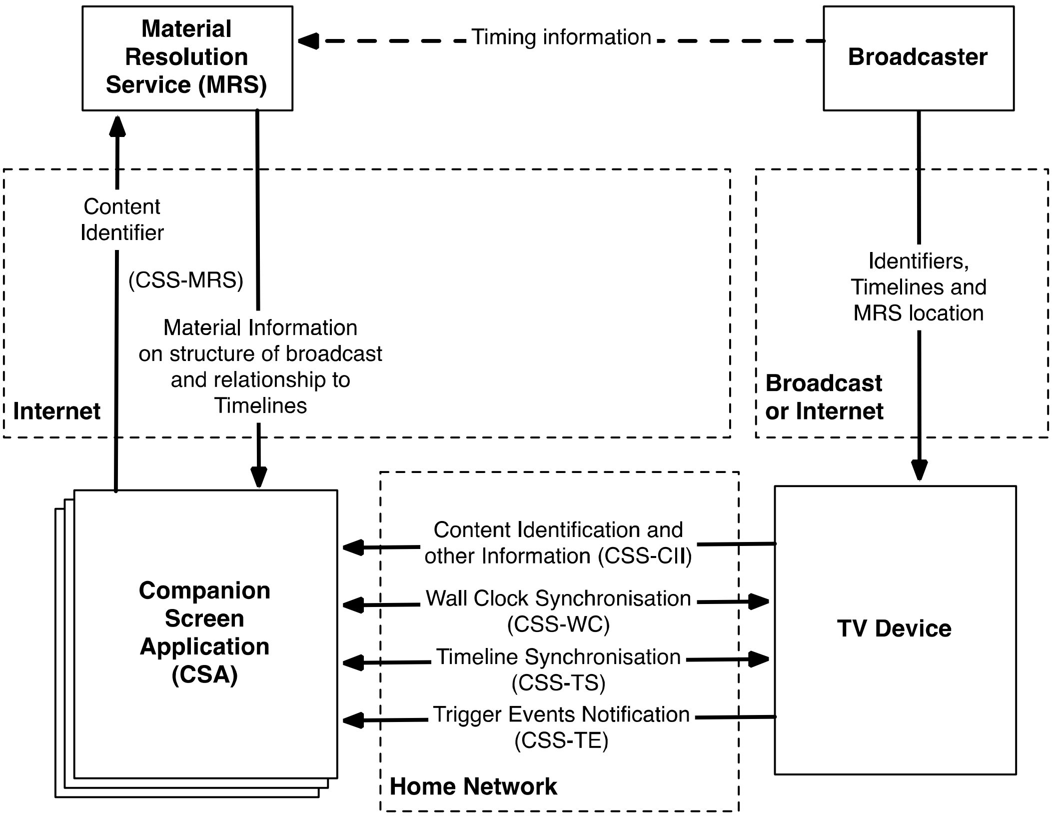


Figure 4.2.1.1: Data flows for companion synchronization architecture

Service endpoints are the locations of implementations of interfaces on a network.

For companion synchronization it is assumed that the CSA has already discovered and (if required) associated with the TV Device and knows the service endpoint of the Content Identification and other Information interface (CSS-CII) that is implemented by the TV Device.

In so far as is possible, an absolute minimum of new signalling and standardization is introduced in the present document with the goal of allowing current broadcasts to work within the framework of the present document. Thus wherever possible existing signalling is re-used and, rather than selecting a single option, multiple signalling mechanisms are supported.

Multiple CSAs may independently be able to synchronize their presentation of Timed Content with a single TV Device.

The TV Device receives broadcasts from a broadcaster. The broadcast includes various broadcast identifiers for the overall stream and the services within it. Content identifying information is communicated via the CSS-CII interface to the CSA from the TV Device.

The CSA may understand this information directly and be able to recognize what content is being presented. Alternatively, it may pass this information to a Material Resolution Service (MRS) via the CSS-MRS interface in order to obtain that understanding. A CSA may already know the location of the service endpoint for the CSS-MRS interface. Alternatively, this can be signalled in the broadcast and communicated from the TV Device to the CSA via the CSS-CII interface.

The MRS provides Material Information. This information describes the editorial structure of the broadcast in terms of a collection of Materials representing editorial segments of the broadcast (such as whole programmes, programme segments, adverts or trailers). It may also include the information needed to relate the timeline of each Material to the timeline of the broadcast.

For a DVB broadcast, the Content Identifier (CI) is typically able to identify to the resolution of individual programmes. For some use cases, this may be sufficient to enable a CSA to display content related to the programme. In this sense it is loosely synchronized.

For other use cases, the CSA may need to know more accurately the time position within the programme. The DVB broadcast can include Broadcast Timelines. The CSA can communicate with the TV Device via the CSS-TS interface to coordinate the synchronized presentation of Timed Content in the reference frame of a Broadcast Timeline. This process is termed Timeline Synchronization, and the timeline used in the communication is termed the Synchronization Timeline. The process involves regular communication to maintain synchronization and cope with changes in presentation timing by the TV Device (such as discontinuities due to the User using a pause function on the TV Device). Clause 4.2.2 describes a generic architecture for this Timeline Synchronization process and clause 4.2.3 maps this generic architecture to the companion synchronization architecture.

Communication across the home network may have significant latency. To take this into consideration, a separate timing reference or "Wall Clock" is established as a common reference on both the TV Device and CSA. A Wall Clock synchronization process takes place via the CSS-WC interface between TV Device and CSA to establish the Wall Clock across both, with the TV Device acting as a master clock. In doing so, the CSA can compensate for network latency. The Timeline Synchronization protocol incorporates Wall Clock Time Values in the messages exchanged, enabling both the TV Device and CSA to compensate for the time taken for the message to travel between them across the home network.

NOTE 1: The Wall Clock synchronization communication process is based on the principles used in NTP [i.2], however the Wall Clock does not represent real date and time.

The mechanisms described above primarily enable coordination of presentation on the TV Device and the CSAs when the sequence is known in advance. In addition Trigger Events allow coordination when the timing is not known in advance. For example, Trigger Events allow a broadcaster to notify a CSA of a score during a live sports competition. It is assumed that the CSA has the capability to respond to the event. The CSA knows (or discovers) the location of the Trigger Event signalling in the broadcast currently being presented by the TV Device. The CSA communicates with the TV Device to subscribe to the Trigger Events and provide the location of the event signalling within the DVB broadcast. If the TV Device detects the event signalling it notifies the CSA. Subscription and notification communication takes place via the CSS-TE interface.

Material Information embodies an abstraction from the specifics of the broadcast or IP delivery platform being used by the TV Device. The values of Content Identifiers, selection mechanisms for Synchronization Timelines and locations of Trigger Events map to the specifics of the delivery platform. However, a CSA can process Material Information at an abstract level as described at in clause 5.5. The CSA uses values from it in the procedures defined in clause 4.3 without needing to understand how the TV Device uses these values. A CSA needs only to recognize the editorial meaning of Materials to determine appropriate behaviour. A CSA can do this by recognizing Material Identifiers (see clause 5.5.3) and names given to Trigger Events (see clause 5.5.7) and by examining private data (see clause 5.9) in Material Information.

NOTE 2: If common Material Identifiers are used across several platforms then it is possible in principle for a single CSA with no delivery platform specific understanding to present synchronized Timed Content across all the platforms.

The present document for companion synchronization therefore defines:

- The means by which a CSA can synchronize with or estimate the Wall Clock of the TV Device.
- The packaging format for the CI carried in the broadcast or IP delivered media stream to the TV Device.
- The packaging format for the Material Resolution Service location(s) carried in the broadcast or IP delivered media stream to the TV Device.
- The means by which a TV Device derives a Broadcast Timeline from the received broadcast or IP delivered media stream.
- The interaction mechanism by which the Content Identifier is made available from and by the TV Device.
- The interaction mechanism by which changes to CI are announced.
- The means by which a CSA can coordinate with a TV Device to synchronize presentation of content, even if there are changes in presentation timing by the TV Device.
- The packaging format and data model for Material Information.
- The interaction format with the Material Resolution Service.

- The mechanism by which the MRS can advise of updates to Material Information.
- The packaging format of Trigger Event messages in the broadcast data.
- The packaging format to describe the Trigger Event message location in the Material Information.
- The interaction mechanism by which the CSA can subscribe to and be notified of Trigger Events.
- The carriage of private data in messages conveyed across interfaces.

Although out of scope of the present document, it is the responsibility of the Broadcaster that a Broadcast Service and associated Timed Content are available at the TV Device and at the CSA. It is also the responsibility of the Broadcaster to determine the timing relationship between the Broadcast Service and associated Timed Content, indicated by the dotted line in figure 4.2.1.1. The Broadcaster has to make this information correctly available to the parts of the system that are specified in the present document. Annex B provides informative guidelines for Broadcasters for this purpose.

4.2.2 Media synchronization architecture

Figure 4.2.2.1 illustrates a generic architecture for synchronization of media streams, based on [i.1]. Clause 4.2.3 describes how this generic architecture maps to the Timeline Synchronization aspect of the companion synchronization architecture already described.

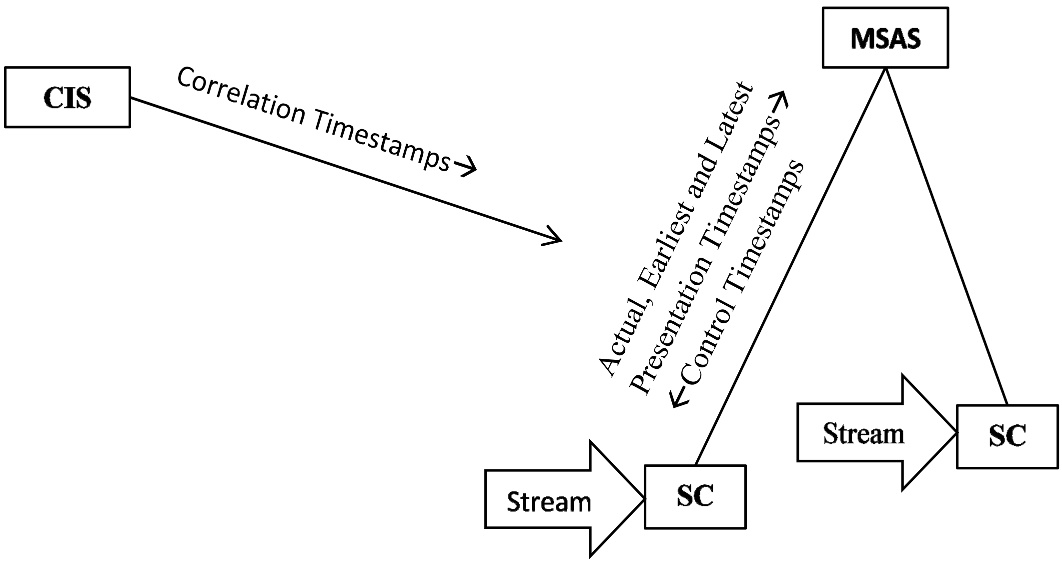


Figure 4.2.2.1: Generic architecture for media synchronization

The following elementary functions are distinguished in this architecture:

- MSAS: Media Synchronization Application Server.
- SC: Synchronization Client.
- CIS: Correlation Information Server (also known as SC' in [i.1]).

The following information flows between elementary functions of this architecture:

- Actual Presentation Timestamps: These timestamps describe the actual moment that a media sample is presented to the user by the elementary function that created them.
- Earliest and Latest Presentation Timestamps: These timestamps describe the earliest and latest moments that it is possible for a media sample to be presented to the user by the elementary function that created them.
- Control Timestamps: These timestamps describe the moment at which a media sample is to be presented if timing of presentation is to be synchronized.

- Correlation Timestamps: These timestamps describe a relationship between two different time reference frames - such as the time reference frame used in Earliest and Latest Presentation Timestamps and Control Timestamps, and the time reference frame that is the timeline for the Timed Content being presented by an SC.

The tasks of the SC are:

- Measuring for a received media stream the actual moment that a media sample is played out to the user, and computing the Actual Presentation Timestamp based on this.
- Measuring for a received media stream the earliest and latest moments that a media sample can be played out to the user, and computing Earliest Presentation Timestamps and Latest Presentation Timestamps based on this.
- Sending this information to the MSAS in the form of Actual Presentation Timestamps, Earliest Presentation Timestamps and Latest Presentation Timestamps.
- Receiving from the MSAS Control Timestamps, which indicate the moment that a media sample should be played out to the user.
- Delaying (buffering) if necessary a media stream according to the received Control Timestamps.
- Optionally receiving Correlation Timestamps from the CIS, and using these to translate the timeline of Earliest Presentation Timestamps, Latest Presentation Timestamps and Control Timestamps.

The tasks of the MSAS are:

- Collecting Actual Presentation Timestamps, Earliest Presentation Timestamps and Latest Presentation Timestamps from SCs.
- Calculating delay differences between the media play out of the SCs, and creating Control Timestamps based on this.
- Distributing Control Timestamps to SCs to suggest the timing of presentation that SCs should align to in order to achieve synchronized timing of presentation across all SCs.
- Optionally receiving Correlation Timestamps from the CIS, and using these to translate the timeline of Actual Presentation Timestamps, Earliest Presentation Timestamps, Latest Presentation Timestamps and Control Timestamps.

The tasks of the CIS are:

- Measuring the timing of different media streams in relation to the Synchronization Timeline, and creating Correlation Timestamps based on this.
- Sending Correlation Timestamps to the MSAS and/or SCs.

To create Control Timestamps, the MSAS may carry out sanity checks on the received Actual, Earliest and Latest Presentation Timestamps to check for excessively leading or lagging SCs and determine the most laggard SC to which the playout of other SCs should be synchronized.

Different timed content items have Timelines that may differ in type (e.g. PTS or ISOBMFF CT), tick rate and origin. The CIS determines the correlation between those Timelines. Depending on which Synchronization Timeline is used in the exchange of Actual Presentation Timestamps, Earliest Presentation Timestamps, Latest Presentation Timestamps and Control Timestamps between SC and MSAS, it may be the SC, the MSAS or both doing the translation between timelines.

NOTE: The concept of Synchronization Timeline can be illustrated with a language metaphor, where the Synchronization Timeline is the language that the SC speaks with the MSAS, and the Correlation Timestamp is the dictionary that is used to translate Timestamps on the Timeline of the Timed Content into Timestamps on the Synchronization Timeline. The dictionary for translation is only needed if the two Timelines are different.

This generic architecture for media synchronization can be applied to a wide variety of use cases, including synchronizing multiple content streams on a single device, synchronizing a single stream on multiple devices (e.g. for social TV) and synchronizing multiple streams across multiple devices. Each application requires a specific mapping of the described elementary functions on functional elements.

4.2.3 Mapping of generic media synchronization architecture to Timeline Synchronization

Figure 4.2.3.1 illustrates the mapping of the generic synchronization architecture to the Timeline Synchronization aspect of the architecture for companion synchronization.

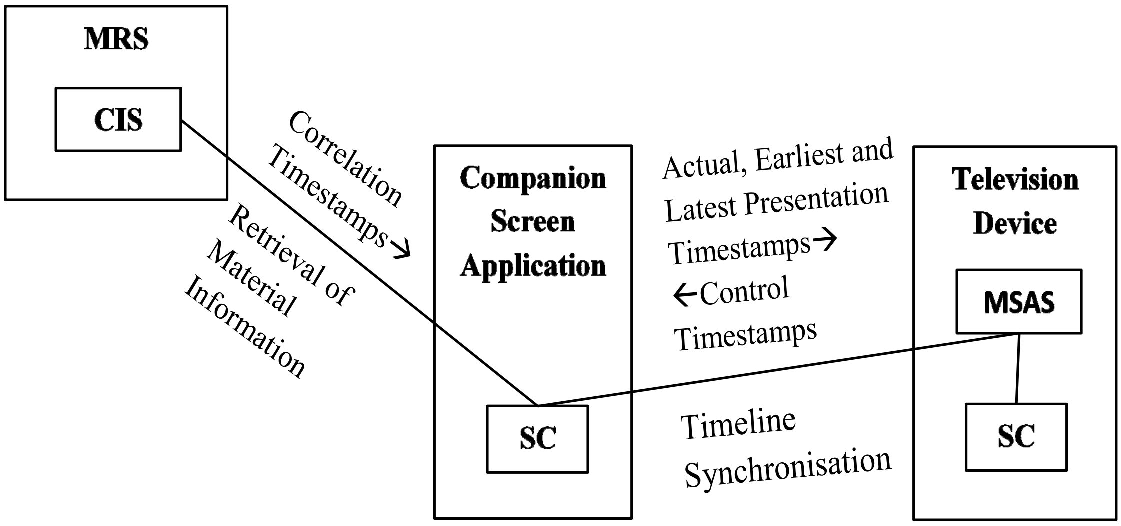


Figure 4.2.3.1: Mapping of Timeline Synchronization architecture

The mapping is as follows.

The TV Device incorporates the following elementary functions:

- MSAS function for coordinating Timeline Synchronization.
- SC for the Timed Content being presented by the TV Device.

The Companion Screen Application incorporates the following elementary functions:

- SC for the Companion Device intending to present Timed Content in synchrony with the TV Device.

The Material Resolution Service incorporates the following elementary functions:

- CIS that provides Correlation Timestamps as Material Information.

Correlation Timestamps are provided only to the SC elementary function of the CSA. The Timestamps that are exchanged between the SC function of the CSA and the MSAS function of the TV Device use a Timeline that derived from the Timed Content being presented by the TV Device and therefore do not require translation at the TV Device. The SC function of the CSA does need to translate between that Timeline and the Timeline of its own Timed Content and so uses the Correlation Timestamps to do this.

Figure 4.2.3.2 shows this integrates with other components of the companion synchronization architecture from figure 4.2.1.1.

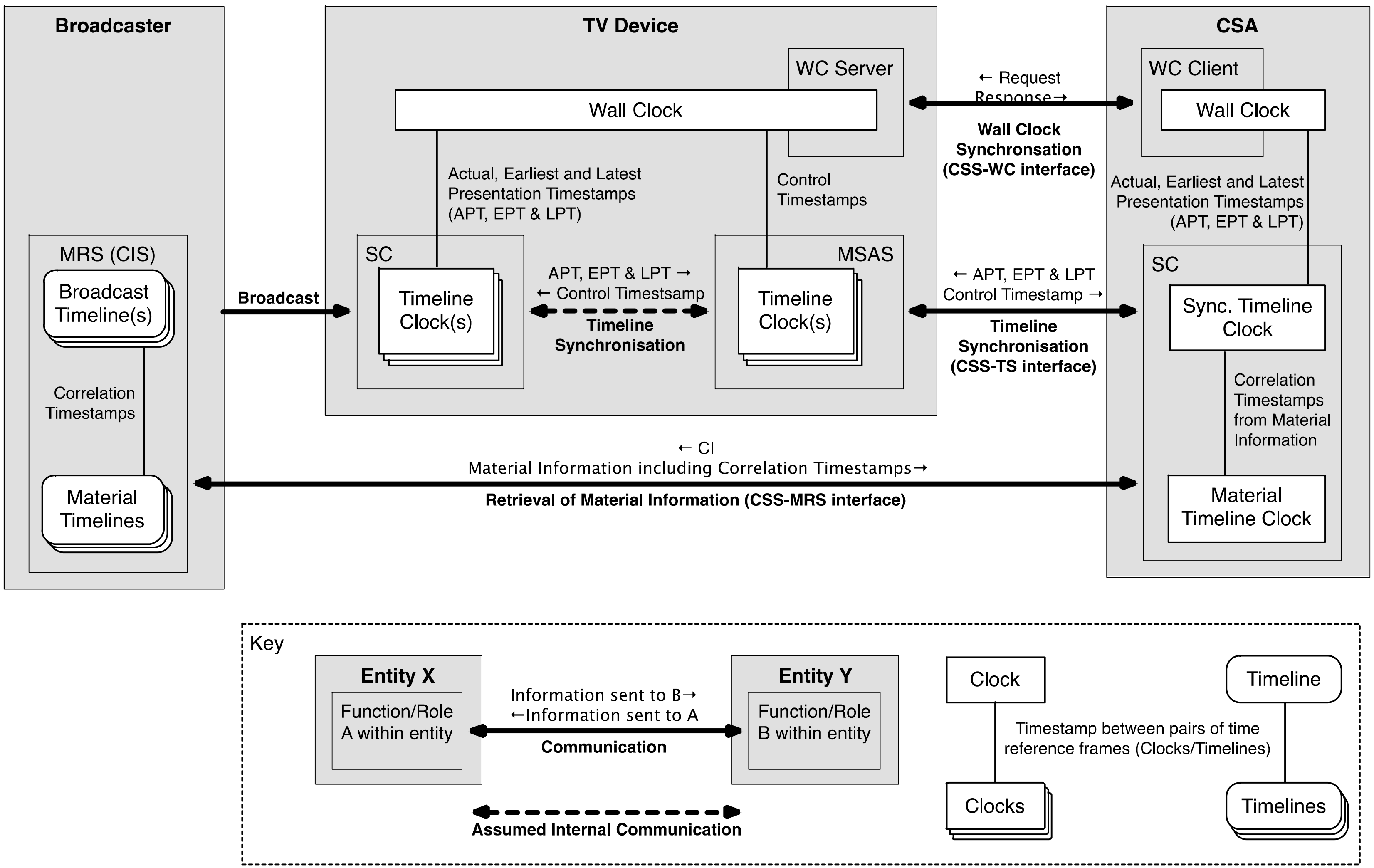


Figure 4.2.3.2: Timeline Synchronization architecture in the context of the wider companion synchronization architecture

A WC Client function in the CSA synchronizes a Wall Clock in the CSA to a Wall Clock in the TV Device by communicating the WC Server function of the TV Device. This is the communication that takes place via the CSS-WC interface.

Broadcast signalling will convey one or more Broadcast Timelines. Timeline Synchronization communication between the SC and MSAS functions internal to the TV Device will be in terms of these Broadcast Timelines. Both TV Device SC and TV Device MSAS will be able to model a clock representing the Broadcast Timeline in relation to the Wall Clock because the TV Device SC generates Actual, Earliest and Latest Presentation Timestamps in relation to the Wall Clock. Internally within the TV Device, the SC function is assumed to communicate Actual, Earliest and Latest Presentation Timestamps to the MSAS function and the MSAS function is assumed to communicate Control Timestamps to the SC function.

The SC function of the CSA will select one of the Broadcast Timelines for use as a Synchronization Timeline for Timeline Synchronization communication between it and the TV Device MSAS. Actual Presentation Timestamps, Earliest Presentation Timestamps, Latest Presentation Timestamps and Control Timestamps relate the Synchronization Timeline to the Wall Clock and therefore enable the SC in the CSA to model a clock representing the Broadcast Timeline. This is the communication that takes place via the CSS-TS interface.

The CSA will obtain Correlation Timestamps as part of Material Information retrieved from the MRS and provide them to the SC function of the CSA. These correlations will enable the SC to model a clock representing the timeline of a Material corresponding to the broadcast. The retrieval of Material Information takes place via the CSS-MRS interface.

In the generic architecture for synchronization, the MSAS can perform the correlations just described. However for the architecture defined in the present document only the CSA needs to receive Correlation Timestamps from the MRS. This is because it is the CSA and its SC function that performs the translation between the Synchronization Timeline and the Material Timeline.

4.2.4 Simplified scenario for the TV Device

The present document identifies a simplified scenario where CSAs receive their Timed Content at least as early as the TV Device receives its Timed Content (such as a broadcast). As a consequence, the TV Device is not required to buffer and delay the playout of its Timed Content. Also as a consequence, the TV Device is not required to process received Earliest Presentation Timestamps and Latest Presentation Timestamps however it may choose to do so.

NOTE: This simplified scenario can be achieved for live media content by delay management in the broadcaster network, see clause B.4. It can be achieved for recorded content by making Timed Content required by the CSA available to it ahead of time.

4.3 Procedures

4.3.1 Introduction

The following clauses describe the main procedures relating to the interfaces defined within the present document. The data model and data formats used in these procedures are defined in clause 5 and the underlying protocol details of each interface are then given in clauses 6 to 10.

Clause 4.3.2 describes the procedure for the CSA to obtain content identification and other information from the TV Device, including information that the CSA needs to perform all the other procedures.

Clause 4.3.3 describes the procedure by which the CSA may resolve Material Information from the content identity provided by the TV Device and obtain the information that the CSA needs to perform the procedure for Timeline Synchronization.

Clause 4.3.4 describes the procedure by which the CSA can synchronize its Wall Clock to the Wall Clock of the TV Device.

Clause 4.3.5 describes the procedure by which the CSA can synchronize its presentation of Timed Content to the presentation of Timed Content by the TV Device.

Clause 4.3.6 describes the procedure by which the CSA can request to be notified by the TV Device of Trigger Events that are signalled in the Timed Content delivered to the TV Device.

4.3.2 Content Identification and other Information (CSS-CII)

This clause is about the Companion Screen Application continuously obtaining Content Identification and other Information (CII) related to the content being presented by the TV Device.

Figure 4.3.2.1 shows the procedures for obtaining the Content Identification and other Information. It is assumed that the discovery, pairing and application-launching have successfully happened and that the Companion Screen Application knows the CSS-CII service endpoint at the TV Device.

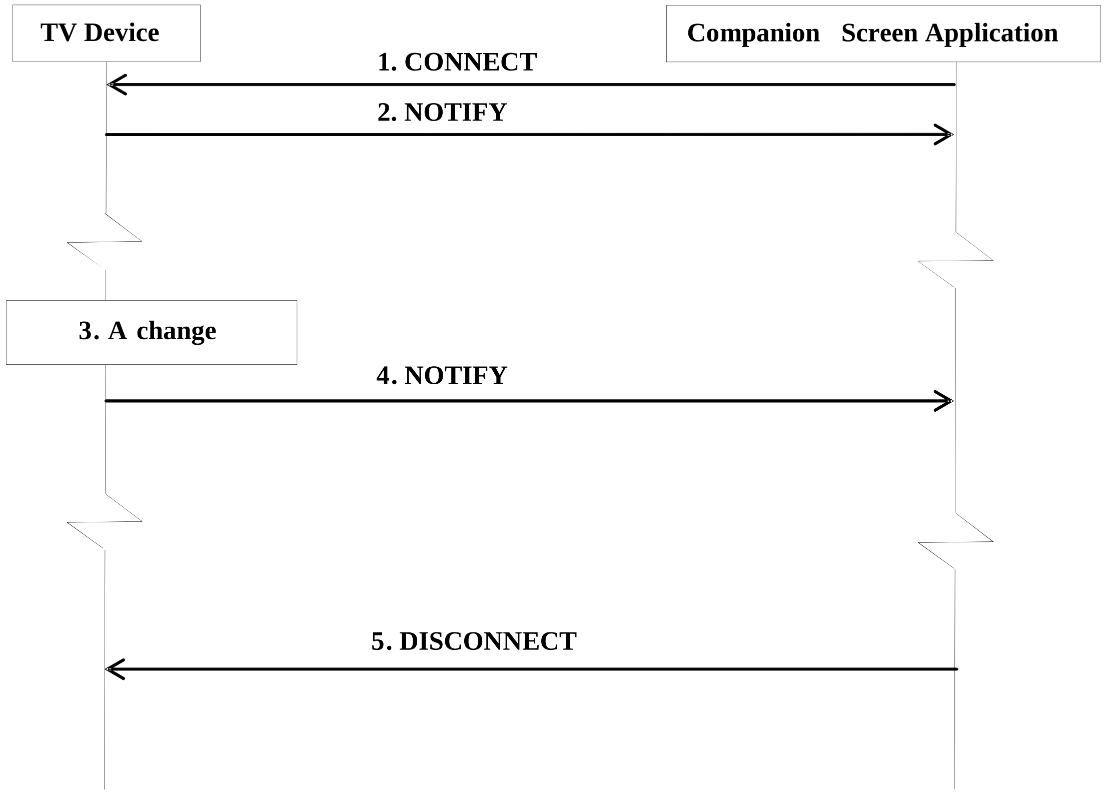


Figure 4.3.2.1: Procedures for obtaining Content Identification and other Information (CSS-CII)

For steps 2 and 4, the format and meaning of the message that conveys this are defined in clause 5.6. The protocol for the whole procedure is defined in clause 6.

1) The Companion Screen Application connects so that it can receive Content Identification and other Information from the TV Device.

2) The TV Device notifies the Companion Screen Application with the current Content Identification and other Information, which includes the Content identifier (CI), the locations (as URLs) of the service endpoints for the CSS-MRS, CSS-WC, CSS-TS and CSS-TE interfaces and some more parameters.

3) At some point in time, there is a change to the Content Identification and other Information at the TV Device. This could be because of the start of a new television program, because of a channel change by the users, because the television is no longer presenting or other reasons.

4) The TV Device notifies the Companion Screen Application of the updated Content Identification and other Information.

Steps 3)- 4) can be repeated when applicable.

5) The Companion Screen Application disconnects when it no longer wishes to receive Content Identification and other Information updates.

4.3.3 Material Resolution Service (CSS-MRS)

Figure 4.3.3.1 shows the procedures for Material resolution. It is assumed that the Companion Screen Application has obtained the Content Identifier and the location of the CSS-MRS interface service endpoint via the procedures of clause 4.3.2 or by some other means outside the scope of the present document.

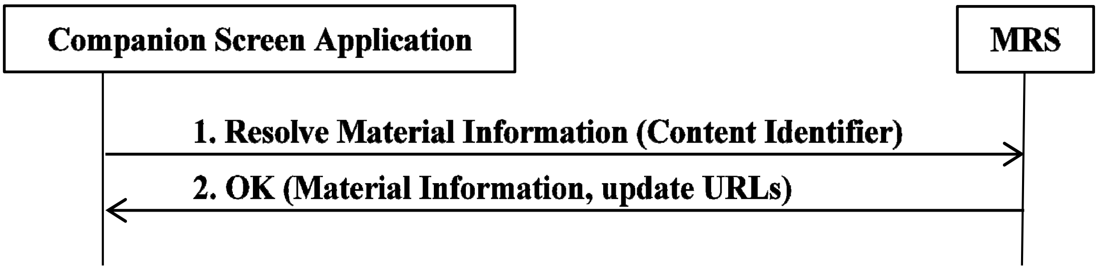


Figure 4.3.3.1: Procedures for material resolution (CSS-MRS)

The format of the response message sent by the MRS is defined in clause 7.4. The protocol for the whole procedure is defined in clauses 7.2 and 7.3:

1) The Companion Screen Application requests the MRS to resolve a Content Identifier.

2) The MRS responds with Material Information and update URLs.

Material Information describes Materials (see clause 5.5.2) that represent the editorial structure of the Timed Content being presented by the TV Device or Timed Content to be presented by the CSA.

Material Information may also include Synchronization Timeline information (see clause 5.5.4) that describes Timelines associated with the Timed Content being presented by the TV Device and how they map to the Timelines for Materials (using one or more Correlation Timestamps). Synchronization Timeline Information also includes Timeline Selectors (see clause 5.3.3) that a CSA uses in the Timeline Synchronization procedure described in clause 4.3.5 to specify which Timeline is to be used for that procedure.

Both parts of Material Information (Materials and Synchronization Timeline Information) can change over time and so can require updates to be conveyed from the MRS to the Companion Screen Application. For example: a broadcaster can change the schedule of programmes for a broadcast service or new Correlation Timestamps may be generated as a live broadcast progresses. The response from the MRS may include the URLs for update services for Materials and the URLs for update services for Synchronization Timeline Information. Figure 4.3.3.2 shows the procedures for obtaining updates.

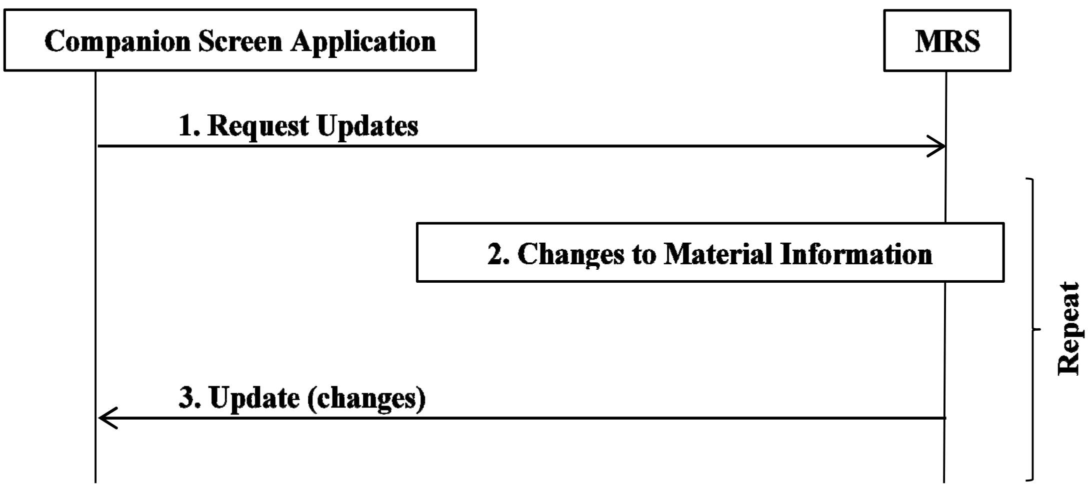


Figure 4.3.3.2: Procedures for material resolution updates (CSS-MRS)

The possible formats for update messages and the protocols over which they are carried are described in clause 7.6. Several different protocols are defined that an MRS can support. Different message formats are defined for updates to Materials and updates Synchronization Timeline Information.

1) The Companion Screen Application requests to be given updates to Materials or updates to Synchronization Timeline Information.

2) At some point, the MRS determines that the information has changed.

3) The MRS sends an update message to the Companion Screen Application describing the changes.

Steps 2) and 3) can be repeated when applicable.

4.3.4 Wall Clock (CSS-WC)

Wall Clock synchronization synchronizes the Wall Clock at the Companion Device to the Wall Clock at the TV Device. Figure 4.3.4.1 illustrates the procedures for Wall Clock synchronization via the CSS-WC interface. It is assumed that the CSA has obtained the location of this interface's service endpoint via the procedures of clause 4.3.2 or by some other means outside the scope of the present document.

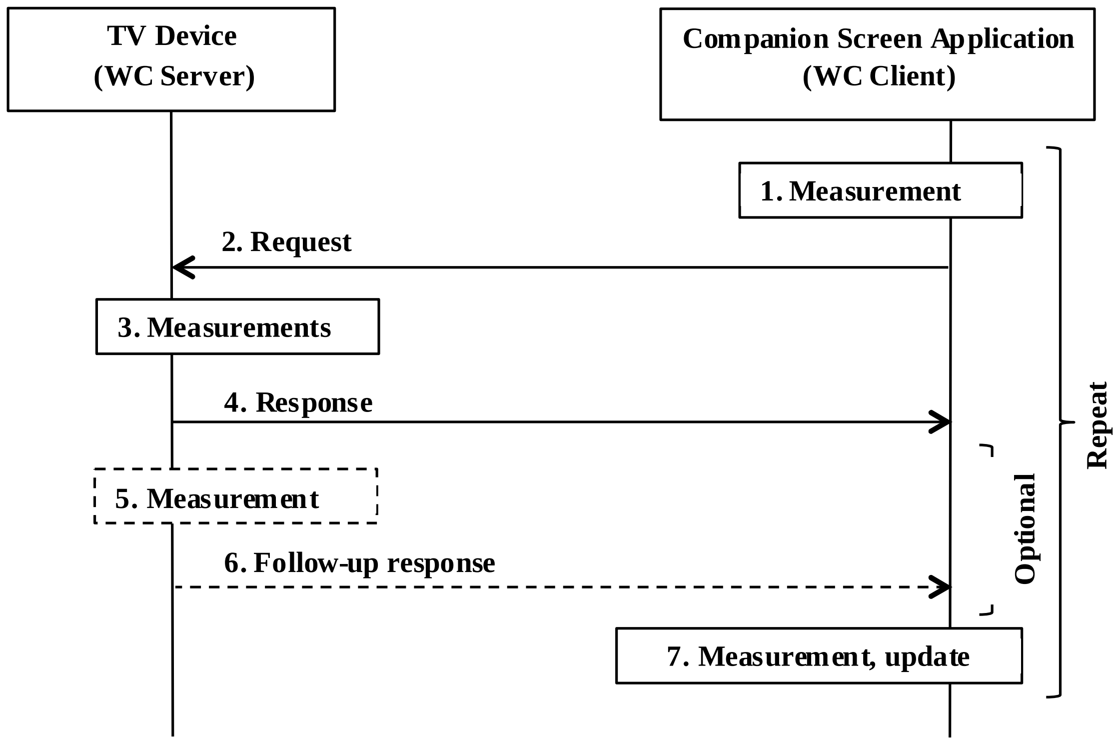


Figure 4.3.4.1: Procedures for Wall Clock synchronization (CSS-WC)

The formats of messages exchanged and the protocol used are defined in clause 8.

1) The WC Client function of the Companion Screen Application measures the current time value of its Wall Clock.

2) The Companion Screen Application immediately sends a Wall Clock protocol (CSS-WC) request to the TV Device.

3) The WC Server function of the TV Device measures the current time value of its Wall Clock immediately when it receives the request and again immediately prior to step 4).

4) The TV Device sends a Wall Clock protocol response to the Companion Screen Application. This includes both measurements made in step 3).

5) The WC Server function of the TV Device records a measurement of the time value of its Wall Clock at the time the response message was sent in step 4).

6) The TV Device sends a follow-up response message to the Companion Screen Application that includes the measurement made in step 5).

Steps 5) and 6) are optional depending on the capabilities of the TV Device.

7) The WC Client function of the Companion Screen Application measures the current time value of its Wall Clock immediately when it receives the response or the follow-up response. It then calculates an estimate of the difference between its Wall Clock and the Wall Clock in the TV Device. It then updates its Wall Clock to more accurately match that of the TV Device.

All steps of this procedure are repeated at regular intervals to ensure that the Wall Clock of the Companion Screen Application remains closely synchronized with the Wall Clock of the TV Device.

The CSA can continuously calculate measures such as dispersion (as detailed in clause C.8.3.2) to estimate how closely synchronized it is to the Wall Clock of the TV Device.

4.3.5 Timeline Synchronization (CSS-TS)

Timeline Synchronization synchronizes the presentation of Timed Content between multiple SCs, coordinated by the MSAS. Figure 4.3.5.1 shows the procedures for Timeline Synchronization via the CSS-TS interface. It is assumed that the CSA has obtained the location of this interface's service endpoint via the procedures of clause 4.3.2 or by some other means outside the scope of the present document.

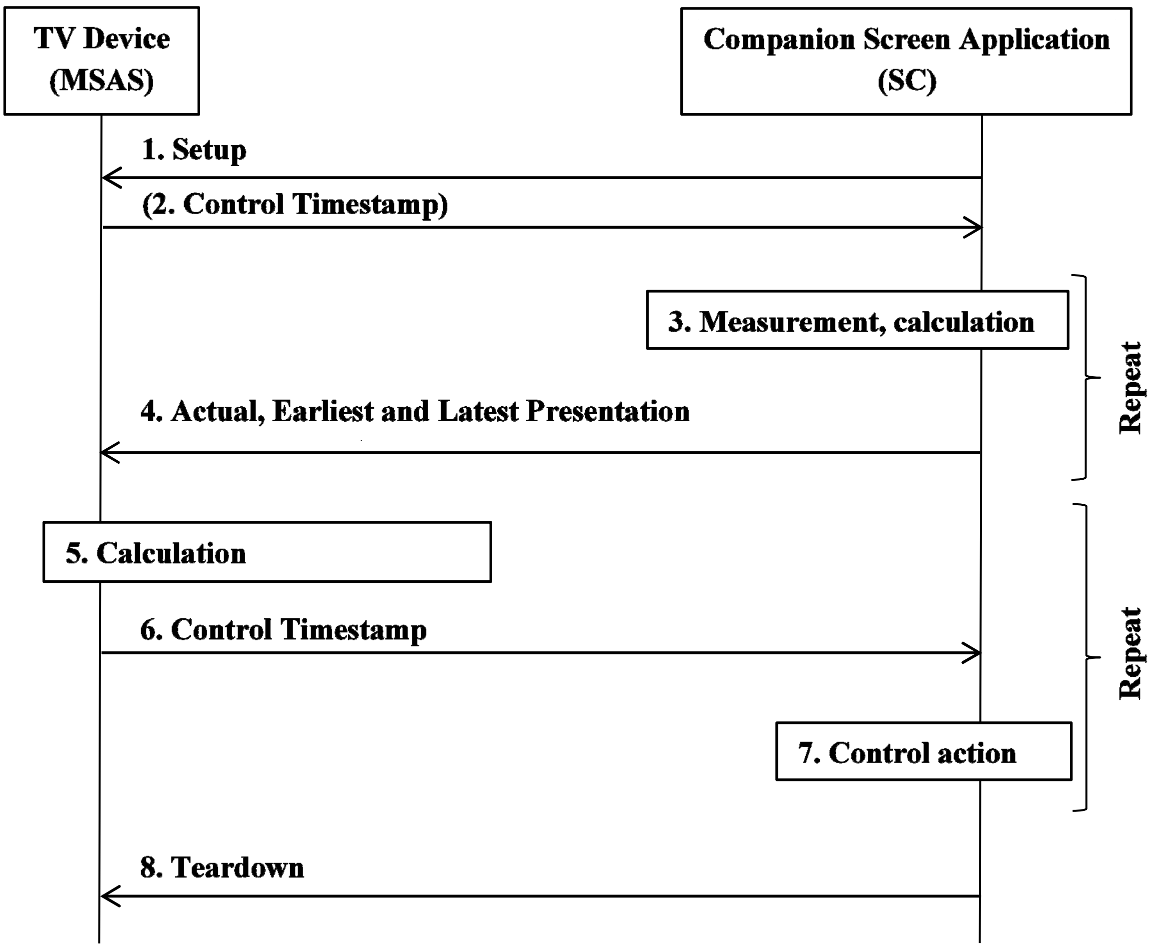


Figure 4.3.5.1: Procedures for Timeline Synchronization

The formats for messages used in each interaction are defined in clause 5.7. The protocol by which the messages are conveyed in defined in clause 9.

1) The Companion Screen Application sets up a Timeline Synchronization session with the TV Device (CSS-TS). At this point it selects which Timeline is used. The format for this message is defined in clause 5.7.3.

2) The TV Device sends a Control Timestamp to the Companion Screen Application, see also steps 5) - 7). The format for this message is defined in clause 5.7.5.

3) The SC function in the Companion Screen Application performs a measurement of its Timed Content presentation and calculates an Actual Presentation Timestamp, an Earliest Presentation Timestamp and a Latest Presentation Timestamp of its Timed Content.

4) The Companion Screen Application sends a message to the TV Device with an Actual, Earliest and Latest Presentation Timestamp. The format for this message is defined in clause 5.7.4.

Steps 3) and 4) are repeated asynchronously, i.e. independent of message received from the TV Device.

5) The MSAS function in the TV Device performs a calculation on Actual Presentation Timestamps, Earliest Presentation Timestamps and a Latest Presentation Timestamps received from one or more SCs and determines a Control Timestamp.

6) The TV Device sends a message to the Companion Screen Application with a new Control Timestamp.

7) The SC function of the Companion Screen Application performs a control action to synchronize the presentation of the Timed Content according to the Control Timestamp.

Steps 5) - 7) are repeated asynchronously, i.e. independent of message received from a particular Companion Screen Application.

8) The Companion Screen Application tears down the Timeline Synchronization Session.

Additional information on step 1).

The CSA can acquire knowledge of the Timelines that are available to be selected via the CSS-MRS interface as described in the procedure in clause 4.3.3. The CSA can also obtain this information by some other means outside the scope of the present document.

Additional information on step 3).

The SC in the Companion Screen Application determines Actual Presentation Timestamps, Earliest Presentation Timestamps and Latest Presentation Timestamps on the Synchronization Timeline with respect to the reference point for timestamping specified in clause 5.7.2.

Correlation Timestamps enable a Companion Screen Application to translate its understanding of the progress of a Synchronization Timeline to an understanding of the progress of the Material Timelines. These Correlation Timestamps are included in the Material Information (see clause 5.5.6) and updates to Material Information, obtained according to the procedures of clause 4.3.3.

Illustrative examples of the above calculation are provided in clause C.4.

NOTE: These procedures are similar to media synchronization procedures from [i.1], clause 8.15.1.

The Actual Presentation Timestamp describes to the current timing with which the SC is presenting Timed Content to the user. If an SC has not yet begun to present Timed Content then it does not need to calculate and include an Actual Presentation Timestamp in the message sent in step 4). An SC may also opt to not provide an Actual Presentation Timestamp if the Timed Content presented by the SC is not intended to progress unless synchronized to presentation by the TV Device.

EXAMPLE: A CSA is presenting Timed Content consisting of a play-along quiz experience. The progress of the quiz is tied to the progress of the quiz show presented on the TV Device and will not progress unless presentation is synchronized with a TV Device. Therefore the SC function of the CSA opts to not provide an Actual Presentation Timestamp.

For a live stream of Timed Content, the Earliest Presentation Timestamp corresponds to the earliest time that the Timed Content can be presented to the user and the Latest Presentation Timestamp corresponds to the latest time that the Timed Content can be presented, taking into account the capacity to buffer (delay) the Timed Content. That buffer may be in the SC, in the network or a combination of both.

In the case of adaptively streamed content using MPEG DASH, the SC computes the Earliest Presentation Timestamp by using information from the Media Presentation Description (MPD). The MPD contains information about the currently available segments of a stream.

If the SC is unable to vary the time at which it presents its Timed Content, then the Latest Presentation Timestamp shall be equal to the Earliest Presentation Timestamp.

If a Timed Content is available in full, either stored or buffered at the SC or in the network and/or if a Timed Content is available indefinitely, then the Earliest Presentation Timestamp and the Latest Presentation Timestamp can reflect this capacity for near infinity flexibility in presentation timing. How this is done is specified in clause 5.7.4.

Additional information for step 5).

The simplest case of the MSAS calculation is analysing the received Earliest Presentation Timestamps, determining the most laggard SC and sending out a Control Timestamp that is equivalent to the Earliest Presentation Timestamp of the most laggard SC. The MSAS can examine the Latest Presentation Timestamps to ensure that all connected SC are able to modify their playout according to this Control Timestamp. If SCs provide Actual Presentation Timestamps, then the MSAS can instead select a Control Timestamp that reflects as closely as possible the presentation timing represented by Actual Presentation Timestamps, while still ensuring that it falls within the achievable presentation timing of all SCs as indicated by the Earliest and Latest Presentation Timestamps they report. This minimizes disruption to existing Timed Content presentations. This is particularly important if the Earliest and Latest Presentation Timestamps indicate an infinite amount of flexibility in presentation timing.

More illustration of the MSAS calculations is provided in clauses C.5.3 and C.7.3.

As part of a Control timestamp, the MSAS can indicate a change to the rate of presentation (e.g. a pause). The MSAS may do this if it determines that an SC has paused its presentation. Typically this could happen if the SC elementary function within the TV Device is paused. The MSAS can have knowledge of this because it too is part of the TV Device. However, the MSAS may also do this in response to deducing that other SCs have paused their presentation by checking if two consecutive Actual Presentation Timestamps correspond to a paused stream, i.e. a changed Time Value on the Wall Clock and an unchanged Time Value on the Synchronization Timeline.

Additional information for step 7).

If the required presentation timing adjustment at the SC is major, then the SC may decide to make a hard jump to adjust the presentation of the Timed Content. Major adjustments may be caused by seek, rewind and fast-forward actions by the user at the TV Device.

If the required presentation timing adjustment at the SC is minor, then the SC may gradually vary its playout rate. This may be achieved by using techniques such adaptive media playout (AMP) to minimize the impact of the adjustment on the user experience (see also clause C.3).

If the SC needs to skip back in time to access content it does not have locally, it may use techniques like IPTV retransmission (RET), IPTV Fast Channel Change (FCC) and dedicated content delivery network (CDN) technologies as discussed in clause C.3.

In case of adaptively streamed content using MPEG DASH, the SC would typically not have its own dedicated buffer space for Timeline Synchronization. Instead, the SC would use the network as its buffer, which it controls by timing its segment requests to the network such that segments arrive at the correct time for synchronized presentation.

4.3.6 Trigger Events interface (CSS-TE)

This clause is about the Companion Screen Application continuously obtaining Trigger Events related to the content being presented by the TV Device via the CSS-TE interface.

Figure 4.3.6.1 shows the procedures for obtaining the Trigger Events. It is assumed that the CSA has obtained the URL for this interface service endpoint via the procedures of clause 4.3.2 or by some other means outside the scope of the present document.

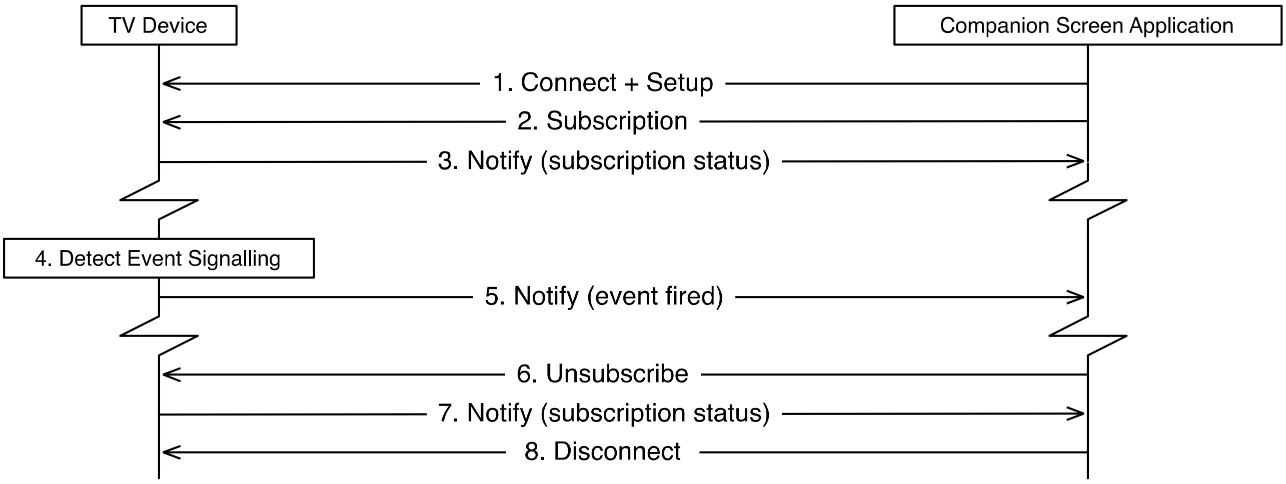


Figure 4.3.6.1: Procedures for obtaining Trigger Events (CSS-TE)

The formats for messages exchanged for this procedure are defined in clause 5.8.5. The protocol for the whole procedure is defined in clause 10.

1) The Companion Screen Application connects to the Trigger Event service endpoint provided by the TV Device and sets up the context of the connection by sending a message as defined in clause 5.8.5.2.

2) The Companion Screen Application subscribes to a Trigger Event notification from the TV Device. The Companion Screen Application may be informed about available Trigger Events in the Material Information from the MRS The format of the subscription message is defined in clause 5.8.5.3.

3) The TV Device sends a Trigger Event notification in response to each Trigger Event subscription request to indicate if the subscription has been successful or not. The format of the message is defined in clause 5.8.5.4.

Step 2) and 3) can be repeated when applicable allowing the Companion Screen Application to subscribe to multiple different Trigger Events.

4) The TV Device detects a successfully subscribed event in the broadcast data.

5) The TV Device notifies the Companion Screen Application with Trigger Event information. The format of the message is defined in clause 5.8.5.4.

Steps 4) and5) depend on the presence of signalling in the Timed Content. The signalling for each subscribed Trigger Event will happen zero or more times.

6) At some point in time the Companion Screen Application decides that a subscribed Trigger Event is no longer required and unsubscribes it. For example, this could be because the current Material Information no longer lists this Trigger Event, indicating that it is no longer applicable to the Timed Content. The format of the message is defined in clause 5.8.5.3.

7) The TV Device sends a Trigger Event notification in response to each Trigger Event subscription cancellation to acknowledge the cancellation of the event subscription. The format of the message is defined in clause 5.8.5.4.

Steps 2) - 7) can be repeated, for example when Content Identification and other Information changes at program boundaries and the set of appropriate Trigger Events changes.

8) The Companion Screen Application decides that the context of the current session is no longer applicable, unsubscribes all current subscriptions and then closes the connection.

Figure 4.3.6.2 shows a variant on figure 4.3.6.1. In this case TV Device detects that the context of a Trigger Event session is no longer applicable, for example, the media being presented changes due to a channel change and the subscribed Trigger Event does not exist in the new media. As a consequence the TV Device cancels any current subscriptions in that session.

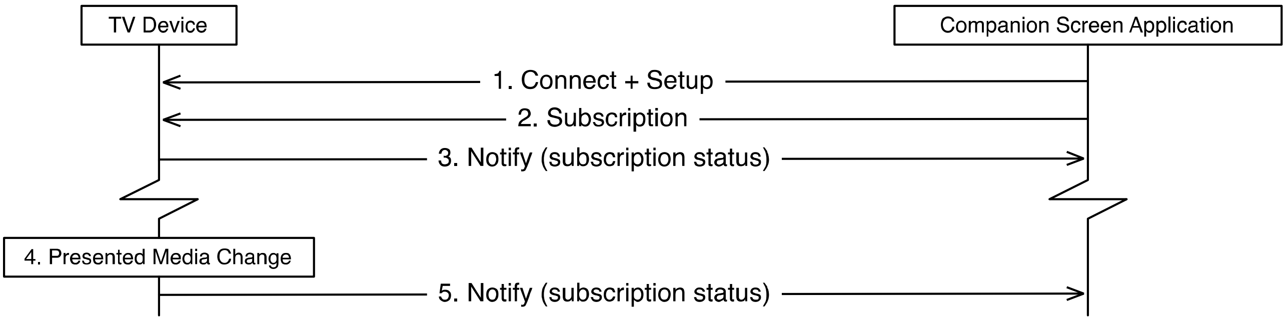


Figure 4.3.6.2: Procedures for obtaining Trigger Events (CSS-TE) continued

The formats of messages exchanged are the same as those detailed above.

5 Data model

5.1 Data model introduction

Clause 5 details the information conveyed in messages across the interfaces between TV Device and CSA and between MRS and CSA. It describes the formats, models and semantics and their representation in JSON as messages that are conveyed.

The procedures that give rise to the exchange of these messages are detailed in clause 4.3 and the protocols that carry them are described in clauses 6, 7, 9 and 10.

Messages, JSON objects, or properties in messages or JSON objects, that are not defined in the present document, or otherwise not understood by the TV Device or CSA, shall be ignored.

Clause 5.2 defines a Content Identifier and how it is derived depending on the Timed Content that is being presented by the TV Device to the user.

Clause 5.3 defines the information used to describe the properties of Timelines and the selector used to select Timelines. The clause also defines how Timelines are derived from the Timed Content being presented by the TV Device to the user.

Clause 5.4 defines how a correlation between two time reference frames (such as timelines) is expressed.

Clause 5.5 defines Material Information that is provided by the MRS. This includes descriptions of how this information is represented in JSON and semantics for how a CSA interprets the Material Information. The procedures in which Material Information is exchanged are detailed in clause 4.3.3 and the protocols in which it is conveyed are detailed in clause 7.

Clause 5.6 defines a Content Identification and other Information message that is sent by the TV Device to the CSA and how this is represented in JSON. The procedure in which these messages are exchanged is detailed in clause 4.3.2 and the protocol in which they are conveyed is described in clause 6.

Clause 5.7 defines the types of Timestamp messages that are exchanged between the TV Device and CSA and the representation in JSON. The procedure in which these messages are exchanged is detailed in clause 4.3.5 and the protocol in which they are conveyed is described in clause 9.

Clause 5.8 defines Trigger Events and the information that is used in the present document to locate them. The clause also defines messages and their JSON representation that are exchanged between the TV Device and CSA to manage subscription to and notification of Trigger Events. The procedure in which these messages are exchanged is detailed in clause 4.3.6 and the protocol in which they are conveyed is described in clause 11.

Clause 5.9 defines the format for private data and how it is to be represented within JSON. This is permitted to occur in most messages and many of the JSON objects defined in this clause.

The diagrams that are used in the subsequent sections employ UML-based structure diagrams that are described in annex D.

5.2 Content Identification

5.2.1 General

The Content Identifier (CI) is a composite identifier that serves to unambiguously identify content to the resolution of an individual programme or presentation. The CI is derived from information that identifies the delivery stream and information contained within it.

The format of a CI is a URI [1] with restrictions applied. It consists of a URI scheme identifier, hierarchical part and optional query part. The format of the CI is hierarchically structured such that the information encoded in it is conveyed in the following order:

1) Stream identifying information that identifies the broadcast transmission or IPTV stream. For example, this may be the DVB URI [3] with the original network ID and transport stream ID of a DVB broadcast, or the URL of an IPTV stream.

2) Service identifying information (if multiple services are carried in a single stream), such as the service ID of a TV channel or Radio station in a DVB Broadcast.

3) Content identifying information (if multiple content items are carried in a stream), such as DVB event identifiers for the programme in the broadcast schedule or a time window or section of an IPTV stream.

4) Supplementary ancillary data (if present).

The process of generating the CI from the underlying sources of information is deterministic. The present document defines how to generate the CI and permits one and only one formulation as a sequence of characters given the same underlying information. To generate a CI URI in a deterministic way, the following rules are applied:

- There is only a single order for the elements in the URI.

- When an optional piece of information is not present, then the separator characters that would enclose it are also not present.

- Numeric values have a single representation (e.g. whether or not a numeric value will be padded by including leading zeros to it is specified).

- Use of upper and lower case is normalized as set out in clause 6.2 of [1] by using upper case in percent encoding and lower case in the scheme name and host name.

NOTE: This means that for a given broadcast or a given stream, a given viewing selection and at a given time, all TV Devices generate the same "final" CI. Two CIs are the same if they pass a simple case-sensitive character-by-character string equality test.

Depending on the type of service the construction of the CI may depend on multiple sources of information some of which may not be available quickly. To enable the TV Device to emit a CI soon after selection of a service zero or more "partial" CIs may be emitted prior to the emission of the "final" complete CI. The following rules apply to "partial" CIs:

- "partial" CIs shall be syntactically valid subsets of a "final" CI;

- "partial" CIs shall be a valid CI stem as defined in clause 5.2.2.

The "partial" or "final" status of the CI is signalled alongside the CI as defined in clause 5.6. The permitted options for emission of "partial" and "final" CIs are defined in clause 5.2.3.6.

For a DVB service carried in a broadcast or IPTV streamed MPEG2 transport stream, the format of the CI is based on a subset of the DVB locator format defined in [3]. This is defined in clause 5.2.3.

For a DVB service carried by DASH [17] the format of the CI is defined in clause 5.2.4.

For any other type of service, the requirements for the format of the CI is defined in clause 5.2.5.

5.2.2 Content Identifier stem

A Content Identifier stem (CI stem) describes a set of CIs. A CI stem is a string representing the first *n* characters of any CI that is in the set.

Given a CI stem string that is n characters long, then a CI is considered to match the CI stem if the first n characters of the CI match the CI stem string in a case sensitive comparison.

NOTE: A CI that is shorter than the CI stem never matches that CI stem. A CI stem that is the empty string matches all CIs.

EXAMPLE: The CI stem "dvb://233a.1004.1044;" will match any CI provided by a TV Device while it is currently presenting that specified DVB service. The CI stem "dvb://233a.1004.1044;35f7~20131004T0930Z--PT01H00M" will match any CI provided by a TV Device while it is currently presenting the specified DVB service and the specified DVB event, scheduled for 9.30am on the 4th October 2013 is being signalled by the broadcaster to be the programme currently being shown.
See clause 5.2.3 for details of the construction of the CIs used in these examples.

5.2.3 DVB Broadcast and IPTV services

5.2.3.1 General

For DVB services with associated DVB Service Information (SI) as defined in ETSI EN 300 468 [13], the CI shall unambiguously and globally identify up to the level of detail denoted by a DVB event. The CI may also carry a TV-Anytime CRID relating to that event, if one is present in the SI, and additional ancillary data relating to that event.

For the programme currently being presented by the receiver to the user, the CI denotes the present event as signalled in Event Information Table (EIT) that carries present/following event information for the actual TS (EIT present/following actual) as defined in clause 5.2.4 of ETSI EN 300 468 [13].

Where data from DVB SI is described below, it has come either from EIT present/following for the actual TS, or from other tables or descriptor loops within that table that apply to that event, or the service within which the event features, or the bouquet in which the service is included, or the network in which the service or bouquet features.

The method by which the TV Device determines the signalling available on a particular platform, and therefore which parts of the CI can be encoded, is outside the scope of the present document. Therefore the presence of the signalling in CI is optional depending on the requirements underlying platform.

NOTE: For a given platform, a TV Device is expected to know whether and how TV-Anytime CRIDs are signalled for events and which SI tables (e.g. BAT, NIT, SDT or EIT) are present for a given platform.

EXAMPLE 1: If a platform does not require TV Devices to process the BAT then the TV Device is not expected to include the anc_bat key value pair in the CI.

EXAMPLE 2: If a platform does not require TV Devices to process TV-Anytime CRIDs associated with DVB events, then the TV Device is not expected to include the ep_crid key value pair in the CI.

The CI URI follows a standard URI syntax for URLs. It shall consist of a scheme identifier followed by a colon ":" character, a hierarchical part and an optional query part prefixed by a question mark "?" character if the query part is present:

scheme ":" hierarchical-part ["?" query-part]

The CI URI scheme identifier is "dvb" and therefore the CI shall begin with the string "dvb" as lower case letters.

The hierarchical part of the CI URI includes a service path (defined in clauses 5.2.3.2 and 5.2.3.3) and an event constraint (defined in clause 5.2.3.4).

Depending on the presence of optional signalling the query part of the CI URI (defined in clause 5.2.3.5) may convey either or both of the following:

- a TV-Anytime episode CRID

- CI ancillary data

Table 5.2.3.1.1 summarizes the syntax of a CI for DVB broadcast and IPTV services as a formal specification expressed in Augmented BNF as defined in IETF RFC 5234 [4] and using the Core Rules defined in Appendix B.1 of IETF RFC 5234 [4].

Table 5.2.3.1.1: Syntax of the CI URI for DVB services

ci-dvb	=	dvb-url [ci-dvb-query]
ci-dvb-query	=	"?" ci-dvb-query-params
ci-dvb-query-params	=	ep-crid ["&" anc-eit] ["&" anc-sdt] ["&" anc-bat] ["&" anc-nit]
ci-dvb-query-params	=/	anc-eit ["&" anc-sdt] ["&" anc-bat] ["&" anc-nit]
ci-dvb-query-params	=/	anc-sdt ["&" anc-bat] ["&" anc-nit]
ci-dvb-query-params	=/	anc-bat ["&" anc-nit]
ci-dvb-query-params	=/	anc-nit ; (see note 1)
dvb-url	=	dvb-scheme ":" dvb-net-path [dvb-event-constraint]
dvb-scheme	=	%x64.76.62 ; "dvb" (see note 4)
dvb-net-path	=	"//" dvb-service-without-event ; (see note 3)
dvb-service-without-event	=	original-network-id "." transport-stream-id "." service-id
dvb-service-without-event	=/	%x27 textual-service-identifier %x27
textual-service-identifier	=	*unreserved ; (see note 2)
dvb-event-constraint	=	";" event-id [";" TVA-id] time-constraint
original-network-id	=	4*4hex-lc
transport-stream-id	=	4*4hex-lc
service-id	=	4*4hex-lc
event-id	=	4*4hex-lc
TVA-id	=	4*4hex-lc
anc-nit	=	%x6e.69.74.5f.61.6e.63 "=" ci-ancillary-data ; "nit_anc" (see note 4)
anc-bat	=	%x62.61.74.5f.61.6e.63 "=" ci-ancillary-data ; "bat_anc" (see note 4)
anc-sdt	=	%x73.64.74.5f.61.6e.63 "=" ci-ancillary-data ; "sdt_anc" (see note 4)
anc-eit	=	%x65.69.74.5f.61.6e.63 "=" ci-ancillary-data ; "eit_anc" (see note 4)
ep-crid	=	%x65.70.5f.63.72.69.64 "=" TVA-episode-crid ; "ep_crid" (see note 4)
TVA-episode-crid	=	1*percent-encoded
ci-ancillary-data	=	hex-byte-string
time-constraint	=	"~" time-duration
time-duration	=	start-time "--" duration
start-time	=	date %x54 time %x5a ; date "T" time "Z" (see note 4)
duration	=	%x50.54 hours %x48 minutes %x4d ; "PT" hours "H" minutes "M" (see note 4)
date	=	year month day
time	=	hours minutes
year	=	digit digit digit digit
month	=	digit digit
day	=	digit digit
hours	=	digit digit
minutes	=	digit digit
hex-byte-string	=	*(hex-lc hex-lc)
percent-encoded	=	unreserved / pct-enc-char
hex-lc	=	digit / %x61-66 ; 0-9 or a-f (see note 4)
hex-uc	=	digit / %x41-46 ; 0-9 or A-F (see note 4)
unreserved	=	alpha / digit / "-" / "." / "_"
pct-enc-char	=	"%" hex-uc hex-uc
NOTE 1: These rules strictly control the situations in which "&" and ";" separator symbols are present or not present. NOTE 2: Textual-service-identifier is defined in ETSI TS 102 034 [5]. NOTE 3: The dvb-net-path is described further in clauses 5.2.3.2 and 5.2.3.3. NOTE 4: Terminal value encoding is used to make the case of characters explicit. Literal text strings are case insensitive in this ABNF syntax.		

5.2.3.2 Net path for DVB broadcast services

The net path of the CI for a DVB broadcast service shall comprise a pair of forward slashes "//" followed by the original network id, a full stop "." character, transport stream id, a full stop "." character and the service id:

"//" original-network-id "." transport-stream-id "." service-id

The original network id, transport stream id, and service id shall all be expressed in hexadecimal as 4 digits where alphabetic characters are in lower case. Where the value of the id results in a hexadecimal number less than 4 digits long, this shall be padded to 4 digits by prefixing one or more zero "0" characters.

5.2.3.3 Net path for DVB IPTV services

Where original network id, transport stream id and service id are known by the receiver the dvb-net-path part of the CI for an IPTV service shall be formatted as defined for broadcast services in clause 5.2.3.2.

If this is not possible, then net path of the CI for a DVB IPTV service shall be a textual service identifier as defined in table 5.2.3.1.1 but following the second definition of the dvb-service-without-event rule (i.e. %x27 textual-service-identifier %x27).

5.2.3.4 Event Constraint

The event constraint conveys the event-id, a TVA_id (if one is exists), and a time constraint describing the UTC start time and duration from the start_time and duration fields for this event (coming from EIT present/following actual):

";" event-id [";" tva-id] "~" time-constraint

If there is EIT present in the DVB SI for the Timed Content being presented then the event constraint shall be included in the CI, otherwise the event constraint shall not be included in the CI.

If there are one or more TVA_ids carried in TVA_id_descriptor [2] descriptors in the EIT present/following actual, then the event constraint shall include the tva-id part (i.e. [";" TVA-id]) in the CI and it shall correspond to the first TVA_id found in the first TVA_id_descriptor descriptor.

The time constraint begins with a tilde "~" character followed by the date and start time, as determined from the EIT signalling for the present event.

For both start time and duration, the seconds are not included in this representation. Minutes shall not be rounded to the nearest minute irrespective of the value of the number of seconds.

5.2.3.5 Query

The query part of the CI, if present, shall be prefixed by a question mark "?" character followed by one or more key-value pairs, separated by ampersand "&" characters:

"?" key-value-pair *("&" key "=" value)

The query part shall be present if there is one or more key-value pair to convey. If there are to be no key-value pairs, then the query part of the CI, including the prefixed question mark character, shall be omitted.

The following key-value pairs are defined and shall be included in the query part in the same order as they are listed below:

1) If the platform (whose requirements are outside the scope of the present document) defines that TV-Anytime CRIDs are signalled for some or all DVB events and if there is one or more TV-Anytime episode CRIDs signalled using content identifier descriptors in the manner described in clause 12.1 of [2] for this event in EIT present/following for the actual TS, then the key "ep_crid" shall be included. The value corresponding with the key shall be the episode CRID. If the signalling defines multiple episode CRIDs for the event, then the first episode CRID shall be selected. The CRID shall include the authority but omit the "crid://" prefix. Number digits, lower and upper case alphabetic characters, dash "-", full-stop "." and underscore "_" shall be included unchanged. All other characters in a CRID shall be substituted with a percent "%" character followed the ASCII [28] value of that character formatted as two hex digits where alphabetic characters are uppercase. If the platform does not define use of TV-Anytime CRIDs in this way, then the "ep_crid" key shall not be included.

2) If the TV Device detects the presence of a ci_ancillary_data descriptor (described in clause 6.4.1 of ETSI EN 300 468 [13]) in the descriptor loop of the event within EIT present/following for the actual TS, then the key "anc_eit" shall be included. The value corresponding with the key will be the payload of the ancillary_data_byte field of the descriptor encoded as hex digits. Each byte is to be encoded as a two digit hexadecimal value where alphabetic characters are lowercase, starting with the first byte of the payload and finishing with the last.

3) If the TV Device detects the presence of a ci_ancillary_data descriptor in the descriptor loop of the SDT for the service to which this event belongs, then the key "anc_sdt" shall be included. The value corresponding with the key will be the payload of the ancillary_data_byte field of the descriptor encoded as hex digits. Each byte is to be encoded as a two digit hexadecimal value where alphabetic characters are lowercase, starting with the first byte of the payload and finishing with the last.

4) For the service to which this event belongs, if the TV Device chose to install that service because it is part of a particular bouquet (BAT), and the TV Device detects the presence of a ci_ancillary_data descriptor in the 1st descriptor loop of the BAT for that, then the key "anc_bat" shall be included. The value corresponding with the key will be the payload of the ancillary_data_byte field of the descriptor encoded as hex digits. Each byte is to be encoded as a two digit hexadecimal value where alphabetic characters are lowercase, starting with the first byte of the payload and finishing with the last.

5) If the TV Device detects the presence of a ci_ancillary_data descriptor in the 1st descriptor loop of the NIT that applies to the service to which this event belongs, then the key "anc_nit" shall be included. The value corresponding with the key will be the payload of the ancillary_data_byte field of the descriptor encoded as hex digits. Each byte is to be encoded as a two digit hexadecimal value where alphabetic characters are lowercase, starting with the first byte of the payload and finishing with the last.

If the value of a key-value pair is not present, then the key-value pair shall not be included in the query. A value that is zero-length (because its value has been derived from a zero-length descriptor payload) is considered to be present.

Clause 5.2.3.6 defines how the TV Device shall signal the CI as the broadcast and IPTV signalling becomes known.

For DVB IPTV services the steps above for constructing the CI are modified as follows:

- If there is an EIT in the stream, then steps 1 and 2 above are not modified. However, it is possible, though unlikely, that a DVB IPTV service will not include an EIT. As such the information normally extracted from the EIT will not be available, and in this case a CI may be generated without this information. However, where the episode CRID of the content is known to the receiver (e.g. through information provided via the Broadband Content Guide (BCG) [27]), then step 1 above shall encode this episode CRID value in the CI.

- Step 3 above shall be modified if there is a ciAncillaryData [5] element in the Service Discovery and Selection (SD&S) XML of IP service of the service for which this CI is being generated. In this case the key anc_sdt shall be included. The value corresponding with the anc_sdt key will be the payload of the AncillaryDataBytes element of the ciAncillaryData element as received. If the IPTV service includes the SDT, then the value from the ciAncillaryData element shall take precedence over the SDT in the stream. If ciAncillaryData element is not present, but there is an SDT, then any ci_ancillary_data descriptor values contained in the SDT shall be encoded as described in step 3.

NOTE 1: The reference to [5] above is intended to be valid for a future edition of ETSI TS 102 034 [5]. Annex F details the relevant parts of the future editions and may be used until the future edition is available.

- Step 4 and 5 above shall be omitted. If a BAT or NIT are present in the IPTV service stream it shall be ignored for the purposes of generating the CI query part.

NOTE 2: For DVB IPTV services do not use the anc_bat and anc_nit keys.

5.2.3.6 Progressive build-up of CIs for DVB Broadcast and IPTV services

5.2.3.6.1 Common

The TV Device may emit a sequence of zero or more "partial" CIs as the TV Device progressively acquires the signalling from which the CI is derived. Once the TV Device has acquired all of the signalling contributing to the CI or determined that the signalling is not present then the TV Device shall emit a "final" CI. The progression of CIs that may be emitted is constrained as described in this clause.

There are four potential forms of the CI URI:

a) CI up to and including net path as defined in clauses 5.2.3.2 and 5.2.3.3.

b) CI up to and including net path as defined in clauses 5.2.3.2 and 5.2.3.3 and if applicable the anc_sdt, anc_nit and anc_bat query keys as defined in clause 5.2.3.5.

c) CI up to and including the event information conveyed in the event constraint as defined in clause 5.2.3.4 and if applicable the ep_crid and anc_eit query keys as defined in clause 5.2.3.5.

d) Complete CI up to and if applicable including the anc_sdt, anc_nit and anc_bat query keys as defined in clause 5.2.3.5.

NOTE: Form A and form B are indistinguishable when the anc_sdt, anc_nit and anc_bat query keys are not present. The same is true of form C and form D. Form A is a subset of forms B, C and D. Form C is a subset of form D.

Two sequences of CI URI forms are allowed.

The first sequence applies if the TV Device has already acquired the signalling from which the anc_sdt, anc_nit and anc_bat query values are derived:

1) Form B, typically emitted on starting a channel change

2) Form D

The second sequence applies if the TV Device has not yet acquired the signalling from which the anc_sdt, anc_nit and anc_bat query values are derived:

1) Form A, typically emitted on starting a channel change

2) Form C

3) Form D

While the TV Device is presenting a service the SI may update to indicate a new current event. Soon after the TV Device detects that the EIT present/following actual has been updated it shall emit a modified CI. The signalling carried in the SDT, NIT and BAT, if present, is assumed to be pseudo static. Once this signalling has been acquired then the CI emitted at event boundaries shall be a single "final" CI with form D.

5.2.3.6.2 DVB Broadcast systems

Forms A, B and C are "partial" CIs. These are allowed transient states when some of signalling has not yet been acquired by the TV Device (or are not yet known to be absent).

Form D is the normal "final" and steady state for the CI once all required signalling has been acquired (or has been determined to be absent).

In some cases the TV Device may emit a CI with status "partial" and then subsequently emit the same CI but with status "final" once all the signalling that might contribute to the CI has been acquired (or has been determined to be absent).

For each DVB SI event there will be a single "final" CI with form D.

5.2.3.6.3 DVB IPTV systems

DVB IPTV systems follow the rules for DVB broadcast systems with the following allowed variants:

1) If the IPTV system does not carry EIT information and the CRID of the current event is not known to the TV Device by some other means then event information in the CI is not available. If additionally the optional ciAncillaryData in the SD&S and optional signalling in the SDT is not present then the "final" CI may only provide the net path information in form A.

2) If the IPTV system does not carry EIT information and the CRID of the current event is not known to the TV Device by some other means but the optional ciAncillaryData in the SD&S or the optional signalling in the SDT are present then the "final" CI will be in form D but the information derived from the EIT will not be present. Form A or form B may also be emitted as a "partial" CI prior to the "final" CI.

5.2.4 DVB DASH services

DVB services may be delivered in an ISO BMFF format using the MPEG DASH specification [16]. As this delivery format does not include the same signalling information as a DVB Broadcast or IPTV service, it is not possible to generate a CI URI according to the rules set out in clause 5.2.3. In this case, the CI URI shall consist of the absolute URL initially used to retrieve the MPD followed by an URI fragment structure [18]. The CI URI shall comply with the MPD anchor definition in clause C.4 of the DASH specification [17], and the URI fragment of the CI URI:

- shall include the **period** parameter; and
- may include a **mpd_ci_ancillary** parameter; and
- may include a **period_ci_ancillary** parameter; and
- shall not include any other parameters.

The **period** parameter shall always be the first parameter, and shall represent the period of the MPD that is currently being presented by the TV Device. If the currently presenting period has an ID defined in the MPD, then the value of this parameter shall be that ID and shall follow the rules set out in clause C.4 of the DASH specification [17]. If the currently presenting period does not have an ID then the value of this parameter shall be an empty string.

If present, the **mpd_ci_ancillary** parameter shall immediately follow the **period** parameter, and shall carry exactly the bytes carried in the **ciAncillaryData** element of the MPD, as defined in clause E.2.3.

NOTE: This means that the base64 encoding is used directly without transformation.

If present, the **period_ci_ancillary** parameter shall immediately follow the **mpd_ci_ancillary** parameter or the **period** parameter if the **mpd_ci_ancillary** parameter is not present, and shall carry exactly the bytes carried in the **ciAncillaryData** element (as defined in clause E.2.3) of the Period signalled in the period parameter.

Even if the MPD is updated or retrieved again, the URL used to retrieve the URL for the first time, prior to any redirections, shall always be used. The DASH CI as described above is shown in table 5.2.4.1, as a formal specification expressed in Augmented BNF as defined in IETF RFC 5234 [4] using the Core Rules defined in Appendix B.1 of [4] and the rules defined previously in table 5.2.3.1.1.

Table 5.2.4.1: Syntax of the CI URI for DVB DASH services

`ci-dvb-dash`	=	mpd-anchor ["&" mpd-ci-ancillary] ["&" period-ci-ancillary]
`mpd-anchor`	=	MPD "#" mpd-period ; see note 1
`mpd-period`	=	%x70.65.72.69.6f.64 "=" periodID ; "period"
`periodID`	=	1*unreserved; see note 2
`mpd-ci-ancillary`	=	%x6d.70.64.5f.63.69.5f.61.6e.63.69.6c.6c.61.72.79 "=" ci-ancillary-data ; "mpd_ci_ancillary"; see note 2
`period-ci-ancillary`	=	%x70.65.72.69.6f.64.5f.63.69.5f.61.6e.63.69.6c.6c.61.72.79 "=" ci-ancillary-data ; "period_ci_ancillary"; see note 2
NOTE 1: MPD is the absolute URI form of the MPD, as defined in IETF RFC 3986 [1]. The use of the "#" rather than the "?" is mandated by the URI fragment syntax, and hence the difference from the form used for the DVB CI in clause 5.2.3.		
NOTE 2: ci-ancillary-data and unreserved are as defined in table 5.2.3.1.1.		

5.2.5 Other types of service

For Timed Content that is not delivered as any of the other types of service defined in the previous sub clauses of clause 5.2, the URI used as the CI:

- should still comply with the general requirements defined in clause 5.2.1;

- and shall not use the URI scheme identifier "dvb" unless it is a DVB defined system for which a DVB locator syntax has been defined which can be guaranteed to not match any CI that could be generated according to the format defined in clause 5.2.3;
- but is otherwise outside the scope of the present document.

NOTE: The CI format is not defined here because the signalling and carriage of non-DVB services is application or platform dependent. It is up to individual platforms or applications to define their own profile for using a URI to uniquely and unambiguously identify the Timed Content and the delivery mechanism.

5.3 Timelines

5.3.1 General

A Timeline is a linear scale against which time can be measured for a particular system. Examples of such systems include:

- a clock, whose tick count can be measured;
- or a broadcast or streamed media system (for audio and/or video) where values of time correspond to particular moments in the media and are either signalled explicitly or derived from the progress of presentation.

Clause 5.3.2 describes concepts of tick rate and accuracy for a Timeline.

Clause 5.3.3 defines sources of Timelines for Timed Content presented by a TV Device and defines Timeline Selectors that are used to specify a particular source of Timeline as part of the CSS-TS interface procedure defined in clause 4.3.5.

Clauses 5.3.4 and later clauses within clause 5.3 define how different sources of Timeline are signalled and how this signalling is to be interpreted by a TV Device to derive a Timeline.

The use of multiple timelines is permissible within the present document.

5.3.2 Tick rate and accuracy properties

A Timeline is a linear scale against which time can be measured for a particular stream or content item (such as progress of a broadcast or position within a content item).

A Timeline has properties that characterize it:

- A Timeline has a scale that defines the rate of ticks as numbers of ticks per second.
- A Timeline may also have a measure of accuracy associated with it.

A Timeline does not have any inherent absolute relationship to real world date and time, unless otherwise stated for a particular Timeline.

A Time Value is a measure of a moment in time for a particular Timeline. Time Values are represented by an integer number of ticks (positive or negative).

The scale of a Timeline is defined as a rational fraction $unitsPerTick \,/\, unitsPerSec$. The scale represents the duration of a single tick, measured in seconds. The reciprocal of the fraction defines the rate measured in ticks per second.

EXAMPLE 1: unitsPerTick = 1
unitsPerSecond = 50

Therefore:
scale = $1/50$ = 0,02 secs per tick
rate = $50/1$ = 50 ticks per second

Given a time period (measured in ticks) = 100
Then the time period (measured in seconds) = $100 \times (1/50)$ = 2 secs

A Timeline may represent the flow of time of some underlying media and the tick rate can be fast enough to discriminate individual access units (such as video frames). However there may be limitations to the accuracy with which Time Values on that Timeline correspond to the actual editorial content of the media.

NOTE: Limitations of Timeline Accuracy do not affect the behaviour of the TV Device. However a Companion Screen Application might take accuracy into account when assessing how precisely it can synchronize its timing of presentation of Timed Content to the TV Device, from the perspective of what the user will see and hear.

This can arise due to practical limitations to the accuracy with which Timeline information is embedded in, or extracted from, the media stream or limitations to the certainty with which the media stream is being authored from an editorial perspective.

EXAMPLE 2: A Timeline for a broadcast stream ticks at 50 ticks per second, which in principle would be sufficient for frame accurate Time Values (and in turn, frame accurate synchronization). However when the Timeline Time Values are embedded into the broadcast stream, this is only done within an accuracy of 0,2 secs.

The effect of this limitation can be modelled as a Gaussian (normal) distribution representing the probability that a time t is inaccurate by a particular amount d with respect to the underlying media.

The limitation on accuracy is then quantified as a number of seconds, equal to at least 2 standard deviations (95 % confidence). So given a Time Value t for a piece of media where the Timeline has accuracy *a*, then there is at least a 95 % likelihood that $-a < d < a$ where d is the amount by which t may be inaccurate.

If two items of underlying media are synchronized, then there will be a correlation between their Timelines. When mapping a Time Value from the Timeline of one to the Timeline of the other, the accuracy of that mapped value will be affected by the accuracies of both of their Timelines. Assuming the sources of uncertainty are independent variables, given two items of media X and Y with respective Timeline accuracies a_x and a_y then the accuracy of a mapped Time Value in either direction (from X to Y or Y to X) will be:

$$\sqrt{{a_x}^2 + {a_y}^2}$$

EXAMPLE 3: A Timeline X has the following properties:

accuracy = 0,5 secs

unitsPerTick = 1

unitsPerSecond = 50

bounds on amount of inaccuracy 95 % of time: $\pm 0{,}5 \times 50 = \pm 25$.

EXAMPLE 4: A second Timeline Y is correlated with a Timeline X:

accuracy_x = 0,5 secs

accuracy_y = 0,2 secs

Possible sources of error/uncertainty are assumed to be independent for both Timelines.

Given a point on Timeline X, then the accuracy of the corresponding point on Timeline Y (or vice versa) shall be:

$\text{accuracy}_{x,y} = \sqrt{0{,}5^2 + 0{,}2^2} = 0{,}539$ secs (to 3 decimal places).

5.3.3 Sources of Timelines and the Timeline Selector

Different types of broadcast services and other Timed Content use different types of signalling mechanism to convey Timelines associated with the progress of presentation of the Timed Content.

Some types of Timeline are signalled in a way that is intrinsic to the Timed Content, such as PTS in MPEG TS and Composition Time in ISOBMFF. Intrinsic types have the advantage that they are always present, but they have the disadvantage that they can be modified when the content is re-multiplexed, transcoded or re-originated by the distribution network.

Some other types of Timeline are signalled in a way that is extrinsic and specially added to the Timed Content. Extrinsic types have the advantage that they are never modified, but some may have the disadvantage that they may be stripped off by the distribution network.

A third class of timeline is embedded within the codec for media streams carried in the audio and or video data of supporting codecs. It has the advantages of always being present, not being modified under re-multiplexing and some forms of transcoding, and not stripped off by distribution networks. It has the disadvantage of relying on the AV codecs to support it and so is not universal across multiple codecs.

No timelines of this third type are currently defined or mandated in the present document, but the specification allows for their use as a timeline.

The Timeline Selector is a URI that specifies the source of a Timeline by indicating its type and information needed to locate the signalling that conveys Time Values on it. A CSA may obtain a Timeline Selector as part of Synchronization Timeline Information (see clause 5.5.9.4) that is provided by an MRS via the CSS-MRS interface (as detailed in the procedure described in clause 4.3.3). The CSA using the CSS-TS interface (as detailed in the procedure described in clause 4.3.5) sends a Timeline Selector to the TV Device to select the Synchronization Timeline used in Timeline Synchronization. The Timeline Selector is conveyed in a message described in clause 5.7.3.

A CSA may also obtain a Timeline Selector by some other means that is not specified in the present document.

Table 5.3.3.1 indicates the values of Timeline Selector that are defined in the present document along with the reference to the clause that describes how that particular type of Timeline is signalled and how to locate the signalling.

Table 5.3.3.1: Specification of values for timelineSelector

Name	Clause	Timeline Selector value	Intrinsic / Extrinsic
MPEG-TS PTS: Presentation Time Stamp	5.3.4	"urn:dvb:css:timeline:pts"	Intrinsic
ISOBMFF: composition time	5.3.5	"urn:dvb:css:timeline:ct"	Intrinsic
TS Adaptation: TEMI	5.3.6	"urn:dvb:css:timeline:temi:<component_tag>:<timeline_id>"	Extrinsic
MPEG DASH: Period relative Timeline	5.3.7	"urn:dvb:css:timeline:mpd:period:rel:<ticks-per-second>" "urn:dvb:css:timeline:mpd:period:rel:<ticks-per-second>:<period-id>"	Intrinsic

Other values of Timeline Selector may be supported by a TV Device or provided by an MRS in order to support future or proprietary Timeline signalling mechanisms. These are outside of the scope of the present document.

EXAMPLE: A TV Device supports use of a timeline embedded within a particular audio codec. Organization "Acme" wishes to standardize its use as a Timeline. The organization could choose any well-formed URI scheme to identify the timeline, such as a URN. The organization has registered a URN namespace "urn:acme" and assigned a URN within that namespace that identifies the timeline. It may also identify the embedding method if there is more than one option for this. Timed Content often comprises several media components with more than one using the same codec (e.g. normal audio and alternative language audio). The URN therefore also includes a field that carries an identifier for the media component and the meaning of that field is profiled for when the codec is carried in MPEG transport streams or ISOBMFF containers.

5.3.4 MPEG-TS PTS: Presentation Time Stamp

If PTS is to be used as a Timeline, then the Time Value is the value of PTS (as defined in [6]) of the media stream being presented to the user.

The unitsPerTick of the timeline shall be 1 and the unitsPerSecond shall be 90 000, corresponding to the tick rate of clock underpinning PTS.

When PTS values wrap then the Time Value shall also wrap and shall never equal or exceed 2^{33}.

5.3.5 ISOBMFF: composition time

Where media is provided in an ISOBMFF [7] encapsulation, the composition time may be used as the Time Value, in a manner analogous to the use of the PTS field of the PES packet header, as described in clause 5.3.4.

NOTE 1: The composition time is derived from the decode time through the addition of a delta to the decode time, where the delta is zero if not present. The decode time is carried in the ISOBMFF as a delta value (differences) from the previous values for the decoding value. Thus deriving the composition time requires performing the arithmetic to sum all the sample duration values which add to the decode time.

In the case of fragmented ISOBMFF movie files the track fragment decode time box (the box identified with the 4CC 'tfdt') provides the sum of all previous fragment decode times, and therefore is the starting point for the calculation of the sum of all sample duration values.

In the case of fragmented ISOBMFF movie files, if the track fragment decode time box is not present then the composition time shall not be used as a Timeline. Fragmented ISOBMFF movie files are those that make use of the movie fragment box (the box identified with the 4CC 'moof').

NOTE 2: Fragmented movie files are used by MPEG DASH, and DASH requires the presence of the track fragment decode time box.

The unitsPerTick of the timeline shall be 1, and the unitsPerSecond shall be taken from the largest timescale value carried in either the timescale element of the movie header box (the box identified with the 4CC 'mvhd') or the timescale element of the media header box (the box identified with the 4CC 'mdhd') within the media box (the box identified with the 4CC 'mdia') within any track box (the box identified with the 4CC 'trak').

NOTE 3: The ISOBMFF allows different tracks to operate at different clock rates, although all operate against the same clock. The above text is to ensure that the most detailed clock is used. However, it is very possible that this clock may not be at a particularly high frequency.

5.3.6 TS Adaptation Packet

Clause 11 defines methods by which an extrinsic Timeline can be carried in a ISO/IEC 13818-1 [6] transport stream packet's adaptation layer. The Time Value conveyed for the Timeline applies to the first access unit that follows the relevant adaptation layer packet, which may be in the same packet.

Time Values for the Timeline for an access unit not immediately preceded by an adaptation layer packet may be inferred.

Clause 11 also defines the format of the Timeline Selector and how a TV Device uses the information embedded within the Timeline Selector string to determine the location and identity of the Timeline in the broadcast signalling and how to derive the Time Values, unitsPerTick and unitsPerSecond of the Timeline.

5.3.7 MPEG DASH: Period relative Timeline

5.3.7.1 General

DVB services may be delivered in MPEG DASH format [17] using the MPEG DASH specification [16]. An MPEG DASH Period relative Timeline is a Timeline that is relative to the start of a chosen Period in an MPEG DASH Presentation. The chosen Period is determined by the Timeline Selector used to select this Timeline and is known as the base Period.

A Time Value of 0 corresponds to the start of the base Period. Points in the presentation before the base Period will result in negative Time Values, and points in the presentation after the start of the base Period will result in positive Time Values.

5.3.7.2 Timeline Selector for a Period relative Timeline

The Timeline Selector for a Period relative Timeline for an MPEG DASH presentation is a URN with one of the following forms:

1) urn:dvb:css:timeline:mpd:period:rel:<ticks-per-second>

2) urn:dvb:css:timeline:mpd:period:rel:<ticks-per-second>:<period-id>

For both forms **urn:dvb:css:timeline:mpd:period:rel** identifies that an MPEG DASH Period relative Timeline is to be used.

For both forms, **<ticks-per-second>** is a base-ten integer number as a string that defines the scale of the Timeline. The unitsPerTick of the Timeline shall be 1. The unitsPerSecond of the Timeline shall be value of <ticks-per-second>.

For form (2) only, **<period-id>** identifies the Period to be used as the base Period. It consists of the value of the Period@ID attribute of the Period and is escaped according to the rules defined for a Name Specific String (NSS) in a URN in section 2 of IETF RFC 2141 [26].

For form (1) only, the base Period is the first Period in the MPEG DASH Presentation Description (MPD). If the MPD is dynamic, the base Period is the first Period in the MPD in its current state as maintained by the TV Device.

NOTE 1: Form (1) can be used when the MPD is static such as for a pre-authored VoD asset. In this situation the CSA and/or MRS do not need to know the Period ID.

NOTE 2: If the MPD is dynamic, then it is recommended to always use form (2) and to not use form (1). It is likely that the CSA and/or MRS will be unable to know which Period is the first in the MPD being used by the TV Device.

5.3.7.3 Determining Time Values and Timestamps for a Period relative Timeline

For an MPEG DASH Period relative Timeline, the Time Value is the total amount of time elapsed since the start of a chosen MPEG DASH Period. The chosen Period is the one chosen as the base Period for the Timed Content currently being presented to the user.

NOTE 1: The value of the Period@start attribute is not included in the calculation.

The Time Value for the Timeline that forms part of the Timestamp shall be converted to match the units of the Timeline scale and shall be able to convey any Time Value t ticks in the range $-2^{63} \leq t < 2^{63}$. The Time Value shall be rounded to the nearest integer number of ticks.

NOTE 2: Care is to be taken with implementations and choice of tick rate for non-integer frame rate video. The colloquially named 29,97 fps is actually 30 000 / 1 001 frames per second. Therefore a TV Device that is presenting 29,97 fps content reports 1 001 ticks passing per frame of video if an MPEG DASH Period Relative Timeline is defined as having a tick rate of 30 000 ticks per second. A broadcaster's choice of tick rate in the Timeline Selector will affect the accuracy of Time Values reported by the TV Device.

EXAMPLE: The TV Device is presenting an MPD presentation.
The Timeline Selector is: urn:dvb:css:timelime:mpd:period:rel:25:3f2a5
This specifies that the timeline should be relative to the start of period with id 3f2a5 and should have a scale of 25 ticks per second.

The MPD at the TV Device includes a sequence of consecutive Periods with attributes shown in table 5.3.7.3.1.

Table 5.3.7.3.1: Example MPD Period IDs and durations

Period@ID	Period Duration
3f2a4	30,00 secs
3f2a5	25,00 secs
3f2a6	22,50 secs
3f2a7	15,76 secs

At the current time:
Period@ID = "3f2a7"
Time since start of Period "3f2a7" = 5,28 secs
Wall Clock time = 1 385 628 462 000 000 000 nanoseconds

Then:
Timeline units per tick = 1
Timeline units per second = 25
Timeline time value = (25,00 + 22,50 + 5,28) × 25 = 1 319,5
Timeline time value (rounded) = 1 320
Timestamp = (1 320 ticks , 1 385 628 462 000 000 000 nanoseconds)

A Timeline indicated by a given Timeline Selector shall be available if it is possible to determine a Time Value on that Timeline that corresponds to the point currently being presented by the TV Device. It is possible to do this if:

- the TV Device is presenting an MPEG DASH presentation
- and the Period ID in the Timeline Selector corresponds to the value of any Period@ID currently in the MPD.

NOTE 3: The MPEG DASH specification [16] only requires support for xlink with actuate set to onLoad within an MPEG DASH MPD and therefore the TV Device resolves all xlink references when the MPD is loaded. However a CSA does not know what Periods xlink references have resolved to and therefore may not know what point in the Timed Content corresponds to a given Time Value on the Timeline. Clause B.7 describes how using different Timelines for Periods before and after an xlink referenced Period enables a CSA to address this.

5.4 Timeline correlation

A correlation between two Timelines expresses a deterministic linear monotonic relationship between them, such that there is a one-to-one mapping between Time Values on each Timeline. This is the basis of how a CSA can translate between a Material Timeline and a Synchronization Timeline, or how a CSA or TV Device can relate Wall Clock Time Values to Timeline Time Values.

A correlation between Timeline X and Timeline Y can be expressed as Correlation Timestamp, which is a pair of Time Values (c_x, c_y) representing a point of correlation where a Time Value c_x from Timeline X is deemed to occur at the same moment as a Time Value c_y from Timeline Y.

The respective scales (or rates) of each Timeline are expressed with respect to the duration of a second and can therefore be related to each other.

Assuming that the rates of both Timelines are known, any Time Value t_x on Timeline X can therefore be mapped to a point t_y on Timeline Y by relating it back to the point of correlation, as illustrated in figure 5.4.1.

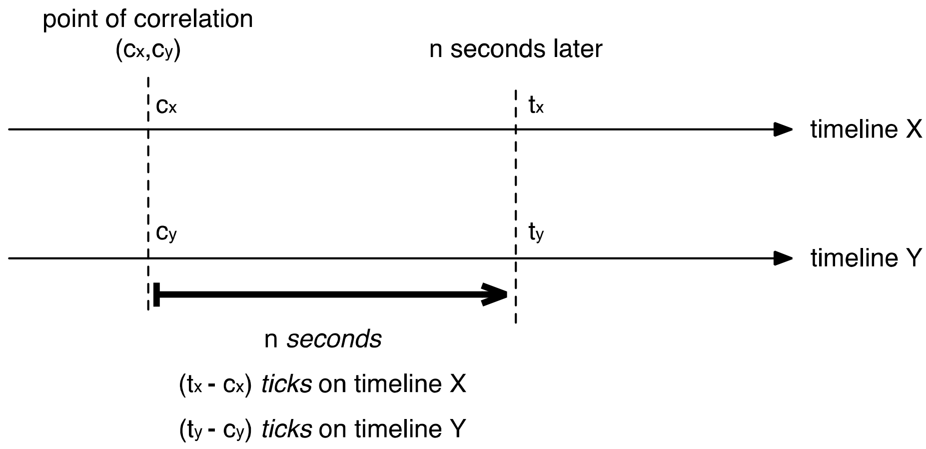


Figure 5.4.1: Correlation between two Timelines

Given the rate (ticks per second) r_x of Timeline X and r_y of Timeline Y, it is possible to derive the number of seconds n that t_x or t_y are later than the point of correlation:

$$n = (t_x - c_x)\frac{1}{r_x}$$

$$n = \left(t_y - c_y\right)\frac{1}{r_y}$$

Therefore a point t_x on Timeline X can be expressed instead as the equivalent point t_y on Timeline Y given by the equation:

$$t_y = c_y + (t_x - c_x)\frac{r_y}{r_x}$$

This can also be expressed as:

$$t_y = c_y + (t_x - c_x)\frac{units\_per\_sec_y \cdot units\_per\_tick_x}{units\_per\_tick_y \cdot units\_per\_sec_x}$$

If the relationship between the Timelines does not drift then the Correlation Timestamp correctly represents the relationship indefinitely. However if the relationship does drift then the Correlation Timestamp is only correct for an instant and becomes incorrect unless continuously updated. How frequently the Correlation Timestamp needs to be revised depends on how much inaccuracy in the Correlation Timestamp can be tolerated and how rapidly the timing relationship between the Timelines drifts.

NOTE: Drift is avoidable if timing reference sources used in different parts of a broadcast chain or content production process are all tightly synchronized, also known as genlock. If such tight genlocked synchronization cannot be achieved, then multiple Correlation Timestamps are used to provide a correlation at different points of correlation. See annex B for a discussion of some ways in which this can be achieved.

5.5 Material Information

5.5.1 General

Material Information is a collection of data describing the properties of one or more Materials including identifiers, private data, a Timeline for the Material and details of Trigger Events that may occur while the Material is being presented. Material Information also details the available Synchronization Timelines depending on the CI reported by the TV Device. Synchronization Timelines are used for the Timeline Synchronization communication between the TV Device and the CSA via the CSS-TS interface (see procedure in clause 4.3.5). Also included are mapping relationships between Synchronization Timelines and Material Timelines and a Timeline Selector value that is provided by the CSA at the beginning of the CSS-TS procedure to select the corresponding Synchronization Timeline as the Timeline used for exchanges of timing information in the procedure.

NOTE: For a DVB Broadcast, the available Synchronization Timelines will typically be the Timelines that are signalled in the DVB broadcast received by the TV Device. For on-demand streamed Timed Content, the available Synchronization Timelines will typically be the Timelines signalled in or derived from the streaming container format.

The MRS provides Material Information to the CSA via the CSS-MRS interface as part of the procedure detailed in clause 4.3.3. It is represented in JSON format as defined in clause 5.5.9 and is conveyed via a protocol defined in clause 7.

The model for Materials and Synchronization Timeline Information that makes up Material Information is illustrated using UML syntax in figure 5.5.1.1 and is explained in more detail in clauses 5.5.2 to 5.5.8.

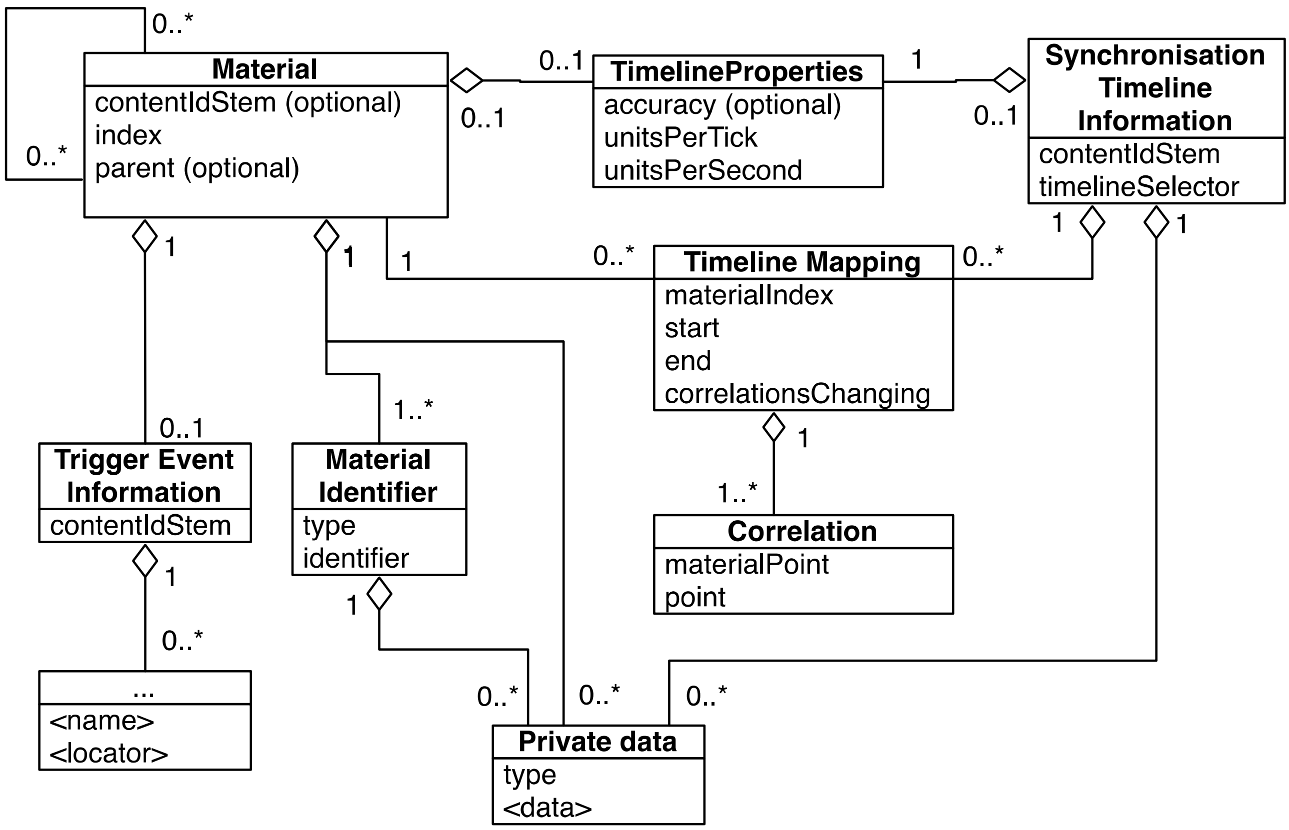


Figure 5.5.1.1: Model for Material Information

A Timeline Mapping from a Synchronization Timeline to a Material Timeline defines an interval of Time Values on the Synchronization Timeline. The Timeline Mapping also defines how every Time Value within that interval maps to a corresponding Time Value on the Material Timeline. Timeline Mappings are described in detail in clause 5.5.5.

A point on a Synchronization Timeline may map to the Timelines of more than one Material. If a point on a Synchronization Timeline does not map to the Timeline of a Material, then it is not possible to determine the time position on that Material's Timeline at that point in time. If there are no mappings then, by definition, it is not possible to accurately determine the time position on the Timeline of any Material.

A Material is considered to be active if it corresponds to what is currently being presented by the TV Device. The process by which a Companion Screen Application determines which Materials are Active Materials given the content currently being presented by the TV Device (as described by the CI and a time value on a Synchronization Timeline) is described in clause 5.5.8.

5.5.2 Material

A Material represents a segment of content with an editorially significant meaning, including, but not necessarily limited to:

- an interval of a broadcast service during which a given DVB event is signalled in EIT present/following actual as being the present event;
- an individual advert, trailer or continuity announcement, or a collection of these;
- the whole of or a segment of a programme;
- the whole of or a segment of content that could be presented by a Companion Application.

The representation of a Material in JSON is defined in clause 5.5.9.2.

Material Identifiers (see clause 5.5.3) associated with a Material can be used to associate editorial meaning with that Material. How Material Identifiers are used to do this is outside the scope of the present document.

NOTE 1: Clause B.2 provides examples and guidance on the use of Materials to represent editorial meaning.

A Material may optionally have a Timeline with properties of scale (tick rate) and the accuracy of the Timeline (see clause 5.3.2).

A Material will be active (see clause 5.5.8) when the CI has certain values and can therefore be considered to be associated with those one or more CIs.

EXAMPLE 1: For a DVB broadcast service a Material may represent a single programme and therefore correspond to a single CI. Another Material might represent the whole broadcast service brand and therefore correspond to any CI corresponding to a programme broadcast on that service.

However a CI does not necessarily provide the resolution needed to discriminate between multiple materials.

EXAMPLE 2: For a DVB broadcast service, the CI only changes at programme junctions, signalled by DVB Events. This is not sufficiently fine-grained to uniquely identify adverts or trailers.

Where CIs provide sufficient resolution, a Material can use a CI stem to determine when that Material is active. The process for checking if a CI matches a CI stem is defined in clause 5.2.2.

If a Material is active, this means that it relates to the point in the Timed Content currently being presented by the TV Device. A CSA can either determine if a Material is active by matching the CI stem for that Material to the CI reported by the TV Device, or by matching the CI to the CI stem of a Synchronization Timeline and then using Timeline Mappings from that Synchronization Timeline to the Material Timeline. If the current point on the Synchronization Timeline reported by the TV Device is within a mapping interval for this Material then the Material is active. When and how to apply these two approaches is described in more detail in clause 5.5.8.

NOTE 2: If a Material has no CI stem and no Timeline Mappings from a Synchronization Timeline the Material's Timeline then a CSA is unable to determine if that Material is active.

A Material may optionally have zero, one or more parent Material and therefore any Material will have zero, one or more children. Hierarchical relationships shall be acyclic. When parent-child relationships are present, this form of hierarchical relationship indicates that a child Material is considered to be editorially a sub-part within a parent Material. However, the bounds of mapping intervals for a child Material may not necessarily lie entirely within the bounds of a mapping interval for a parent Material. CSA behaviour because of an indicated hierarchical relationship is outside the scope of the present document.

NOTE 3: A parent-child relationship does not necessarily guarantee that a child Material will only be active when its parent Material is also active. This relationship only signals an editorial intent. Issues such as practical limitations in the generation of broadcast signalling and Material Information or late editorial changes may result in situations where (according to the process described in clause 5.5.8) a child Material is determined to be active but its parent Material is not. Clause B.2.2 describes scenarios in which this situation can arise.

A Material may optionally have information that lists Trigger Events that may occur during presentation of the Material. This is described in clause 5.5.7.

5.5.3 Material Identifiers

A Material shall have one or more Material Identifiers. A Material Identifier identifies a Material. A Material Identifier has type and value. The value shall consist of a single token string of characters with no separators (space characters or newline characters). The type is encoded as a URI and shall scope the identifier globally.

EXAMPLES: Type: urn:schemename:eidr
Value: 10.5240/5FD4-FEE1-22F5-583E-FECC-0

Type: tag:bbc.co.uk/programmes/episode
Value: p00t8qnw

Type: urn:tva
Value: crid://fp.bbc.co.uk/182af2

For a Material Identifier of a given type, the interpretation of the value, its scope and its uniqueness are outside the scope of the present document.

NOTE: The meaning, scope and uniqueness of the values are implied by the type, e.g. for one type of identifier, values could be permanently and globally unique, whereas for a different type the values could be reused.

The representation of a Material Identifier in JSON is defined in clause 5.5.9.3.

5.5.4 Synchronization Timeline information

An item of Synchronization Timeline information describes:

- the properties of a Synchronization Timelines (see clause 5.3.2);
- the circumstances in which that Synchronization Timeline may be used;
- optionally the lead-in circumstances just prior to when that Synchronization Timeline may be used;
- the mappings between that Synchronization Timelines and the Timelines of Materials;
- and the means of selecting that Synchronization Timeline for use.

The representation of an item of Synchronization Timeline information in JSON is defined in clause 5.5.9.4.

The properties of a Synchronization Timeline are its scale (tick rate) and information about its accuracy as defined in clause 5.3.2. The representation of Timeline properties in JSON is defined in clause 5.5.9.5.

The circumstances are defined in terms of the set of CIs for which the Synchronization Timeline can be used. A CI stem describes this set of CIs. If the CI stem matches the CI reported by the TV Device, then the Synchronization Timeline is available for use. If it does not match then that Synchronization Timeline is not available. The process for matching is defined in clause 5.2.2.

Synchronization Timeline information may optionally describe the lead-in circumstances prior to when a Synchronization Timeline is anticipated to become available for use. The lead in circumstances is a set of CIs, represented by a CI stem. When the CI reported by the TV Device matches the CI stem, this indicates to the CSA that the corresponding Synchronization Timeline will be available shortly.

NOTE: If this is provided, a CSA can interpret it as a cue to establish the connections it needs to start receiving Control Timestamps for the Synchronization Timeline. The CI stem for the lead-in circumstances will match the reported CI just prior to it becoming available and so not miss the start.

The Timeline selector is a string that uniquely identifies a Timeline within the scope of the set of CIs described by the CI stems. The CSA supplies the Timeline selector in the Timeline Synchronization procedure in order to request that the TV Device use that particular Timeline as the Synchronization Timeline. This procedure is described in clause 4.3.5 and the protocol used is defined in clause 7.3.

Timeline selectors corresponding to each type of Timeline are defined in clause 5.3.3.

5.5.5 Timeline mappings

For Material Information a Timeline Mapping is a relationship between a Synchronization Timeline and a Material Timeline. It is associated with a set of one or more Correlation Timestamps (see clause 5.5.6).

The representation of a Timeline Mapping in JSON is defined in clause 5.5.9.6.

A Timeline Mapping also indicates if the Correlation Timestamps associated with it are anticipated to change. The present document defines update mechanisms that a CSA may use to obtain any changes that are made to the set of Correlation Timestamps when carried as part of Synchronization Timeline information. This is described in clause 7.6.

NOTE 1: A likely use case for indicating that Correlation Timestamps are anticipated to change, and providing updates to the set of Correlation Timestamps, is where a live broadcast is accompanied by a stream of Timed Content to be presented by the CSA that is not genlocked to the live broadcast.

A Mapping defines a finite interval on the Synchronization Timeline for which a Material's Timeline is considered to apply. The lower bound of the interval is inclusive, but the upper bound is exclusive. Within that interval, one or more Correlation Timestamps map from Time Values on the Synchronization Timeline to Time Values on the Material Timeline. This is explained in more detail in clause 5.5.6.

The underlying way in which Time Values are encoded and conveyed in a media stream for either Synchronization Timeline or Material Timeline may impose limits on the minimum and maximum values that can be conveyed.

The interval expressed on the Synchronization Timeline in a mapping shall not have a value for its lower bound that is greater than its upper bound.

NOTE 2: This implies that if a Timeline's values wrap, a mapping that spans the wrapping point is instead expressed by splitting it into two separate mappings, one covering the interval before the point where the wrap occurs and the other covering the interval after the wrap. Given a situation where wrapping occurs because Time Values t on the Synchronization Timeline are limited to the interval $min \leq t < max$ then the mapping is not expressed as a single mapping. It is instead expressed as two separate mappings: the first with interval $start \leq t < max$ and the second with interval $min \leq t < end$ where $start$ and end are the start and end Time Values of the original mapping.

NOTE 3: Clause B.2.4 provides examples and guidance on handling of wrapping situations and other possible sources of ambiguity in timeline mapping.

5.5.6 Correlation Timestamps

A Timeline Mapping is associated with one or more Correlation Timestamps. As described in clause 5.4, a Correlation Timestamp between two Timelines (combined with knowledge of the tick rates of the timelines) enables points on one Timeline to be converted to and from a point on another Timeline.

The representation of a Correlation Timestamp in JSON is defined in clause 5.5.9.7.

One of the Correlation Timestamps defined for a Timeline Mapping will apply for any given Time Value t on the Synchronization Timeline that lies within the interval defined by the mapping. The Correlation Timestamp that applies is the one with the largest Time Value on the Synchronization Timeline that is less than t. If no Correlation Timestamp meets this criteria, then the Correlation Timestamp with the smallest Time Value on the Synchronization Timeline applies.

If a Time Value on the Synchronization Timeline lies outside of that interval then none of the Correlation Timestamps associated with that mapping will apply.

EXAMPLE: Figure 5.5.6.1 illustrates which Correlation Timestamps apply given a set of 4 Correlation Timestamps C1, C2, C3 and C4 associated with a mapping interval.

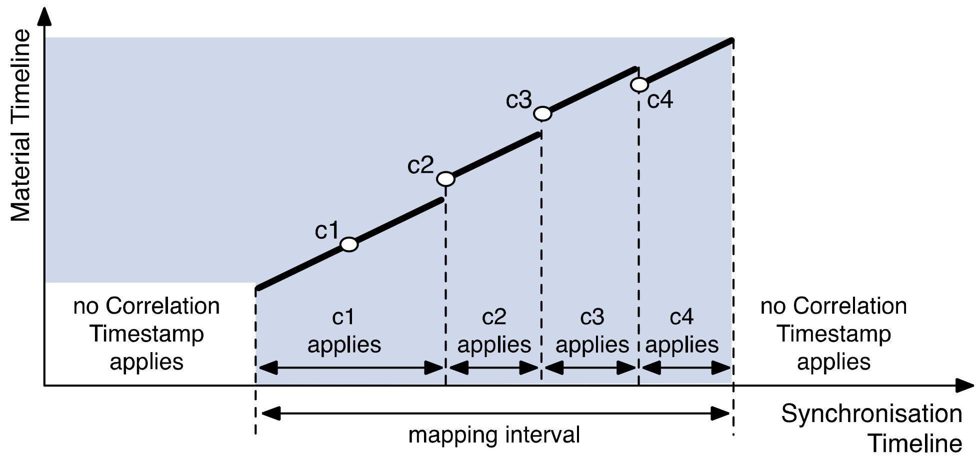


Figure 5.5.6.1: Example of when a Correlation Timestamp within a mapping interval applies

NOTE: The variation from one Correlation Timestamp to the next will result in small ambiguities where multiple points on the Synchronization Timeline map to the same point on the Material Timeline. This is unavoidable in situations where the Material Timeline is not perfectly genlocked to the Synchronization Timeline and cannot be resolved by splitting a mapping into multiple mappings as described in clause 5.5.5.

5.5.7 Trigger Event info

A Material may have Trigger Event info. Trigger Event info describes Trigger Events that may occur while a Material is being presented. This information provides details needed to request that the TV Device notify the CSA if a Trigger Event occurs via the CSS-TE interface (see procedures defined in clause 4.3.6).

The representation of Trigger Event Info in JSON is defined in clause 5.9.8. Trigger Event info also describes the circumstances (context) in which the Trigger Events may be detectable by the TV Device in terms of the set of CIs describing when the Trigger Events can be detected. A CI stem describes the set of CIs and is defined and matched to the CI reported by the TV Device in the manner described in clause 5.2.2.

Trigger Event info shall list zero, one or more Trigger Events. Each Trigger Event is described by a name that identifies it and a locator that can be passed to the TV Device to inform it that the CSA wishes to be notified if that Trigger Event occurs. A CSA should recognize Trigger Events by their name. The locator is specific to the mechanism by which Trigger Events are signalled.

NOTE: The name therefore provides a decoupling to ensure that the CSA does not need to understand Trigger Event signalling mechanism specifics.

The Trigger Event name is a string restricted to be one or more characters in length and starting with a lower case or upper case letter followed by lower case or upper case letters, numbers or underscore characters.

The locator is a URI string.

5.5.8 Process for determining which Materials are Active Materials

A Material is an Active Material if it relates to the editorial content currently being presented by the TV Device. Zero, one or more Materials may be active at any given moment. The CI reported by the TV and the current time position on a Synchronization Timeline are used by the CSA to determine whether a Material is active or not according to the process described here.

A CSA can use the CSS-TS interface (see procedures in clause 4.3.5) to obtain Time Values from the TV Device that are on a Synchronization Timeline of the CSA's choosing. If the CSA has not obtained a Time Value from the TV Device (e.g. because it is not using the CSS-TS interface) then the CSA can determine if any Material with a CI stem is active by matching it with the CI currently reported by the TV Device. If there is a match then the Material is considered to be active. The process for matching a CI stem to a CI is defined in clause 5.2.2. All other Materials are inactive.

If the CSA has obtained a Time Value on a Synchronization Timeline then whether a Material is active is determined according to the following process:

> For each Material under consideration and given a CI currently reported by the TV Device and a Time Value on the Synchronization Timeline known to correspond to Timed Content currently being presented by the TV Device, then:
>
> 1) If there exists one or more mappings between the Synchronization Timeline and the Material's Timeline, then the Material is active if the current Time Value lies within the interval of at least one of the mappings. Otherwise the Material is inactive. If the Material has a CI stem, it shall be ignored.
>
> 2) Otherwise: if the Material has a CI stem and it matches the current CI (according to the process defined for matching in clause 5.2.2), then the Material is active. However it will not be possible to determine a position on a Timeline for the Material.
>
> 3) Otherwise: the Material is not active.

Given several Materials, this process can result in the decision that more than one Materials are considered to be active.

The decision process described in this clause is summarized in the flow chart in figure 5.5.8.1.

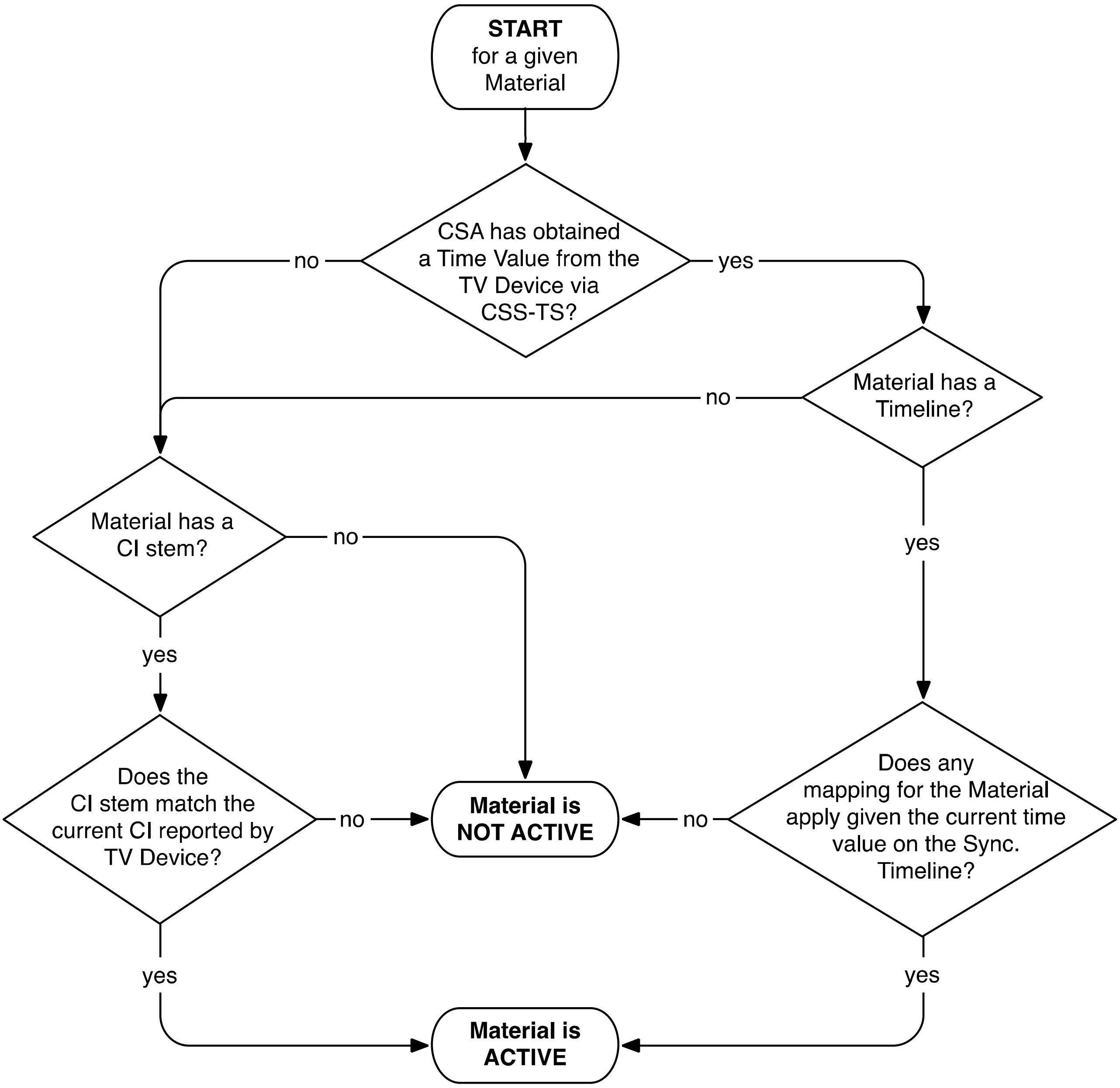


Figure 5.5.8.1: Flow chart decision process for determining if a Material is active

5.5.9 JSON syntax for representing Material Information

5.5.9.1 General

Material Information shall be represented as JSON objects as defined in clause 5.5.9. The JSON schema defined in clause A.1.3 can be used to validate Material Information that is part of an MRS response JSON Document as defined in clause 7.3.

The Material Information is delivered as part of a response from an MRS. The Materials and Synchronization Timeline information objects will be contained within an MRS response JSON object as described in clause 7.4 or as part of an MRS update JSON object as described in clause 7.6. Timeline Mappings are conveyed as part of Synchronization Timeline information. Clause A.2.2 provides an example of a full JSON Material Information response.

5.5.9.2 JSON for Material

A Material object represents a Material as defined in clause 5.5.2. It is represented by a JSON object with properties defined in this clause. If such a JSON object is correctly formed, it can be validated by the schema definition for "material" in the JSON schema in clause A.2.2. The template below illustrates this JSON object:

```
{
    "ids"                : [ ... ],
    "parents"            : [ <string>, ... ],
    "contentIdStem"      : <beginning-of-uri>,  (optional)
    "timelineProperties" : <timelineProperties> (optional)
    "triggerEventInfo"   : <triggerEventInfo>   (optional)
}
```

See the "materials" property in clause 7.4 for the mechanism for associating a unique name with a Material object.

Required properties:

- **ids** is a JSON array containing one or more Material Identifier objects as defined in clause 5.5.9.3.
- **parents** is an array of zero or more arbitrary string values. Each value shall match the name of one of the properties held in the "materials" JSON object (see clause 7.4). That property's value is another Material object, known as the parent Material, within the same JSON document. Values in this array indicate that this Material is considered a child of the parent Material.

Optional properties:

- **contentIdStem** is a string value. Its value is a CI stem as defined in clause 5.2.2.
- **timelineProperties** is a JSON object representing the properties of the Timeline of this Material, as defined in clause 5.5.9.5. The timelineProperties JSON object shall include an accuracy property in this context.
- **triggerEventInfo** is a JSON object (as defined in clause 5.5.9.8) representing the Trigger Events that may be available for this Material.

5.5.9.3 JSON for Material Identifier

A Material Identifier object represents a Material Identifier as defined in clause 5.5.3. It is represented by a JSON object with properties defined in this clause. If such a JSON object is correctly formed, it can be validated by the sub schema defined under the **"ids"** property in the schema definition for "material" in the JSON schema in clause A.2.2. The template below illustrates this JSON object:

```
{
    "type"    : <uri>,
    "id"      : <string>
}
```

Required properties:

- **type** is a URI string value representing the identifier's type.
- **id** is an arbitrary string value containing the identifier's value.

5.5.9.4 JSON for Synchronization Timeline information

A Synchronization Timeline information object represents Synchronization Timeline information, as defined in clause 5.5.4, for a single Synchronization Timeline. It is represented by a JSON object with properties defined in this clause. If such a JSON object is correctly formed, it can be validated by the schema definition for "syncTimelineInformation" defined in the JSON schema in clause A.1.3. The template below illustrates this JSON object:

```
{
    "contentIdStem"       : <beginning-of-uri>,
    "leadInContentIdStem" : <beginning-of-uri>,   (optional)
    "timelineSelector"    : <uri>,
```

```
    "timelineProperties"  : <timelineProperties>,
    "mappings"            : [ ... ]
}
```

Required properties:

- **contentIdStem** is a string value. Its value is a CI stem that defines the circumstances in which the Synchronization Timeline is available as defined in clause 5.2.2.
- **timelineSelector** is a string value consisting of a URI that indicates the Synchronization Timeline. Some possible values are specified in clause 5.3.3.
- **timelineProperties** is a JSON object (defined in clause 5.5.9.5) describing the properties of the Synchronization Timeline as defined in clause 5.5.9.5. The timelineProperties JSON object shall include an accuracy property.
- **mappings** is a JSON array containing zero or more Timeline Mappings. Each is represented by a JSON object. This JSON object is defined in clause 5.5.9.6.

Optional properties:

- **leadInContentIdStem** is a string value. Its value is a CI stem that defines the lead-in circumstances just prior to the availability of the Synchronization Timeline as defined in clause 5.2.2.

5.5.9.5 JSON for Timeline properties

A Timeline properties object defines the scale (rate) of a Timeline and may define its accuracy (as defined in clause 5.3.2). It is represented by a JSON object with properties defined in this clause. If such a JSON object is correctly formed, it can be validated by the schema definition for "timelineProperties" defined in the JSON schema in clause A.1.2. The template below illustrates this JSON object:

```
{
    "unitsPerTick"    : <nonZeroPositiveInteger>,
    "unitsPerSecond"  : <nonZeroPositiveInteger>,
    "accuracy"        : <zeroOrPositiveNumber>   (optional)
}
```

Required properties:

- **unitsPerTick** and **unitsPerSecond** are integer values greater than zero that define the Timeline scale. unitsPerTick divided by unitsPerSecond defines the duration (in seconds) of a single integer tick of the Timeline.

Optional properties:

- **accuracy** is a decimal (non integer) number of seconds quantifying the broadcaster's understanding of how accurately Time Values on this Timeline represent the timing of underlying media content. The value should represent the 95 % confidence interval (2 standard deviations) or better. The minimum value is zero, indicating perfect accuracy.

The accuracy property is optional when the Timeline properties object is part of a CII message (see clause 5.6) but the property shall be defined if the Timeline properties object is part of Material Information (see clause 5.5.9.2 and clause 5.5.9.4).

NOTE: The value accuracy may be non-integer (e.g. 0,001 representing 1 millisecond).

5.5.9.6 JSON for Timeline Mapping

A Timeline Mapping defines a mapping interval (described in clause 5.5.5) to the Timeline of a Material from the Synchronization Timeline for the Synchronization Timeline information in which the Timeline Mapping is aggregated.

A Timeline Mapping is represented by a JSON object as defined in this clause. If such a JSON object is correctly formed, it can be validated by the schema definition for "mapping" defined in the JSON schema in clause A.1.2. The template below illustrates this JSON object:

```
    {
        "materialIndex"        : <string>,
        "start"                : <integerAsString>,
        "end"                  : <integerAsString>
        "correlations"         : [ <correlationTimestamp>, <correlationTimestamp, ... ],
        "correlationsChanging" : <boolean>
}
```

Required properties:

- **materialIndex** is a string value matching the name of one of the properties held in the "materials" JSON object (see clause 7.4). The value of that property is the Material JSON object to which this mapping applies.
- **start** is a string value representation of an integer value in the units of the Synchronization Timeline that is the start of the mapping interval on the Synchronization Timeline. The mapping interval is inclusive of this start value.
- **end** is a string value representation of an integer value in the units of the Synchronization Timeline that is the end of the mapping interval on the Synchronization Timeline. The mapping interval is exclusive of this end value.
- **correlations** is an array of one or more Correlation Timestamp objects as defined in clause 5.5.9.7. The array shall be ordered in ascending order by their "point" property.
- **correlationsChanging** is a boolean value. When the value is true this indicates that the correlations are anticipated to change. When the value is false this indicates that the correlations are not anticipated to change.

5.5.9.7 JSON for Correlation Timestamp

A Correlation Timestamp (as defined in clause 5.5.6) is represented by a JSON object with properties defined in this clause. If such a JSON object is correctly formed, it can be validated by the schema definition for "correlation" defined in the JSON schema in clause A.1.2. The template below illustrates this JSON object:

```
    {
        "materialPoint"  : <integerAsString>,
        "point"          : <integerAsString>
    }
```

Required properties:

- **materialPoint** is a string value representation of an integer value in the units of the Material Timeline that is the correlation point on the Material Timeline that corresponds with a point of correlation on the Synchronization Timeline.
- **point** is a string value representation of an integer value in the units of the Synchronization Timeline that is that point of correlation on the Synchronization Timeline.

5.5.9.8 Trigger Event Info

A Trigger Event Info object represents a set of Trigger Events, as defined in clause 5.5.7, that are applicable to a Material and the context in which to make the subscription. The object is represented by a JSON object with the properties defined in this clause. If such a JSON object is correctly formed, it can be validated by the schema definition for "triggerEventInfo" defined in the JSON schema in clause A.1.3. The template below illustrates this JSON object:

```
    {
        "contentIdStem" : <start-of-uri>,
        "events"  : {
            <event-name-as-string> : <uri>,
            <event-name-as-string> : <uri>,
            ...
            }
    }
```

Required properties:

- **contentIdStem** is a string value. Its value is a CI stem as defined in clause 5.2.2.
- **events** is a JSON object containing zero or more properties. The name of each property is the name of an event and the value of that property is a URI for that event. The event name is a string restricted to be one or more characters in length and starting with a lower case or upper case letter followed by lower case or upper case letters, numbers or underscore characters.

NOTE: How event names are allocated to events is outside the scope of the present document. The mapping between this event name and any event name signalling that might be present in the broadcast signalling is outside the scope of the present document.

5.6 Content Identification and other Information (CII)

5.6.1 General

Content Identification and other Information (CII) is reported by the TV Device to the CSA via the CSS-CII interface as part of the procedure defined in clause 4.3.2. CII provides the CSA with a Content Identifier for the Timed Content that is currently being presented by the TV Device and other information needed to enable the CSA to commence Wall Clock, Timeline Synchronization and Trigger Event procedures.

CII consists of the following pieces of information:

- The protocol version implemented by the TV Device.
- The URL of the MRS service endpoint known by the TV Device to relate to the Timed Content currently being presented. How this URL is determined is defined in clause 5.6.2.
- The Content Identifier corresponding to the Timed Content currently being presented by the TV Device as defined in clause 5.6.3.
- The status of the Content Identifier as defined in clause 5.6.3.
- The status of presentation of Timed Content by the TV Device as defined in clause 5.6.4.
- The URL of the Wall Clock Service endpoint as defined in clause 5.6.5.
- The URL of the Timeline Synchronization Service endpoint as defined in clause 5.6.5.
- The URL of the Trigger Event Service endpoint as defined in clause 5.6.5.
- An unordered list of zero or more timeline options (each consisting of a Timeline Selector and properties of the corresponding Timeline) as defined in clause 5.6.6.
- Private data as defined in clause 5.9.

The protocol version reported in the CII designates the specification that describes the encoding of CII and the specification with which the service endpoints are compatible.

5.6.2 Reporting the MRS URL

To determine the URL of the MRS for a DVB broadcast service, the TV Device shall look for a uri_linkage_descriptor with uri_linkage_type 0x02 that signals an MRS URL (as defined in clause 6.4.14 of ETSI EN 300 468 [13]). The TV Device shall look for this in the descriptor loops listed in table 5.6.2.1 that correspond to the DVB service currently being presented by the TV Device. The BAT descriptor loops shall be looked in if, and only if, the TV Device chose to install that DVB service because it is part of that bouquet. If and when a uri_linkage_descriptor signals an MRS URL, then it shall be interpreted under the scoping rules shown in table 5.6.2.1 such that the uri_linkage_descriptor that is not overridden by any another shall be the one from which the MRS URL is obtained. A receiver may also obtain an MRS URL by other means (such as private data) that are outside the scope of the present document, otherwise there is no MRS URL value available.

Table 5.6.2.1: Scoping rule for uri_linkage_descriptor when uri_linkage_type=0x02

Location of uri_linkage_descriptor	Scope of definition	Scope this definition overrides
First descriptor loop of NIT	Network	None
First descriptor loop of BAT	Bouquet	Network (see note)
Transport stream descriptor loop of NIT	Transport stream	Bouquet or network
Transport stream descriptor loop of BAT	Transport stream	Bouquet or network (see note)
Service descriptor loop of SDT	Service	Transport stream, bouquet or network
Event descriptor loop of EIT present event	Event	Service, Transport stream, bouquet or network
NOTE: A uri_linkage_descriptor in the BAT descriptor loops is ignored if the TV Device did not install the service because it is part of the bouquet.		

To determine the URL of the MRS for a DVB IPTV service [5], the TV Device shall look first for the element URILinkage as defined in clause 5.2.12 of [5], i.e. the element carried in the service description and selection information. Should it fail to find such an element it shall then look for a uri_linkage_descriptor with uri_linkage_type 0x02 that signals an MRS URL (as defined in clause 6.4.14 of ETSI EN 300 468 [13]). The TV Device shall look for this in the descriptor loops listed in table 5.6.2.1 that correspond to the DVB IPTV service currently being presented by the TV Device. The BAT descriptor loops shall be looked in if, and only if, the TV Device chose to install that DVB IPTV service because it is part of that bouquet. If and when the uri_linkage_descriptor signals an MRS URL, then it shall be interpreted under the scoping rules shown in table 5.6.2.1 such that the uri_linkage_descriptor that is not overridden by any another shall be the one from which the MRS URL is obtained. A receiver may also obtain an MRS URL by other means (such as private data) that are outside the scope of the present document, otherwise there is no MRS URL value available.

NOTE: The reference to [5] above is intended to be valid for a future edition of ETSI TS 102 034 [5]. Annex F details the relevant parts of the future editions and may be used until the future edition is available.

To determine the URL of the MRS for a DVB DASH service [16], the TV Device shall look for a mrsUrl element in the MPD (as defined in annex E) that describes the content currently being presented by the TV Device. If this element is not present, and an MRS URL is not available through other means (such as private data) that are outside the scope of the present document, then there is no MRS URL value available.

5.6.3 Reporting the Content Identifier

The format of the Content Identifier shall be defined according to clause 5.2 according to the type of Timed Content being presented by the TV Device. If the TV Device does not know the Content Identifier, or if the TV Device is not presenting Timed Content (e.g. it is displaying a menu screen instead) then there is no Content Identifier value available.

The status of the content identifier shall indicate whether the content identifier is "partial" or "final " according to the process defined in clause 5.2.

5.6.4 Status of presentation

The status of presentation of Timed Content is provided by the TV Device to inform the CSA of the status of presentation so that the CSA can adjust its behaviour and presentation if the CSA considers it appropriate to do so.

The status of presentation shall be represented by a primary aspect of status followed by zero or more extended aspects. Table 5.6.4.1 summarizes the syntax in a formal specification expressed in Augmented BNF as defined in IETF RFC 5234 [4] and using the Core Rules defined in Appendix B.1 of IETF RFC 5234 [4]. Any permitted status of presentation shall validate against the rule named status-of-presentation in table 5.6.4.1.

Table 5.6.4.1: Syntax of the status of presentation in CII

status-of-presentation	=	primary-aspect extended-aspects
primary-aspect	=	%x6f.6b.61.79 ; "okay"
primary-aspect	=/	%x74.72.61.6e.73.69.74.69.6f.6e.69.6e.67 ; "transitioning"
primary-aspect	=/	%x66.61.75.6c.74 ; "fault"
primary-aspect	=/	1*(%x21-7E); (see note)
extended-aspects	=	*(" " extended-aspect)
extended-aspect	=	1*(%x21-7E)
NOTE: A primary aspect of status is always included and is at least 1 character long.		

If the primary aspect of status is one of the following lower-case string values then it shall convey the corresponding described meaning:

- **okay:** The TV Device is presenting Timed Content to the user. Some portion of the Timed Content (such as video and/or audio) is being presented at any play speed, including paused, faster than normal, slower than normal or reverse.
- **transitioning:** The TV Device is in the process of starting or changing what Timed Content is being presented to the user but has not yet begun presenting the Timed Content it is changing to.
- **fault:** The TV Device is not currently presenting Timed Content to the user due to a problem either in receiving the Timed Content or in trying to present it.

The status of presentation reported by the TV Device should reflect the user's perception of whether and how the TV Device is presenting Timed Content. The value of the primary aspect of status should be one of the values defined above where the corresponding meaning is true given what the user perceives. The TV Device should not use a different value of the primary aspect in such situations. In all other situations the value of the primary aspect is outside the scope of the present document.

Behaviour of a CSA in the presence of other primary aspect values is outside of the scope of the present document.

NOTE 1: For other primary aspect values, the CSA can infer that Timed Content is not being presented (with or without faults) or being transitioned to, otherwise the primary aspect would be one of the values defined above. Therefore if a TV Device defines and uses a new primary aspect value then some or all CSAs could cease presenting Timed Content. Clause C.9 suggests an appropriate primary aspect of status of presentation to be reported by the TV Device for a range of situations.

No extended aspects are defined in the present document. A CSA may ignore any extended aspects in the status of presentation.

NOTE 2: Extended aspects enable a TV Device to signal variations on the broad theme that the primary aspect represents while still enabling any CSA that only recognizes the primary aspects defined in the present document to make a best effort to behave appropriately in all situations.

When reporting any status, the TV Device may still provide Control Timestamps via the Timeline Synchronization protocol and notifications of Trigger Events via the Trigger Event Notification protocol. If the primary aspect of status is any value other than "okay" then the TV Device may endeavour to provide best-effort estimates of Control Timestamps and Trigger Event Notifications, however the CSA is expected to understand that these Timestamps and notifications may be unreliable.

5.6.5 Reporting Wall Clock, Timeline Synchronization and Trigger Event Notification service endpoint URLs

The TV Device shall provide both the URLs of the Wall Clock and Timeline Synchronization Service endpoints if the TV Device is currently providing Timeline Synchronization functionality. Where no Timeline Synchronization service endpoint is available there may optionally be a Wall Clock Service endpoint value available.

NOTE: It is recommended that, if possible, a TV Device provides a functioning Wall Clock Synchronization service endpoint both when the Timeline Synchronization Service endpoint is and is not available and that this service endpoint and the underlying Wall Clock is kept the same at all times. Clause C.8.2 describes why this is desirable.

The TV Device shall provide the URL of the Trigger Event Notification service endpoint if Trigger Event Notification functionality is currently being provided by the TV Device. If this functionality is not currently available then there is no Trigger Event Notification service endpoint value available.

5.6.6 Reporting a list of timeline options

The unordered list of Timeline Selectors and corresponding properties of Timelines shall contain zero, one or more options for Timelines that can be used as the Synchronization Timeline. Each option shall be expressed as a Timeline Selector with corresponding Timeline properties. The options represent suggestions by the TV Device of Timelines that can be used as the Synchronization Timeline for the procedures described in clause 4.3.5. What options are listed is therefore outside the scope of the present document. If a Timeline is listed, this does not guarantee that the Timeline will be available if it is selected by the CSA. If a Timeline is not listed this does not indicate that a particular Timeline will be unavailable if the CSA attempts to select it.

NOTE: This is a mechanism to allow a TV Device to suggest to a CSA the Timelines that may be selected and their properties. This can be useful if, for example, a CSA does not have access to an MRS.

5.6.7 JSON Representation of Content Identification and other Information

Content Identification and other Information is represented by a JSON object with the properties defined in this clause. If such a JSON object is correctly formed, it can be validated by the root object of the JSON schema in clause A.1.4. The template below illustrates this JSON object:

```
{
    "protocolVersion"    : "1.1",                 (optional)
    "mrsUrl"             : <url>,                 (optional)
    "contentId"          : <contentId>,           (optional)
    "contentIdStatus     : <"partial" or "final">, (optional)
    "presentationStatus" : <string>,              (optional)
    "wcUrl"              : <url>,                 (optional)
    "tsUrl"              : <url>,                 (optional)
    "teUrl"              : <url>,                 (optional)
    "timelines"          : [ <timelineOption>,
                             <timelineOption>,
                             ...
                           ]                      (optional)
}
```

Required properties:

- There are no required properties.

Optional properties:

- **protocolVersion** is a string value that conveys the protocol version implemented by the TV Device. When complying with the present document this shall have the value "1.1".
- **mrsUrl** is a string value consisting of the URL of the MRS endpoint known by the TV Device to relate to the Timed Content currently being presented, as defined in clause 5.6.2. The mrsUrl shall not end in "/".
- **contentId** is a string value consisting of the Content Identifier for the Timed Content currently being presented by the TV Device, as specified in clause 5.6.3.
- **contentIdStatus** is a string with value "partial" or "final" that provides the status of the current contentId as specified in clause 5.6.3. This property shall be present in the JSON object if contentId property is present.
- **presentationStatus** is a string value describing the current status of presentation of Timed Content by the TV Device as defined in clause 5.6.4.
- **wcUrl** is a string value consisting of the URL of the Wall Clock Service endpoint as defined in clause 5.6.5.
- **tsUrl** is a string value consisting of the URL of the Timeline Synchronization Service endpoint as defined in clause 5.6.5.
- **teUrl** is a string value consisting of the URL of the Trigger Event Service endpoint as defined in clause 5.6.5.

- **timelines** is an array consisting of zero or more timeline options as defined in clause 5.6.6. Each option is represented by a JSON object that is described below.

Where a property is not present, this means that the last reported value still applies. Where the value of a property is not available, the value null shall be sent, with the following exceptions:

- The protocolVersion property shall be present in at least the first CII communicated where it shall be set to the specified value; it shall never have the value null. Within a protocol transport connection the value of protocolVersion property shall not change but may optionally be repeated with the same value or the property may be omitted from all but the initial CII JSON object.

- The presentationStatus property shall be present and set to the specified value in at least the first CII communicated. The presentationStatus property shall never have the value null.

A timeline option is represented by a JSON object as described by schema definition for "timelineOption" defined in the JSON schema in clause A.1.4. The template below illustrates this JSON object:

```
{
    "timelineSelector"   : <uri>,
    "timelineProperties" : <timelineProperties>
}
```

Required properties:

- **timelineSelector** is a string value consisting of a URI that indicates the Synchronization Timeline. Some possible values are specified in clause 5.3.3.

- **timelineProperties** is a JSON object describing the properties of the Synchronization Timeline as defined in clause 5.5.9.5.

5.7 Timestamps and Timeline Synchronization

5.7.1 General

Timestamps are exchanged between the MSAS function of the TV Device and the SC function of the CSA to achieve Timeline Synchronization. This exchange takes place via the CSS-TS interface using the procedure defined in clause 4.3.5 using the protocol described in clause 9 and following the architectural concepts described in clauses 4.2.2 and 4.2.3.

The setup message sent by the SC to the MSAS at the beginning of the procedure is detailed in clause 5.7.3.

All Timestamps contain a pair of Time Values representing a point of correlation between the Synchronization Timeline and the Wall Clock. They are expressed in relation to a reference point detailed in clause 5.7.2. Actual, Earliest and Latest Presentation Timestamps are sent from SCs to the MSAS and are detailed in clause 5.7.4. A Control Timestamp is sent from MSAS to SC and contains additional information and is detailed in clause 5.7.4.

A Timestamp is represented by a JSON object with the properties defined in this clause 5.7. If such a JSON object is correctly formed, it can be validated by the schema definition appropriate to the type of timestamp (see clause 5.7.4 and clause 5.7.5). The template below illustrates the aspects of this JSON object that are common to all types of Timestamp:

```
{
    "contentTime"   : <integerAsString>,
    "wallClockTime" : <integerAsString or "plusinfinity" or "minusinfinity">
}
```

A Control Timestamp extends this object with additional properties that are described below in clause 5.7.5.

Required properties:

- **contentTime** is a string value representation of an integer value in the units of the selected Synchronization Timeline, indicating a specific point in the Timed Content.

- **wallClockTime** is a string value representation of the Wall Clock time value as an integer number of nanoseconds. This value is the Wall Clock time corresponding to that specific point in the content, using the reference point for timestamping specified in clause 5.7.2. It can also have the value "*minusinfinity*" or "*plusinfinity*", representing respectively the beginning (-∞) and end (+∞) of time.

The value "minusinfinity" shall only be used for Earliest Presentation Timestamps. It signals that the Timed Content is available in full and can be presented at any time. The value "plusinfinity" shall only be used for the Latest Presentation Timestamp to signal that the Timed Content can be buffered or delayed indefinitely. In those cases, the value of contentTime is not defined.

5.7.2 Reference point for timestamping

This clause provides a reference model for the timestamping of Timed Content. Timestamping is the process in which a media sample (e.g. video frame or audio sample), identified by a Time Value on the Synchronization Timeline (e.g. PTS or MPEG-TEMI), is associated with a Wall Clock time.

Having a well-defined reference point is important, as ambiguity about the location of the reference points would lead to synchronization errors and in the worst case even to situations, where some devices introduce run-away delays to remain in sync with others.

Figure 5.7.2.1 illustrates the reference point for timestamping.

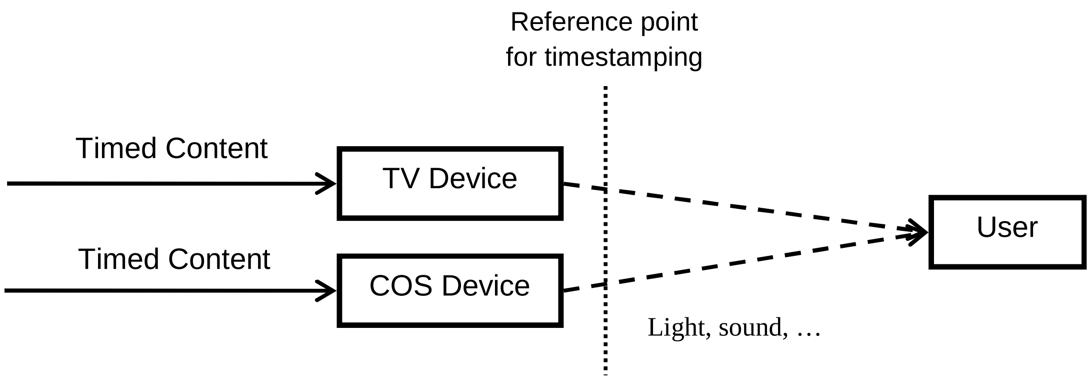


Figure 5.7.2.1: Reference point for timestamping

The reference point for timestamping is at the output of the TV Device and COS device in the physical domain such as light or sound. This means that the SC in the TV Device and COS Device shall perform calculations such that Timestamps (see clauses 5.7.4 and 5.7.5) relate to that reference point. In practice, the SC may only have access to a reference point within the device decoder so will need to compensate for delays due to output buffers, frame buffers, quality enhancement technologies, etc. Clause C.7 provides examples of such calculations. If no accurate values are known (or knowable), then the SC shall make a best effort estimation of the extra delays and compensate accordingly.

NOTE: HDMI 2.0 [i.9] provides functionality for dynamic synchronization of video and audio streams. Information from the HDMI can be used to make a best-effort estimate of the extra travel time between a set-top box and the light and sound output of TV screen.

5.7.3 Setup data

Setup data specifies the timeline that a CSA wishes to be used for Timestamps communicated via the CSS-TS interface as part of the Timeline Synchronization procedure defined in clause 4.3.5 using the protocol described in clause 9. It consists of the following information:

- A Content Identifier stem.
- A Timeline Selector.

The above information can be obtained from Synchronization Timeline Information (see clause 5.5.4) provided by an MRS via the CSS-MRS interface using the procedures described in clause 4.3.3. It can also be obtained by some other means outside the scope of the present document.

The Timeline Selector describes the type and location of timeline signalling to be derived from the Timed Content currently being presented by the TV Device (see clause 5.3.3) but only while the CI provided matches the CI stem (according to the process described in clause 5.2.2). This Timeline is used as the Synchronization Timeline for Timestamps exchanged during the remainder of the procedure. The derivability of the Timeline and matching of the CI determines the availability of the timeline, and is described in more detail in clause 9.2.

The Content Identifier stem is considered to match the timed content currently being presented by the TV Device if it matches according to the rules defined in clause 5.2.2.

Setup data is represented by a JSON object with properties defined in this clause. If such a JSON object is correctly formed, it can be validated by the schema defined by the root object of the JSON schema in clause A.1.5. The template below illustrates this JSON object:

```
{
    "contentIdStem"                : <beginning-of-uri>,
    "timelineSelector"             : <uri>
}
```

Required properties:

- **contentIdStem** is a string value consisting of a CI stem.
- **timelineSelector** is a string value consisting of a URI that indicates which Synchronization Timeline is to be used for Timestamps. Some possible values are specified in clause 5.3.3.

5.7.4 Actual, Earliest and Latest Presentation Timestamp

An SC provides Actual, Earliest and Latest Presentation Timestamps to an MSAS via the CSS-TS interface as part of the Timeline Synchronization procedure described in clause 4.3.5 via the protocol described in clause 9. These Timestamps convey the earliest and latest times at which the SC is capable of presenting a given point on the Synchronization Timeline and the actual time at which the SC is currently presenting a given point on the Synchronization Timeline.

An Actual, Earliest or Latest Presentation Timestamp consists of a time value in terms of the Synchronization Timeline (a content time) and a time value in terms of the Wall Clock (a wall clock time).

The meaning of an Actual Presentation Timestamp where the content time is CT_0 and Wall Clock time is WC_0 is equivalent to the SC saying to the MSAS that "I am currently presenting the media sample identified by time value CT_0 when the Wall Clock has the time value WC_0".

The meaning of an Earliest Presentation Timestamp where content time is CT_1 and Wall Clock time is WC_1 is equivalent to the SC saying to the MSAS that "I can present the media sample identified by time value CT_1 at the very earliest when the Wall Clock has the time value WC_1".

The meaning of a Latest Presentation Timestamp where content time is CT_2 and Wall Clock time is WC_2 is equivalent to the SC saying to the MSAS that "I can present the media sample identified by the time value CT_2 at the very latest when the Wall Clock has the time value WC_2".

Actual, Earliest and Latest Timestamps are represented by a JSON objects with properties defined in this clause. If such a JSON object is correctly formed, it can be validated by the schema defined by the root object of the JSON schema in clause A.1.7. The template below illustrates this JSON object:

```
{
    "actual"    : <timestamp>,      (optional)
    "earliest"  : <timestamp>,
    "latest"    : <timestamp>
}
```

Required properties:

- **earliest** is a JSON object representing the Earliest Presentation Timestamp. The representation is defined in clause 5.7.1, except that the property wallClockTime shall not take the value "plusinfinity".

- **latest** is a JSON object representing the Latest Presentation Timestamp. The representation is defined in clause 5.7.1, except that the property wallClockTime shall not take the value "minusinfinity".

Optional properties:

- **actual** is a JSON object representing the Actual Presentation Timestamp. The representation is defined in clause 5.7.1, except that the property wallClockTime shall not take the values "plusinfinity" or "minusinfinity".

5.7.5 Control Timestamps

An MSAS provides Control Timestamps to an SC via the CSS-TS interface as part of the Timeline Synchronization procedure described in clause 4.3.5 via the protocol described in clause 9. These Timestamps convey the times at which the SC is recommend to present a given point on the Synchronization Timeline.

A Control Timestamp consists of a Time Value in terms of the Synchronization Timeline (a content time), a Time Value in terms of the Wall Clock (a wall clock time) and a play speed.

The meaning of a Control Timestamp where contentTime is CT_3 and wallClockTime is WC_3 is equivalent to the MSAS saying to the SC "please present the media sample identified by contentTime CT_3 at wallClockTime WC_3 to the user".

A Control Timestamp also includes a timeline speed multiplier. This is a hint to the SC that the Control Timestamps may no longer represent a rate of progression consistent with normal playback speed, but may be running at a multiple represented by this value, which can be negative. This value shall be "1" when the Synchronization Timeline advance is coincident with normal playback speed. When paused, this value shall be "0" and the Control Timestamp shall feature a time value on the Synchronization Timeline corresponding to the point at which the TV Device has paused its media presentation. How the timeline speed multiplier is determined by the MSAS is out of scope of the present document.

If the Synchronization Timeline is currently unavailable then the contentTime and timeline speed multiplier shall both be unknown and wallClockTime shall represent the moment in time at which the determination was made that the Synchronization Timeline was no longer available.

NOTE 1: The situations in which a Synchronization Timeline is available or unavailable are explained in clause 9.2.

A Control Timestamp is represented by a JSON object with properties defined in this clause. If such a JSON object is correctly formed, it can be validated by the schema defined by the root object of the JSON schema in clause A.1.6. This extends the representation of a Timestamp object as described in clause 5.7.1 by defining an additional property. The template below illustrates this JSON object:

```
{
    "contentTime"             : <integerAsString or null>,
    "wallClockTime"           : <integerAsString>,
    "timelineSpeedMultiplier": <number or null>
}
```

NOTE 2: The circumstances in which contentTime and timelineSpeedMultiplier properties have null values are identical. There will therefore never be a situation when only one of these two properties is a null value and the other is not.

Required properties:

- **contentTime** is a string value representing the content time part of a Control Timestamp and is in the format defined in clause 5.7.1. However if the content time is unknown (because the timeline is unavailable) then contentTime shall be a null value instead.

- **wallClockTime** is a string value representing the Wall Clock time value part of a Control Timestamp and is in the format defined in clause 5.7.1.

- **timelineSpeedMultiplier** is a string value representing a number providing a hint of current rate of progression of Timed Content presentation. However if the timeline speed multiplier is unknown (because the timeline is unavailable) then timelineSpeedMultiplier shall be a null value instead.

5.8 Trigger Events

5.8.1 General

JSON objects (defined in clause 5.8.5) are exchanged between the CSA and the TV Device via the CSS-TS interface using the procedures described in clause 4.3.6 via a protocol defined in clause 10. These objects allow the CSA to manage subscriptions for Trigger Events and the TV Device to notify the Companion Screen Application of the status of subscriptions and event notifications when the event signalling is detected.

The technologies for signalling Trigger Events are defined in clause 5.8.4.

Trigger Event locations are described using a URI. The URI formats are summarized in clause 5.8.2.

5.8.2 Encoding Trigger Event locations

Trigger Event locations are encoded using a URI. The format depends on the broadcast encoding used for Trigger Events. The formats defined by the present document are summarized in table 5.8.2.1.

Table 5.8.2.1: Trigger Event location URI format

Trigger Event source	Clause	Event location URI
DSM-CC "do it now" stream events	5.8.4.2	urn:dvb:css:triggerevent:dsmcc:<component_tag>:<eventid>
DASH	5.8.4.3	<schemeIdUri>:< base64 encoded value>

NOTE: Other URI formats may be present to support either future or proprietary schemes.

5.8.3 Reference point for Trigger Events

Trigger Event notifications carry a presentationWallClockTime property. This represents presentation time for the event in terms of wall clock time. This allows the CSA to compute when it should act on the event.

The time of presentation is in terms of the same reference point as is used for timestamping (see clause 5.7.2) but is calculated assuming a 1x playback speed. However playback speed may not be x1 or, between the time the Trigger Event notification is received and the time the CSA should act on the device, the playback speed may have changed. To enable the CSA to correctly compensate for changes to playback speed, the notification includes the wall clock time at which the presentationWallClock Time was calculated. By monitoring changes in playback speed through the control timestamps described in clause 5.7.5, the CSA is able to make the necessary adjustments to the presentation time.

NOTE: It is recommended that the calculation be performed and the notification be sent only a short period of time before the Trigger Event is due to be presented so that the CSA can achieve a reasonable (less than 1 second) accuracy of presentation of the Trigger Event without having to use the CSS-TS interface to obtain Control Timestamps.

The calculations to obtain the value for the property depend on the encoding standard used for the Trigger Event signalling.

5.8.4 Trigger Event Sources

5.8.4.1 General

This clause defines how Trigger Events are signalled in the content. References to the Trigger Events in messages between the CSA and TV Device are encoded as URIs. The URIs for the Trigger Event signalling mechanisms defined in the present document are defined in clause 5.8.4.

5.8.4.2 DSM-CC "do it now" Stream Events

5.8.4.2.1 Encoding in MPEG TS

The present document references the definitions of DSM-CC "do it now" stream events in MPEG Transport Streams in the following specifications:

- ETSI ES 202 184 [21].
- ETSI TS 102 727 [22].
- ETSI TS 102 809 [23].

These have compatible definitions of how the Stream Descriptor aspect of DSM-CC "do it now" events are encoded (but may have differences in other aspects of the DSM-CC Object Carousel and other DSM-CC Stream Event types).

The present document only requires that the content include DSM-CC sections carrying a DSM-CC Descriptor List carrying a Stream Event Descriptor to implement DSM-CC "do it now" events in the content. The content may also carry DSM-CC Object Carousel data including Stream Event objects referencing the stream events. However, this is optional as the present document references the Stream Descriptor directly, i.e. the Stream Descriptor is not located using a look-up of the stream event name in the Object Carousel.

NOTE 1: In [21], [22] and [23] the eventId is also recorded in the table id extension of the section carrying the Stream Event Descriptor.

NOTE 2: [21], [22] and [23] allow multiple transmissions of stream events (as detection of a single is message may be unreliable) but only single firing.

5.8.4.2.2 Encoding the URI referencing the Trigger Event

URI references to DSM-CC "do it now" stream events in messages between the CSA and TV Device are encoded as URNs as follows:

- urn:dvb:css:triggerevent:dsmcc:<component_tag>:<eventId>

urn:dvb:css:triggerevent:dsmcc identifies a Trigger Event according to clause 5.8.4.

component_tag is the base-ten integer representation of the value of the component_tag (declared by a stream identifier descriptor as defined in ETSI EN 300 468 [13]) associated with stream of TS packets that carry the DSM-CC Stream Descriptors.

eventId is the base-ten integer representation of the value of the eventId of a Stream Event Descriptor on the component identified by the component tag.

5.8.4.2.3 Trigger Event data

The DSM-CC Stream Event Descriptor can optionally carry event data in its privateDataByte field. If event data is present in the Stream Event Descriptor, it shall be base64 encoded and conveyed in the triggerEventData property of the Trigger Event Notification message. If there is no data in the privateDataByte field then the triggerEventData property shall be null.

5.8.4.2.4 Reference point

The presentationWallClockTime property of the event notification shall be the Wall Clock time at the reference point (see clause 5.7.2) corresponding to when the surrounding content reaches the reference point. The surrounding content is defined as being the audio and/or video content which surrounds the Transport Stream packet carrying the Stream Event.

NOTE: It may not be possible for TV Devices to accurately determine the timing of the surrounding content due to the lack of explicit timing information in the Stream Event Transport Stream packet.

The event notification message shall be sent on detection of the event signalling in the content and so the presentationWallClockTime will normally be in the near future when the notification is received by the CSA.

5.8.4.3 DASH Events

5.8.4.3.1 Encoding in the DASH Content

DASH defines two methods for carrying events, as described in clause 9 of ETSI TS 103 285 [16], in-band events as described in clause 5.10.3 of DASH [17] that use the 'emsg' box and MPD events as described in clause 5.10.2 of DASH [17]. Both of these mechanisms use a URI to scope and identify the event stream, and both require that the URI is signalled in advance in the MPD.

DASH events include identifiers that allow for the filtering of repeated events. The TV device shall filter events such that a given event, as identified by its identifier, is only transmitted once.

5.8.4.3.2 Encoding the URI referencing the event

In both cases of DASH events, a URI is used to indicate the event stream. This URI shall be comprised of the @schemeIdUri attribute value followed by a colon (":") followed by the base64 encoded @value attribute value of the EventStream or InBandEventStream element within the MPD that correspond to the desired events. Base 64 encoding shall be performed using the "Base 64 Encoding with URL and Filename Safe Alphabet" as specified in IETF RFC 4648 [24].

NOTE: It is possible for some event streams to be Representation specific, and so event streams may disappear through the nature of a DASH stream. Such cases are handled by the use of the TEN message as described in clause 5.8.5.4 that indicates a change to the subscription status of a subscribed event stream.

5.8.4.3.3 Event Data

When an event is received, a Trigger Event Notification message shall be generated, with properties as defined below.

Where MPD events are used, then:

- the @id attribute of the Event element, if present, shall be converted to a string and carried in the triggerEventId property. If this attribute is not present, then the triggerEventId property shall not be present;
- the content of the Event element (excluding the start and end tags and any attributes) shall be base64 encoded and conveyed in the triggerEventData property. If the Event element is empty, or only contains whitespace characters, the triggerEventData property shall not be present.

Where the 'emsg' box is used to carry inband event, then:

- if the message_data field is present (i.e. it has a length greater than 0), the data carried in it shall be base64 encoded and communicated in the triggerEventData property of the Trigger Event Notification Message. If there is a zero length message_data field then the triggerEventData property shall not be present;
- the id field of the 'emsg' box shall be converted to a string and carried in the triggerEventId property.

5.8.4.3.4 Reference Point

DASH events include explicit presentation and duration time information. The presentationWallClockTime property of the trigger event notification message shall be computed as:

- For events conveyed by the 'emsg' box, the Wall Clock time corresponding to the presentation time derived, assuming a timelineSpeedMultiplier of 1, from the signalling in the 'emsg' box.
- For events conveyed by the MPD, the Wall Clock time corresponding to the presentation time derived, assuming a timelineSpeedMultiplier of 1, from the @presentationTime attribute of the Event element.

For events whose presentationWallClockTime has already past, the value shall be calculated assuming a timelineSpeedMultiplier of 1 occurred for the elapsed duration of the event.

As DASH events also include a duration, the trigger event notification message property triggerEventDuration shall use the same units as used for the presentationWallClockTime property, assuming a timelineSpeedMultiplier of 1 and be set to:

- For events conveyed by the 'emsg' box the duration signalled in the 'emsg' box.
- For events conveyed by the MPD, the duration signalled in the @duration attribute of the Event element, converted to the same units as used for the presentationWallClockTime property.

The trigger event notification message shall be sent when the event data is received. I.e. the trigger event notification message shall not be delayed until the calculated presentation time.

NOTE: Where a CSA wishes to retain synchronization with the Timed Content in the event of different timelineSpeedMultiplier values the CSA may wish to use the CSS-TS service. This will allow the CSA to receive Control Timestamps that allow it to adjust the received duration and presentationWallClockTime values to compensate for changes in the timelineSpeedMultiplier value.

5.8.4.4 Other Event Sources

It is possible that other event sources may be present, and that some of these sources may be proprietary. These sources will be identified by URIs, and these may be communicated from the MRS, or via other mechanisms that are outside the scope of the present document. The URIs that define these other event sources may be passed to the TV Device via the protocols outlined in the present document. The mapping and precise timing and interpretation of the relevant messages will be defined by these proprietary schemes. As there is no expectation of support for any such scheme, a TV Device may refuse an attempt to subscribe to such events.

5.8.5 Messages

5.8.5.1 General

This clause defines messages for Trigger Events.

5.8.5.2 Trigger Event Session Setup (TESS) message

After the protocol transport connection to the CSS-TE endpoint is successfully opened from the CSA to the TV Device (see clause 10.2), the first message sent from the CSA to the TV Device shall be the Trigger Event Session Setup. This message establishes the context of this connection as being the presentation of Timed Content with a Content Identifier that the CI stem provided according to the process defined in clause 5.2.2. This message shall be the first message sent on a new connection and shall only be sent once during the life of the connection.

If the CI stem is valid for multiple media presentations, then successful Trigger Event subscriptions made during one media presentation shall continue to be valid if the media presentation changes directly to another media presentation that is compatible with same CI stem. For example, changing channel between services with a compatible CI stem would not invalidate the Trigger Event subscriptions.

If the media presentation changes to one that is not compatible with the session CI stem, then all Trigger Event subscriptions made by the CSA so far on that connection shall cease to be valid. This shall cause the TV Device to cancel these Trigger Event subscriptions.

A Trigger Event Session Setup message is represented by a JSON object with properties defined in this clause. If such a JSON object is correctly formed, it can be validated by the schema defined by the root object of the JSON schema in clause A.1.8. The template below illustrates this JSON object:

```
{
    "contentIdStem" : <beginning-of-uri>
}
```

Required properties:

- **contentIdStem** is a string value consisting of the beginning or whole of a CI URI.

5.8.5.3 Trigger Event Subscription Management (TESM) messages

A single message is defined to encode the actions required by the CSA to manage Trigger Event subscriptions:

- requesting a new subscription;
- removing a previous subscription.

The message carries:

- the identifier for the Trigger Event;
- the action to be performed on that event subscription.

The TESM message is represented by a JSON object with properties defined in this clause. If such a JSON object is correctly formed, it can be validated by the schema defined by the root object of the JSON schema in clause A.1.9. The template below illustrates this JSON object:

```
{
    "triggerEvent" : <uri>,
    "subscribed"   : <boolean>
}
```

Required properties:

- **triggerEvent** is a string carrying a URI identifying a Trigger Event. See clause 5.8.2.
- **subscribed** is a Boolean. "true" indicates a request to subscribe to an event, "false" a request to release a previously requested subscription.

5.8.5.4 Trigger Event Notification (TEN) message

A single message is defined to encode the notification of a Trigger Event by the TV Device to the CSA or changes in availability of the Trigger Event listener in the TV Device.

The message carries:

- the identifier for the Trigger Event
- the Trigger Event data (if present, null otherwise) from the event signalling
- the wall clock time at which the Trigger Event is associated
- the wall clock time at which the calculation of the time the Trigger Event is associated took place
- the subscription status of the Trigger Event listener

It is represented by a JSON object with the properties defined in this clause. If such a JSON object is correctly formed, it can be validated by the schema defined by the root object of the JSON schema in clause A.1.10. The template below illustrates this JSON object:

```
{
    "triggerEvent"                 : <uri>,
    "triggerEventData"             : <base64-as-string or null>,
    "presentationWallClockTime"    : <integerAsString or null>,
    "calculationWallClockTime"     : <wallclock-as-string or null>,
    "subscribed"                   : <boolean>,
    "triggerEventId"               : <string>,                   (optional)
    "triggerEventDuration"         : <integerMinimumZeroAsString>  (optional)
}
```

Required properties:

- **triggerEvent** is a string carrying a URI identifying the Trigger Event. See clause 5.8.2.
- **triggerEventData** is a string carrying Trigger Event data from the Trigger Event signalling. If there is no Trigger Event data this property will have the value null. If the string is not null it shall be encoded Base 64 using the "Base 64 Encoding with URL and Filename Safe Alphabet" as specified in IETF RFC 4648 [24].

- **presentationWallClockTime** is a string value representing the Wall Clock time with which the event is associated. See clauses 5.8.3 and 5.8.4. This property shall only be null if the notification is due to a change in subscription status of the Trigger Event listener.

- **calculationWallClockTime** is a string value representing the Wall Clock time at which the calculation of the **presentationWallClockTime** property took place and that was used as input for the calculation of the **presentationWallClockTime**. Further informative discussion of the use of this property is given in clause C.10. This property shall only be null if the notification is due to a change in subscription status of the Trigger Event listener.

NOTE: This property is needed to allow the CSA to compensate correctly when the playback speed is not x1 or for any changes in play speed between the time the TEN message is sent and the event is due to be presented.

- **subscribed** is a Boolean value. "true" indicates that the Trigger Event listener is active, "false" indicates that the Trigger Event listener cannot be made active or has ceased to be active and is now removed.

Optional properties:

- **triggerEventID** is a string that carries the id associated with an event, if such is provided. Currently only DASH events, as described in clause 5.8.4.3 make use of this optional field.

- **triggerEventDuration** is a number represented as a string value that conveys the duration of the event in the same units as the presentationWallClockTime field. Currently only DASH events, as described in clause 5.8.4.3 make use of this optional field.

A TEN message shall be sent by the TV Device to the CSA in the following four circumstances:

- Notifying that an event listener has successfully been made active, following a TESM message. In this case the required properties are as follows:

  - triggerEvent is a string carrying a URI identifying the Trigger Event;
  - subscribed is "true";
  - presentationWallClockTime is "null";
  - calculationWallClockTime is "null";
  - triggerEventData is "null".

- Notifying that an event listener has successfully been removed, following a TESM message. In this case the required properties are as follows:

  - triggerEvent is a string carrying a URI identifying the Trigger Event;
  - subscribed is "false";
  - presentationWallClockTime is "null";
  - calculationWallClockTime is "null";
  - triggerEventData is "null".

- Notifying that an event listener has been removed, following a change to the content being presented such that the CI stem no longer matches. In this case the required properties are as follows:

  - triggerEvent is a string carrying a URI identifying the Trigger Event;
  - subscribed is "false";
  - presentationWallClockTime is "null";
  - calculationWallClockTime is "null";
  - triggerEventData is "null".

- Notifying that an event has triggered. In this case the required properties are as follows:
    - triggerEvent is a string carrying a URI identifying the Trigger Event;
    - subscribed is "true";
    - presentationWallClockTime represents Wall Clock time and shall not be "null";
    - calculationWallClockTime represents Wall Clock time and shall not be "null";
    - triggerEventData shall carry the Trigger Event data or "null" (depending on what the Trigger Event signalling carries).

A TEN message shall be sent by the TV Device to the CSA one time for each change in status of the Trigger Event listener and once each time event signalling for a subscribed Trigger Event causes the event to fire.

5.9 Private data

5.9.1 General

The objects defined in clause 5 are used to convey information between parts of the ecosystem that support a companion experience. The information contained within them represents only that information needed for the generic ecosystem. In a given use of this system, e.g. to support a specific application, it may be desirable to be able to extend the information carried in these objects in a private fashion. This clause defines a mechanism that can be used safely to do this without risking a clash of property names caused by any future updates to the present document.

NOTE: The mechanism defined here is not intended to be used for updates to the present document.

Certain objects can include private data. This private data follows a common format. This data shall only occur in the objects:

- Material, defined in clause 5.5.9.2.
- Material Identifier, defined in clause 5.5.9.3.
- Synchronization Timeline Information, defined in clause 5.5.9.4.
- Timeline Mapping, defined in clause 5.5.9.6.
- Content Identification and other Information, defined in clause 5.6.
- Setup Data, defined in clause 5.7.3.

Private data has type and value. The type is encoded as a URI. The type shall scope the private data globally. The format of the value is proprietary and limited only by the JSON representation format.

Thus every JSON definition listed above shall include the following definition.

```
"private"  : [ ... ],            (optional)
```

Optional properties:

- **private** is a JSON array containing one or more private data objects as defined in clause 5.9.2.

It is recommended that there are no more than 10 objects in the array.

5.9.2 JSON for an item of private data

A private data object represents private data as defined in clause 5.9.1. It is represented by a JSON object with the properties defined here. If such a JSON object is correctly formed, it can be validated by the schema definition for "private" defined in the JSON schema in clause A.1.2. The template below illustrates this JSON object:

```
{
    "type"    : <uri>,
```

```
        <string>  : <any>,
        ...
    }
```

Required property:

- **type** is a URI string value representing the type of the private data.

Optional property:

- **<string>** represents any valid JSON property name of any format, including JSON objects.

The optional properties are outside of the scope of the present document. A particular private data format may choose to require certain properties and this is not prohibited by the present document as such definitions are outside the scope of the present document.

It is recommended that any single instance of this private data object does not exceed 1 024 bytes.

6 Content Identification and other Information (CSS-CII)

6.1 General

The CSA obtains Content Identification and other Information from the TV Device over the CSS-CII interface that enables the CSA to determine what Timed Content is currently being presented by the TV Device. It also provides any other information required from the TV Device in order to proceed with Timeline Synchronization.

Clauses 6.2 and 6.3 describe the protocol and protocol transport used for delivering Content Identification and other Information from the TV Device to the CSA and the requirements for the implementation of a Content Identification and other Information service endpoint in the TV Device.

An overview of the procedures of the protocol exchanges is described in clause 4.3.2. The semantics, data model and representation of the Content Identification and other Information carried in the protocol are described in clause 5.6.

6.2 Protocol

An overview of the procedures of the protocol (subscribe, notify and unsubscribe) are described in clause 4.3.2.

The TV Device shall allow multiple CSAs to be connected to the CSS-CII service endpoint simultaneously.

When a CSA tries to connect to the CSS-CII service endpoint, the TV Device may decline the request because either the service endpoint is currently unavailable or because it would exceed the number of simultaneous connections to this service endpoint that the TV Device is able to support at the time.

When a CSA connects successfully to the CSS-CII service endpoint, the TV Device shall send a Content Identification and other Information (CII) message. Clause 5.6 defines this message and which properties shall be included in the first of these messages that is sent over the connection. Any optional properties that are not included shall be assumed by the CSA to have the value null.

When the TV Device detects a change that would result in a change to one or more of the properties included in the Content Identification and other Information message, the TV Device shall send to all connected CSAs a message containing updated Content Identification and other Information. This message shall include the properties that have changed and may omit properties that have not.

The TV Device may ignore any messages received over this protocol from the CSA that are not described in this clause.

The CSA may ignore any properties in the Content Identification and other Information message that are not defined in the description of the representation format provided in clause 5.6.

In normal operation, the TV Device should not disconnect the CSA until the CSA disconnects itself. If a TV Device is currently not providing Companion Synchronization functionality, it should continue to provide the CSS-CII service endpoint but instead report some or all of the properties in Content Identification and other Information as null values indicating unavailability of that information.

NOTE: An unavailable property is represented in the JSON representation of Content Identification and other Information by giving the property a value of null.

The TV Device should only disconnect a CSA if it is no longer able to provide Companion Synchronization functionality because, for example, the TV Device is powering down or is out of resources.

6.3 Protocol transport

The TV Device shall implement a CSS-CII service endpoint that implements the server side of the WebSocket protocol version 13 as defined in IETF RFC 6455 [8]. The service endpoints provided by the TV Device in the CII message is the WebSocket URL of the WebSocket server. As defined in section 3 of the WebSocket protocol specification [8], a WebSocket URL defines the host, port, security, and resource-name of the service endpoint.

All messages sent by CSA and TV Device, as defined in clause 5.6, shall be WebSocket data frames (as required by the WebSocket protocol specification [8]) with the payload in text format.

If the CSS-CII service endpoint is currently unavailable then the TV Device shall refuse the connection request by responding with an HTTP response code of 403 "Forbidden".

If the TV Device has reached the limit of the number of simultaneous connections to the CSS-CII service endpoint that it can handle, then the TV Device shall refuse the connection request by returning the HTTP response code 503 "Service Unavailable".

The CSA shall subscribe for Content Identification and other Information by establishing a connection to the CSS-CII service endpoint in the role of a client of the WebSocket protocol.

The TV Device may examine the "Origin" header if it is present in the HTTP Request header from the CSA used in the WebSocket protocol to establish the connection. On the basis of the value of the "Origin" header, the TV Device may choose to refuse to establish the connection and respond with a 403 "Forbidden" response code, as described in section 10.2 of IETF RFC 6455 [8]. The rules used by a TV Device to determine when to allow and when to refuse a connection are outside of the scope of the present document.

The TV Device may return 400 and 500 series HTTP response codes (including code 403 "Forbidden" and 503 "Service Unavailable") for other reasons that are out of the scope of the present document.

NOTE 1: The "Origin" header is not recommended as a mechanism for restricting which CSAs can access the CSS-CII service as it can easily be faked by rogue CSAs. See section 10.2 of IETF RFC 6455 [8] for a discussion of the scenarios for which the Origin header is intended to provide security.

The CSA shall disconnect by closing the connection according to the process described in the WebSocket protocol specification section 7 [8].

If the TV Device closes the connection, it shall do so according to the process described in the WebSocket protocol specification [8] section 7. This is not considered a normal closure, and so the TV Device shall provide an appropriate status code in the Close Frame to indicate the reason for the connection closure.

NOTE 2: Section 7.4 of the WebSocket protocol specification [8] provides a list of defined status codes.

Both TV Devices and CSAs shall gracefully handle the unexpected closing of a connection in the event that no WebSocket Close Frame is sent or received or in the event of the underlying TCP socket connection disconnecting, for example because of timeout.

7 Material Resolution Service (CSS-MRS)

7.1 General

The content identifier, retrieved as described in clause 6, can be translated into Material Information, potentially including Timeline Information, by the Material Resolution Service. The following clauses define the means by which the CSA requests the MRS to perform the translation and the format of the information returned.

7.2 Material resolution protocol

The protocol used to interact with the MRS shall be a REST API running over HTTP 1.1, as defined in IETF RFC 2616 [9] or HTTPS (specifically HTTP 1.1 running over TLS), as defined in IETF RFC 2818 [10]. The MRS URL indicates which protocol is to be used.

Whilst the HTTP response codes would normally be status code 200 OK, other status codes may also be returned, and these shall be handled appropriately. Notably, other successful (2xx), redirection (3xx) and Not-Modified (304) codes may be returned and shall be supported.

Both client error codes (4xx) and server error codes (5xx) may be returned. The behaviour under these cases is not defined, but it is recommended that the behaviour is the same as if no Content Identifier had been received from the TV Device.

NOTE 1: These error codes include those used to initiate authentication processes. The present document does not define authentication, but it may be used by other specifications or in later revisions of the present document.

The CSA shall support the "Expires" header. In the event that the "Expires" header indicates that the value has already expired, a new query shall not be issued for at least 2 secs. The CSA should not repeat a request until after the time indicated in the "Expires" header, unless otherwise indicated as described in clause 7.6.3.

NOTE 2: If the encoder clocks at the Broadcaster side are not genlocked, then the "Expires" header can be used to reflect how long a Correlation Timestamp is expected to remain valid. For example, the Broadcaster can measure by what time a differential drift of 1 ms would occur, and use that value to populate the "Expires" header.

The "max-age" directive may also be present. If present, as per the HTTP specification [9], it shall take precedence over the "Expires" header.

A CSA could be implemented as an HTML application and that application might not have originated from the MRS. As a result the protocol requests outlined in this clause potentially represent Cross Origin Resource Sharing, as outlined in CORS [14]. Such requests can be blocked or limited by the HTML environment (or user agent, in W3C terminology). To prevent such request blocking or limitations, the COS Device and MRS shall support CORS.

NOTE 3: Although the MRS is required to support CORS, it is at liberty to return headers that prevent or limit resource sharing.

7.3 Material resolution protocol query

7.3.1 Material resolution protocol general requirements

The CSA shall issue GET requests to the MRS_REST_API URL defined in clause 7.3.2.

The CSA shall include an "Accept: application/json" header field in the request.

The CSA shall support both gzip and identity (for non-gzipped JSON) encoding of responses, and include in the request an "Accept-Encoding" header field including gzip and identity.

The CSA shall support entity tags, or Etags, as defined in IETF RFC 2916 [12] for the request.

The CSA should include a "Referer" header in the request that shall be set to identify the CSA.

The CSA should include an "Origin" header, in the request as defined in IETF RFC 6454 [15].

NOTE: Where the CSA is not an HTML-5 application and does not have an Origin or Referer in HTML terms, if a value is used for the Origin or Referer, it is done so at the discretion of the application author. However, it is recommended that this value is an HTTP URL where the host part of the URL is a host that the CSA author has permission to use and that for the Referer header the remainder uniquely identifies the CSA.
It should be noted that these values are easily altered, faked or copied and so they should not be used for authentication or verification.
As certain non-browser based applications may prefer not to send these values, they are not mandated but instead recommended. They are recommended as including them provides a closer match to the native use of HTTP transfers.

Any repeat of a query shall use the "If-Modified-Since" header field.

Any repeat of a query that included entity tags shall use the "If-None-Match" header field.

7.3.2 Material resolution protocol URL

The URL that forms the MRS HTTP REST query shall be constructed by the concatenation, in order, of three parts:

- The mrsUrl as identified by the TV Device as described in clause 5.6.2 and as returned by the TV Device in the mechanism described in clause 6.2.
- The fixed string "/v1.1/MRS".
- The one material resolution protocol parameter as detailed in clause 7.3.3.

The format is defined in Augmented BNF (as defined in IETF RFC 5234 [4]) in table 7.3.2.1.

Table 7.3.2.1: Syntax of the MRS HTTP REST Query

MRS-REST-API	=	mrsUrl "/v1.1/MRS" "?" parameter "=" value
mrsUrl	=	(see note 1)
parameter	=	"contentId"
value	=	(see note 2)
NOTE 1: The mrs url is a well formed URL as received from the TV Device. As defined in clause 5.6, it does not end in "/".		
NOTE 2: As the value may include reserved or unsafe characters, as defined in IETF RFC 3986 [1], the value may be escape, or percent, encoded as described in IETF RFC 3986 [1].		

NOTE: The format of the URL and the parameter is deliberately specified so as to ensure that all CSAs generate the same URL for the same content identifier. This is intended to improve the cacheabilty of the response to the request.

7.3.3 Material resolution protocol parameter

The parameter name for the Content Identifier shall be the string "contentId" and the value shall be a Content Identifier (as defined in clause 5.2).

The parameter value shall be represented as an ASCII character string. The parameter value shall comply with the URI encoding rules defined in IETF RFC 3986 [1]. Specifically if any characters that make up the value fall in the reserved set (regardless of context), they shall be encoded using an escaped encoding, i.e. a "%" character followed by two upper case hexadecimal digits. Characters falling into the unreserved character set shall not be encoded.

NOTE: Limiting the use of escape coding helps to minimize the length of the URL and helps ensure a consistent URL for all instances of the same parameter.

7.4 Material resolution protocol response

The HTTP headers of the MRS response shall include the "Expires" header.

The HTTP headers of the MRS response shall also include the following CORS [14] headers, as defined in table 7.4.1.

Table 7.4.1: Required CORS headers in MRS protocol

CORS Header Field	Recommended Value	Notes
`Access-Control-Allow-Origin`	"*"	Values other than "*" may limit the availability of the returned data depending on the execution environment of the CSA.
`Access-Control-Allow-Method`	"GET"	Only the GET method is defined in the present document. If the GET method is omitted, certain execution environments may block attempts to perform the GET operations defined in the present document.
`Access-Control-Allow-Headers`	"X-Requested-With, Origin, If-Modified-Since, Accept, If-None-Match, Content-Type"	This allows access to key header fields in certain execution environments.
`Access-Control-Max-Age`	60	This allows some execution environments to remember the access control header fields thereby reducing communications. Some environments may well choose a larger value.

The HTTP headers of the MRS response may include the "max-age" Cache-Control header field.

The MRS response shall use the MIME type "application/json" as per IETF RFC 4627 [11].

The body of the MRS response may gzip encode the returned data.

The body of the MRS response is a JSON Document consisting of an MRS response object described by the root object of the JSON schema defined in clause A.1.3. The template below illustrates this JSON object:

```
{
    "type"                      : "response",
    "version"                   : "1.1",
    "rev"                       : <string>,
    "repollingInterval"         : <zeroOrPositiveInteger>,
    "materials"                 : { ... },
    "syncTimelineInformation"   : [ ... ],
    "updateMaterial"            : [ ... ],              (optional)
    "updateTimelineSync"        : [ ... ]               (optional)
}
```

Required properties:

- **type** with the string value "response".
- **version** with the string value "1.1". The version represents the specification against which this JSON document was authored.
- **rev** is an arbitrary string value representing the revision of the response. When comparing the response to two requests for resolution of the same CI, the response with the rev value that is greater (when performing a string comparison) will be the more recent and more accurate response. If both rev attributes take the same value then the responses can be assumed to be the same.

NOTE: Many possible schemes can be used to assign 'rev' values. For example: a UTC date-time of the form 'YYYY-MM-DDThh:mm:ssZ' (as defined in an IETF RFC 3339 [i.3]) would serve as a suitable scheme.

- **repollingInterval** is an integer with minimum value 0 that represents the recommended integer number of seconds after which a CSA should repeat the same request to the MRS in order to obtain an updated version of the Material Information. These repeat requests shall use the updateMaterial or updateTimelineSync URL(s), if provided, otherwise the repeat request shall use the same mechanism and URL as the original request.

A value of zero indicates that no re-polling should be performed. The relationship between the use of repollingInterval and the HTTP headers Expires and MaxAge is defined in clause 7.6.3.

- **materials** is a JSON object holding zero or more properties representing JSON Material objects as defined in clause 5.5.9.2. Each property name is arbitrary and uniquely identifies its associated Material object within the scope of all JSON Documents and updates to JSON Documents from the MRS. This name is used to reference this Material object from Timeline Mappings and other Material objects.

- **syncTimelineInformation** is a JSON array of zero or more JSON Synchronization Timeline information objects as defined in clause 5.5.9.4.

Optional properties:

- **updateMaterial** is an JSON array of zero or more URLs that provide the location at which updates to the returned information can be retrieved. If this property is omitted or has no elements, then there are no MRS update protocols implemented by the MRS that provide updates to Materials.

- **updateTimelineSync** is an JSON array of zero or more URLs that provide the location at which updates to the timeline synchronization information. If this property is omitted or has no elements, then there are no MRS update protocols implemented by the MRS that provide updates to Synchronization Timeline Information.

7.5 Material resolution protocol example (informative)

The following is an example of the interaction based on the case where the returned MRS URL is http://mrs.example.com and the content identifier is dvb://233a.1004.1044;35f7~20131004T0930Z--PT01H00M (from the example in clause C.2.1). This example represents the required subset of header fields; other header fields in both the request and response will normally be present.

EXAMPLE Request:

```
GET http://mrs.example.com/v1.1/MRS?contentId=dvb%3A%2F%2F233a.1004.1044%3B35f7~20131004T0930Z--
         PT01H00M HTTP/1.1

Accept: application/json

Accept-Encoding: identity, gzip

Referer: http://csa.example2.com/csa-apps/example

Origin: http://csa.example2.com
```

EXAMPLE CORS Related Response Headers:

```
Access-Control-Allow-Origin: *

Access-Control-Allow-Method: GET, OPTIONS, HEAD

Access-Control-Allow-Headers: X-Requested-With, Origin, Referer, If-Modified-Since, If-None-
         Match, Content-Type, Accept, Accept-Encoding

Access-Control-Max-Age: 86400;
```

NOTE: Some systems can use a "pre-flight" request [14] whereby CORS headers are only sent once, and cached for a time indicated by the Access-Control-Max-Age header. Consequently they may not appear on all transfers.

EXAMPLE Generic Response Headers:

```
HTTP/1.1 OK

Content-Encoding: gzip

Content-Type: application/json
```

7.6 MRS Update Protocols

7.6.1 General

The MI, or parts of it, as returned by the MRS may change whilst the content is viewed and so may require updates. Where the updateMaterial array is present and populated, it means that updates may, but need not, occur to the Materials detailed in the materials property. Where the updateTimelineSync array is present and populated, it means that updates may, but need not, occur to the Synchronization Timeline Information detailed in the syncTimelineInformation property.

The updateMaterial and updateTimelineSync properties are arrays of locations from which updates may be received or retrieved. The location is encoded as a URL and partially identifies the update protocol to use. For each location, a MIME type is also conveyed which completes the identification of the update protocol. These arrays may also contain locations and MIME types for other mechanisms that are not defined in the present document.

A CSA shall implement the long poll mechanism defined in clause 7.6.4. A CSA should implement the web socket mechanism defined in clause 7.6.5 and the server sent events mechanism defined in clause 7.6.6. A CSA may implement the BOSH/XMPP mechanism defined in clause 7.6.7.

When present, array element 0 shall locate and identify a Long Poll service as defined in clause 7.6.4. All other array elements are optional.

The order of the elements in the array indicates the preferred order of the locations, where the highest index indicates the most preferred choice. The CSA shall use the array elements in descending order, starting at the highest element present. The CSA may ignore any array entry whose URL and mimeType it does not understand or implement. Where a CSA fails to establish a connection using a given array entry, the CSA shall continue to try the remaining elements in descending order. Once a CSA has a successful connection, it shall continue to use the same the same update location, unless:

- an error condition occurs; or
- a new update array is received that no longer contains the update location currently in use.

If either case occurs, the CSA shall repeat the process of selecting an update location.

7.6.2 JSON syntax for update array element

The objects that make up the updateMaterial array and updateTimelineSync array shall follow the template:

```
{
    "url"                    : <url>,
    "mimeType"               : <string>,
    "protocolSpecificData"   : <string>     (optional)
}
```

Required properties:

- **url:** this carries the URL representation of the location of update information. It shall be complete and the client shall not modify this URL.
- **mimeType:** this shall further identify the URL protocol, as defined in the subsequent clauses of the present document.

Optional properties:

- **protocolSpecificData:** this shall carry any data that needs to be provided by the CSA when obtaining updates from the location described by the url property. This shall be specific to, and defined by, a given protocol. The format and any encoding necessary to encapsulate this information as a JSON string shall be defined by that protocol.

7.6.3 Update response JSON

7.6.3.1 Update response JSON types

The JSON object returned by the update protocols conveys changes to Materials or Synchronization Timeline Information. The JSON object shall be as defined in clause 7.4, but all properties apart from "type" are optional and it shall have additional properties as defined below in this clause. The type property shall have the value "update".

For an update to Materials the JSON object shall have the semantics defined in clause 7.6.3.2. For an update to Synchronization Timeline Information the JSON object shall have the semantics defined in clause 7.6.3.3.

The JSON object shall have additional properties as illustrated by the following template:

```
"updateVersionNo"   : <integerAsString>,
"patchData"         : <jsonPatchSyncTimelineInformation>    (optional)
```

Required properties:

- **updateVersionNo:** a string carrying a number that increments by one with each new update that is transmitted.

NOTE: The purpose of the updateVersionNo is to allow a CSA to identify if a given update is a repetition of one already received. This functionality is required if the CSA has to reconnect to the update server. In the case of LongPoll as defined in clause 7.6.4, the IfModifiedSince and IfMatch headers can provide a similar functionality, and can be used where the type is "response".

Optional properties:

- **patchData:** the data for a timeline sync update as defined in clause 7.6.3.3. The patchData property shall only be present for this type of update.

7.6.3.2 Semantics of Material update responses

The updates that are received through the URLs carried in the updateMaterial array property shall be a JSON object where if a property of that JSON object is present, it indicates an update to the value received in the original response, and shall be used to replace the value that property held in the original response. Where a property is not present, the original value is still valid and shall be used. The only property for which this does not apply is the "type" property, as detailed in clause 7.6.3.1.

NOTE 1: An update response that updates the collection of Materials is therefore expected to include the materials property. The value of that property is a complete replacement for all Materials signalled in any previous update or the original request.

NOTE 2: An update response can include the syncTimelineInfo property. An MRS that does not implement the Timeline Sync update mechanism can use this to provide a coarse update to the Synchronization Timeline Information. Use in conjunction with a Timeline Sync update mechanism is not recommended due to the possibility of race conditions.

7.6.3.3 Semantics of Timeline Sync update responses

The updates that are received through the URLs carried in the updateTimelineSync array property shall be JSON Patch objects as defined in IETF RFC 6902 [20]. These patch objects convey updates to the syncTimelineInformation object returned by the MRS according to the JSON patch object syntax. The JSON patch objects shall be interpreted in the order they are received. The "op" property shall only contain the operation "add" or "remove".

NOTE: How the CSA interprets the value of materials or syncTimelineInfo properties, if present in the JSON object, is outside the scope of the present document.

7.6.4 Long poll

A long poll is a normal request issued to the specified URL, where the response is as per clause 7.4 but where the JSON object in the response complies with the semantics defined in clauses 7.6.3.2 and 7.6.3.3 as appropriate.

NOTE 1: The use of the relevant If-Modified-Since or If-Match headers as defined in clause 7.3.1 are important for some implementations of long poll. These headers allow for the identification and removal of replicated updates in the presence of reconnections.

The response will not normally be received immediately and the server will normally hold the connection open, only returning the response when an update occurs. As such connections are commonly closed by the underlying network after a period of time, the CSA shall re-issue the command if the connection is unexpectedly closed.

NOTE 2: Such closure of the connection does not constitute an error condition in the terms used in clause 7.6.1 that trigger the selection of an update server.

When the connection closes, whether unexpectedly or as a result of completing the return of an update, a new request shall not be issued any earlier than indicated by the lesser of the value of the repollingInterval JSON object property in responses and updates (measured from the start of the last request), or the "Expires" header of the last successful HTTP response.

NOTE 3: The timing limits set out in the above allow for a server that does not implement long poll to utilize normal HTTP causing the CSA to implement a manual polling mechanism at a controlled rate. This clearly reduces the responsiveness of updates, but may be acceptable in some circumstances. A true long poll implementation can correctly set either of the fields to support immediate reconnection and ensure responsive updates.

The mimeType shall take the format application/json for updateMaterial, or application/json-patch for updateTimelineSync and the protocol component of the URL shall be "http" or "https".

7.6.5 WebSocket protocol

The mimeType shall take the format application/json for updateMaterial or application/json-patch for updateTimelineSync and the protocol component of the URL shall be "ws" or "wss". This URL shall refer to the server-side of the WebSocket protocol version 13 as defined in IETF RFC 6455 [8].

All messages sent shall be WebSocket data frames (as required by the WebSocket protocol specification [8]) with a payload in text format.

The CSA shall send no data to the server over the websocket.

The server will send update responses in accordance with clause 7.6.3.

7.6.6 Server Sent Events protocol

Server-sent events (SSE) shall comply with the W3C specification [19].

The mimeType shall be "text/event-stream" and the protocol component of the URL shall be "http" or "https".

The event type as signalled by the "event:" line of the event stream (as defined in SSE [19]) shall be set to "MIupdate" (i.e. the line "event: MIupdate" shall precede the event data) for material updates or be set to "TSupdate" (i.e. the line "event:TSupdate" shall precede the event data) for timeline sync updates.

The data shall be sent as a named data field (i.e. the string "data:" shall precede the event data).

The data for the event shall be the JSON object as defined in clause 7.6.3.2 for the "MIupdate" events, and a JSON Patch object as described in clause 7.6.3.3 for "TSupdate" events. To ensure correct transfer, the encoding of the JSON object shall not include the end of line characters as defined in SSE [19].

NOTE: The data field of an SSE is conveyed as a line of data terminated by a newline or linefeed or combination. Therefore these values cannot occur in the representation of the JSON object without breaking this update mechanism. As JSON does not require the presence of newline characters, this is not an issue for the JSON representation, but may represent an issue for code generating JSON objects.

7.6.7 BOSH/XMPP protocol

The mimeType for XMPP IETF RFC 6120 [29] based updates shall be application/jabber+xml. The protocol part of the URL is unspecified by IETF RFC 6120 [29], but should normally be "http" or "https".

The protocolSpecificData property shall be present and shall contain the base64 encoded XML stanza that shall be sent by the client to the server to subscribe to updates.

The returned body element of a message shall contain a JSON object as defined in clause 7.6.3.2, for updates to the Material Information, or a JSON object as defined in clause 7.6.3.3, for updates to the syncTimelineInformation. The JSON representation shall be encoded in base64 [24] in the body element to avoid risk of corruption of the XML document.

8 Wall Clock (CSS-WC)

8.1 General

To compensate for latencies in network communication between TV Device and CSA, the Timestamps exchanged in the Timeline Synchronization procedure make reference to a Wall Clock that is assumed to be synchronized between the TV Device and the CSA. Wall Clock Synchronization establishes a best effort approximation of a synchronized Wall Clock between them.

This clause describes the protocol used for Wall Clock Synchronization via the CSS-WC interface and the requirements for the implementation of a Wall Clock Synchronization service endpoint in the Wall Clock Server (WC Server) elementary function of the TV Device. It also describes the expected protocol interaction and the role of the Wall Clock Client (WC Client) elementary function of the CSA as outlined by the procedure described in clause 4.3.4. Clause C.8 provides informative guidance for implementers of both the WC Server and WC Client.

The WC Server elementary function of the TV Device shall implement a Wall Clock and support the Wall Clock Synchronization service described in clause 8.2 using the message format specified in clause 8.3 and protocol transport specified in clause 8.4.

The Wall Clock of the TV Device shall be monotonic and without discontinuities for the duration that the TV Device is powered and able to perform its role in Wall Clock Synchronization. The clock shall have a stable constant rate within the tolerance limits of a practical implementation (such as a crystal oscillator).

NOTE 1: A real time clock that adjusts for leap seconds, or which may abruptly change value because an NTP client adjusts it, is unlikely to be suitable for use as a Wall Clock in a TV Device.

NOTE 2: A clock that is monotonic (does not abruptly change value) but which slews its frequency under the control of an NTP client can be suitable provided that the maximum possible amount of slew is limited and reported as part of the maximum frequency error in protocol messages.

8.2 Protocol overview

8.2.0 Protocol Introduction

The protocol between a WC Client and WC Server consists of a regular exchange of messages between them. The WC Client periodically sends a request message to which the WC Server shall respond.

The protocol carries the following information:

- Time values, used to estimate the offset between WC Server Wall Clock and WC Client Clock (described in clause 8.2.1).
- Measurement precision, representing the maximum amount of precision possible in the process of reading a value from the Wall Clock by the Wall Clock Server (described in clause 8.2.2).

- Maximum frequency error, representing the maximum amount by which the frequency of the WC Server Wall Clock may deviate from its nominal value over time (described in clause 8.2.3).

The Time Values carried in the protocol from WC Server to WC Client are measurements of the Wall Clock in the TV Device.

The Time Values carried in the protocol from WC Client to WC Server are measurements of either the Wall Clock in the CSA or some other clock in the CSA from which the Wall Clock is derived. A CSA can derive a Wall Clock from some other clock by applying offsets and from frequency adjustments that are calculated as a result of information obtained from earlier Wall Clock protocol message exchanges.

NOTE: Clause C.8.3 provides examples of approaches to the design of the WC Client function and explains when it is appropriate to measure the Wall Clock or some other clock in the WC Client for the purposes of the Wall Clock protocol.

8.2.1 Time values and estimating Wall Clock offset

The WC Client records a time value T_1 for the moment at which the request message is assembled and sent to the WC Server and the time value T_4 for the moment at which the response message is received. The WC Server shall, similarly, record its own Wall Clock time value T_2 at which the request message is received and its own Wall Clock time value T_3 at which the response message is assembled and transmitted. This is illustrated in figure 8.2.1.1.

- T1 is known as the originate time value.
- T2 is known as the receive time value.
- T3 is known as the transmit time value.
- T4 is known as the response time value.

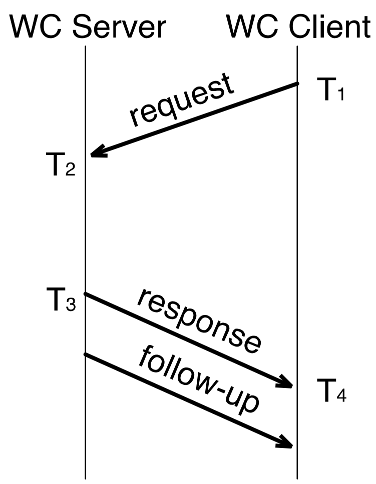


Figure 8.2.1.1: Overview of Wall Clock Synchronization service

Assuming that a request message and response message take equal amounts of time to be transmitted between WC Client and WC Server, it is possible for the WC Client to estimate the offset between the clock at the WC Client and the Wall Clock at the WC Server:

$$offset\ \theta = \frac{(T_3+T_2)-(T_4+T_1)}{2}$$

It is also possible for the WC Client to estimate the total time that the request and response messages were in flight - known as the round trip time:

$$round\_trip\_time\ \delta = (T_4 - T_1) - (T_3 - T_2)$$

The uncertainty bounds of the estimated offset (known as dispersion) are wider than or equal to plus or minus half the round trip time.

NOTE: Dispersion is explained in more detail for WC Client implementers in clause C.8.3.2.

When the WC Server sends its response, it may send a follow-up response that updates the WC Client with a revised response where the transmit time value T_3 is more accurate. The condition under which this behaviour is required of the WC Server is described in clause 8.3. If a WC Client receives a follow-up response message, it shall always substitute this in place of the response message that it is a follow up to when applying its filtering and clock adjustment processes.

8.2.2 Measurement precision

Measurement precision is the maximum error that there may be in a measurement of the Wall Clock due to the granularity by which the ticks of the Wall Clock increment, and by the time it takes to measure its value.

If the smallest regular increment of the Wall Clock is N ticks at a nominal tick rate of M ticks per second, then a reading of the Wall Clock at any moment in time is limited in precision to being within N/M seconds.

If the process of taking a reading takes ρ seconds, then the measurement precision is therefore:

$$measurement\ precision = \rho + N/M$$

NOTE: In a practical implementation, this is effectively the smallest reliably observable difference between two readings of the Wall Clock taken immediately one after the other.

8.2.3 Maximum frequency error

A Wall Clock may vary in frequency over time (slew from its nominal frequency) for a range of reasons. The maximum frequency error quantifies the maximum amount by which this may happen.

Frequency error is quantified in units of parts per million (ppm). A frequency error of N ppm means that a period of time t seconds measured according to the Wall Clock (assuming perfect precision) may be inaccurate by $\pm t \times N/1\ 000\ 000$ secs.

An underlying hardware oscillator is characterized in terms of the maximum expected frequency error, F ppm, given the expected operating conditions of the device.

A system where a clock discipline process is used (such as an NTP client) may intentionally slew the clock frequency. The maximum amount of slew that may be applied by such a process is ± L ppm and shall also be considered when determining the maximum frequency error.

If a clock discipline process (such as an NTP client) is intended to compensate for the frequency error of the underlying hardware oscillator, it is reasonable to assume that the effect of slew due to a clock discipline process is to reduce frequency error. Therefore maximum frequency error is defined as:

$$maximum\ frequency\ error\ (ppm) = \max(F, L)$$

If this is not the case then the maximum frequency error will be the sum of F and L.

NOTE: If there is uncertainty as to the exact maximum error frequency of a Wall Clock, it is recommended that a sensible upper bound value be selected for which there is reasonable confidence that it is no smaller than the true maximum error frequency. For example: a typical crystal oscillator for consumer applications usually has an accuracy considerably better than 100 ppm; and NTP clients on a typical Unix or Linux environment usually limit their maximum slew rate to 500 ppm. A maximum error frequency can, in these circumstances, be estimated with confidence to be no worse than 100 ppm if there is no clock discipline process, or 500 ppm if an NTP client may be running.

8.3 Wall Clock protocol

The syntax of the Wall Clock Synchronization messages is described in table 8.3.1.

Table 8.3.1: Syntax of Wall Clock synchronization message

Syntax	Value	No. bits	Identifier
wall_clock_sync_message {			
version	0	8	uimsbf
message_type		8	uimsbf
precision		8	tcimsbf
reserved	0	8	bslbf
max_freq_error		32	uimsbf
originate_timevalue {			
originate_timevalue_secs		32	uimsbf
originate_timevalue_nanos		32	uimsbf
}			
receive_timevalue {			
receive_timevalue_secs		32	uimsbf
receive_timevalue_nanos		32	uimsbf
}			
transmit_timevalue {			
transmit_timevalue_secs		32	uimsbf
transmit_timevalue_nanos		32	uimsbf
}			
}			

Time values in this syntax are conveyed as a 64 bit field comprising two 32 bit unsigned integer most significant byte first sub fields. The first is a number of seconds and the second field is a number of nanoseconds. The value of the nanosecond field shall be within the range 0 to 999 999 999 inclusive for **receive_timevalue** and **transmit_timevalue**, but this is not necessarily the case for **originate_timevalue** where the nanoseconds field can have any value.

version identifies the protocol version and shall have the value 0.

message_type determines the type of message and shall take one of the values shown in table 8.3.2.

Table 8.3.2: message_type field

message_type value	Meaning
0	Request from CSA.
1	Response from TV Device that will not be followed by a follow-up response
2	Response from TV Device that will be followed by a follow-up response
3	Follow-up response from TV Device
4 to 255	reserved

A request sent by the WC Client shall have **message_type** value of 0. The WC Server shall only send messages with **message_type** 1, 2 or 3 to a CSA.

precision is an 8-bit twos-compliment signed integer log base 2 of the measurement precision of the Wall Clock at the WC Server measured in seconds. If the value of this field is N, then the Wall Clock measurement precision achieved by the WC Server is 2^N seconds or less. The value of this field shall only be defined in this way for **message_type** 1, 2 and 3. For all other values of **message_type** the value of this field is undefined and should be ignored by the message recipient.

EXAMPLE 1: A precision value of -20 indicates a precision of about one microsecond or better.

8 bits are **reserved** and shall be set to zero.

max_freq_error is an unsigned 32-bit integer of the maximum frequency error of the Wall Clock at the WC Server measured in 1/256ths of parts per million (ppm). If the value of this field is N, then the maximum frequency error of the Wall Clock in the WC Server is N/256 ppm or less. The value of this field shall only be defined in this way for **message_type** 1, 2 and 3. For all other values of **message_type** the value of this field is undefined and should be ignored by the message recipient.

EXAMPLE 2: A max_freq_error value of 7 680 indicates a maximum frequency error of 30 ppm or less.

originate_timevalue is a time value of the clock at the WC Client at the time a **message_type** 0 request message was sent. A WC Server shall include it, unmodified in **message_type** 1, 2 or 3 response messages. Because this value is relayed back to the WC Client but not intended for used by the WC Server, the WC Client can use it to carry other information instead of a time value; such as any other unique numeric value that allows the WC Client to match up a response from the WC Server to the original request.

NOTE 1: If a WC Client is able to determine after sending a request message the accurate time at which the request was sent, then there is no need for the WC Client to measure the Clock before sending the request. The WC Client can therefore, for example, use a simple sequence number as the **originate_timevalue**.

receive_timevalue is a time value of the Wall Clock at the WC Server at the time a **message_type** 0 request message was received. The value of this field shall be set by the WC Server for **message_type** 1, 2 and 3. For all other values of **message_type** the value of this field is undefined and should be ignored by the message recipient.

transmit_timevalue is a time value of the Wall Clock in the WC Server at the time a **message_type** 1 message was transmitted. This value shall be set by the WC Server for **message_type** 1, 2 and 3. For **message_type** 3 (a follow up message) this shall be a more accurate measurement of the time of transmission of the previous response from the WC Server to the WC Client. All other fields in a follow up message shall be the same values as they were for the previous response sent to the WC Client. For all other values of **message_type** this field is undefined and should be ignored.

The WC Server function of the TV Device may ignore a received message where:

- the message is not 32 bytes in length;
- or the **version** is not equal to 0;
- or the **message_type** is not equal to 0.

In all other cases, the WC Server shall respond to the sender of a message where the **message_type is** 0 with a response message where the **message_type** is 1 or 2. In the response message, the **originate_timevalue** shall be the same as the **originate_timevalue** from the received request message. The **receive_timevalue** shall be the Wall Clock time value logged at the moment when the request message was received. The **transmit_timevalue** shall be the Wall Clock time value logged when the response message was assembled and transmitted.

If a WC Server is capable of determining a more accurate value for the **transmit_timevalue** after a response has been sent, then the response message **message_type** shall be 2 otherwise it shall be 1.

If a WC Server is capable of determining a more accurate value for the **transmit_timevalue** after a response has been sent, then the WC Server shall also send a follow-up message back to the sender. The follow-up message shall contain the same field values as the response message except the **message_type** shall be 3 and the **transmit_timevalue** shall be the more accurate value that has been determined.

NOTE 2: It is desirable for a TV Device to send responses that are both accurate and speedy. The revised transmit_timevalue in a follow-up response reduces the estimate of round trip time calculated by the WC Client and therefore reduces the dispersion. On congested or slow networks this could be milliseconds. Speedy responses minimize the contribution of frequency error to dispersion. If a TV Device waits 1 second before sending a response, then 500 ppm frequency error at both Server and Client will add 1 millisecond to the dispersion.

8.4 Wall clock protocol transport

A Wall Clock Synchronization message shall be transported as the payload of a single UDP packet.

A WC Server shall provide a Wall Clock Synchronization service endpoint (CSS-WC). The WC Server shall listen for Wall Clock Synchronization protocol request messages from CSAs on the IP interface and port number corresponding to the service endpoint.

The CSS-WC service endpoint (clause 5.6.7) is represented as a URL of the form "udp://<ip-address>:<port-number>", where <ip-address> and <port-number> are the IP address and port number, at which the WC Server is providing the service endpoint.

A WC Server shall send any response message and any follow-up message back to the same IP address and port number from which the original request was received. A WC Client shall therefore listen for response messages on the same IP interface and port number that it transmits request messages from. However if the number of request messages received per second exceeds the capacity of the WC Server to respond to them than the WC Server may ignore some request messages.

9 Timeline Synchronization (CSS-TS)

9.1 General

Timeline Synchronization is the process by which the SC elementary function of the CSA and the MSAS elementary function of the TV Device exchange Timestamp information via the CSS-TS interface in order to coordinate the synchronized presentation of Timed Content.

A Synchronization Timeline is agreed upon, and then Timestamps are exchanged that relate time values on the Synchronization Timeline to time values of the Wall Clock of the TV Device.

Clauses 9.2 and 9.3 describe the protocol and protocol transport used for Timeline Synchronization (CSS-TS) between the TV Device MSAS and CSA SC and the requirements for the implementation of a Timeline Synchronization service endpoint by the MSAS elementary function of the TV Device.

An overview of the procedures of the protocol (setup, exchange of timestamps and teardown) are described in clause 4.3.5. The semantics, data model and representation for the setup information and Timestamps carried in protocol exchanges are described in clause 5.7.

9.2 Protocol

The MSAS shall allow multiple CSAs to have open sessions to the CSS-TS service endpoints simultaneously. The MSAS shall allow a CSA to have multiple open sessions to a single CSS-TS service endpoint.

When a CSA begins the session the MSAS may decline to begin the session if the number of SCs communicating with the MSAS has exceeded any limit that the MSAS may impose. The MSAS may also decline to begin the session if Timeline Synchronization functionality is currently unavailable.

NOTE 1: Timeline Synchronization functionality might become unavailable because, for example, the TV Device is no longer presenting Timed Content, or because it is powering down. Other mechanisms that might cause Timeline Synchronization functionality to become unavailable are out of scope of the present document.

Once the session is established, the MSAS will be expecting to receive a setup data message (as described in clause 5.7.3) from the SC. While in this state, the MSAS shall ignore any message that does not conform to the representation of a setup data message defined in clause 5.7.3.

NOTE 2: Ignoring a message that does not conform to the representation of a setup data message means that the MSAS:

i) will not select the timeline to be used for Timeline Synchronization;

ii) will not send out a control timestamp; and

iii) will still expect to receive a (well conformed) setup data message from the SC.

After a setup data message is received, the MSAS shall ignore any message that does not conform to the representation of an Actual, Earliest and Latest Presentation Timestamp message defined in clause 5.7.4.

NOTE 3: Ignoring a message that does not conform to the representation of an Actual, Earliest and Latest Presentation Timestamp message means that the MSAS will not update its internal values of the Actual, Earliest and Latest Presentation Timestamp relative to that SC.

Once a setup data message has been received by the MSAS the Content Identifier stem and timeline selector provided in the setup data message shall determine the Timeline to be used for Timeline Synchronization for the rest of the session and whether the Timeline is currently available.

NOTE 4: A CSA can switch to using a different Synchronization Timeline by initiating a new session and tearing down the existing session.

The MSAS may ignore any properties in a setup data message that are not defined in the description of the representation format provided in clause 5.7.3.

The MSAS shall respond to receiving the setup data message by sending a Control Timestamp message to the SC. The SC may ignore any properties in a Control Timestamp that are not defined in the description of the representation format provided in clause 5.7.5.

NOTE 5: The quicker a response is sent by the TV Device, the quicker a CSA can align the timing of presentation of Timed Content.

At any given point in time, if the contentIdStem matches the CI of the Timed Content currently being presented by the TV Device (using matching process defined in clause 5.2.2) and if the Timeline specified by the timeline selector is derivable for the Timed Content currently being presented by the TV Device, then the Timeline is considered to be available.

A Timeline is derivable if it is possible to calculate a Time Value on that Timeline that corresponds to the point in the Timed Content currently being presented (using the process appropriate to the type of timeline requested as defined in clause 5.3).

In all other situations (including if the MSAS does not recognize or understand the provided timeline selector), the Timeline is considered to be unavailable.

NOTE 6: The reasons for a Timeline becoming unavailable include: the TV Device ceasing to present Timed Content; a channel change or broadcast programme junction on the TV Device resulting in the CI no longer matching the CI stem; or a Timeline in a broadcast or stream no longer being signalled.

The MSAS shall send an updated Control Timestamp message to the SC if any of the following conditions occur:

- The Timeline becomes available.
- The Timeline becomes unavailable.
- The Timeline is available and the value of the timeline speed multiplier changes.
- The Timeline is available and the timeline speed multiplier is a non-zero value and the updated Control Timestamp would represent a change in the timing of presentation relative to the Wall Clock of 1 ms or more when compared to the previous Control Timestamp sent to the SC.

NOTE 7: Possible causes of a change in timing relationship include: movements in clock drift; discontinuities in the timeline itself or in the carriage of the timeline; the addition or removal of SCs; or changes in the Actual, Earliest and Latest Presentation Timestamps received from an SC.

NOTE 8: Discontinuities signalled in the carriage of the Timed Content can indicate a possible change in timing relationship, such as jumps in the continuity_counter of a Transport Stream packet header or changes of the adaptation field discontinuity_indicator in a Transport Stream packet header for a stream.

A need for frame accurate synchronization at 50 fps would suggest that the CSA present a frame within an accuracy of ±10 ms of the TV Device presenting an associated frame. The TV Device should therefore send an updated Control Timestamp to the CSA when the change in timing relationship is substantially less than half of 10 ms. If the oscillator driving the Wall Clock in the TV Device is accurate with respect to a broadcast stream that the TV Device is presenting to within 1 000 ppm, then 1ms drift in timing relationship will occur no more frequently than once per second. This is a reasonable rate for sending updated Control Timestamps to CSAs.

If the required accuracy of synchronization is more demanding (e.g. ±1 ms or ±2 ms), then it is recommended that the MSAS send updated Control Timestamps if the timing relationship differs by an amount substantially less than 1 ms and substantially less than the required accuracy.

From the start of the session, the MSAS shall assume that the initial values for the Actual Presentation Timestamp, Earliest Presentation Timestamp and Latest Presentation Timestamp for the SC are those listed in table 9.2.1.

Table 9.2.1: Initial values for an SC's Actual, Earliest and Latest Presentation Timestamps

Timestamp	content time	Wall Clock time
Actual Presentation Timestamp	undefined	undefined
Earliest Presentation Timestamp	undefined	-∞
Latest Presentation Timestamp	undefined	+∞

The values of content time and Wall Clock Time shown in table 9.2.1 correspond to the actual values that the MSAS shall assume and are not encoded in a message representation format (such as JSON).

If the MSAS receives an Actual, Earliest and Latest Presentation Timestamp message (as described in clause 5.7.4) from the SC then the values within the received message shall be used in place of the initial values or any previous values received from that SC. The MSAS may ignore any properties in an Actual, Earliest and Latest Presentation Timestamp message that are not defined in the description of the representation format provided in clause 5.7.4.

While the Timeline is available, the MSAS will provide Control Timestamps with the aim of achieving synchronized timing of presentation between SCs currently engaging in Timeline Synchronization with the MSAS. This process shall take into account the restrictions on timing of presentation of each SC as indicated by the most recent Actual, Earliest and Latest Presentation Timestamps provided by each SC or the initial values in table 9.2.1 if they are still applicable. However the MSAS may choose to not take into account the Actual, Earliest and Latest Presentation Timestamp communicated by the SC in a session where the Timeline is currently unavailable. If the Timeline becomes available again, the MSAS shall have remembered the most recently provided Actual, Earliest and Latest Presentation Timestamp (even if provided during a period in which the timeline was unavailable) and shall take them into account again.

NOTE 9: Clauses C.5.3 and C.6.3 provide examples of how an MSAS can calculate a Control Timestamp given Actual, Earliest and Latest Presentation Timestamps reported by one or more SCs.

If the Timeline Synchronization functionality is to become unavailable, the MSAS shall initiate the termination of the established session. The CSA may teardown an established session at any time.

9.3 Protocol transport

The MSAS elementary function of the TV Device shall implement a CSS-TS service endpoint that implements the server-side of the WebSocket protocol version 13 as defined in IETF RFC 6455 [8]. The service endpoint provided by the MSAS function in the CII message is the WebSocket URL of the WebSocket server.

NOTE 1: As defined in section 3 of the WebSocket protocol specification [8], the WebSocket URL defines the host, port, security and resource name of the service endpoint.

All messages sent by SC and MSAS shall be WebSocket data frames (as required by the WebSocket protocol specification [8]) with a payload in text format.

The SC shall begin the session of protocol exchanges by establishing a connection to the CSS-TS service endpoint in the role of a client of the WebSocket protocol.

If the CSS-TS service endpoint is currently unavailable then the MSAS function of the TV Device shall refuse the connection request by responding with an HTTP response code of 403 "Forbidden".

If the MSAS function of the TV Device has reached the limit of the number of simultaneous connections to the CSS-TS service endpoint that it can handle, then it shall refuse the connection request by returning the HTTP response code 503 "Service Unavailable".

The MSAS may examine the "Origin" header if it is present in the client handshake used in the WebSocket protocol to establish the connection. On the basis of the value of the "Origin" header, the MSAS may choose to refuse to establish the connection and respond with a 403 "Forbidden" response code, as described in section 10.2 of IETF RFC 6455 [8]. The rules used by a MSAS to determine when to allow and when to refuse a connection are outside of the scope of the present document.

NOTE 2: The Origin header is only supplied by web browser based SCs. It is unlikely to be provided by non-browser based SCs. Its value can also be easily faked by non-browser based SCs. Therefore this is not recommended as a mechanism for restricting which SCs can access the CSS-TS service. The likely most appropriate behaviour is for a MSAS to permit connections to be established irrespective of the value of the Origin header. See section 10.2 of IETF RFC 6455 [8] for a discussion of the scenarios for which the Origin header is intended to provide security.

The MSAS function of the TV Device may return 400 and 500 series HTTP response codes (including code 403 "Forbidden" and 503 "Service Unavailable") for other reasons that are out of the scope of the present document.

The SC shall teardown the session of protocol exchanges according to the process described in the WebSocket protocol specification section 7 [8].

The MSAS shall teardown the session of protocol exchanges by closing the connection as described in the WebSocket protocol specification [8] section 7. If the MSAS is tearing down the connection because the CSS-TS service endpoint is to become unavailable, then the MSAS shall provide the reason code for closure of 1001 indicating that the service endpoint is "going away".

Both MSAS and SC shall gracefully handle the closing of a connection in the event that no WebSocket Close Frame is sent or received or in the event of the underlying TCP socket connection timing out.

10 Trigger Events (CSS-TE)

10.1 General

The CSA obtains Trigger Event notifications from the TV Device to enable the CSA to discover if there are editorially significant occurrences in the media currently being presented by the TV Device. The Trigger Event notification and the optional data it carries allow the CSA to respond to the event. This functionality can be highly scalable as the CSA does not need to communicate with the broadcaster's servers at the time of the event.

Clauses 10.2 and 10.3 describe the protocol and protocol transport used for delivering Trigger Events from the TV Device to the CSA via the CSS-TE interface and the requirements for the implementation of a Trigger Event service endpoint in the TV Device.

10.2 Protocol

An overview of the procedures of the protocol are described in clause 4.3.6 and the message exchanged are described in clause 5.8.

Subject to the number of sessions not excluding any limit that the TV Device may impose, the TV Device shall allow multiple CSAs to have simultaneously open sessions to the CSS-TE service endpoint and the TV Device shall allow each CSA to have multiple simultaneously open sessions to the CSS-TE service endpoint. Subject to the number of event subscriptions not excluding any limit that the TV Device may impose, the TV Device shall allow multiple simultaneous event subscriptions on each session.

The TV Device may decline to begin a session if the Trigger Event functionality is currently unavailable.

NOTE: Trigger Event functionality might become unavailable because, for example, the TV Device is powering down.

In normal operation, the TV Device should not disconnect the CSA until the CSA disconnects itself.

If a TV Device is functioning but currently not providing Trigger Event functionality, it should continue to provide the CSS-TE service endpoint but instead respond to any subscriptions with a "not available" notification.

10.3 Protocol Transport

A TV Device that supports the a CSS-TE service endpoint shall implement the server-side of the WebSocket protocol version 13 as defined in IETF RFC 6455 [8]. The service CSS-TE endpoint provided by the TV Device is describable by a WebSocket URL.

NOTE 1: As defined in section 3 of the WebSocket protocol specification [8], the WebSocket URL defines the host, port, security and resource name of the service endpoint.

All messages sent by CSA and CSS-TE service shall be WebSocket data frames (as required by the WebSocket protocol specification [8]) with a payload in text format.

To use the CSS-TE service the CSA shall establish a connection to the CSS-TE service endpoint using the WebSocket protocol.

If the CSS-TE service endpoint is currently unavailable then the TV Device shall refuse the connection request by responding with an HTTP response code of 403 "Forbidden".

If the CSS-TE service has reached the limit of the number of simultaneous connections to the CSS-TE service endpoint that it can handle, then the TV Device shall refuse the connection request by returning the HTTP response code 503 "Service Unavailable".

The CSA and the CSS-TE service shall manage Trigger Event subscriptions using Trigger Event Subscription Management messages communicated via the connection between the CSA and the CSS-TE service.

The CSS-TE service may examine the "Origin" header if it is present in the client handshake used in the WebSocket protocol to establish the connection. On the basis of the value of the "Origin" header, the CSS-TE service may choose to refuse to establish the connection and respond with a 403 "Forbidden" status code, as described in section 10.2 of IETF RFC 6455 [8]. The rules used by a CSS-TE service to determine when to allow and when to refuse a connection are outside of the scope of the present document.

NOTE 2: The Origin header is only supplied by web browser based CSAs. It is unlikely to be provided by non-browser based CSAs. Its value can also be easily faked by non-browser based CSAs. Therefore this is not recommended as a mechanism for restricting which CSAs can access the CSS-TE service. The likely most appropriate behaviour is for a CSS-TE service to permit connections to be established irrespective of the value of the Origin header. See section 10.2 of IETF RFC 6455 [8] for a discussion of the scenarios for which the Origin header is intended to provide security.

The CSS-TE service may return 400 and 500 series HTTP response codes (including code 403 "Forbidden" and 503 "Service Unavailable") for other reasons that are out of the scope of the present document.

When the CSA no longer requires the Trigger Event service it shall first unsubscribe any current Trigger Event subscriptions and then shall close the connection according to the process described in the WebSocket protocol specification section 7 [8].

If the CSS-TE service or the TV Device needs to close the connection, it should first cancel any current Trigger Event subscriptions and then shall close the connection according to the process described in the WebSocket protocol specification [8] section 7. This is not considered a normal closure, and so the TV Device shall provide an appropriate status code in the Close Frame to indicate the reason for the connection closure.

NOTE 3: Section 7.4 of the WebSocket protocol specification [8] provides a list of defined reason codes.

Both CSS-TE service and CSA shall gracefully handle the closing of a connection if no WebSocket Close Frame is sent or received or in the event of the underlying TCP socket connection timing out.

11 Timelines in Transport Stream adaptation fields

11.1 General

This clause defines a mechanism for the carriage of a timeline in the adaptation header of transport stream packets:

- Clause 11.3 defines the use of a mechanism published by MPEG in July 2015 as an amendment to ISO/IEC 13818-1 [6].

NOTE: Clause 11.2 in an earlier version of the present document defined another mechanism that is now deprecated.

11.2 Void

11.3 Timed External Media Information (TEMI)

11.3.1 General

Adaption field descriptors for Timed External Media Information (TEMI) is a draft amendment (DAM) to the MPEG Systems specification ISO/IEC 13818-1 [6] that defines a mechanism for carriage of a timeline in the adaptation field of a transport stream packet that contains a PES stream with PTS declared in the PES header. It is defined in [25]. This clause (11.3) defines its Timeline Selector and the requirements that a TV Device shall meet if the TV Device supports the use of a TEMI timeline.

11.3.2 Timeline Selector for an MPEG TEMI timeline

The format for a timeline selector for a temi_timeline is shown by the following Augmented BNF (as defined in IETF RFC 5234 [4]) rule definition for temi-timeline-selector:

temi-timeline-selector = "urn:dvb:css:timeline:temi:" component-tag ":" timeline-id

component-tag is the base-ten integer representation of the value of the component_tag (declared by a stream identifier descriptor as defined in ETSI EN 300 468 [13]) associated with stream of TS packets that carry the temi _timeline_descriptor.

timeline-id is the base-ten integer representation of the value of the timeline_id field in the temi_timeline_descriptor.

The timeline to be selected shall be determined from the most recently received temi_timeline_descriptor whose timeline_id matches that specified in the Timeline Selector and which is carried in the af_descriptor() in the adaptation_field for TS packets identified by the component tag specified in the Timeline Selector.

11.3.3 Interpretation of a temi_timeline_descriptor

The TV Device shall decode a temi_timeline_descriptors as defined in [25].

If the has_timestamp field of a temi_timeline_descriptor is 1 or 2 then the timeline tick value is taken from the media_timestamp field and the tick rate is determined from the timescale field.

The unitsPerTick of the timeline shall be 1 and the unitsPerSecond of the timeline shall be equal to the value of the timescale field.

If the has_timestamp field is neither a value 1 or 2, then the timeline tick rate and tick value are undefined.

The TV Device may ignore the value of the has_ntp, has_ptp and has_timecode fields.

If the paused field is 1 then the tick rate is unchanged, but the TV Device shall behave for the purposes of calculating Timestamps as if the presentation has been paused. Specifically, a TEMI timeline with a paused field set to 1 may still be selected and used, and the corresponding Control Timestamps will include a constant contentTime field and a timelineSpeedMultiplier field that is set to 0.

EXAMPLE: The MSAS in a simple TV Device sends Control Timestamps to SCs in CSAs based solely on the behaviour of the SC within the TV Device that is presenting broadcast content containing a TEMI timeline. If that TEMI timeline paused field is 1, then a Control Timestamp sent by the MSAS will have a timeline speed multiplier set to the value 0.

If the TV Device detects a change in the pause field or a discontinuity in a timeline by observing a change in the discontinuity field from a 0 to a 1 from one to the next, then the SC elementary function of the TV Device should report a revised Earliest and Latest Presentation Timestamps to the MSAS function. This is because it will likely result in a significant change to the correlation between the Wall Clock and the timeline. In turn, this will normally generate the transmission of a new Control Timestamp from the TV Device.

Extrapolated Time Values on the TEMI timeline shall not be disrupted by a wrap of PTS values. A disruption consists of one or more Control Timestamps being sent to CSAs that represent a large shift in presentation timing when no such shift has actually taken place at the TV Device.

NOTE: temi_timeline_descriptors are not always carried on every access unit. Extrapolation of Time Values for access units without a temi_timeline_descriptor (following an earlier access unit that did have a temi_timeline_descriptor) is involves calculating the difference between the PTS of the two access units. This calculation needs to correctly handle a situation where PTS has wrapped between the two access units.

11.3.4 Determining Disappearance of a TEMI timeline (informative)

A TV Device is faced with the challenge of deciding both when a previously seen TEMI timeline has disappeared, and when a TEMI timeline that it is looking for is not present. Firstly, as the use of PMT signalling to indicate the TEMI timeline is not required, there is no explicit in stream signalling of the presence or absence of a TEMI timeline. Next as a timeline may come and go at any point during a content item, it is not possible to assume that the timeline presence is related to the content identifier. Further, TEMI timeline descriptors are not required to be present on every access unit and are expected to be interpreted meaning that the absence of a descriptor on an access unit does not imply the absence of a timeline. Finally, in the case of a broadcast where splices may occur the TEMI timeline may disappear and reappear over periods potentially measured in minutes.

It is recommended that a TV Device uses a time threshold of approximately 2,5 secs to decide if a timeline has disappeared. I.e. 2,5 secs after it last saw a TEMI timeline descriptor, or after it started looking for a timeline descriptor, a TV Device may decide that the timeline has disappeared, and inform any interested CSA appropriately.

Further, it is recommended that a CSA is not designed to make use of the disappearance of a timeline.

11.3.5 TEMI timelines and bitstream manipulation (informative)

Certain operations in the transmission of content, e.g. splicing of a video component might result in the temporary disappearance of a TEMI timeline. As a TV Device is unable to differentiate between the end of a TEMI timeline (especially if it has only just tuned to a channel) and one that is temporarily disappeared whilst alternative content is overlaid, any such operation needs to preserve the original timeline. It is therefore recommended that any equipment manipulating the stream ensures that the timeline from the original source is still present in the output when it is expected that the original source will eventually be re-instated. This allows a TV Device to determine that the timeline is present.

12 Connection and playback session management

A TV Device and a CSA engaged in synchronized playback normally use the CSS-WC interface (see clause 8), in conjunction with at least either the CSS-CII interface (see clause 6), the CSS-TS interface (see clause 9), or the CSS-TE interface (see clause 10). Under certain circumstances, they may also use all four interfaces in parallel. Table 12.1 summarizes these semantics.

Table 12.1: Combinations of CSS network connections active state

CSS-WC connection	CSS-CII connection	CSS-TS connection	CSS-TE connection	Description
not established	don't care	don't care	don't care	No synchronized playback is happening.
established	not established	don't care	don't care	
established	established	don't care	don't care	A synchronized playback session is possible.

As opposed to the other interfaces, the CSS-WC interface uses connection-less UDP transport; hence no "established" or "not established" connection status can be inferred from the underlying transport connection. To detect a failure of the use of this interface, a CSA shall hence use the calculated dispersion of its estimate of the TV Device Wall Clock (see clause C.8.3.2). If the calculated dispersion exceeds a threshold value, the use of CSS-WC interface shall be deemed to be interrupted. Since an appropriate threshold value will depend on the context (e.g. the running application) in which the playback session is used, CSAs should provide suitable interfaces to running applications, allowing them to indicate a desired threshold.

If at least one of the CSS-CII, CSS-TS or CSS-TE interface connections fails the respective WebSocket connection as defined in clause 7.1.7 of the WebSocket protocol specification [8], or the CSS-WC connection is to be deemed failed as defined above, then the whole synchronized playback session shall be deemed interrupted.

How a TV Device or a CSA should react to any of the types of interruption described in this clause will depend on the context (e.g. the running application) in which the playback session is used. TV Devices and CSAs should therefore provide suitable interfaces to running applications, allowing them to detect such an interruption, and to react in an appropriate way.

Annex A (informative): JSON representation

A.1 JSON Schemas

A.1.1 General

The schemas here will validate the JSON representations of protocol messages described in the present document. The schemas are written as a JSON Schema according to the JSON Schema draft 4 specifications for core [i.7] and validation [i.8].

A core schema in clause A.1.2 defines common types. Individual separate schemas combined with the core schema will validate correctly formed JSON representations of each message.

Normative text in other parts of the present document place additional requirements on the content and format of messages. Validation of a message against the appropriate schema provided here is therefore not sufficient to guarantee the correctness of a message.

A.1.2 Core schema

```
{
    "$schema"     : "http://json-schema.org/draft-04/schema#",
    "id"          : "https://www.dvb.org/metadata/css/css-core-1.1.1#",
    "title"       : "DVB Companion Screens and Streams - Core schema types",
    "description" : "",


    "definitions" : {

        "contentId" : { "type" : "string", "format" : "uri" },

        "contentIdStem" : { "type" : "string", "format" : "uri" },

        "id" : {
            "type"       : "object",
            "properties" : {
                "type"    : { "type" : "string", "format" : "uri" },
                "id"      : { "type" : "string" },
                "private" : { "$ref" : "#/definitions/private" }
            },
            "required" : [ "type", "id" ],
            "additionalProperties" : true
        },

        "private" : {
            "type"     : "array",
            "minItems" : 0,
            "items"    : {
                "type"       : "object",
                "properties" : {
                    "type" : { "type" : "string", "format" : "uri" }
                },
                "required" : [ "type" ],
                "additionalProperties" : true
            }
        },

        "timelineProperties" : {
            "type" : "object",
            "properties" : {
                "unitsPerTick"   : { "$ref" : "#/definitions/nonZeroPositiveInteger" },
                "unitsPerSecond" : { "$ref" : "#/definitions/nonZeroPositiveInteger" },
                "accuracy"       : { "$ref" : "#/definitions/zeroOrPositiveNumber" }
            },
            "required" : [ "unitsPerTick", "unitsPerSecond" ],
            "additionalProperties" : true
        },
```

```
        "timelineSelector" : { "type" : "string", "format" : "uri" },

        "mapping" : {
            "type"       : "object",
            "properties" : {
                "start"                : { "$ref" : "#/definitions/integerAsString" },
                "end"                  : { "$ref" : "#/definitions/integerAsString" },
                "materialIndex"        : { "type" : "string" },
                "correlationsChanging" : { "type" : "boolean" },
                "correlations" : {
                    "type"     : "array",
                    "minItems" : 1,
                    "items"    : { "$ref" : "#/definitions/correlation" }
                }
            },
            "required" : [ "start", "end", "materialIndex", "correlationsChanging", "correlations"
],
            "additionalProperties" : true
        },

        "correlation" : {
            "type"       : "object",
            "properties" : {
             "point"         : { "$ref" : "#/definitions/integerAsString" },
             "materialPoint" : { "$ref" : "#/definitions/integerAsString" }
            },
            "required" : [ "point", "materialPoint" ],
            "additionalProperties" : true
        },

        "integerAsString" : {
            "type"    : "string",
            "pattern" : "^(0|(-?[1-9])[0-9]*)$"
        },

        "integerMinimumZeroAsString" : {
            "type"    : "string",
            "pattern" : "^(0|([1-9])[0-9]*)$"
        },

        "zeroOrPositiveNumber" : {
            "type"             : "number",
            "minimum"          : 0,
            "exclusiveMinimum" : false
        },

        "nonZeroPositiveInteger" : {
            "type"             : "integer",
            "minimum"          : 0,
            "exclusiveMinimum" : true
        },

        "zeroOrPositiveInteger" : {
            "type"             : "integer",
            "minimum"          : 0,
            "exclusiveMinimum" : false
        },

        "triggerEvents" : {
            "type" : "object",
            "patternProperties" : {
                "^[a-zA-Z][0-9a-zA-Z_]*$" : { "$ref" : "#/definitions/triggerEventLocator" }
            }
        },

        "triggerEventLocator" : { "type" : "string", "format" : "uri" },

        "jsonPatchSyncTimelineInformation" : {
            "type"     : "array",
            "minItems" : 0,
            "items"    : {
                "type"  : "object",
                "oneOf" : [
                    { "$ref" : "#/definitions/jsonPatch_add_op" },
                    { "$ref" : "#/definitions/jsonPatch_remove_op" }
                ],
                "properties" : {
```

```
                "path" : { "$ref" : "#/definitions/jsonPointer" }
            },
            "required" : [ "op", "path" ],
            "additionalProperties" : true
        }
    },

    "jsonPatch_add_op" : {
        "properties" : {
            "op" : { "enum" : [ "add" ] }
        },
        "required" : [ "value" ]
    },

    "jsonPatch_remove_op" : {
        "properties" : {
            "op" : { "enum" : [ "remove" ] }
        }
    },

    "jsonPointer" : {
        "type"    : "string",
        "pattern" : "^(/([^/~]|~[01])*)*$"
    }
  }
}
```

A.1.3 Material Information (MI) schema

This schema will validate a correctly formed JSON representation of a Material Information response payload:

```
{
    "$schema"     : "http://json-schema.org/draft-04/schema#",
    "id"          : "https://www.dvb.org/metadata/css/css-mrs.mi-1.1.1#",
    "title"       : "DVB Companion Screens and Streams - CSS-MRS Material Information message
representation Schema",
    "description" : "Material Resolution Service response or update response containing Material
Information",

    "type"       : "object",
    "properties" : {
        "version" : { "enum" : [ "1.1" ] },
        "rev"     : { "type" : "string" },
        "repollingInterval" : { "$ref" : "css-core-1.1.1#/definitions/zeroOrPositiveInteger" },
        "materials" : {
            "type"     : "object",
            "minItems" : 0,
            "patternProperties" : {
                "^.+$" : { "$ref" : "#/definitions/material" }
            }
        },
        "syncTimelineInformation" : {
            "type"     : "array",
            "minItems" : 0,
            "items"    : { "$ref" : "#/definitions/syncTimelineInformation" }
        },
        "updateMaterial" : {
            "type"     : "array",
            "minItems" : 0,
            "items"    : { "$ref" : "#/definitions/updateElement" }
        },
        "updateTimelineSync" : {
            "type"     : "array",
            "minItems" : 0,
            "items"    : { "$ref" : "#/definitions/updateElement" }
        }
    },
    "oneOf" : [
        {
            "properties" : {
                "type" : { "enum" : [ "response" ] }
            },
            "required" : [ "type", "version", "rev", "repollingInterval", "materials",
"syncTimelineInformation" ],
            "not"      : {
                "properties" : { "patchData" : { } },
```

```
                "required"   : [ "patchData" ]
            }
        },
        {
            "properties" : {
                "type"             : { "enum" : [ "update" ] },
                "updateVersionNo" : { "$ref" : "css-core-1.1.1#/definitions/integerAsString" },
                "patchData"        : { "$ref" : "css-core-
1.1.1#/definitions/jsonPatchSyncTimelineInformation" }
            },
            "required" : [ "type", "updateVersionNo" ]
        }
    ],
    "additionalProperties" : true,

    "definitions" : {

        "material" : {
            "type"       : "object",
            "properties" : {
                "parents" : {
                    "type"     : "array",
                    "minItems" : 0,
                    "items"    : { "type" : "string" }
                },
                "contentIdStem" : { "$ref" : "css-core-1.1.1#/definitions/contentIdStem" },
                "ids" : {
                    "type"     : "array",
                    "minItems" : 1,
                    "items"    : { "$ref" : "css-core-1.1.1#/definitions/id" }
                },
                "timelineProperties" : {
                    "allOf" : [
                        { "$ref"     : "css-core-1.1.1#/definitions/timelineProperties" },
                        { "required" : [ "accuracy" ] }
                    ]
                },
                "triggerEventInfo" : {
                    "type"       : "object",
                    "properties" : {
                        "contentIdStem" : { "$ref" : "css-core-1.1.1#/definitions/contentIdStem" },
                        "triggerEvents" : { "$ref" : "css-core-1.1.1#/definitions/triggerEvents" }
                    },
                    "required" : [ "contentIdStem", "triggerEvents" ]
                },
                "private" : { "$ref" : "css-core-1.1.1#/definitions/private" }
            },
            "required" : [ ""ids", "parents" ],
            "additionalProperties" : true
        },

        "syncTimelineInformation" : {
            "type"       : "object",
            "properties" : {
                "contentIdStem"       : { "$ref" : "css-core-1.1.1#/definitions/contentIdStem" },
                "leadupContentIdStem" : { "$ref" : "css-core-1.1.1#/definitions/contentIdStem" },
                "timelineSelector"    : { "$ref" : "css-core-1.1.1#/definitions/timelineSelector" },
                "timelineProperties"  : {
                    "allOf" : [
                        { "$ref"     : "css-core-1.1.1#/definitions/timelineProperties" },
                        { "required" : [ "accuracy" ] }
                    ]
                },
                "mappings" : {
                    "type"     : "array",
                    "minItems" : 0,
                    "items"    : { "$ref" : "css-core-1.1.1#/definitions/mapping" }
                },
                "private" : { "$ref" : "css-core-1.1.1#/definitions/private" }
            },
            "required" : [ "contentIdStem", "timelineSelector", "timelineProperties", "mappings" ],
            "additionalProperties" : true
        },

        "updateElement" : {
            "type"       : "object",
            "properties" : {
                "url"                   : { "type" : "string", "format" : "uri" },
```

```
            "mimeType"             : { "type" : "string" },
            "protocolSpecificData" : { "type" : "string" }
        },
        "required" : [ "url", "mimeType" ],
        "additionalProperties" : true
    }
  }
}
```

A.1.4 Content Identification and other Information (CII) schema

This schema will validate a correctly formed JSON representation of a Content Identification and other Information (CII) message:

```
{
    "$schema"     : "http://json-schema.org/draft-04/schema#",
    "id"          : "https://www.dvb.org/metadata/css/css-cii.cii-1.1.1#",
    "title"       : "DVB Companion Screens and Streams - CSS-CII Content Identification and other
Information message representation schema",
    "description" : "",

    "type"       : "object",
    "properties" : {
        "protocolVersion" : { "enum" : [ "1.1" ] },
        "mrsUrl"          : {
            "oneOf" : [
                { "type" : "string", "format" : "uri" },
                { "type" : "null" }
            ]
        },
        "contentId" : {
            "oneOf" : [
                { "$ref" : "css-core-1.1.1#/definitions/contentId" },
                { "type" : "null" }
            ]
        },
        "contentIdStatus" : {
            "oneOf" : [
                { "enum" : [ "partial", "final" ] },
                { "type" : "null" }
            ]
        },
        "presentationStatus" : { "$ref" : "#/definitions/presentationStatus" },
        "wcUrl" : {
            "oneOf" : [
                { "type" : "string", "format" : "uri" },
                { "type" : "null" }
            ]
        },
        "tsUrl" : {
            "oneOf" : [
                { "type" : "string", "format" : "uri" },
                { "type" : "null" }
            ]
        },
        "teUrl" : {
            "oneOf" : [
                { "type" : "string", "format" : "uri" },
                { "type" : "null" }
            ]
        },
        "timelines" : {
            "oneOf" : [
                { "type" : "array",
                 "items" : { "$ref" : "#/definitions/timelineOption" },
                 "minItems" : 0
                },
                { "type" : "null" }
            ]
        },
        "private" : { "$ref" : "css-core-1.1.1#/definitions/private" }
    },
    "additionalProperties" : true,

    "definitions" : {
        "presentationStatus" : {
```

```
            "type"    : "string",
            "pattern" : "^(okay|transitioning|fault|[^ ]+)( [^ ]+)*$"
        },
        "timelineOption" : {
            "type"       : "object",
            "properties" : {
                "timelineSelector"   : { "$ref" : "css-core-1.1.1#/definitions/timelineSelector" },
                "timelineProperties" : { "$ref" : "css-core-1.1.1#/definitions/timelineProperties"
},
                "private"            : { "$ref" : "css-core-1.1.1#/definitions/private" }
            },
            "required" : [ "timelineSelector", "timelineProperties" ],
            "additionalProperties" : true
        }
    }
}
```

A.1.5 Setup-data schema

This schema will validate a correctly formed JSON representation of a setup-data message:

```
{
    "$schema"     : "http://json-schema.org/draft-04/schema#",
    "id"          : "https://www.dvb.org/metadata/css/css-ts.setup-data-1.1.1#",
    "title"       : "DVB Companion Screens and Streams - CSS-TS Setup Data message representation
schema",
    "description" : "",

    "type"       : "object",
    "properties" : {
        "contentIdStem"    : { "$ref" : "css-core-1.1.1#/definitions/contentIdStem" },
        "timelineSelector" : { "$ref" : "css-core-1.1.1#/definitions/timelineSelector" },
        "private"          : { "$ref" : "css-core-1.1.1#/definitions/private" }
    },
    "required" : [ "contentIdStem", "timelineSelector" ],
    "additionalProperties" : true
}
```

A.1.6 Control Timestamp schema

This schema will validate a correctly formed JSON representation of a Control Timestamp message:

```
{
    "$schema"     : "http://json-schema.org/draft-04/schema#",
    "id"          : "https://www.dvb.org/metadata/css/css-ts.ct-1.1.1#",
    "title"       : "DVB Companion Screens and Streams - CSS-TS Control Timestamp message
representation schema",
    "description" : "",

    "oneOf" : [
        {
            "type"       : "object",
            "properties" : {
                "contentTime"             : { "$ref" : "css-core-1.1.1#/definitions/integerAsString"
},
                "timelineSpeedMultiplier" : { "type" : "number" }
            },
            "required" : [ "contentTime", "timelineSpeedMultiplier" ]
        },
        {
            "type"       : "object",
            "properties" : {
                "contentTime"             : { "type" : "null" },
                "timelineSpeedMultiplier" : { "type" : "null" }
            },
            "required" : [ "contentTime", "timelineSpeedMultiplier" ]
        }
    ],
    "properties" : {
        "wallClockTime"           : { "$ref" : "css-core-1.1.1#/definitions/integerAsString" },
        "contentTime"             : { },
        "timelineSpeedMultiplier" : { }
    },
```

```
    "required" : [ "wallClockTime", "contentTime", "timelineSpeedMultiplier" ],
    "additionalProperties" : true
}
```

A.1.7 Actual, Earliest and Latest Presentation Timestamp schema

This schema will validate a correctly formed JSON representation of Earliest and Latest Presentation Timestamp message:

```
{
    "$schema"     : "http://json-schema.org/draft-04/schema#",
    "id"          : "https://www.dvb.org/metadata/css/css-ts.apteptlpt-1.1.1#",
    "title"       : "DVB Companion Screens and Streams - CSS-TS Actual, Earliest and Latest
Presentation Timestamp message representation schema",
    "description" : "",

    "type" : "object",
    "properties" : {
        "actual" : {
            "type"       : "object",
            "properties" : {
                "contentTime"   : { "$ref" : "css-core-1.1.1#/definitions/integerAsString" },
                "wallClockTime" : { "$ref" : "css-core-1.1.1#/definitions/integerAsString" }
            },
            "required" : [ "contentTime", "wallClockTime" ],
            "additionalProperties" : true
        },
        "earliest" : {
            "type"       : "object",
            "properties" : {
                "contentTime"   : { "$ref" : "css-core-1.1.1#/definitions/integerAsString" },
                "wallClockTime" : {
                    "oneOf" : [
                        { "$ref" : "css-core-1.1.1#/definitions/integerAsString" },
                        { "enum" : [ "minusinfinity" ] }
                    ]
                }
            },
            "required" : [ "contentTime", "wallClockTime" ],
            "additionalProperties" : true
        },
        "latest" : {
            "type"       : "object",
            "properties" : {
                "contentTime"   : { "$ref" : "css-core-1.1.1#/definitions/integerAsString" },
                "wallClockTime" : {
                    "oneOf" : [
                        { "$ref" : "css-core-1.1.1#/definitions/integerAsString" },
                        { "enum" : [ "plusinfinity" ] }
                    ]
                }
            },
            "required" : [ "contentTime", "wallClockTime" ],
            "additionalProperties" : true
        }
    },
    "required" : [ "earliest", "latest" ],
    "additionalProperties" : true
}
```

A.1.8 Trigger Event Session Setup (TESS) schema

This schema will validate a correctly formed JSON representation of a setup-data message:

```
{
    "$schema"     : "http://json-schema.org/draft-04/schema#",
    "id"          : "https://www.dvb.org/metadata/css/css-te.tess-1.1.1#",
    "title"       : "DVB Companion Screens and Streams - CSS-TE Trigger Event Session Setup message
representation schema ",
    "description" : "",

    "type"       : "object",
    "properties" : {
        "contentIdStem" : { "$ref" : "css-core-1.1.1#/definitions/contentIdStem" },
```

```
        "private"       : { "$ref" : "css-core-1.1.1#/definitions/private" }
    },
    "required" : [ "contentIdStem" ],
    "additionalProperties" : true
}
```

A.1.9 Trigger Event Subscription Management (TESM) schema

This schema will validate a correctly formed JSON representation of a Trigger Event Subscription Management (TESM) message:

```
{
    "$schema"     : "http://json-schema.org/draft-04/schema#",
    "id"          : "https://www.dvb.org/metadata/css/css-te.tesm-1.1.1#",
    "title"       : "DVB Companion Screens and Streams - CSS-TE Trigger Event Subscription 
Management message representation schema",
    "description" : "",

    "type"       : "object",
    "properties" : {
        "triggerEvent" : { "$ref" : "css-core-1.1.1#/definitions/triggerEventLocator" },
        "subscribed"   : { "type" : "boolean" },
        "private"      : { "$ref" : "css-core-1.1.1#/definitions/private" }
    },
    "required" : [ "triggerEvent", "subscribed" ],
    "additionalProperties" : true
}
```

A.1.10 Trigger Event Notification (TEN) schema

This schema will validate a correctly formed JSON representation of a Trigger Event Notification (TEN) message:

```
{
    "$schema"     : "http://json-schema.org/draft-04/schema#",
    "id"          : "https://www.dvb.org/metadata/css/css-te.ten-1.1.1#",
    "title"       : "DVB Companion Screens and Streams - CSS-TE Trigger Event Notification message 
representation schema",
    "description" : "",

    "type"       : "object",
    "properties" : {
        "triggerEvent"     : { "$ref" : "css-core-1.1.1#/definitions/triggerEventLocator" },
        "triggerEventData" : {
            "oneOf" : [
                {
                    "type"    : "string",
                    "pattern" : "^([A-Za-z0-9-_]{4})*([A-Za-z0-9-_]{2}[A-Za-z0-9-_=]=)?$"
                },
                { "type" : "null" }
            ]
        },
        "presentationWallClockTime" : {
            "oneOf" : [
                { "$ref" : "css-core-1.1.1#/definitions/integerAsString" },
                { "type" : "null" }
            ]
        },
        "subscribed"           : { "type" : "boolean" },
        "triggerEventId"       : { "type" : "string" },
        "triggerEventDuration" : { "$ref" : "css-core-1.1.1#/definitions/integerMinimumZeroAsString" 
},
        "private"              : { "$ref" : "css-core-1.1.1#/definitions/private" }
    },
    "required" : [ "triggerEvent", "triggerEventData", "presentationWallClockTime", "subscribed" ],
    "additionalProperties" : true
}
```

A.2 Example JSON representation (informative)

A.2.1 General

The examples provided here are examples of correctly formed JSON representations of protocol message described in the present document.

A.2.2 Examples of Material Information

A.2.2.1 Example response from the MRS


```
{
    "type"            : "response",
    "version"         : "1.1",
    "rev"             : "20130815095215",
    "repollingInterval" : 30,
    "updateMaterial" : [
        {
            "url"      : "http://mrs.bbc.co.uk/material-updates/233a/1004/1044?20130815095215",
            "mimeType" : "application/json"
        },
        {
            "url"      : "ws://mrs.bbc.co.uk/material-updates/233a/1004/1044?20130815095215",
            "mimeType" : "application/json"
        }
    ],
    "updateTimelineSync" : [
        {
            "url"      : "http://mrs.bbc.co.uk/correlation-updates/233a/1004/1044?20130815095215",
            "mimeType" : "application/json-patch"
        },
        {
            "url"      : "ws://mrs.bbc.co.uk/correlation-updates/233a/1004/1044?20130815095215",
            "mimeType" : "application/json-patch"
        }
    ],
    "materials"       : {

        "1bced38994" : {
            "parents"       : [ ],
            "contentIdStem" : "dvb://233a.1004.1044;35f7~20130218T0830Z--PT00H45M",
            "ids" : [
                { "type" : "tag:bbc.co.uk/programmes/series",  "id" : "b00cg66y" },
                { "type" : "tag:bbc.co.uk/programmes/episode", "id" : "b00cr16g" }
            ],
            "timelineProperties" : {
                "unitsPerTick"   : 1,
                "unitsPerSecond" : 1000,
                "accuracy"       : 0.04
            }
        },
        "1bced39100" : {
            "parents" : [ "1bced38994" ],
            "ids" : [
                { "type" : "tag:bbc.co.uk/programmes/clips",  "id" : "d015tfya" }
            ],
            "timelineProperties" : {
                "unitsPerTick"   : 1,
                "unitsPerSecond" : 1000,
                "accuracy"       : 0.04
            },
            "private" : [
                {
                    "type"     : "tag:bbc.co.uk/programmes/clips/link-url",
                    "clip-url" : "http://www.bbc.co.uk/programmes/b1290532/"
                }
            ]
        },
        "1bced40562" : {
            "parents"       : [ ],
            "contentIdStem" : "dvb://233a.1004.1044;363a~20130218T0915Z--PT00H45M",
```

```
        "ids" : [
            { "type" : "tag:bbc.co.uk/programmes/series",  "id" : "b01bmj28" },
            { "type" : "tag:bbc.co.uk/programmes/episode", "id" : "b01bzrjp" }
        ],
        "timelineProperties" : {
            "unitsPerTick"   : 1,
            "unitsPerSecond" : 1000,
            "accuracy"       : 0.04
        },
        "triggerEventInfo" : {
            "contentIdStem" : "dvb://233a.1004.1044",
            "triggerEvents" : {
                "alert1"  : "urn:dvb:css:triggerevent:dsmcc:56:0",
                "askUser" : "urn:dvb:css:triggerevent:dsmcc:15:9"
            }
        }
    },
    "1bced92125" : {
        "parents"       : [ ],
        "contentIdStem" : "dvb://233a.1004.1044;3653~20130218T1000Z--PT01H00M",
        "ids" : [
            { "type" : "tag:bbc.co.uk/programmes/series",  "id" : "b01g5y69" },
            { "type" : "tag:bbc.co.uk/programmes/episode", "id" : "b01j0qtd" }
        ]
    }
],
"syncTimelineInformation" : [
    {
        "contentIdStem"    : "dvb://233a.1004.1044",
        "timelineSelector" : "urn:dvb:css:timeline:pts",
        "timelineProperties" : {
            "unitsPerTick"   : 1,
            "unitsPerSecond" : 90000,
            "accuracy"       : 3.0
        },
        "mappings" : [
            {
                "materialIndex"        : "1bced38994",
                "start"                : "2972394000",
                "end"                  : "3213144000",
                "correlationsChanging" : false,
                "correlations" : [
                    { "point" : "2972394000", "materialPoint" : "0" }
                ]
            },
            {
                "materialIndex"        : "1bced39100",
                "start"                : "3213144000",
                "end"                  : "3214944000",
                "correlationsChanging" : false,
                "correlations" : [
                    { "point" : "3213144000", "materialPoint" : "0" }
                ]
            },
            {
                "materialIndex"        : "1bced40562",
                "start"                : "3215844000",
                "end"                  : "3457404000",
                "correlationsChanging" : true,
                "correlations" : [
                    { "point" : "3215844000", "materialPoint" : "0" },
                    { "point" : "3220344000", "materialPoint" : "50004" },
                    { "point" : "3223944000", "materialPoint" : "90007" }
                ]
            },
            {
                "materialIndex"        : "1bced92125",
                "start"                : "3455874000",
                "end"                  : "3461814000",
                "correlationsChanging" : false,
                "correlations" : [
                    { "point" : "3455874000", "materialPoint" : "0" }
                ]
            }
        ]
    },
    {
```

```
            "contentIdStem"       : "dvb://233a.1004.1044",
            "leadupContentIdStem" : "dvb://233a.1004.1004",
            "timelineSelector"    : "urn:dvb:css:timeline:temi:8:150",
            "timelineProperties" : {
                "unitsPerTick"   : 1,
                "unitsPerSecond" : 25,
                "accuracy"       : 0.0
            },
            "mappings" : [
                {
                    "materialIndex"        : "1bced38994",
                    "start"                : "765650",
                    "end"                  : "832525",
                    "correlationsChanging" : false,
                    "correlations" : [
                        { "point" : "765650", "materialPoint" : "0" }
                    ]
                },
                {
                    "materialIndex"        : "1bced39100",
                    "start"                : "832525",
                    "end"                  : "833025",
                    "correlationsChanging" : false,
                    "correlations" : [
                        { "point" : "832525", "materialPoint" : "0" }
                    ]
                },
                {
                    "materialIndex"        : "1bced40562",
                    "start"                : "833275",
                    "end"                  : "900375",
                    "correlationsChanging" : true,
                    "correlations" : [
                        { "point" : "833275", "materialPoint" : "0" },
                        { "point" : "834525", "materialPoint" : "50004" },
                        { "point" : "835525", "materialPoint" : "90007" }
                    ]
                },
                {
                    "materialIndex"        : "1bced92125",
                    "start"                : "899950",
                    "end"                  : "901600",
                    "correlationsChanging" : false,
                    "correlations" : [
                        { "point" : "899950", "materialPoint" : "0" }
                    ]
                }
            ]
        }
    ]
}
```


A.2.2.2 Example Materials update response from the MRS


```
{
    "type"              : "update",
    "version"           : "1.1",
    "rev"               : "20130815100004",
    "updateVersionNo" : "1",
    "repollingInterval" : 30,
    "materials"         : {

        "1bced39100" : {
            "parents" : [ "1bced38994" ],
            "ids" : [
                { "type" : "tag:bbc.co.uk/programmes/clips",  "id" : "d015tfya" }
            ],
            "timelineProperties" : {
                "unitsPerTick"   : 1,
                "unitsPerSecond" : 1000,
                "accuracy"       : 0.04
            },
            "private" : [
                {
                    "type"     : "tag:bbc.co.uk/programmes/clips/link-url",
                    "clip-url" : "http://www.bbc.co.uk/programmes/b1290532/"
                }
```

```
            ]
        },
        "1bced40562" : {
            "parents"       : [ ],
            "contentIdStem" : "dvb://233a.1004.1044;363a~20130218T0915Z--PT00H45M",
            "ids" : [
                { "type" : "tag:bbc.co.uk/programmes/series",  "id" : "b01bmj28" },
                { "type" : "tag:bbc.co.uk/programmes/episode", "id" : "b01bzrjp" }
            ],
            "timelineProperties" : {
                "unitsPerTick"   : 1,
                "unitsPerSecond" : 1000,
                "accuracy"       : 0.04
            },
            "triggerEventInfo" : {
                "contentIdStem" : "dvb://233a.1004.1044",
                "triggerEvents" : {
                    "alert1"  : "urn:dvb:css:triggerevent:dsmcc:56:0",
                    "askUser" : "urn:dvb:css:triggerevent:dsmcc:15:9"
                }
            }
        },
        "1bced92125" : {
            "parents"       : [ ],
            "contentIdStem" : "dvb://233a.1004.1044;3653~20130218T1000Z--PT01H00M",
            "ids" : [
                { "type" : "tag:bbc.co.uk/programmes/series",  "id" : "b01g5y69" },
                { "type" : "tag:bbc.co.uk/programmes/episode", "id" : "b01j0qtd" }
            ]
        }
    ]
}
```


A.2.2.3 Example Synchronization Timeline Information update response from the MRS


```
{
    "type"            : "update",
    "version"         : "1.1",
    "rev"             : "20130815100315",
    "updateVersionNo" : "5",
    "patchData" : [
        {
            "op"    : "add",
            "path"  : "/syncTimelineInformation/0/mappings/2/correlation/-",
            "value" : { "point" : "3227544000", "materialPoint" : "130008" }
        }
    ]
}
```


A.2.3 Example of Content Identification and other Information (CII)


```
{
    "protocolVersion"    : "1.1",
    "mrsUrl"             : "http://css.bbc.co.uk/dvb/233A/mrs",
    "contentId"          : "dvb://233a.1004.1044;363a~20130218T0915Z--PT00H45M",
    "contentIdStatus"    : "partial",
    "presentationStatus" : "okay",
    "wcUrl"              : "udp://192.168.1.5:5800",
    "tsUrl"              : "ws://192.168.1.8:5815",
    "timelines" : [
        {
            "timelineSelector"   : "urn:dvb",
            "timelineProperties" : {
                "unitsPerTick"   : 5,
                "unitsPerSecond" : 10
            }
        }
    ]
}
```

A.2.4 Example of Setup Data

```
{
    "contentIdStem"    : "dvb://233a.1004.1044",
    "timelineSelector" : "urn:dvb:css:timeline:pts"
}
```

A.2.5 Example of Control Timestamp

```
{
    "contentTime"             : "834188",
    "wallClockTime"           : "116012000000",
    "timelineSpeedMultiplier" : 1.0
}
```

A.2.6 Example of Actual, Earliest and Latest Presentation Timestamp

```
{
    "actual" : {
        "contentTime"   : "834190",
        "wallClockTime" : "115992000000"
    },
    "earliest" : {
        "contentTime"   : "834190",
        "wallClockTime" : "115984000000"
    },
    "latest" : {
        "contentTime"   : "834190",
        "wallClockTime" : "plusinfinity"
    }
}
```

A.2.7 Example of Trigger Event Session Setup

```
{
    "contentIdStem" : "dvb://1234.5678"
}
```

A.2.8 Example of Trigger Event Subscription Management

```
{
    "triggerEvent" : "urn:dvb:css:triggerevent:dsmcc:56:0",
    "subscribed"   : true
}
```

A.2.9 Trigger Event Notification

```
{
    "triggerEvent"                : "urn:dvb:css:triggerevent:dsmcc:15:9",
    "triggerEventData"            : "SGVscCEgV2UgYXJlIHRyYXBwZWQgaW4gYSBzdGFuZGFyZHMgbWVldGluZyE=",
    "presentationWallClockTime" : "897698657643434",
    "subscribed"                  : true
}
```

Annex B (informative): Implementation guidelines for broadcasters

B.1 General

Annex B provides informative guidelines for broadcasters for implementing Timeline Synchronization. Whereas the present document specifies the interfaces between TV Device, Companion Screen Applications and Material Resolution Service, it has a deeper impact on broadcaster systems and broadcast distribution networks.

Clause B.2 discusses the use of Material Information.

Clause B.3 discusses how a timeline may be derived from a broadcast chain.

Clause B.4 introduces the concept of a "synchronization centre" to manage and equalize delays in the broadcast chain.

Clause B.5 suggests a solution for broadcasters how to deal with distribution networks that re-originate intrinsic timelines such as PTS.

Clause B.6 illustrates how the broadcaster can generate Correlation Timestamps for various use cases.

B.2 Use of Material Information and Material Resolution

B.2.1 General

Clause B.2.2 provides among others illustrations of Material Information for various use cases and suggests solution for handling potential ambiguities in timeline mappings.

Clause B.2.3 discusses the issues involved in determining the precise start and end times of programmes broadcast via a DVB broadcast service.

Clause B.2.4 explains how to handle wrapping of a Synchronization Timeline.

B.2.2 Material Information

B.2.2.1 General

The Material Information data model described in clause 5.5 is intended to be sufficiently flexible and generic to be able represent the editorial structure of a broadcast for most purposes and to facilitate broadcaster or CSA developer specific identifiers and other data to be conveyed without creating interoperability issues.

The data model provides a layer of abstraction that insulates a CSA from the specifics of a television delivery platform such as DVB broadcast or IPTV services. A CSA can concern itself with the editorial meaning of Materials signalled by Material Identifiers and private data and treat other data such as Timeline Selectors and Content Identifiers as opaque data to be matched or provided in interactions between the CSA and TV Device without need to examine their actual value and how it maps to broadcast platform specifics (e.g. how a content identifier string is constructed).

Material Information can be used to represent any editorially useful period on the Synchronization Timeline. The providers of Material Information can decide whether Materials represent the programmes, adverts, trails and segments being presented by the TV Device, or whether Materials represent Timed Content to be presented by a Companion Screen Application, or whether Materials are used for all of these purposes.

Figure B.2.2.1.1 illustrates a possible scenario where a Synchronization Timeline has multiple mappings to the Timelines of various Materials representing programmes, 2 adverts halfway through a programme and an editorial segment within a programme.

There are periods on the Synchronization Timeline that map to more than one Material Timeline simultaneously, such as sponsorship adverts that merge into the programme content and an editorial segment that is part of a programme.

The second programme has two parts due to an advert break and so has multiple mappings from the Synchronization Timeline.

There are periods on the Synchronization Timeline for which there are no mappings to Materials, and during these periods the corresponding position on any of the Materials' Timelines is not defined.
Not all Timelines for Materials start at zero, such as that for the 2nd advert.

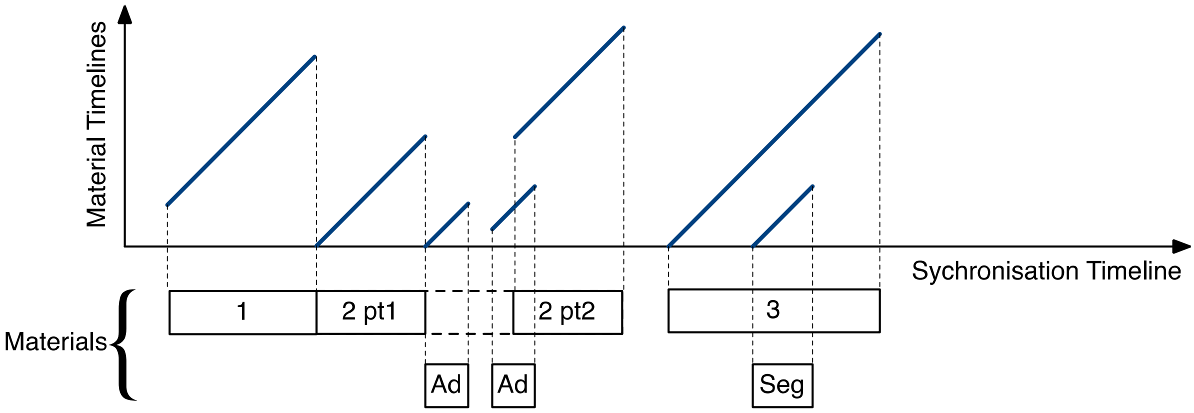


Figure B.2.2.1.1: Illustration of scenario for mappings from a Synchronization Timeline to Timelines of Materials

B.2.2.2 Hierarchical relationships

Material Information can indicate an editorial hierarchical relationship between Materials. This can be used, for example, to indicate to a CSA that it should present Timed Content relating to a sponsorship advert in a way that is visually framed within branding for the programme that is being sponsored. One Material represents the programme and another, indicated to be its child, represents the sponsorship advert.

Note however that if a parent Material is active, this does not necessarily indicate that a child Material is active. Programme 2 and the Adverts in figure B.2.2.1.1 illustrate this. The Material representing Programme 2 has two mappings from the Synchronization Timeline corresponding to part 1 and part 2 of the programme. During the gap between the two parts, the Timeline of the programme does not progress and so there are no mappings during this period when the Adverts will be shown.

B.2.2.3 Use of Material Identifiers

Material Identifiers are globally scoped by a URI defining its type. Because of this, multiple independent parties, such as advertisers, channel operators or content creators can utilize their own identifier schemes, or choose to cooperate in the use of common schemes as they deem appropriate. Companion Screen Applications can distinguish between identifiers that they are intended to recognize and identifiers that have been allocated by another independent organization.

Because multiple Material Identifiers may be associated with a Material, this permits for multiple independent parties to all separately allocate IDs to a particular Material and to have them all delivered in Material Information.

There is no requirement regarding the degree of uniqueness that a Material Identifier possesses. A Material Identifier can therefore be used to identify a class of Materials; in effect providing type information.

For example an advertiser may agree with a broadcaster to allow the advertiser's Companion Screen Application to synchronize with the presentation of their advert. The broadcaster also has a Companion Screen Application that separately recognizes that the TV Device is currently showing an interstitial and therefore ceases to display information relating to the television programme. The Material Information provided by the MRS includes both a Material Identifier provided by the broadcaster and another provided by the advertiser. Each has a separate type URI. The partial JSON fragment below illustrates this:

```
"materials" : {
    "material_012843" : {
        "ids" : {
            { "type" : "tag:advertiser.com/tv/ad_ids", "id" : "55783158" },
            { "type" : "tag:broadcaster.com/material", "id" : "interstitial" }
        }
        ...
    },
    ...
}
```

B.2.2.4 Use of private data

Like Material Identifiers, private data is globally scoped by a URI. This URI identifies the nature of the private data and is intended to give the Companion Screen Application sufficient information to know how to interpret the contents of the private data object (if it recognizes this particular type).

B.2.2.5 Ambiguities in timeline mappings

B.2.2.5.1 Example of Handling wrapping of a Synchronization Timeline

Time Values on the Synchronization Timeline are 32 bit unsigned numbers and are therefore limited to between 0 and 2^{32}-1 (the upper bound is 2^{32}). The scale for both the Synchronization Timeline and Material Timeline are assumed to be the same.

Shortly after a programme begins, the Synchronization Timeline wraps. There is then a short advert break in the programme before it finishes. A single Material represents the programme. Mapping 1 corresponds to the first part of the programme before the break and Mapping 2 corresponds to the second part. Mapping 1 straddles the point at which the Synchronization Timeline wraps and therefore has to be split into two parts - mapping 1a and mapping 1b. This is illustrated in figure B.2.2.5.1.1.

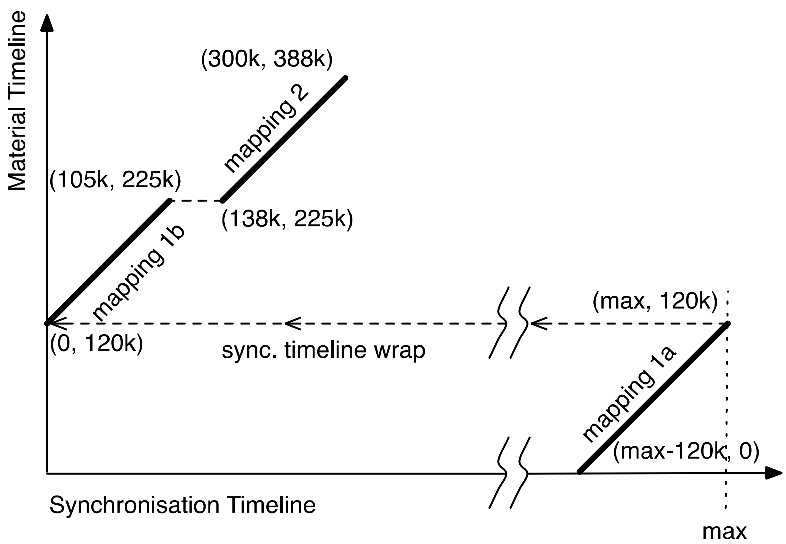


Figure B.2.2.5.1.1: Example of mappings that correctly handle wrapping

Table B.2.2.5.1.1 shows the Time Values that characterize each mapping For each mapping, the pair of values $start_{material}$ and $start_{sync}$ represent the initial Correlation Timestamp that would apply at the start of that mapping interval.

Table B.2.2.5.1.1: Example of mappings that correctly handle wrapping

Mapping	$start_{sync}$	end_{sync}	$start_{material}$
1a	2^{32} - 120 000	2^{32}	0
1b	0	105 000	120 000
2	138 000	300 000	225 000

B.2.2.5.2 A point on the Synchronization Timeline maps to multiple points on the Material Timeline

For a given CI, if Time Values on the Synchronization Timeline are re-used for different points in the content, then this leads to an ambiguity that cannot be resolved practically by a CSA.

This situation may also occur on very small timescales (typically fractions of a second) where the Material Timeline is not genlocked to the Synchronization Timeline. This is unavoidable (except by genlocking). However for larger timescales this situation should be avoided if possible.

If it is safe to assume that the content is being broadcast and consumed live then it may be possible to resolve the ambiguity by reference to external sources of information and comparisons against a real time clock; however for any form of time shifted viewing (delayed, recording playback or video on demand) this is not possible.

B.2.2.5.3 Ambiguities due to long duration Materials

Limitations on the range of Time Values that can be conveyed for a Timeline (before it wraps) also limit the maximum practical duration of a programme represented by a single Material and identified by a single CI.

For example: if Synchronization Timeline Time Values wrap after 12 hours then for a 13 hour long programme there will be some Time Values on the Synchronization Timeline that map to two different times on the Material Timeline. For a 1 hour period at the start and end of the programme, it will be ambiguous as to whether the TV Device is presenting the first hour or the last hour of the programme. This is illustrated in figure B.2.2.5.3.1.

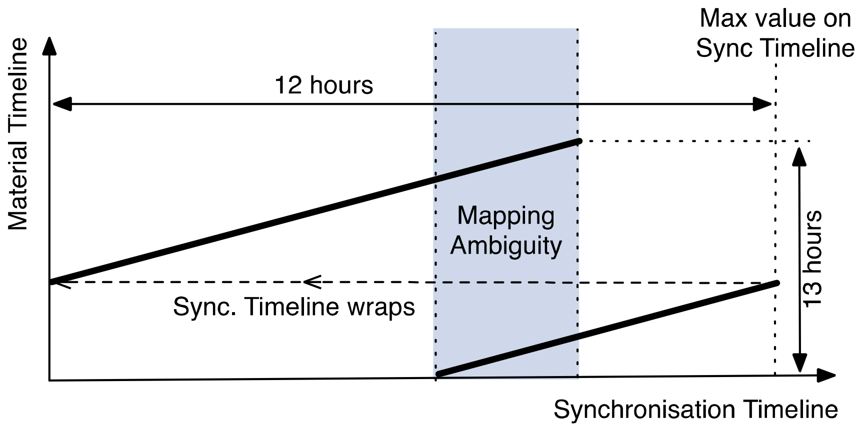


Figure B.2.2.5.3.1: Example of wrapping leading to ambiguity in Timeline Mappings for a long duration programme

Ambiguities are still possible even if the duration of a Material representing a programme is less than the range expressible on the Synchronization Timeline. A programme that is 11 hours long but which is interrupted by 10 minute of adverts every hour spans an effective duration of 12 hours and 50 minutes. If this is represented by a single Material and identified by a single CI and if a Synchronization Timeline wraps over a 12 hour interval then there will be ambiguity for the first and last 1 hour and 50 minutes of the programme.

B.2.2.5.4 Ambiguities due to Synchronization Timeline discontinuities

A Synchronization Timeline (such as a PTS) may feature a discontinuity such as a jump forwards or backwards in Time Values. If this occurs during a period when a TV Device is reporting a particular CI there is potential for ambiguity if some Time Values on the Synchronization Timeline are revisited after the discontinuity before the CI changes. This is illustrated in figure B.2.2.5.4.1.

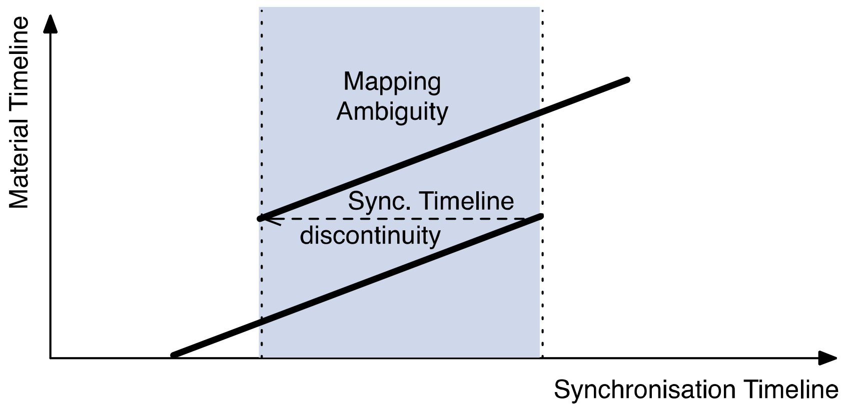


Figure B.2.2.5.4.1: Example of ambiguity due to Synchronization Timeline discontinuity

An ambiguity due to discontinuity can also occur at programme boundaries where a Synchronization Timeline (e.g. PTS) discontinuity is intentional at a programme junction, but it is not precisely co-timed with the change in CI (which is derived from non-precisely timed DVB SI data).

If it is important for a CSA to be able to synchronize precisely at points close to a programme junction, it is recommended to avoid timeline discontinuities unless a change in CI can be guaranteed to be sufficiently precisely co-timed.

B.2.3 Precisely identifying the start and end of programmes on a DVB broadcast service

The Content Identifier for a DVB broadcast service includes information about the present event as signalled in DVB EIT. This changes at the junction between one programme and the next. However in practice it is impractical for a broadcaster to achieve frame accurate alignment of EIT signalling with respect to the start of a programme. Therefore, at times close to the programme junction, there will be periods for which the CI reported by the TV Device does not accurately reflect the programme content currently being presented to the user.

In the explanation that follows, it is assumed (for simplicity) that there is a one to one correspondence between events signalled in DVB EIT, programmes and Materials.

This issue can be solved if the broadcast includes an accurate Synchronization Timeline and the MRS provides Material information not only for the event indicated in the CI, but also Material Information for the previous and next events in the schedule. This needs to include mapping information for all Materials in the Material Information. The CSA can then check the current time value of the Synchronization Timeline reported by the TV Device against the mappings. The CSA can then unambiguously determine which Material is currently being presented. This determination process performed by the CSA is described in clause 5.5.8.

EXAMPLE: The TV Device reports a CI of "B" to the CSA. The CSA sends this to the MRS and receives Material Information describing Materials "a", "b" and "c" and a Synchronization Timeline with mappings to all three Materials. If the CSA were to match the CI stems for each Material, it would conclude that Material "b" is currently being presented. However, when it performs Timeline Synchronization, the mappings reveal that it is actually Material "a" that is being presented. EIT signalling has changed early from "A" to "B" before the programme represented by Material "a" has finished.

In the more complex case, a DVB event may correspond to multiple Materials. For example, Materials may represent individual adverts or trailers between programmes as well as the programme itself. In these situations all Materials that will occur during the period of time encompassed by the present DVB events should be returned as well as Materials encompassed by the previous and next DVB events that occur close to the programme junctions. The same determination process can be used by the CSA to determine which Material or Materials correspond to what is currently being presented by the TV Device.

B.2.4 Content Identification and regular wrapping of the Synchronization Timeline

If a Synchronization Timeline is anticipated to regularly wrap, the Content Identifier will need to change at least as frequently as the wrapping occurs, otherwise it will not be possible for an MRS to unambiguously determine which cycle of the Synchronization Timeline the TV Device is presenting Timed Content for.

A possible example would be a Synchronization Timeline derived from timecode derived from time of day, such as that described in clause B.3. Such a Timeline will wrap every 24 hours. The Synchronization Timeline combined with knowledge of which broadcast service it belongs to is insufficient to disambiguate which Material the user is watching if they could be watching content that is delayed by 24 hours or more, such as a PVR recording.

For a DVB broadcast service, the Content Identifier not only identifies the broadcast service being viewed, but also provides the scheduled start date and time of the present DVB EIT event for that service. This will provide sufficient information to enable an MRS to disambiguate which 24 hour period of the Synchronization Timeline is being observed by the CSA when it provided the CI to the MRS. The MRS can then return the relevant Materials, Timeline mappings and Correlations Timestamps.

For DVB EIT, event ids are allocated from a small finite pool and so may also regularly be recycled. The event id is therefore not necessarily sufficient to provide the disambiguation. This is why the event scheduled start date and time is included in the CI.

B.3 Derivation of timelines from the broadcast chain

B.3.1 General

Figure B.3.1.1 illustrates one possible way that a broadcaster can generate and insert a timeline into a broadcast MPEG transport stream and enable CSAs to resolve it using Material Information with the potential for frame accurate synchronization. The approach here embeds the timeline in the video stream immediately after playout and preserves it, in some form, throughout the broadcast chain. There will be many equally valid variations on the architecture described here.

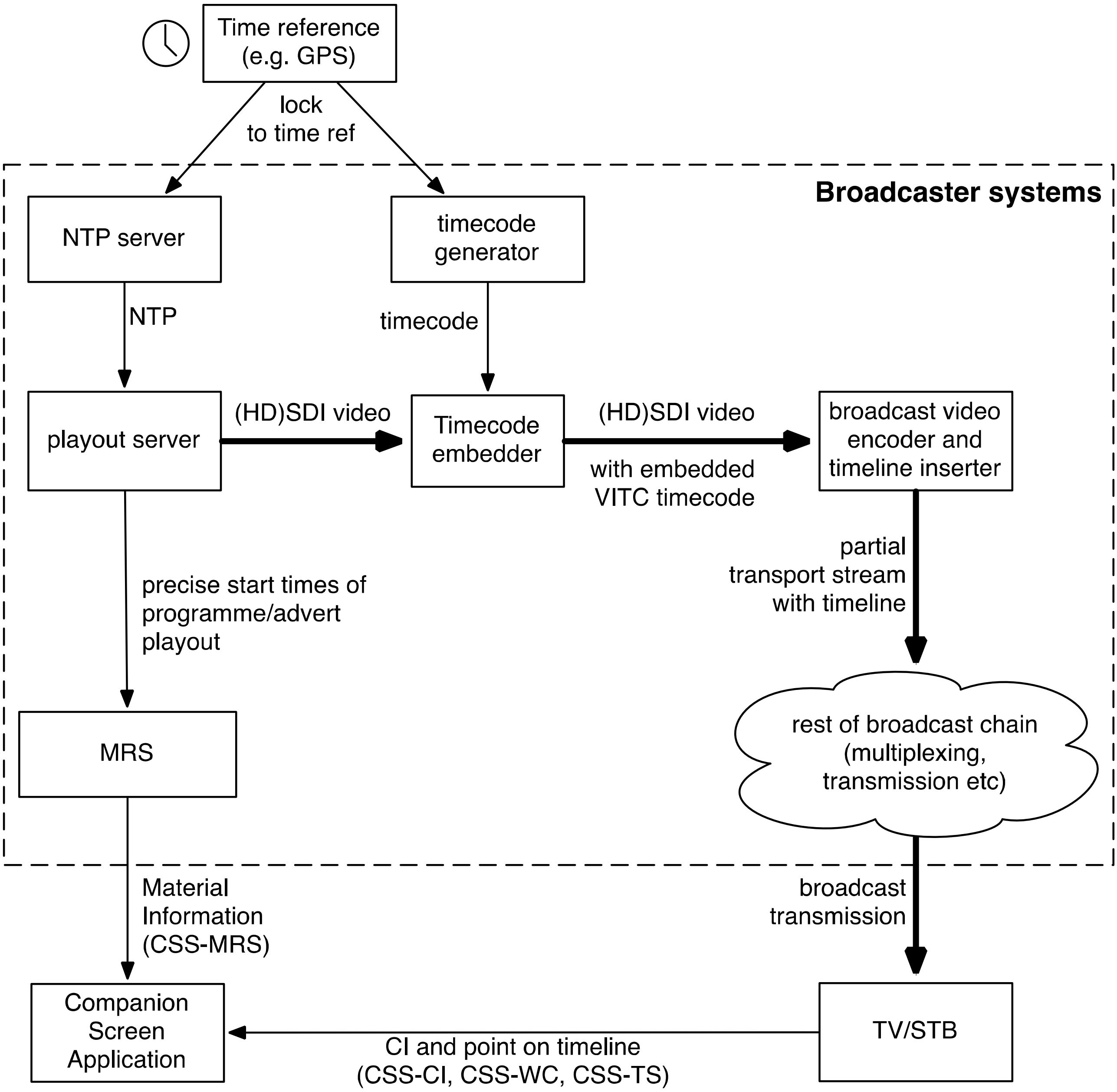


Figure B.3.1.1: Example broadcast chain for frame accurate synchronization

In this scenario it is assumed that the broadcast service is editorially run according to a schedule and that playout of programmes, adverts and other content is performed by a playout server that follows that schedule.

B.3.2 Time synchronization at playout

The real time clock of the playout server and the frame rate of playout are locked to a master time reference, such as a GPS time receiver device, via NTP over a private low latency IP network. A video timecode generator is also locked to the same reference and drives a timecode inserter. The uncompressed video output from the playout server is immediately fed into the timecode inserter with no additional sources of delay in-between. Uncompressed video will emerge from the timecode inserter with embedded VITC timecode. The hours, minutes, seconds and fractions of a second of the GPS time correspond directly to the hours, minutes, seconds and frames of the timecode.

B.3.3 Coding and multiplexing

The broadcast video encoder is assumed to have built-in support for reading VITC timecode from the incoming uncompressed video and using it to generate timeline signalling that is embedded in the encoded video.

It is assumed that the timeline signalling format used for final delivery embeds the timeline in the transport stream alongside, or within, the transport stream packets carrying the media components of the service. It is also assumed that this timeline signalling will be preserved by downstream multiplexing processes between insertion and final arrival of the transport stream in the TV Device. This includes preserving the relationship between the timeline signalling and the PTS of the media component(s).

If the timeline signalling is in the format required (e.g. the signalling format described in clause 11.3) for final delivery, then no further processing is required.

It may be the case that an encoder is not available that outputs timeline signalling in the required format. In these situations, signalling can be generated by the encoder in some other supported format such as an SEI Picture Timing message in an H.264 video elementary stream (see clauses D.1.2 and D.2.2 of Recommendation ITU-T H.264 [i.6]). A separate device can then be placed downstream of the broadcast encoder that decodes the embedded timeline and converts it to the required signalling format. For the format described in clause 11.3, the device could inject it into the existing transport stream packets of the stream carrying the media components and repacketize it as required. Alternatively, for the format described in clause 11.3, it could generate a separate stream of transport stream packets carrying a PES stream containing empty payloads, but with the timeline signalling embedded. The separate stream is then multiplexed into the service.

B.3.4 Generation of Material Information

The playout server also provides a feed of precise (frame-accurate) times at which it begins playout of each programme. This is expressed relative to its internal real time clock. A programme playout time should therefore match the embedded timecode on the first frame of video of that programme. It will therefore also match the timeline value associated with that video frame in the transport stream that arrives at the TV Device.

The feed is used to generate the mappings and Correlation Timestamps served by the MRS. For example:

1) GPS originated UTC time is used as a timing reference. Timecode is generated counting at 25 fps where time zero is at midnight UTC.

2) This is signalled as a timeline in the broadcast stream timeline ticking at 25 ticks per second. A timeline tick value is calculated from any timecode value as $\big((hours \times 60 + minutes) \times 60 + seconds\big) \times 25 + frames$.

3) A programme begins playout at precisely 9 hours, 15 minutes, 1 secs and 0 milliseconds (timecode 09:15:01:00) after midnight UTC starting with frame 0 of the programme content.

4) The Material Information mapping will have a start time of $\big((h9 \times 60 + 15) \times 60 + 1\big) \times 25 + 0 =$ 832 525 ticks on the Synchronization Timeline, and a Correlation Timestamp correlating 832 525 to frame 0. For example:

```
{
    "materialIndex"        : "1bced39100",
    "start"                : "832525",
    "end"                  : "833025",
    "correlationsChanging" : false,
    "correlations"         : [
        { "point" : "832525", "materialPoint" : "0" }
    ]
```

B.3.5 Handling non integer frame rates and dropped frame timecode

For non-integer frame rates such as 29,97 fps or 59,94 fps, dropped-frame timecode is typically used. In dropped-frame timecode, the frame count skips 1 or 2 frames periodically at defined absolute points in the count.

However, the tick values of the Synchronization Timeline conveyed by the TV Device to the CSA is supposed to be a precise representation of the presentation time of the Timed Content. For example: a timeline for a 29,97 fps video stream with a tick count reported every frame will increment by precisely 1 001/30 000 secs for each frame. This can be accurately represented by 1 001 ticks of the Synchronization Timeline per frame at a tick rate of 30 000 ticks per second.

A timeline tick count that is being derived from dropped-frame timecode therefore needs to be derived with care. A precise understanding is needed of the algorithm used to decide the points in the timecode frame count where skips occur.

B.4 Managing delay throughout distribution network

There are several reasons why a broadcaster may want to manage and equalize delays throughout the distribution network(s).

- DVB broadcast streams have typically much lower latency than OTT streams.
- Delays are different in different network segments, e.g. due to transcoding.
- Media-stream buffer capacity is limited in TV devices.

Especially for live broadcasts with live companion streams, it is important that media streams arrive at similar times such that there are no buffer overflows or underflows at the user side. If the TV Device is not equipped with a media buffer, then the broadcaster should make sure that the "master" broadcast stream is the most laggard of the group of associated streams. Equalizing delays between head-ends can also be beneficial to social TV use cases (out of scope for the present document), where friends or groups of people communicate with each other while watching the same content at different locations, a.k.a. "watching apart together".

Figure B.4.1 sketches an architecture to achieve the required delay management and equalization.

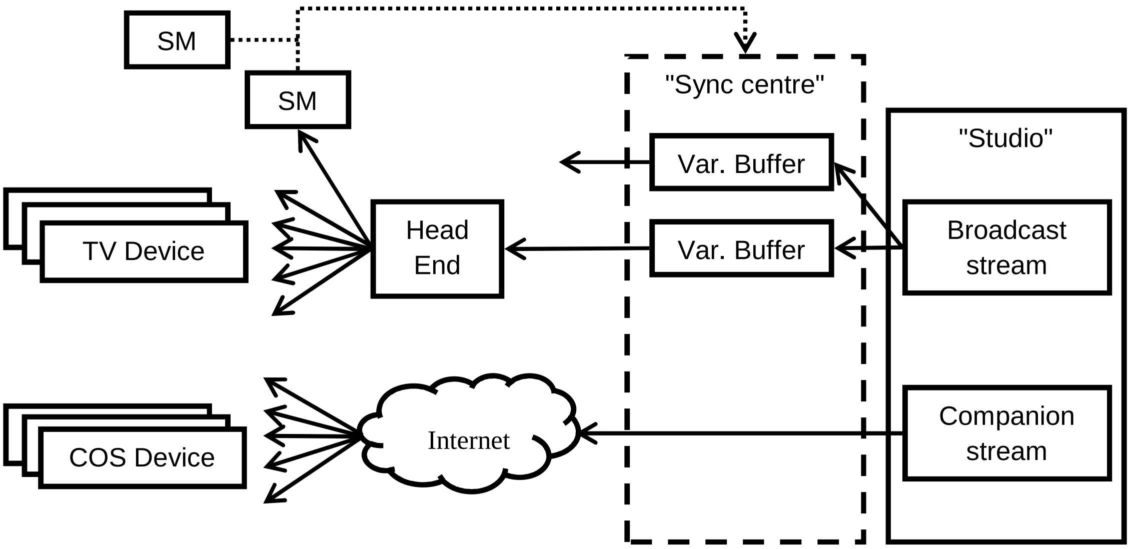


Figure B.4.1: Architecture for delay management and equalization

The architecture has a broadcaster studio that provides broadcast and companion streams. The broadcaster is assumed to have a "synchronization centre" where synchronization is managed. Stream monitors (SM) are placed at strategic points in the distribution networks to monitor the playout timing of the different network segments, typically at a head end or at a special TV device. The reports from the stream monitors are used at the synchronization centre to control variable-delay buffers per network segment and per channel, resulting in a coarse delay equalization of the different streams. The fine synchronization will happen in/between the TV device and COS device(s) in the home.

B.5 Managing multiple content timelines

The existence of multiple timelines will be a fact of life for a broadcaster, until all its distribution networks support immutable timelines such as MPEG TEMI. Distribution networks typically re-multiplex, transcode and even re-originate broadcast streams. In the process, the broadcast timeline may be stripped and a new broadcast timeline is created. This means at least an unknown offset between PTS values of the original stream and the new stream(s). Also, there may be subtle variations between the tick rates of the original-stream PCR clock and the new-stream(s) PCR clock(s). The broadcaster will need to handle the situation of having different PCR and therefore different PTS in different distribution-network segments.

Figure B.5.1 sketches an architecture to manage multiple content timelines.

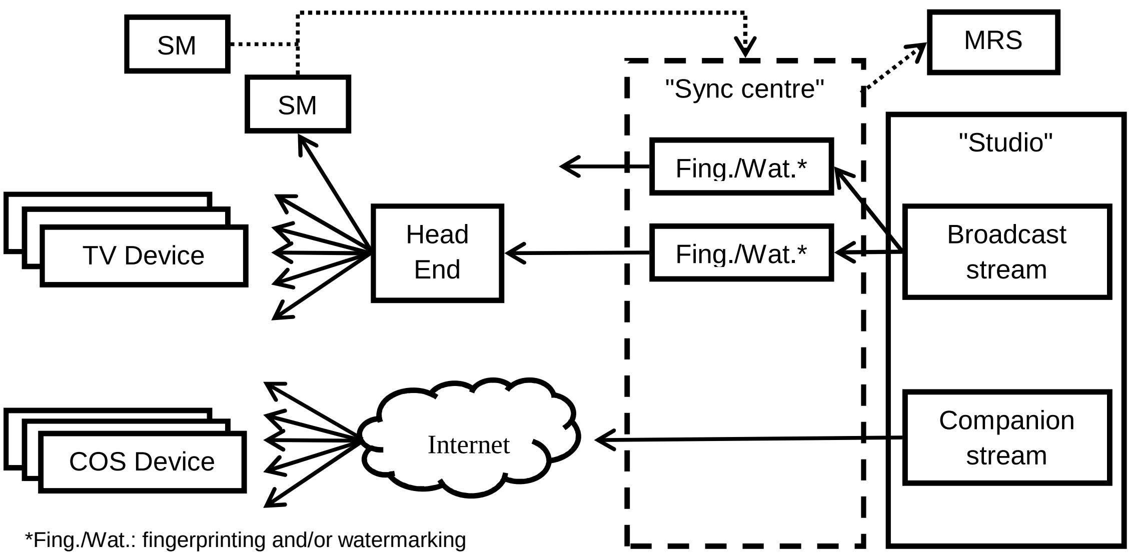


Figure B.5.1: Architecture for managing multiple timelines

The architecture has a broadcaster studio that provides broadcast and companion streams. The broadcaster is assumed to have a "synchronization centre" where synchronization is managed. Stream monitors (SM) are placed at strategic points in the distribution networks to monitor the relationship between the playout timing and PTS, typically at a head end or at a special TV device. The broadcasters may fingerprint and/or watermark the broadcast content such that the stream monitors can correlated the measured timeline (e.g. PTS) values with a specific point in the content, identified by a fingerprint or watermark. The result is passed to the Material Resolution Service (MRS), such that the MRS can provide TV devices material information (MI) expressed in the appropriate broadcast timeline.

B.6 Generating Correlation Timestamps

The present document requires Broadcasters to generate Correlation Timestamps between Timelines, see also clause 5.4. Timed Content can be created in several ways.

- Parallel sources, for example:
  - Recording of lip-sync audio and video streams.
  - Live at-the-scene commentary.

  - Video streams from multiple cameras.

- Reactive sources, for example:
  - Commentary or audio translation of a received A/V stream.
  - Transcoding to a different display resolution, bitrate or frame rate.

Figure B.6.1 sketches an architecture to generate and collect the Correlation Timestamps.

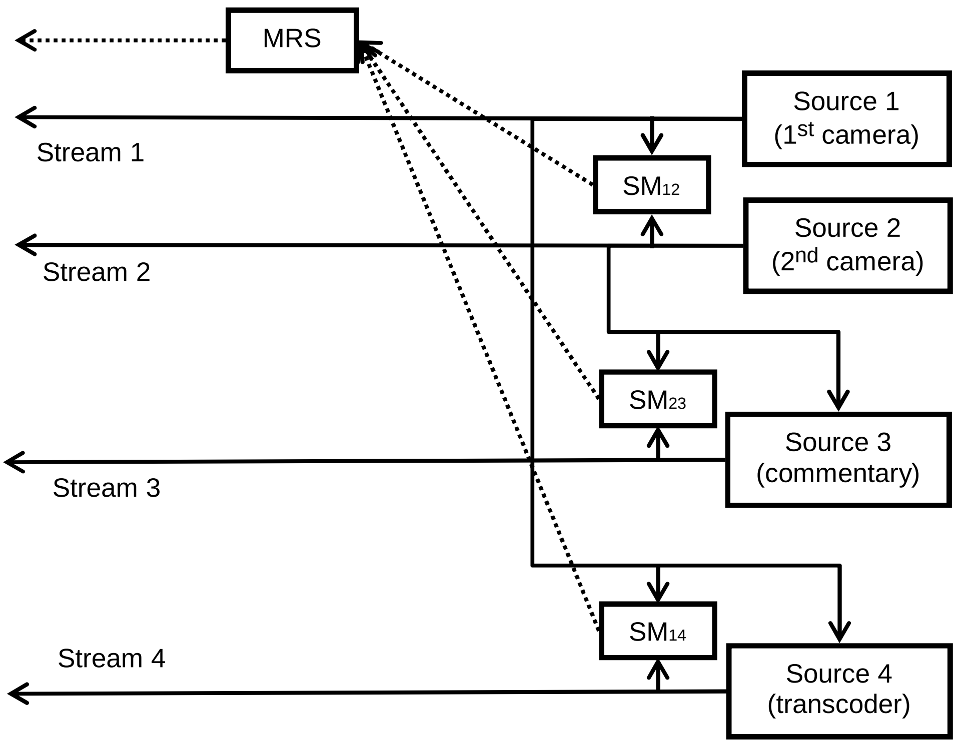


Figure B.6.1: Generating Correlation Timestamps

The architecture has two parallel sources and two reactive sources:

- Sources 1 and 2 are two cameras that are recording a live event in parallel. Stream Monitor SM_{12} measures tuples of Time Values at the Timelines of the two sources, and reports these to the MRS. If there are any delay differences between the two sources, then these are corrected in an automated or editorial way, such that the tuple of Time Values correspond to the same point in time of the two streams.

- Source 3 is a reactive source that provides commentary on Steam 2. Stream Monitor SM_{23} measures tuples of Time Values at the Timelines of Source 2 and Source 3, and reports these to the MRS. The tuples are compensated for the delay in presentation of Stream 2 and the generation of Stream 3.

- Source 4 is a reactive source as well, that provides a transcoded version of Steam 1. Stream Monitor SM_{14} measures tuples of Time Values at the Timelines of Source 1 and Source 4, and reports these to the MRS. The tuples are compensated for the time that transcoding process takes.

The MRS receives the tuples of Time Values. The CIS function of the MRS uses these tuples to calculate Correlation Timestamps for all stream Timelines against a common Synchronization Timeline, which is typically the Timeline of the stream that is used as broadcast stream. However, it can also be another Timeline, e.g. the broadcaster wall-clock that provides a stream-independent Synchronization Timeline.

B.7 Timelines in MPEG DASH streams

B.7.1 Handling xlink references

The MPEG DASH specification [16] only requires support for xlink with actuate set to onLoad within an MPEG DASH MPD and therefore the TV Device resolves all xlink references when the MPD is loaded. The TV Device will know the durations of all Periods in the presentation and can therefore always calculate the time that has elapsed between the start of any base Period and the point currently being presented by the TV Device.

However the CSA and MRS do not know what Periods xlink references have resolved to. The duration of a resolved Period can depend on what Period an xlink reference has resolved to. In this situation the CSA and MRS are unable to know what point in the Timed Content corresponds to a given Time Value on the Timeline. This is illustrated by an example in figure B.7.1.1.

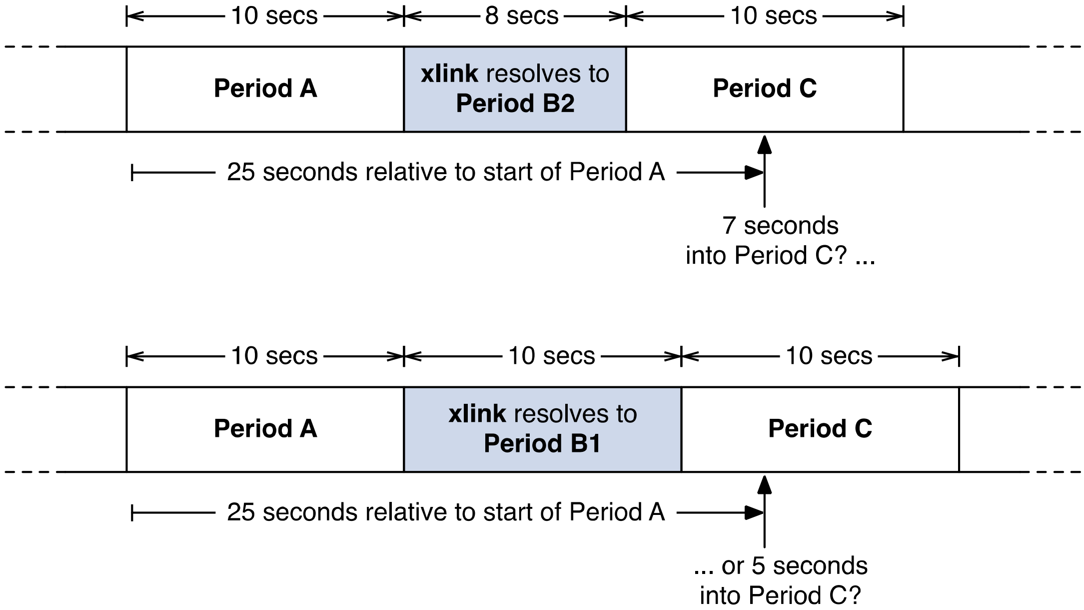


Figure B.7.1.1: Example of Timeline ambiguity from the perspective of the MRS and CSA caused by not knowing what an xlink reference has resolved to

If a broadcaster intends to use xlink references that may resolve to Periods of different durations, then the broadcaster can avoid this ambiguity by providing a different Timeline Selector for use for the Periods before and after the xlink reference.

By making use of the fact that the CI incorporates the Period ID, the broadcast can allocate Period IDs and set the CI stem associated with a given Timeline Selector such that the CI stem will only match the CI for the set of consecutive Periods between occurrences of xlink references. In effect, a new Timeline is being used after any occurrence of an xlink reference that is relative to the first Period after that xlink reference. This is illustrated by example in figure B.7.1.2.

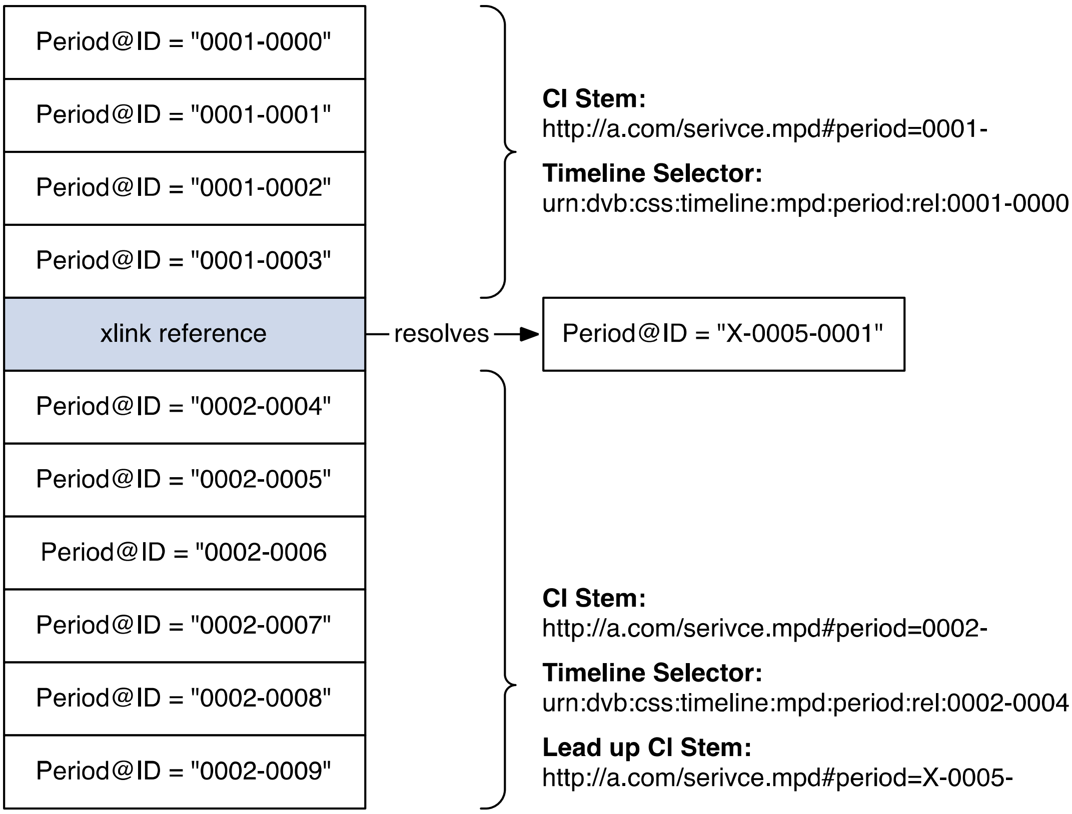


Figure B.7.1.2: Defining Period IDs such that a different Timeline Selector is used for the Periods before or after the use of an xlink reference

For the Period resolved by xlink reference, the broadcaster can also choose to use a different Timeline or continue using Timeline that was being used for the previous Periods. Which is appropriate to choose depends on the needs of the CSA and the design of the broadcaster playout systems.

In this situation, it may be desirable for a CSA to know which Timeline Selector is anticipated to be used next and thus maintain apparently seamless synchronization across the changeover from one Timeline to the next. The CSA can pre-emptively establish a second connection to the CSS-TS service endpoint and specify the anticipated Timeline Selector. The CSA can then immediately begin receiving Control Timestamps as soon as the TV Device moves into presenting the first Period for which the specified Timeline is available. To provide the CSA with this knowledge, the MRS can include a leadInContentIdStem property in the Synchronization Timeline information it provides. This CI stem could, for example, match the Period@ID attribute of the final Period before the changeover.

B.7.2 Timeline wrapping

A given Period relative MPEG DASH Timeline cannot wrap, and can reliably convey a Timeline that spans thousands of years even at comparatively high tick rates such as 90 000 ticks per second or 300 000 000 ticks per second.

However if a broadcaster wishes to notionally reset the Timeline back to zero, the broadcaster has to do so by instead periodically switching to a new Timeline that is relative to a more recent Period.

The broadcaster can use the same approaches as described in clause B.7.1 to limit the applicability of a Timeline Selector to a group of consecutive Periods and to then indicate that a new Timeline Selector should be used for the next group of consecutive Periods.

Annex C (informative): Implementation guidelines for TV Devices and Companion Screen Applications

C.1 General

Annex C provides informative guidelines for manufacturers of TV Devices and developers of Companion Screen Applications for implementing Timeline Synchronization.

Clause C.2 provides examples of the content identifier (CI), both correctly formed and malformed examples.

Clause C.3 suggests solutions to improve the user experience for cases where media streams are added to or removed from a group of synchronized media streams.

Clause C.4 provides generic calculation examples to provide correct measurement and handling of different types of timestamps, focussing on a simplified model of a set-top-box.

Clause C.5 provides calculation examples for a TV Device that cannot vary the presentation time of content.

Clause C.6 provides calculation examples for a TV Device that can vary the presentation time of content.

Clause C.7 provides calculation examples for the SC elementary function in the Companion Screen Application.

Clause C.8 provides guidance on the implementation of the Wall Clock.

Clause C.9 provides guidance on the use of the Status presentation information.

Clause C.10 provides calculation guidance and examples for the CSA and TV Device on the use of the calculationWallClockTime of the TEN message.

C.2 CI examples

C.2.1 Examples of correctly formed CIs

This clause provides some examples of CIs that have been correctly formed in the manner defined in clause 5.2.

```
dvb://233a.1004.1044;35f7~20131004T0930Z--PT01H00M
```

The above CI conveys the following information:

- Original network id: 9018
- Transport stream id: 4100
- Service id: 4164
- Event id: 13815
- Start: 09h30m UTC on 4th October 2013
- Duration: 1 hour 0 minutes

```
dvb://233a.1004.1044;35f7;0080~20131004T0930Z--PT01H00M
```

The above CI conveys the following information:

- Original network id: 9018
- Transport stream id: 4100

- Service id: 4164
- Event id: 13815
- TVA_id: 128
- Start: 09h30m UTC on 4th October 2013
- Duration: 1 hour 0 minutes

```
dvb://233a.1004.1044;35f7~20131004T0930Z--PT01H00M?eit_anc=6230306372313667
```

The above CI conveys the following information:

- Original network id: 9018
- Transport stream id: 4100
- Service id: 4164
- Event id: 13815
- Start: 09h30m UTC on 4th October 2013
- Duration: 1 hour 0 minutes
- CI private data in EIT: "b00cr16g"

```
dvb://233a.1004.1044;35f7~20131004T0930Z--PT01H00M?eit_anc=6230306372313667&nit_anc=495254
```

The above CI conveys the following information:

- Original network id: 9018
- Transport stream id: 4100
- Service id: 4164
- Event id: 13815
- Start: 09h30m UTC on 4th October 2013
- Duration: 1 hour 0 minutes
- CI private data in NIT: "IRT"
- CI private data in EIT: "b00cr16g"

```
dvb://233a.1004.1044;35f7~20131004T0930Z--PT01H00M?ep_crid=fp.bbc.co.uk%2Fa72x6pl
```

The above CI conveys the following information:

- Original network id: 9018
- Transport stream id: 4100
- Service id: 4164
- Event_id: 13815
- Start: 09h30m UTC on 4th October 2013
- Duration: 1 hour 0 minutes
- TV Anytime CRID: crid:// fp.bbc.co.uk/a72x6pl

```
dvb://'bbcone.bbc.co.uk';35f7~20131004T0930Z--PT01H00M
```

The above CI conveys the following information:

- Service: "bbcone"
- Service provider: "bbc.co.uk"
- Event_id: 13815
- Start: 09h30m UTC on 4th October 2013
- Duration: 1 hour 0 minutes

```
http://dash.example.com/content/mpds/test.mpd#period=Period42
```

The above CI conveys the following information:

- The MPD for the content was retrieved from http://dash.example.com/content/mpds/test.mpd
- The Period that the content refers to is Period42

C.2.2 Examples of malformed CIs

This clause provides some examples of CIs that are not formed in manner defined in clause 5.2 and which are therefore not considered to be valid CIs.

```
dvb://233a.1004.1044;35f7
```

The above CI is malformed because it is missing start time and duration.

```
dvb://233A.1004.126;35F7~20131004T0930Z--PT01H00M
```

The above CI is malformed because it uses upper case in the original network ID and event ID.

```
dvb://233a.1004.0126;35f7~20131004T093015Z--PT01H00M20S?nit_anc=00AF13&ep_crid=239F14
```

The above CI is malformed because start time and duration should not contain seconds.

```
dvb://233a.1004.1044;35f7~20131004T0930Z--PT01H00M?nit_anc=495254&
```

The above CI is malformed because the trailing ampersand '&' character should not be present.

```
dvb://233a.1004.126;35f7~20131004T0930Z--PT01H00M
```

The above CI is malformed because the service id is missing leading '0' digits to pad it to 4 digits length.

```
dvb://233a.1004.0126;35f7;~20131004T0930Z--PT01H00M
```

The above CI is malformed because it is missing a TVA_id, or because the second ";" should not be present.

```
dvb://233a.1004.0126;35f7~20131004T0930Z--PT01H00M?
```

The above CI is malformed because the query separator question mark "?" character should not be present if there is no query part.

```
dvb://233a.1004.0126;35f7~20131004T0930Z--PT01H00M?nit_anc=00AF13&ep_crid=239F14
```

The above CI is malformed because the key-value pairs in the query part occur in the wrong order.

```
dvb://132.185.129.195;35f7~20131004T0930Z--PT01H00M
```

The above CI is malformed because it contains an IPv4 address instead of either a DVB triplet (original network id, transport stream id, service id) or a fully qualified domain name identifying the service and the service provider.

```
dvb://bbcone.bbc.co.uk;35f7~20131004T0930Z--PT01H00M
```

The above CI is malformed because it the textual service identifier "bbcone.bbc.co.uk" is not enclosed within single-quote "'" characters.

```
http://dash.example.com/content/mpds/test.mpd?t=65728
```

The above CI is malformed because it does not use the '#' character for the URI Media fragment, does not include the period parameter, and includes other parameters, in this case the t parameter.

C.3 Handling dynamics of media synchronization by the Synchronization Client

Media synchronization has its own dynamics, where users choose to add and remove broadcast streams and companion streams to a synchronized group of media streams. This clause provides some suggestions for the behaviour of a Synchronization Client (SC) to improve the user experience in dynamic situations where there could be discontinuous in the content presentation. The following cases are distinguished:

- New stream is behind the currently synchronized streams.
- New stream is ahead of the currently synchronized streams.
- Removed stream is the most laggard of the currently synchronized streams.

If a new stream is behind the currently synchronized streams, then the SCs for the other streams need to be synchronized against this new most laggard stream, as signalled by the MSAS. The simplest option is to freeze the presentation of currently synchronized streams by those SCs until they are in sync with the new stream. However, this would interrupt the flow of watching for the user if the timing difference is of the order of multiple seconds or more. An alternative is to rewind the currently synchronized streams by the SCs to the point that they are synchronized. As a result, the user would see again part of a previously-seen scene again, which may be less disturbing than a freeze. A more advanced option would be to slow down the presentation speed of the currently synchronized streams by the SCs, and gradually bring them in sync with the new stream, while processing audio to scale it in time without affecting its pitch. Informal tests have shown that playout speed variations of up to 25 % are often not noticeable, and depending on the content, rate variations up to 50 % are sometimes acceptable [i.4].

If a new stream is ahead of the currently synchronized streams, then the SC of the new stream needs to buffer the stream to the point that it is in sync with the other streams, as signalled by the MSAS. This buffering would be perceived by the user as annoying waiting time for the new stream to appear, similar to a slow channel change. A solution can be found in technologies where the client retrieves earlier parts of the content from a cache node in the network via unicast until the point where the other streams have caught up. Examples of such technologies are IPTV retransmission (RET) (see clause F.3.2 of [5]), IPTV Fast Channel Change (FCC) and dedicated content delivery network (CDN) [i.5] technologies.

Figure C.3.1 outlines an architecture for achieving such instant synchronization.

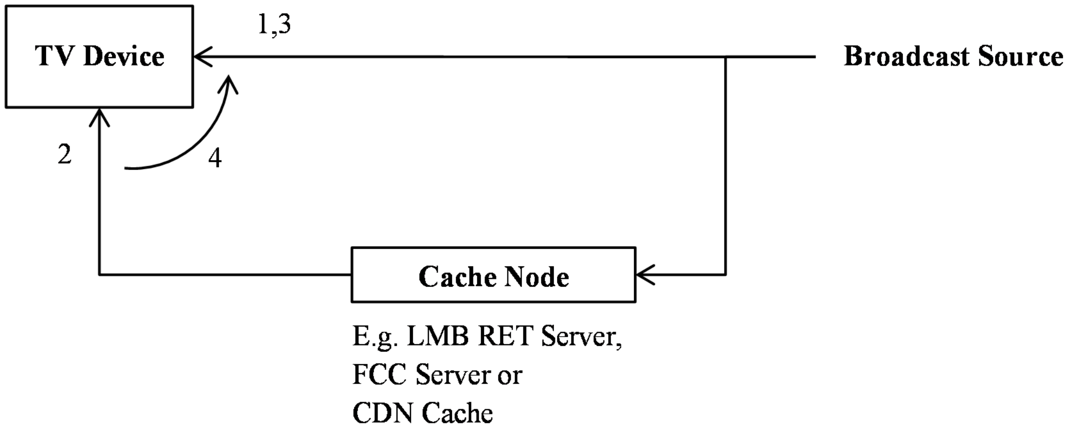


Figure C.3.1: Using a cache node for instant synchronization

1) The TV Device starts receiving a media stream from the broadcast source in synchronized mode with other device(s). The TV Device detects that it is ahead of the other(s). Instead of just waiting until the others have caught up, it obtains the URL of the Cache Node, e.g. as a RetransmissionCacheDescriptor parameter in the AIT table from the broadcast source, or the URL of MPD if the Cache Node uses adaptive streaming (e.g. MPEG DASH).

2) The TV Device retrieves earlier parts of the media stream from the Cache Node and presents that to the user.

3) The TV Device simultaneously buffers the media stream that is directly received from the broadcast source.

4) The TV Device playout catches up with the start of the broadcast stream that it has in its buffers. It switches its presentation to the media stream from the broadcast source and it stops retrieving from the Cache Node.

If a removed stream was the most laggard of a set of synchronized streams, then its removal would result in another stream becoming the most laggard. A simple MSAS implementation would provide Control Timestamps that follows the timeline of this new most-laggard steam, resulting in a timeline skip. The remaining SCs can respond to this by abruptly skipping ahead in the content, which may cause confusion with the user. A more advanced option for the SC would be to speed up the playout speed of the currently synchronized streams using adaptive media playout (AMP), and gradually bring them in sync with the current most laggard stream. Alternatively, the MSAS could prevent a skip by following the timeline of the new most-laggard stream with an offset, resulting in a presentation that is more delayed than necessarily needed.

C.4 Example calculations: Reference point for timestamping

C.4.1 General

Figure C.4.1.1 provides a simplified model of a Set-Top Box (STB) and a TV screen. This simplified model is used for calculation examples on Timestamps, using the reference point for timestamping specified in clause 5.6.2.

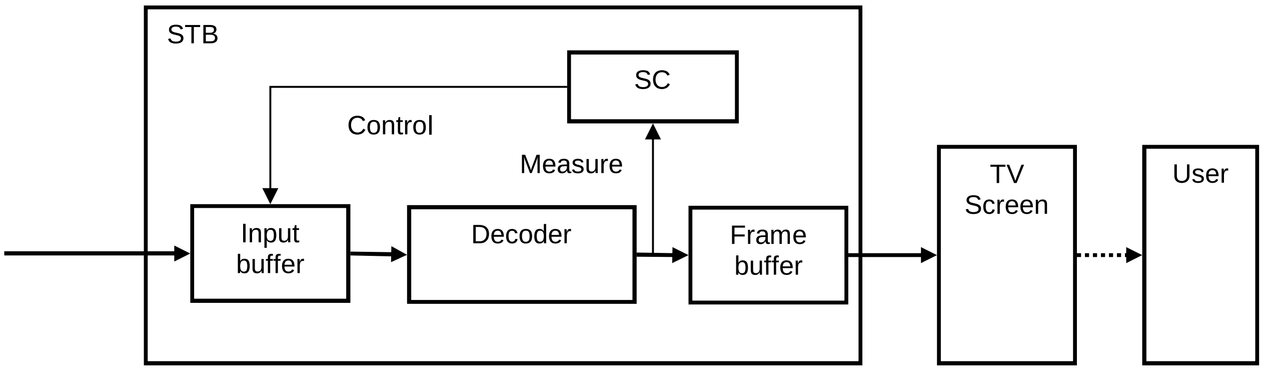


Figure C.4.1.1: Simplified model of a set-top box

In this simplified model, the SC performs media synchronization control via the input buffer as content takes significantly less buffer space in the encoded domain. Measurements are performed at the output of the decoder. The STB has a frame buffer after the decoder into which decoded video frames are assembled. Decoded frames are read serially from the frame buffer and relayed via an HDMI 2.0 [i.9] connection to a TV screen where they emerge as light. The frame buffer and TV screen both introduce delays between decoded video frames emerging from the decoder and emerging from the TV screen as light.

C.4.2 Actual Presentation Timestamp

This clause provides an example calculation of an Actual Presentation Timestamp. In this example, the decoder outputs a frame with a content time $t_{contentTimeline}$ = 5 233 342 (tick) at wall-clock time 49 813,654 (sec). The SC knows (e.g. through measurement) that the identified frame will take 413 ms to pass the frame buffer. It determines (e.g. by querying the HDMI 2.0 interface [i.9]) that the screen adds 153 ms more delay.

The actual Wall-Clock time at which the identified media sample is played out to the user is then calculated as follows:

Measured wall-clock time:	49 813,654	(sec)
Plus frame-buffer delay:	+0,413	(sec)
Plus screen delay:	+0,153	(sec)
-- +		
Reported wall-clock time: 49 814,220 (sec)		

The following values are assumed for tick rates of the Timelines and the Correlation Timestamp / Timeline Mapping.

$UnitsPerTick_{contentTimeline}$	= 1	(unit/tick)
$UnitsPerSecond_{contentTimeline}$	= 90 000	(unit/sec)
$UnitsPerTick_{synchronizationTimeline}$	= 1 001	(unit/tick)
$UnitsPerSecond_{synchronizationTimeline}$	= 24 000	(unit/sec)
Correlation Timestamp ($c_{contentTimeline}$,$c_{synchronizationTimeline}$)	= (4 490 561; 1 285)	(tick,tick)

Using the equation from clause 5.4:

$$t_y = c_y + (t_x - c_x)\frac{units\_per\_sec_y \cdot units\_per\_tick_x}{units\_per\_tick_y \cdot units\_per\_sec_x}$$

with x=contentTimeline and y=synchronization Timeline results in the following calculation:

$t_{synchronizationTimeline}$= 1 285 + (5 233 342 - 4 490 561) × (24 000 × 1)/(1 001 × 90 000) = 1 482; 877 056 277 056 (tick)

This value is rounded to the nearest integer, which is **1 483**.

So the SC sends to the MSAS the following Actual Presentation Timestamp.

```
{
    "contentTime"   : "1483",
    "wallClockTime" : "49814220000000"
}
```

Note that the rounding of contentTime to the nearest integer can be compensated by adjusting the wallClockTime, if greater accuracy is desirable.

C.4.3 Earliest Presentation Timestamp

This clause provides an example calculation of an Earliest Presentation Timestamp. All values are the same as in the previous example of clause C.4.2. In this example, the SC knows (e.g. through its previous instructions) that input buffer is delaying the input stream by 920 ms for media synchronization.

The earliest Wall-Clock time at which the identified media sample could be played out to the user is then calculated as follows:

Measured wall-clock time:	49 813,654	(sec)
Plus frame-buffer delay:	+0,413	(sec)
Plus screen delay:	+0,153	(sec)
Minus SC-added media sync delay:	-0,920	(sec)
-- +		
Reported wall-clock time:	49 813,300	(sec)

The SC sends to the MSAS the following Earliest Presentation Timestamp.


```
{
    "contentTime"   : "1483",
    "wallClockTime" : "49813300000000"
}
```


C.4.4 Latest Presentation Timestamp

This clause provides an example calculation of a Latest Presentation Timestamp. All values are the same as in the previous examples of clauses C.4.2 and C.4.3. In this example, the size of the input buffer is such that it can buffer at least 12 154 ms of the Timed Content. This value may be the result of a worst-case calculation, assuming a maximum bitrate.

The latest Wall-Clock time at which the identified media sample could be played out to the user is then calculated as follows:

Measured wall-clock time:	49 813,654	(sec)
Plus frame-buffer delay:	+413	(sec)
Plus screen delay:	+153	(sec)
Minus SC-added media sync delay:	-920	(sec)
Plus buffer size	+12 154	(sec)
-- +		
Reported wall-clock time:	49 825,454	(sec)

The SC sends to the MSAS the following Latest Presentation Timestamp.


```
{
    "contentTime"   : "1483",
    "wallClockTime" : "49850454000000"
}
```


C.4.5 Control Timestamp

This clause provides an example calculation for executing a Control Timestamp. All values are the same as in the previous examples of clauses C.4.2 and C4.3. In this example, the SC receives from the MSAS the following Control Timestamp.


```
{
    "contentTime"   : "1731",
    "wallClockTime" : "49814165000000"
}
```

The calculation uses the unrounded values of the Earliest Presentation Timestamp, see clause C.4.2, the values from the Control Timestamp, and the unitsPerTick and unitsPerSecond of the Synchronization Timeline.

t_{CTearl}	= 1 482; 877 056 277 056	(tick)
t_{WCearl}	= 49 813,300	(sec)
t_{CTcont}	= 1 487	(ticks)
t_{WCcont}	= 49 814,721	(sec)
$unitsPerTick_{syncTL}$	= 1 001	(unit/tick)
$unitsPerSecond_{syncTL}$	= 24 000	(unit/sec)

The total required synchronization delay is calculated with this equation:

$$\delta_{delay} = (t_{WCcont} - t_{WCearl}) + (t_{CTearl} - t_{CTcont}) \frac{units\_per\_tick_{syncTL}}{units\_per\_sec_{syncTL}}$$

Resulting into the following calculation:

δ_{delay} = (49 814,721 - 49 813,300) + (1 482; 877 056 277 056 - 1 487) × 1 001 / 24 000 = 1,249038889 (sec)

This means that the synchronization delay at input buffer should be set to 1 249 ms. As this value is larger than zero and smaller than 12 154 ms (see clause C 7.3), the STB can do this. As the current input buffer value is 920 ms (see clause C.7.2), this means that the STB should increase the delay by 1 249 - 920 = 329 ms.

C.5 Example calculations: Where TV Device cannot vary the presentation time of content

C.5.1 General

A TV Device may have no ability to vary the moment at which it presents content. This is the simplest scenario. Clause C.6 describes the alternative scenario where a TV Device can vary the presentation time of content.

Clauses C.5.2 and C.5.3 describe and provide examples of the calculation processes that could take place in a TV Device, from the perspective of both the SC and MSAS elementary functions that it implements.

C.5.2 SC elementary function in the TV device

C.5.2.1 Calculation Explanation

The example calculations take place in the context of a TV Device receiving a DVB broadcast service with a timeline for it signalled in the broadcast using TEMI.

The SC elementary function of the TV Device is observing the timing of (measuring) the received broadcast stream at a point after received broadcast input buffering and decoding, but before decoded video frame buffering, and display by the screen. This is illustrated in figure C.5.2.1.1.

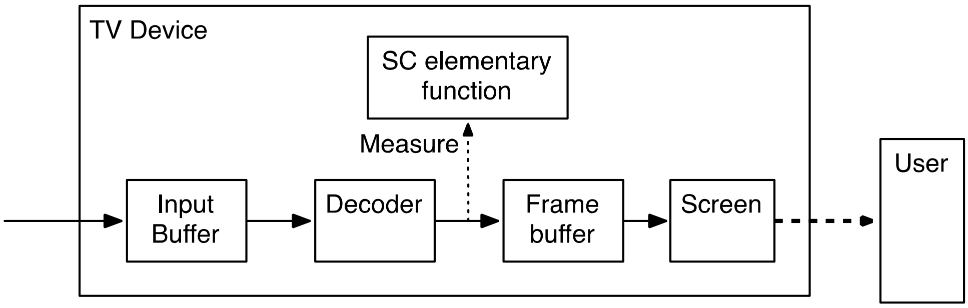


Figure C.5.2.1.1: Model of TV Synchronization Client measurement process

Measurement is of the decode time of the media and is made relative to the Wall Clock. For a DVB broadcast this is in terms of the Presentation Timestamp (PTS) of the decoded video frame.

NOTE: For alternative content delivery mechanisms, the analogous timestamp, such as Composition Time for ISOBMFF, is appropriate.

The SC function of the TV device also observes the signalled TEMI timeline and its relationship to the decode time (PTS). From this information and knowledge of the delays between the point of measurement and the display of content to the user it is possible to calculate the Actual, Earliest and Latest Presentation Timestamps to be sent to the MSAS elementary function. Because the TV Device has no ability to vary the presentation time of content, Control Timestamps received back from the MSAS will be ignored.

The sequence just described is illustrated in figure C.5.2.1.2 and is now demonstrated with an example of the calculations involved.

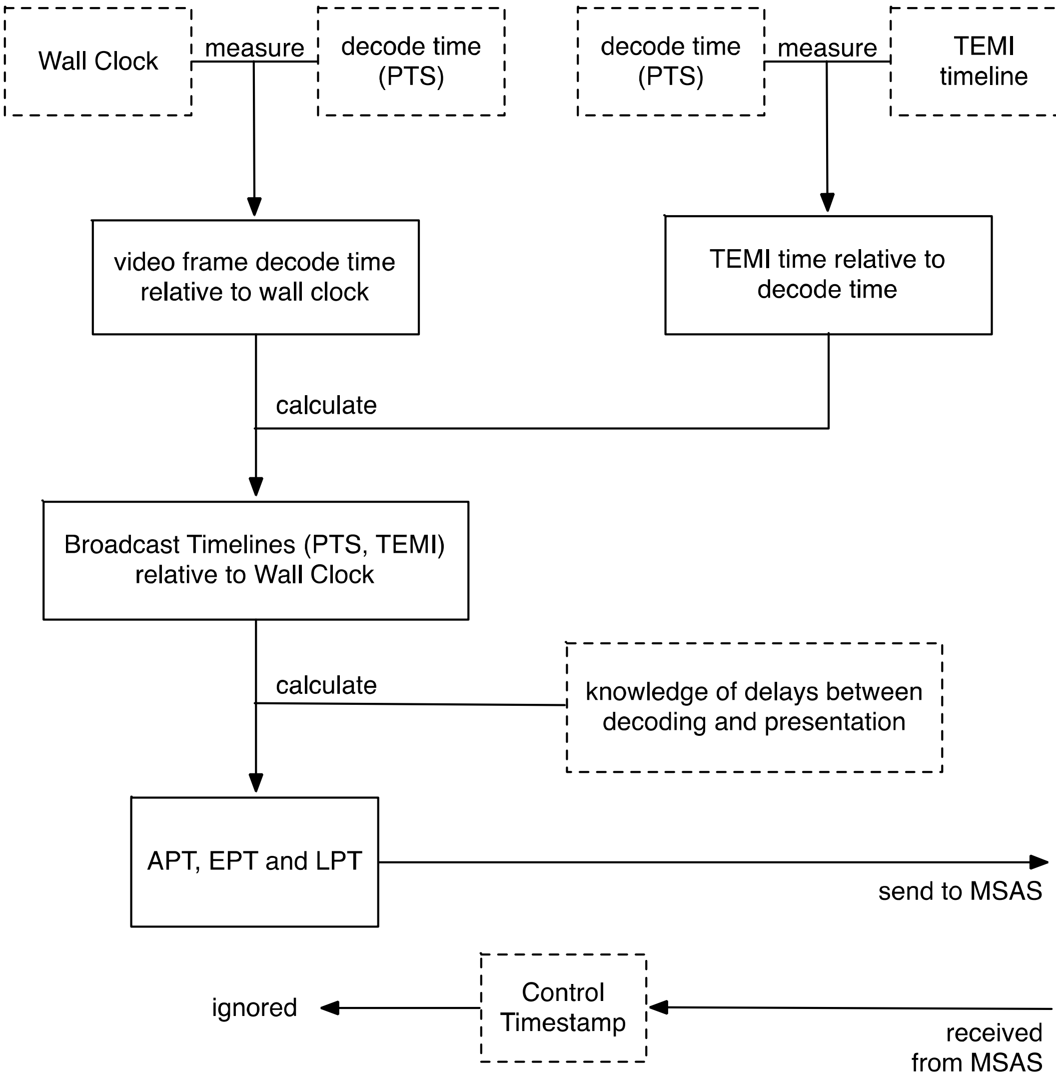


Figure C.5.2.1.2: Example overview of calculation processes for a TV SC

When a frame emerges from the video decoder, the following measurements are taken:

- m_{wall}: measurement of the Wall Clock measured in seconds.
- m_{decode}: measurement of the decode time (e.g. PTS) of the video frame in integer ticks.
- r_{decode} = (unitsPerSecond$_{decode}$ / unitsPerTick$_{decode}$): the tick rate (ticks per second) of the decode time measurement (e.g. 90 000 / 1 for PTS).

The DVB broadcast also contains an additional timeline conveyed as a TEMI timeline with its own tick rate and time values. TEMI timeline values are associated with PTS values derived from the same PCR clock as is used for the PTS for the audio and video. Therefore when a TEMI timeline descriptor is received and processed, the following measurements are taken:

- t_{TEMI}: the integer tick value of the TEMI timeline.
- t_{decode}: the value of the decode time (PTS) corresponding correlating with the timeline tick.
- r_{TEMI} = (unitsPerSecond$_{TEMI}$ / unitsPerTick$_{TEMI}$): the tick rate (ticks per second) of the timeline.

The video decode time can now be converted from being in terms of PTS to being in terms of the TEMI timeline:

$$m_{TSAP} = \text{round}\left((m_{decode} - t_{decode}) \frac{r_{TSAP}}{r_{decode}} + t_{TSAP} \right) m_{TSAPEMI} = \text{round}\left((m_{decode} - t_{decode}) \frac{r_{TSAPEMI}}{r_{decode}} + t_{TEMISAP} \right)$$

To preserve numerical precision, the calculation can be reformulated in terms of unitsPerTick and unitsPerSecond, as shown below, and ensuring that the numerator and denominator of the division operation are fully evaluated before performing the division:

$$m_{TSAPEMI} = \text{round}\left(\frac{(m_{decode} - t_{decode}) units\_per\_tick_{decode} units\_per\_sec_{TSAPEMI}}{units\_per\_sec_{decode} units\_per\_tick_{TSAPEMI}} + t_{TSAPEMI} \right)$$

The Actual, Earliest and Latest Presentation Timestamps to be reported to the MSAS will take the same value (because the TV Device has no ability to vary the time of presentation). The reference point for the APT, EPT and LPT (see clause 5.7.2) are at the point of light emission from the screen. So to calculate them the following also needs to be known:

- F: the frame buffer delay.
- S: the screen delay.

If the Synchronization Timeline used in communication with the MSAS is PTS, then APT, EPT and LPT will take the following values:

$$APT_{PTS} = EPT_{PTS} = LPT_{PTS} = (m_{decode}, m_{wall} + F + S)$$

If the Synchronization Timeline used in communication with the MSAS is TEMI, then APT, EPT and LPT will take the following values:

$$APT_{TSAPEMI} = EPT_{TTSAEMIP} = LPT_{TSAPEMI} = (m_{TEMISAP}, m_{wall} + F + S)$$

The SC elementary function of the TV Device will ignore any Control Timestamp it receives from the MSAS (again because the TV Device has no ability to vary the time of presentation).

C.5.2.2 Calculation Example

A TV Device is receiving a DVB broadcast. The most recently decoded video frame of the service has a PTS decode time of 15 682. The service also has a TEMI timeline with tick rate of 24 000 (23,97 fps with 1 001 ticks per frame) where the most recent timeline descriptor is associated with a PTS value of 8 173 and had tick value 2 304 302 (2 302 frames).

- m_{decode} = 15 682
- r_{decode} = 90 000 / 1
- t_{TEMI} = 2 304 302
- t_{decode} = 8 173
- r_{TEMI} = 24 000 / 1

Then:

$$m_{TEMISAP} = \text{round}\left((15\,682 - 8\,173) \frac{1}{90\,000} \frac{24\,000}{1} + 2\,304\,302 \right)$$

$$= \text{round}\left(\left((15\,682 - 8\,173) \times 1 \times 24\,000 \right) \div (90\,000 \times 1) + 2\,304\,302 \right)$$
$$= 2\,306\,304$$

The measurement of m_{decode} was made at wall clock time 33 300,280 secs. The SC elementary function of the TV Device estimates its Frame buffer delay as 0,5 secs and its screen delay as 0,1 secs.

- m_{wall} = 33 300,280
- F = 0,5
- S = 0,1

The Actual, Earliest and Latest Presentation Timestamps to be reported to the MSAS are therefore:

$$APT_{PTS} = EPT_{PTS} = LPT_{PTS} = (15\,862;\ 33\,300{,}280 + 0{,}5 + 0{,}1) = (15\,862;\ 33\,300{,}88)$$

$$APT_{TSEMIAP} = EPT_{TSAPEMI} = LPT_{TSAPEMI} = (2\,306\,304;\ 33\,300{,}88)$$

C.5.3 MSAS elementary function in the TV device

C.5.3.1 Calculation Explanation

An MSAS receives Earliest and Latest Presentation Timestamps from all Synchronization Clients including the SC elementary function of the TV Device. It then calculates a Control Timestamp and disseminates it back to all SCs as a recommendation of the timing of presentation.

The choice of policy applied by the MSAS to guide its calculation is an implementation detail of an MSAS. Given an MSAS elementary function that knows that the SC elementary function of the TV Device cannot vary its timing of presentation; such an MSAS may choose to adopt a policy of directing all SCs to match the presentation time of the SC elementary function of the TV Device. This is illustrated in figure C.5.3.1.1.

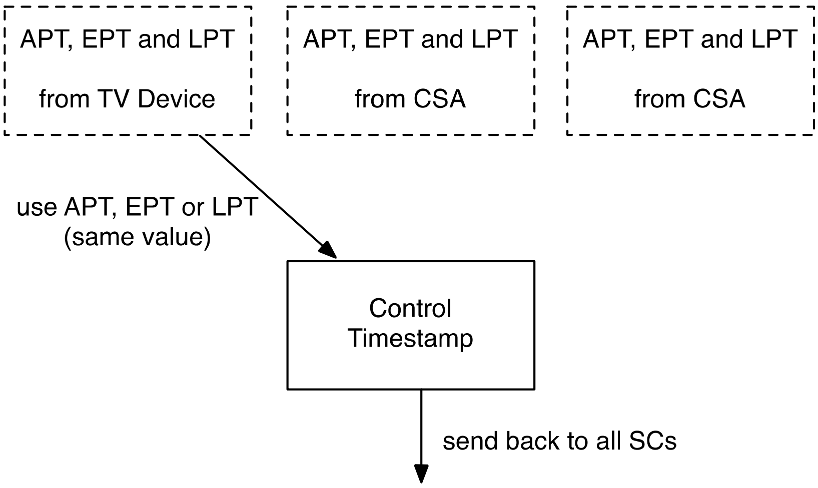


Figure C.5.3.1.1: Example overview of Calculation Process for a TV Device MSAS

Given this policy, the MSAS will ignore the EPT and LPT received from all SCs except that received from the SC elementary function of the TV Device. The MSAS will therefore take the APT, EPT or LPT most recently received from the TV Device and send it to all SCs (including the TV Device). It does not matter whether the APT, EPT or LPT is used because the TV has not ability to vary the timing of presentation and therefore they will have the same value.

C.5.3.2 Calculation Example

An MSAS is coordinating synchronization between 3 Synchronization Clients: SC{1}, SC{2} and SC{3}. SC{1} is the SC elementary function of the TV Device. Each SC{n} has recently reported an Actual Presentation Timestamp APT{n}, an Earliest Presentation Timestamp EPT{n} and a Latest Presentation Timestamp LPT{n}, using a Synchronization Timeline with a tick rate of 25 ticks per second.

- APT{1} = (1 005; 115 821,30)
- EPT{1} = (1 005; 115 820,50)

- LPT{1} = (1 005; 115 820,50)
- APT{2} = (1 002; 115 820,85)
- EPT{2} = (1 002; 115 820,70)
- LPT{2} = (1 002; 115 823,00)
- APT{3} = (1 003; 115 820,40)
- EPT{3} = (1 003; 115 818,00)
- LPT{3} = (1 003; 115 821,30)
- unitsPerSecond = 25
- unitsPerTick = 1

The Control Timestamp sent back to all SCs will equal APT{1}, EPT{1} or LPT{1}:

$$Control\_timestamp = (1\ 005;\ 115\ 820{,}5)$$

C.6 Example calculations: Where TV Device can vary the presentation time of content

C.6.1 General

A TV Device may be able to vary the moment at which it presents content. This is a more complex scenario. Clause C.5 describes the simpler alternative where a TV Device cannot vary the presentation time of content.

Clauses C.6.2 and C.6.3 describe and provide examples of the calculation processes that could take place in a TV Device, from the perspective of both the SC and MSAS elementary functions that it implements.

C.6.2 SC elementary function in the TV device

C.6.2.1 Calculation Explanation

The example calculations take place in the context of a TV Device receiving a DVB broadcast service with a timeline for it signalled in the broadcast using TEMI.

The SC elementary function of the TV Device is observing the timing of (measuring) the received broadcast stream at a point after received broadcast input buffering and decoding, but before decoded video frame buffering, and display by the screen. It changes the timing of presentation of content by controlling the input buffer prior to the decoder. This is illustrated in figure C.6.2.1.1.

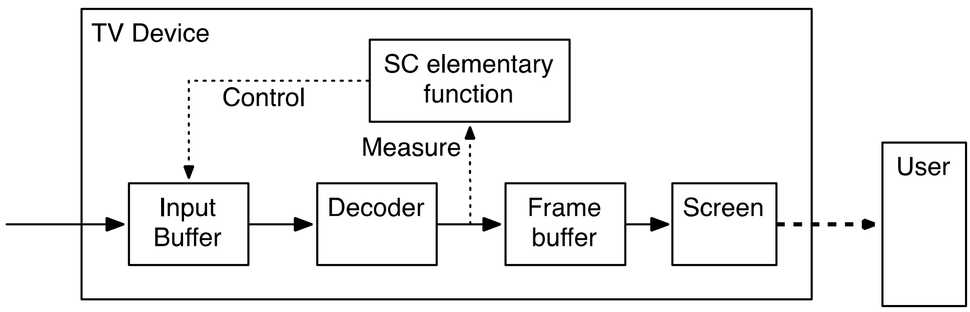


Figure C.6.2.1.1: Model of TV Synchronization Client measurement and control process

Measurement is of the decode time of the media and is made relative to the Wall Clock. For a DVB broadcast this is in terms of the Presentation Timestamp (PTS) of the decoded video frame.

NOTE 1: For alternative content delivery mechanisms, the analogous timestamp, such as Composition Time for ISOBMFF, is appropriate.

NOTE 2: For on-demand streamed content (such as content delivered via MPEG DASH), some or all of the control over timing of presentation could be achieved by varying the timing of fetching of the content stream from the Service Provider. In effect, the Service Provider is providing controllable buffering. In this scenario, the input buffer can be considered to be a black box comprised of a combination of buffering within the TV Device and buffering outside of the TV Device that is provided by the Service Provider.

The SC elementary function of the TV device also observes the signalled TEMI timeline and its relationship to the decode time (PTS). From this information, knowledge of the delays between the point of measurement and the display of content to the user and knowledge of how much delay is being introduced for synchronization purposes, it is possible to calculate the Earliest and Latest Presentation Timestamps to be sent to the MSAS elementary function.

The MSAS sends back Control Timestamps that recommend the timing of presentation of content. To act upon the recommendation, the SC elementary function of the TV Device will adjust the amount of input delay buffering being applied.

The sequence just described is illustrated in figure C.6.2.1.2 and is now demonstrated with an example of the calculations involved.

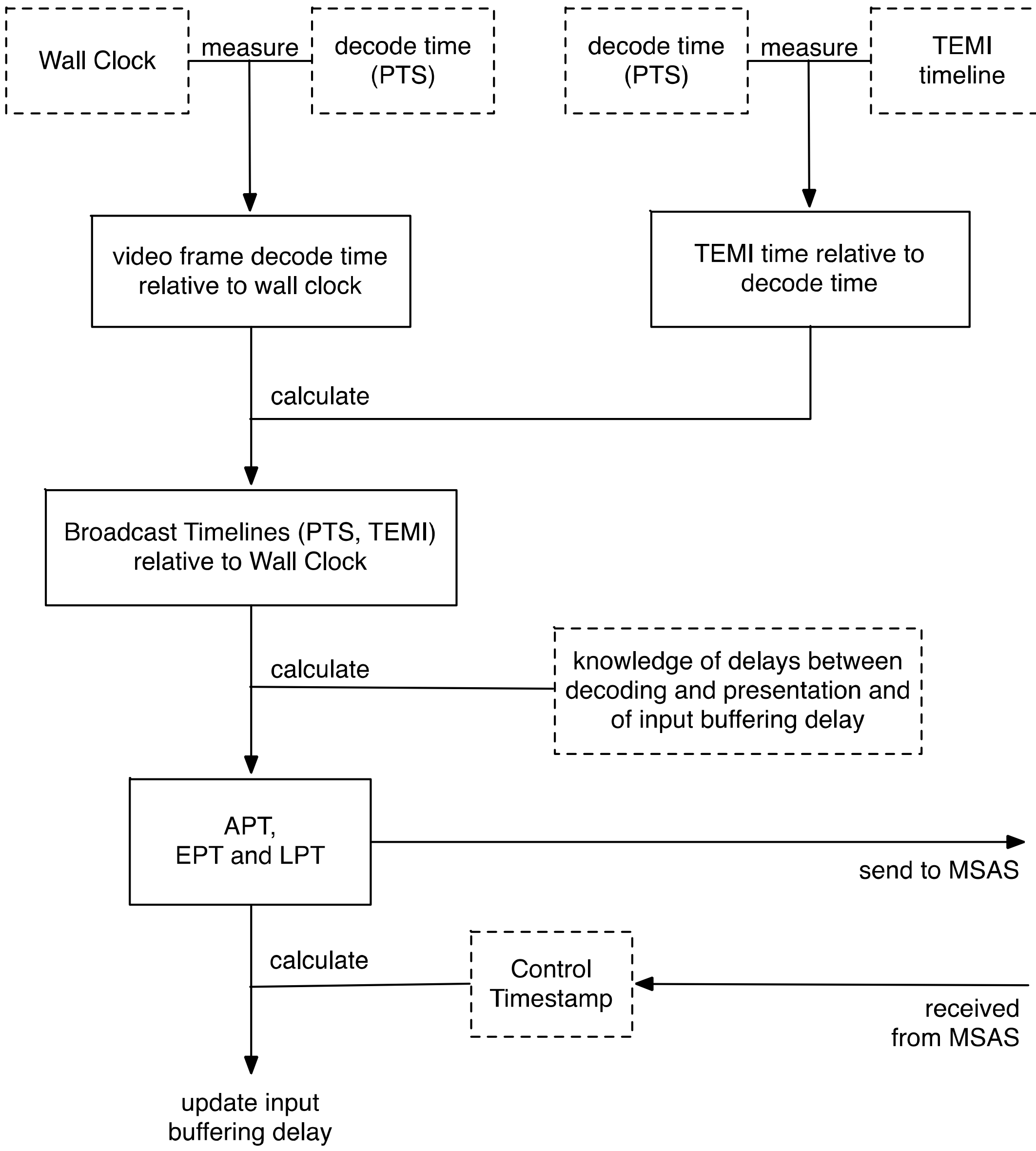


Figure C.6.2.1.2: Example overview of calculation processes for a TV SC

When a frame emerges from the video decoder, the following are measurements are taken:

- m_{wall}: measurement of the Wall Clock measured in seconds.
- m_{decode}: measurement of the decode time (e.g. PTS) of the video frame in integer ticks.
- r_{decode} = (unitsPerSecond$_{decode}$ / unitsPerTick$_{decode}$): the tick rate (ticks per second) of the decode time measurement (e.g. 90 000 / 1 for PTS).
- d: the delay (in seconds) currently being added by the SC for media synchronization.
- d_{max}: the maximum amount of delay (in seconds) that the SC can apply.

The DVB broadcast also contains an additional timeline conveyed as a TEMI timeline with its own tick rate and time values. TEMI timeline values are associated with PTS values derived from the same PCR clock as is used for the PTS for the audio and video. Therefore when a TEMI timeline descriptor is received and processed, the following measurements are taken:

- t_{TEMI}: the integer tick value of the TEMI timeline.

- t_{decode}: the value of the decode time (PTS) corresponding correlating with the timeline tick.
- r_{TEMI} = ($unitsPerSecond_{TEMI}$ / $unitsPerTick_{TEMI}$): the tick rate (ticks per second) of the timeline.

The video decode time can now be converted from being in terms of PTS to being in terms of the TEMI timeline:

$$m_{TSEMIAP} = \text{round}\left((m_{decode} - t_{decode})\frac{r_{TEMISAP}}{r_{decode}} + t_{TSAPEMI}\right)$$

To preserve numerical precision, the calculation can be reformulated in terms of unitsPerTick and unitsPerSecond, as shown below, and ensuring that the numerator and denominator of the division operation are fully evaluated before performing the division:

$$m_{TSAPEMI}$$

The measurement that has been taken is affected by the delay d currently being applied at the input buffer and is used for the Actual Presentation Timestamp. In contrast, the Earliest and Latest Presentation Timestamps to be reported to the MSAS need to represent the range of possible delay that the TV Device can apply (with an upper bound determined by d_{max}).

The reference point for the EPT and LPT (see clause 5.7.2) are at the point of light emission from the screen. So to calculate them the following also needs to be known:

- F: the frame buffer delay.
- S: the screen delay.

If the Synchronization Timeline used in communication with the MSAS is PTS, then APT, EPT and LPT will take the following values:

$$APT_{PTS} = (m_{decode}, m_{wall} + F + S)$$

$$EPT_{PTS} = (m_{decode}, m_{wall} + F + S - d)$$

$$LPT_{PTS} = (m_{decode}, m_{wall} + F + S - d + d_{max})$$

If the Synchronization Timeline used in communication with the MSAS is TEMI, then EPT and LPT will take the following values:

$$APT_{TEMISAP} = (m_{TEMISAP}, m_{wall} + F + S)$$

$$EPT_{TSAPEMI} = (m_{TSAPEMI}, m_{wall} + F + S - d)$$

$$LPT_{TSAPEMI} = (m_{TSAPEMI}, m_{wall} + F + S - d + d_{max})$$

The TV Device is able to vary the timing of presentation of content, so when a Control Timestamp is received from the MSAS it is acted upon by calculating a new value d' (in seconds) for the amount of delay the SC elementary function should be applying to the content the TV Device is presenting. This replaces the previous delay d.

Given a Control Timestamp ($x_{timeline}$, x_{wall}) with rate $r_{timeline}$ the SC calculates x'_{wall} which is an estimate of what x_{wall} would be given most recent current measurements within the TV Device but excluding the amount of delay d currently being applied. The new value d' can then be calculated:

$$d' = x_{wall} - x'_{wall}$$

If TEMI is being used as the Synchronization Timeline, then $r_{timeline} = r_{TEMI}$ and:

$$x'_{wall} = (x_{timeline} - m_{TSAP})\frac{1}{r_{TSAP}} + m_{wall} + F + S - d$$

And therefore:

$$d' = x_{wall} - \left[(x_{timeline} - m_{TSAP})\frac{1}{r_{TSAP}} + m_{wall} + F + S - d\right]$$

If PTS is being used as the Synchronization Timeline, then $r_{timeline} = r_{decode}$ and:

$$x'_{wall} = (x_{timeline} - m_{decode})\frac{1}{r_{decode}} + m_{wall} + F + S - d$$

And therefore:

$$d' = x_{wall} - \left[(x_{timeline} - m_{decode})\frac{1}{r_{decode}} + m_{wall} + F + S - d\right]$$

Where the TV Device incorporates both SC and MSAS elementary functions, and where the MSAS elementary function chooses $x_{timeline}$ to be equal to m_{decode} or m_{TEMI} the calculations are simplified:

$$d' = x_{wall} - (m_{wall} + F + S - d)$$

This can also be calculated in terms of a most recent Earliest Presentation Timestamp with tick rate $r_{timeline}$:

$$EPT = (m_{timeline}, m'_{wall})$$

$$d' = x_{wall} - \left((x_{timeline} - m_{timeline})\frac{1}{r_{timeline}} + m'_{wall}\right)$$

Again, if $x_{timeline}$ is equal to $m_{timeline}$ then the calculations are simplified:

$$d' = x_{wall} - m'_{wall}$$

C.6.2.2 Calculation Example

A TV Device is receiving a DVB broadcast. The most recently decoded video frame of the service has a PTS decode time of 15 682. The service also has a TEMI timeline with tick rate of 24 000 (23,97 fps with 1 001 ticks per frame) where the most recent timeline descriptor occurred at PTS 8 173 and had tick value 2 304 302 (2 302 frames).

- m_{decode} = 15 682
- r_{decode} = 90 000 / 1
- t_{TEMI} = 2 304 302
- t_{decode} = 8 173
- r_{TEMI} = 24 000 / 1 001

Then:

$$m_{TSAPEMI} = \text{round}\left((15\,682 - 8\,173)\frac{1}{90\,000}\frac{24\,000}{1} + 2\,304\,302\right)$$

$$= \text{round}\left(\left((15\,682 - 8\,173) \times 1 \times 24\,000\right) \div (90\,000 \times 1) + 2\,304\,302\right)$$
$$= 2\,306\,304$$

The measurement of m_{decode} was made at wall clock time 33 300,280 secs. The SC elementary function of the TV Device estimates its Frame buffer delay as 0,5 secs and its screen delay as 0,1 secs.

- m_{wall} = 33 300,280
- F = 0,5
- S = 0,1

The Actual Presentation Timestamp to be reported to the MSAS is therefore:

$$APT_{PTS} = (15\,862;\ 33\,300{,}280 + 0{,}5 + 0{,}1)$$
$$= (15\,862;\ 33\,300{,}88)$$

The SC elementary function of the TV Device knows that it is currently adding 0,05 secs delay for media synchronization and that it could delay a maximum of 4 secs:

- d = 0,05
- d_{max} = 4,0

The Earliest and Latest Presentation Timestamps to be reported to the MSAS are therefore:

$$EPT_{PTS} = (15\ 862;\ 33\ 300{,}280 + 0{,}5 + 0{,}1 - 0{,}05) = (15\ 862;\ 33\ 300{,}83)$$

$$LPT_{PTS} = (15\ 862;\ 33\ 300{,}280 + 0{,}5 + 0{,}1 - 0{,}05 + 4{,}0) = (15\ 862;\ 33\ 304{,}83)$$

$$EPT_{TEMI} = (2\ 306\ 302;\ 33\ 300{,}280 + 0{,}5 + 0{,}1 - 0{,}05) = (2\ 306\ 302;\ 33\ 300{,}83)$$

$$LPT_{TSAPEMI} = (2\ 306\ 302;\ 33\ 300{,}280 + 0{,}5 + 0{,}1 - 0{,}05 + 4{,}0) = (2\ 306\ 302;\ 33\ 304{,}83)$$

Given PTS being used as the Synchronization Timeline, a Control Timestamp (15 870; 33 301,10) is received back from the MSAS. Therefore the new delay d' is:

$$\left[(15\ 870 - 15\ 862)\frac{1}{90\ 000} + 33\ 300{,}83\right] = 0{,}27$$

This is an increase in delay of 0,22 secs.

C.6.3 MSAS elementary function in the TV device

C.6.3.1 Calculation Explanation

An MSAS receives Earliest and Latest Presentation Timestamps from all Synchronization Clients, including the SC elementary function of the TV Device. It then calculates a Control Timestamp and disseminates it back to all SCs as a recommendation of the timing of presentation. This is illustrated in figure C.6.3.1.1.

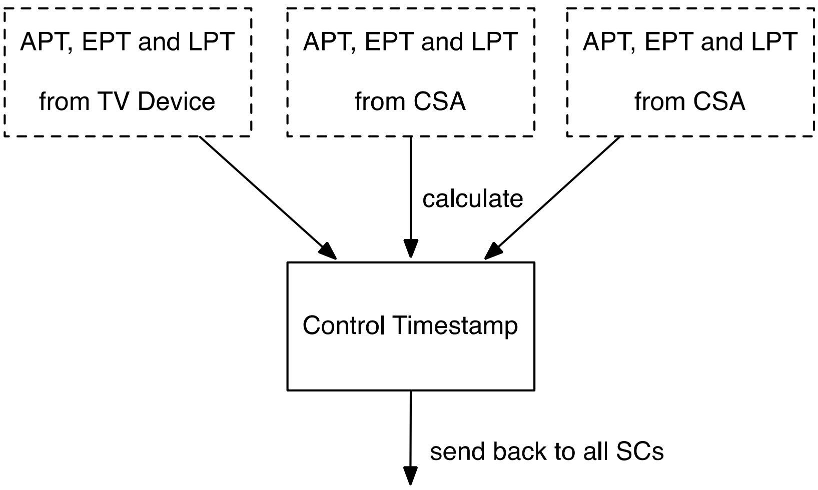


Figure C.6.3.1.1: Example overview of Calculation Process for a TV Device MSAS

The choice of policy applied by the MSAS to guide its calculation is an implementation detail of an MSAS.

To determine if synchronization is possible between any given set of SCs (either all SCs or a subset of SCs) the MSAS determines the window of overlap between possible presentation timing of the SCs in question. Each individual SCs Earliest and Latest Presentation Timestamps describe the bounds for what is possible for that SC.

Given an arbitrary time value x on a Synchronization Timeline with rate r, it is possible to determine the earliest possible Wall Clock time $x_{wall}\{k\}$ and latest possible Wall Clock time $x'_{wall}\{k\}$ at which SC{k} could present. These are calculated from the Earliest and Latest Presentation Timestamps EPT{k} and LPT{k} for SC{k}:

$$EPT\{k\} = [e\{k\}, e_{wall}\{k\}]$$

$$LPT\{(k)\} = [l\{k\}, l_{wall}\{k\}]$$

$$x_{wall}\{k\} = (x - e\{k\})\frac{1}{r} + e_{wall}\{k\}$$

$$x'_{wall}\{k\} = (x - l\{k\})\frac{1}{r} + l_{wall}\{k\}$$

This can also be written as:

$$x_{wall}\{k\} = \frac{(x - e\{k\})\, units\_per\_tick}{units\_per\_sec} + e_{wall}\{k\}$$

$$x'_{wall}\{k\} = \frac{(x - l\{k\})\, units\_per\_tick}{units\_per\_sec} + l_{wall}\{k\}$$

If there exists an interval x_{wall} to x'_{wall} that is entirely within the interval $x_{wall}\{k\}$ to $x'_{wall}\{k\}$ for SC{k} all SCs, then synchronization is possible across all SCs if timeline time value x is presented at any Wall Clock time between x_{wall} and x'_{wall}. This is illustrated in figure C.6.3.1.2.

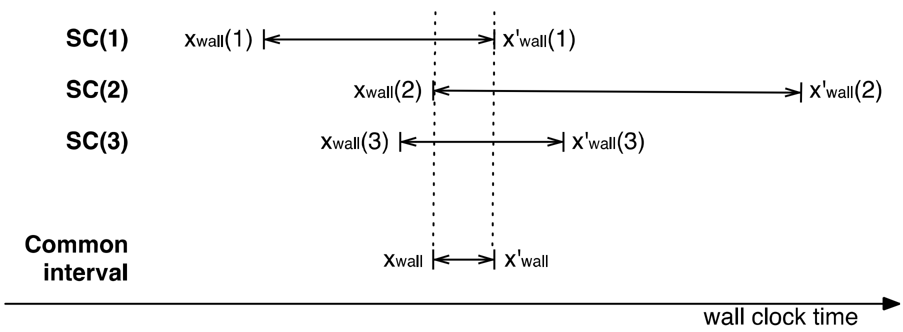


Figure C.6.3.1.2: Determining the interval of presentation timing possible for all SCs

The earliest possible presentation time achievable across all SCs x_{wall} will be the latest of the earliest possible presentation times for each SC:

$$x_{wall} = \max_{1 \le k \le n} x_{wall}\{k\}$$

Similarly, the latest possible presentation time achievable across all SCs x'_{wall} will be the earliest of the latest possible presentation times for each SC:

$$x'_{wall} = \min_{1 \le k \le n} x'_{wall}\{k\}$$

If there is an overlap across all SCs, then x_{wall} will be less than or equal to x'_{wall}. If this is not the case, then synchronized presentation across all SCs is not achievable.

Any Control Timestamp (x, c_{wall}) where $x_{wall} \le c_{wall} \le x'_{wall}$ will therefore represent a recommendation of presentation timing that all SCs are able to achieve.

If some SCs are providing an Actual Presentation Timestamp then this indicates they are already presenting Timed Content. In this situation it is preferable to select a Control Timestamp from within the range of achievable Control Timestamps that is as close as possible to the Actual Presentation Timestamps. An MSAS can choose to favour the Actual Presentation Timestamp from a particular SC, such as the SC residing in the same TV Device as the MSAS.

Again, given the same Time Value x on the Synchronization Timeline, the actual presentation Wall Clock Time Value $y_{wall}\{k\}$ can be calculated for an Actual Presentation Timestamp APT{k}:

$$APT\{(k)\} = [a\{k\}, a_{wall}\{k\}]$$

$$y_{wall}\{k\} = (x - a\{k\})\frac{1}{r} + a_{wall}\{k\}$$

This can also be written as:

$$y_{wall}\{k\} = \frac{(x - a\{k\})units\_per\_tick}{units\_per\_sec} + a_{wall}\{k\}$$

The MSAS can verify that the Actual Presentation Timestamps are consistent with the reported bounds of achievable presentation and select a Control Timestamp that is close to or equal to an Actual Presentation Timestamp. This is illustrated in figure C.6.3.1.3.

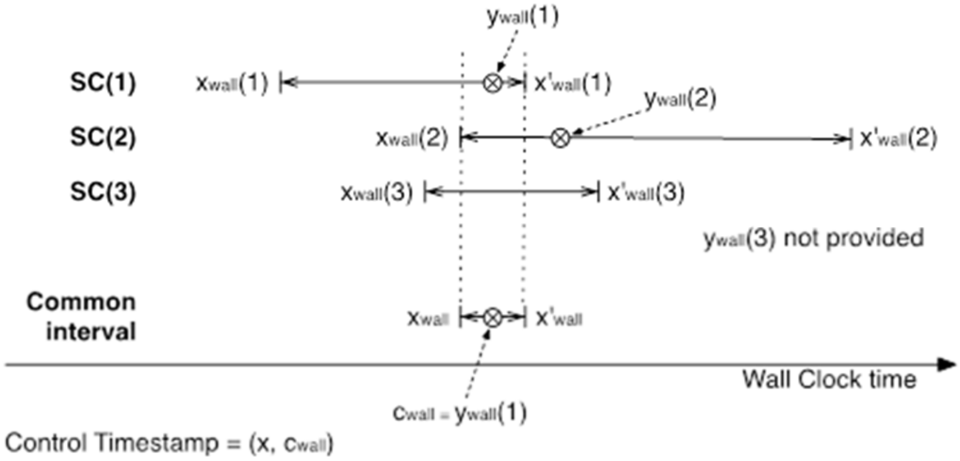


Figure C.6.3.1.3: Determining the Control Timestamp from Actual Presentation Timestamps from some SCs

If no SCs are providing Actual Presentation Timestamps, then when first commencing a synchronization, the MSAS may choose to minimize the delay in presenting content to the user. This is achieved by using the earliest possible presentation time x'_{wall} as the value for c_{wall}.

For later Control Timestamps (particularly if new SCs join the synchronization, or existing SCs leave) a Control Timestamp that maintains the same presentation timing the previous Control Timestamp will ensure that the presentation to the user does not feature a discontinuity as all SCs try to change their presentation timing. Given a previous Control Timestamp (p, p_{wall}) this can recalculated as (x, p'_{wall}) representing a Control Timestamp that represents the same timing of presentation but which is in terms of x instead of p:

$$p'_{wall} = (x - p)\frac{1}{r} + p_{wall}\{k\}$$

If $x_{wall} \leq p'_{wall} \leq x'_{wall}$ then the previous Control Timestamp still represents a timing of presentation that is achievable by all SCs and could therefore continue to be used by them.

C.6.3.2 Calculation Example

An MSAS is coordinating synchronization between 3 Synchronization Clients: SC{1}, SC{2} and SC{3}. Each SC{k} has recently reported an Earliest Presentation Timestamp EPT{k}, a Latest Presentation Timestamp LPT{k}, and in some cases an Actual Presentation Timestamp APT{k}, using a Synchronization Timeline with a tick rate of 25 ticks per second:

- APT{1} = (1 005; 115 821,02)
- EPT{1} = (1 005; 115 820,50)
- LPT{1} = (1 005; 115 821,20)

- APT{2} = (1 002; 115 822,00)
- EPT{2} = (1 002; 115 820,70)
- LPT{2} = (1 002; 115 823,00)
- APT{3} not provided by the SC
- EPT{3} = (1 003; 115 818,00)
- LPT{3} = (1 003; 115 821,30)
- unitsPerSecond = 25
- unitsPerTick = 1

An arbitrary point x on the Synchronization Timeline is selected, in this case by taking the lowest of all time values from the reported Earliest Presentation Timestamps:

$$x = \min(1\,005;\ 1\,002;\ 1\,003) \\ = 1\,002$$

$x_{wall}\{k\}$ and $x'_{wall}\{k\}$ are calculated for each SC{k}:

$$x_{wall}\{1\} = (1\,002 - 1\,005)\frac{1}{25} + 115\,820{,}5 = 115\,820{,}38$$

$$x'_{wall}\{1\} = (1\,002 - 1\,005)\frac{1}{25} + 115\,821{,}2 = 115\,821{,}08$$

$$x_{wall}\{2\} = (1\,002 - 1\,002)\frac{1}{25} + 115\,820{,}7 = 115\,820{,}70$$

$$x'_{wall}\{2\} = (1\,002 - 1\,002)\frac{1}{25} + 115\,823{,}0 = 115\,823{,}00$$

$$x_{wall}\{3\} = (1\,002 - 1\,003)\frac{1}{25} + 115\,818{,}0 = 115\,817{,}96$$

$$x'_{wall}\{3\} = (1\,002 - 1\,003)\frac{1}{25} + 115\,821{,}3 = 115\,821{,}26$$

The MSAS determines that there does exist a common interval of achievable synchronization across all SCs, with a lower bound x_{wall} and upper bound x'_{wall}:

$$x_{wall} = \max(115\,820{,}38;\quad 115\,820{,}70;\quad 115\,817{,}96) \\ = 115\,820{,}7$$

$$x'_{wall} = \min(115\,821{,}08;\quad 115\,823{,}00;\quad 115\,821{,}26) \\ = 115\,821{,}08$$

This MSAS has previously sent an earlier Control Timestamp (950; 115 818,72) to all SCs. Given this earlier Control Timestamp, the MSAS calculates when the Wall Clock component of this Control Timestamp would be if the timeline time value were x:

$$p'_{wall} = (1\,002 - 950)\frac{1}{25} + 115\,828{,}72 \\ = 115\,820{,}80$$

This lies within the bounds set by x_{wall} and x'_{wall} and therefore can also be used as a Control Timestamp:

$$Control\_Timestamp = (1\,002;\ 115\,820{,}8)$$

This Control Timestamp will not cause SCs to adjust the timing of presentation.

If the MSAS had not sent an earlier Control Timestamp then the MSAS could select any value between x_{wall} and x'_{wall}. However since SC{1} and SC{2} have both provided Actual Presentation Timestamps, these can be used to guide the decision. $y_{wall}\{k\}$ is calculated for each Actual Presentation Timestamp APT{k}:

$$y_{wall}\{1\} = (1\,002 - 1\,005)\frac{1}{25} + 115\,821{,}02 = 115\,820{,}90$$

$$y_{wall}\{2\} = (1\,002 - 1\,002)\frac{1}{25} + 115\,822{,}00 = 115\,822{,}00$$

Only $y_{wall}\{1\}$ is within the range of presentation timing achievable by all SCs, and so the Control Timestamp is derived from it:

$$Control\_Timestamp = (1\,002;\ 115\,820{,}9)$$

If no Actual Presentation Timestamps had been provided by any SC, then the MSAS can arbitrarily select (x, x_{wall}) as the Control Timestamp:

$$Control\_Timestamp = (1\,002;\ 115\,820{,}7)$$

C.7 Example calculations: SC elementary function in the Companion Screen Application

C.7.1 General

Timed Content presented by a CSA can include any combination of audio or video media content or application generated content.

Clause C.7.2 describes and provides examples of the calculation processes that could take place in a CSA when timing the presentation of audio or video media content. Clause C.7.3 provides examples of the calculation processes that could take place in the SC elementary function of a CSA when timing the presentation of application generated content.

It is assumed that the CSA knows the identity of the content being presented by the TV Device and has already selected a Synchronization Timeline for its communication with the MSAS. The CSA knows the relationship between the Material Timeline and the progress of time of the Timed Content it is presenting. The CSA has retrieved Material Information from the MRS and therefore knows the Correlation Timestamps that relate the Material Timeline to the Synchronization Timeline.

C.7.2 SC elementary function in the CSA for media content

C.7.2.1 Calculation Explanation

The example calculations take place in the context of a CSA playing an ISOBMFF media stream. The exact way in which a CSA controls media playback will of course depend on the facilities provided by the platform on which the CSA is executing, however the process can be considered as broadly analogous to what may happen within a TV device (as discussed in clauses C.5.2.1 and C.5.3.1). The SC elementary function of the CSA is observing the timing of (measuring) the point at which decoded video frames are to be rendered to frame buffers; after input buffering and decoding but before frame buffering and display by the screen. This is illustrated in figure C.7.2.1.1.

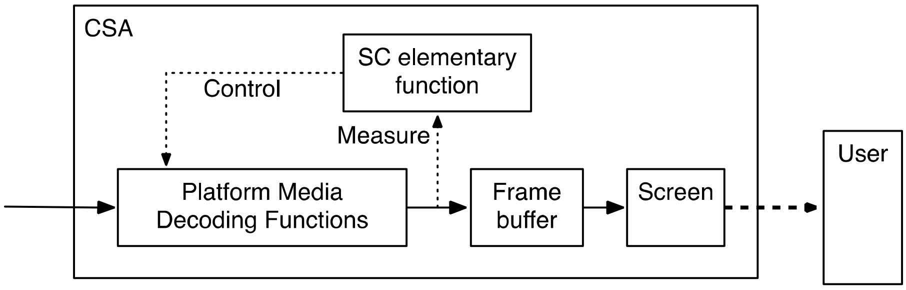


Figure C.7.2.1.1: Model of CSA Synchronization Client measurement process

Measurement is of the decode time of the media and is made relative to the Wall Clock. For an ISOBMFF container this is in terms of the Composition Time (CT) of the decoded video frame.

The SC function of the CSA can translate a CT time value to a time value on the Material Timeline and then to the Synchronization Timeline. Using knowledge of the delays between the point of measurement and the display of content to the user it is possible to calculate the Actual, Earliest and Latest Presentation Timestamps to be sent to the MSAS.

The MSAS sends back Control Timestamps that recommend the timing of presentation of Timed Content. To act upon the recommendation, the SC elementary function of the CSA will adjust the timing of presentation of Timed Content.

The sequence just described is illustrated in figure C.7.2.1.2 and is now demonstrated with an example of the calculations involved.

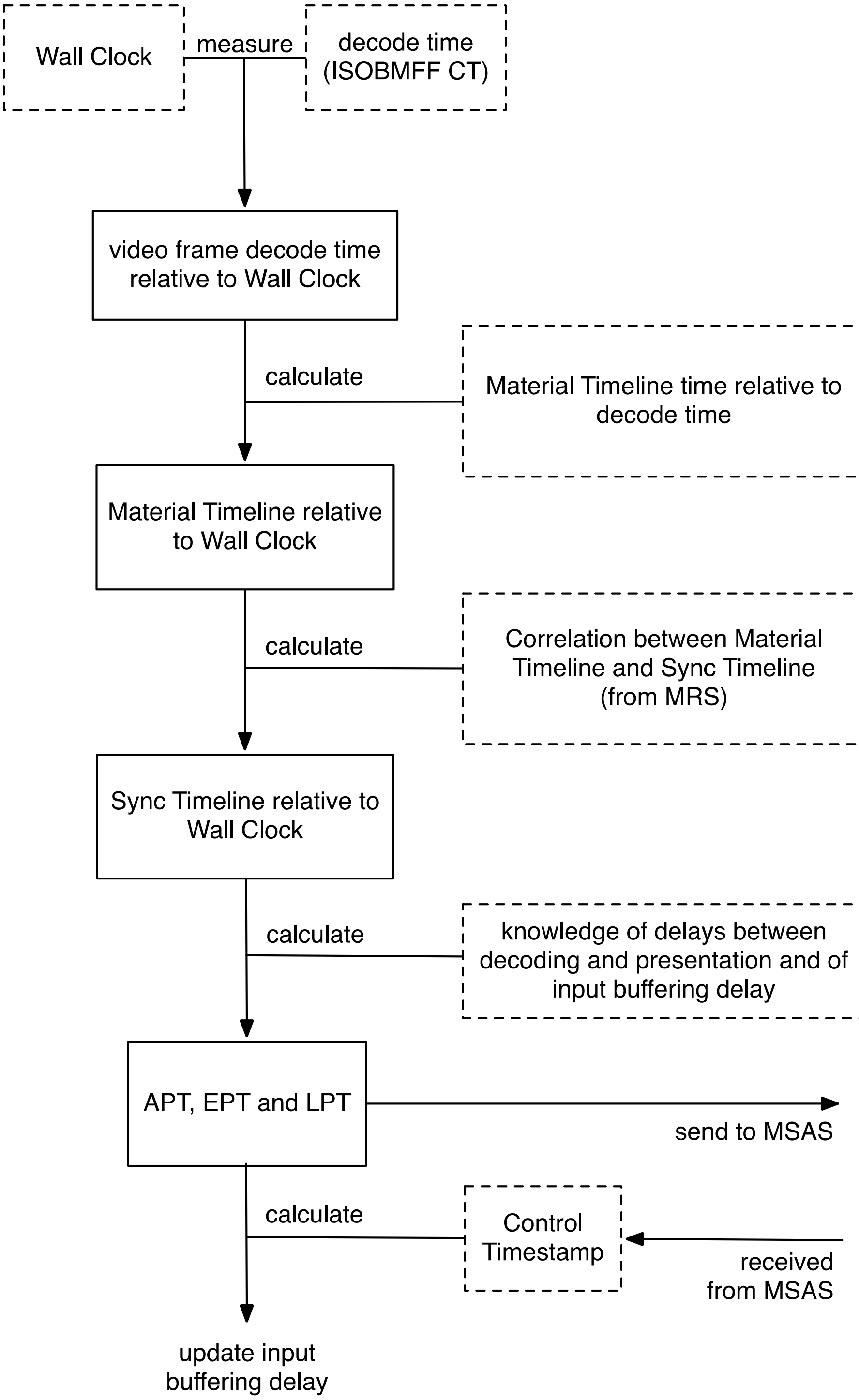


Figure C.7.2.1.2: Example overview of calculation processes for a TV SC

When a decoded frame is to be rendered to the frame buffer, the following are measurements are taken:

- m_{wall}: measurement of the Wall Clock measured in seconds.
- m_{decode}: measurement of the decode time (ISOBMFF CT) of the video frame in integer ticks.
- r_{decode} = ($unitsPerSecond_{decode}$ / $unitsPerTick_{decode}$): the tick rate (ticks per second) of the decode time measurement (e.g. 25 / 1 for 25 fps ticks).
- d: the delay adjustment (in seconds) currently being applied by the SC for media synchronization.

- d_{max}: the maximum amount of delay adjustment (in seconds) that the SC can apply.

The decode time is correlated to the Material Timeline by noting the decode time that corresponds to a point on the Material Timeline and knowledge of the rate of the Material Timeline:

- (t_{decode}, $t_{material}$): point of correlation between time values t_{decode} in terms of CT and $t_{material}$ in terms of the Material Timeline.
- $r_{material}$ = ($unitsPerSecond_{material}$ / $unitsPerTick_{material}$): the tick rate (ticks per second) of the Material Timeline.

The video decode time can now be converted from being in terms of CT to being in terms of the Material Timeline:

$$m_{material} = (m_{decode} - t_{decode})\frac{r_{material}}{r_{decode}} + t_{material}$$

To preserve numerical precision, the calculation can be reformulated in terms of unitsPerTick and unitsPerSecond, as shown below, and ensuring that the numerator and denominator of the division operation are fully evaluated before performing the division:

$$m_{material} = \frac{(m_{decode} - t_{decode})units\_per\_tick_{decode}units\_per\_sec_{material}}{units\_per\_sec_{decode}units\_per\_tick_{material}} + t_{material}$$

From Material Information, the SC elementary function of the CSA has knowledge of how the Material Timeline correlates with the Synchronization Timeline:

- ($t'_{material}$, t'_{sync}): point of correlation between time values $t'_{material}$ in terms of the Material Timeline and t'_{sync} in terms of the Synchronization Timeline.
- r_{sync} = ($unitsPerSecond_{sync}$ / $unitsPerTick_{sync}$): the tick rate (ticks per second) of the Synchronization Timeline.

The video decode time can now be converted from being in terms of the Material Timeline to being in terms of the Synchronization Timeline:

$$m_{sync} = \text{round}\left((m_{material} - t'_{material})\frac{r_{sync}}{r_{material}} + t'_{sync}\right)$$

The measurement that has been taken is affected by the delay d currently being applied for synchronization, whereas the Earliest and Latest Presentation Timestamps to be reported to the MSAS need to represent the range of possible delay that the SC elementary function of the CSA can apply (with an upper bound determined by d_{max}).

The reference point for the APT, EPT and LPT (see clause 5.7.2) are at the point of light emission from the screen. So to calculate them the following also needs to be known:

- F: the frame buffer delay.
- S: the screen delay.

The APT, EPT and LPT to be reported to the MSAS will therefore take the following values:

$$APT = (m_{sync}, m_{wall} + F + S)$$

$$EPT = (m_{sync}, m_{wall} + F + S - d)$$

$$LPT = (m_{sync}, m_{wall} + F + S - d + d_{max})$$

Given a Control Timestamp (x_{sync}, x_{wall}) received from the MSAS, the SC calculates x'_{wall} which is an estimate of what x_{wall} would be given the most recent current measurements within the CSA but excluding the amount of delay d currently being applied. The new value for the delay adjustment d' can then be calculated:

$$d' = x_{wall} - x'_{wall}$$

Where:

$$x'_{wall} = \left(x_{sync} - m_{sync}\right)\frac{1}{r_{sync}} + m_{wall} + F + S - d$$

And therefore:

$$d' = x_{wall} - \left[\left(x_{sync} - m_{sync}\right)\frac{1}{r_{sync}} + m_{wall} + F + S - d\right]$$

A platform with support already built in to synchronize media playback may provide the facility to adjust playback timing by directly setting a Timestamp value that is equivalent to an Actual Presentation Timestamp minus the Frame buffer and screen delays. The Control Timestamp can therefore be used to derive this playback timing adjustment Timestamp (x_{decode}, x''_{wall}) where x_{decode} is in terms of the decode time (ISOBMFF CT in this situation) and x''_{wall} is in terms of the Wall Clock:

$$x_{decode} = \text{round}\left(\left(x_{material} - t_{material}\right)\frac{r_{decode}}{r_{material}} + t_{decode}\right)$$

Where:

$$x_{material} = \left(x_{sync} - t'_{sync}\right)\frac{r_{material}}{r_{sync}} + t'_{material}$$

And x''_{wall} is calculated by taking into account F and S:

$$x''_{wall} = x_{wall} - F - S$$

C.7.2.2 Calculation Example

A CSA is streaming video. The most recently decoded video frame that was rendered to the frame buffer has a decode time of 15 000 measured in milliseconds since the beginning of the video.

- m_{decode} = 15 000
- r_{decode} = 1 000 / 1

The Material Timeline is expressed in terms of frames at a frame rate of 29,97 fps, and decode time of 10 000 correlates to time 0 on the Material Timeline:

- $r_{material}$ = 30 000 / 1 001
- t_{decode} = 10 000
- $t_{material}$ = 0

Then:

$$m_{material} = (15\,000 - 10\,000)\frac{30\,000}{1\,001 \times 1} + 0$$
$$= 149\,850{,}15$$

The Synchronization Timeline is expressed in terms of PTS and a Material Timeline time value of 8 008 correlates to a PTS time value of 2 470 100:

- r_{sync} = 90 000 / 1
- $t'_{material}$ = 8 008
- t'_{sync} = 2 470 100

Then:

$$m_{sync} = \text{round}\left((149\,850{,}15 - 8\,008)\frac{90\,000 \times 1\,001}{30\,000} + 2\,470\,100\right) \\ = 428\,422\,076$$

The measurement of m_{decode} was made at wall clock time 48 100,58 secs. The SC elementary function of the CSA estimates its Frame buffer delay as 0,2 secs and its screen delay as 0,1 secs.

- m_{wall} = 48 100,58
- F = 0,2
- S = 0,1

The Actual Presentation Timestamp to be reported to the MSAS is therefore:

$$APT = (428\,422\,076;\ 48\,100{,}58 + 0{,}2 + 0{,}1) \\ = (428\,422\,076;\ 48\,100{,}88)$$

The SC elementary function of the CSA also knows that it is currently buffer 0,80 secs of video beyond the minimum it needs to cope with network jitter and that the maximum buffering it could perform would be 30 secs:

- d = 0,80
- d_{max} = 30,00

The Earliest and Latest Presentation Timestamps to be reported to the MSAS are therefore:

$$EPT = (428\,422\,076;\ 48\,100{,}58 + 0{,}2 + 0{,}1 - 0{,}8) \\ = (428\,422\,076;\ 48\,100{,}08)$$

$$LPT = (428\,422\,076;\ 48\,100{,}58 + 0{,}5 + 0{,}1 - 0{,}8 + 30{,}0) \\ = (428\,422\,076;\ 48\,130{,}08)$$

A Control Timestamp (428 428 376; 48 100,85) is received back from the MSAS. Therefore the buffering being done should be adjusted to a new value d':

$$d' = 48\,100{,}85 - \left[(428\,428\,376 - 428\,422\,076)\frac{1}{90\,000} + 48\,100{,}58 + 0{,}2 + 0{,}1 - 0{,}8\right] \\ = 0{,}70$$

This will result in presentation of Timed Content 0,1 secs earlier than currently being done by the CSA.

C.7.3 SC elementary function in the CSA for application generated content

C.7.3.1 Calculation Explanation

The example calculations take place in the context of a CSA generating and rendering Timed Content that has been authored against the Material Timeline. The Timed Content could, for example, be rendered HTML pages displayed at certain points along the Timeline or quiz questions presented to the user in sync with the quiz programme being presented on the TV Device. The CSA is therefore modelling the progress of time on the Material Timeline as a clock. This is illustrated in figure C.7.3.1.1.

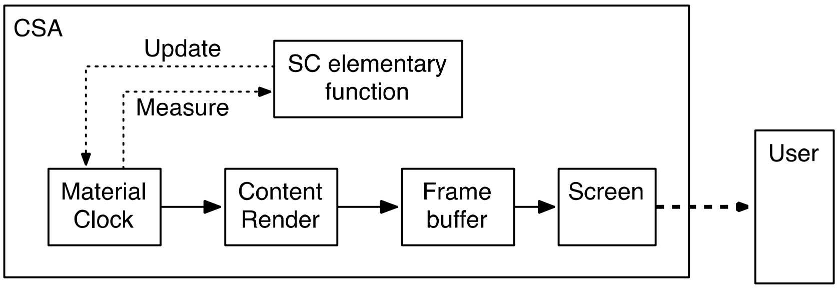


Figure C.7.3.1.1: Model of CSA Synchronization Client measurement process

The Material Clock is relative to the Wall Clock and so can be measured at any moment relative to it (or simply the Correlation Timestamp describing its relationship to the Wall Clock can be read). Because Timed Content is application generated on demand, the CSA has complete flexibility over the timing of presentation. Earliest and Latest Presentation Timestamps to be sent to the MSAS therefore require no calculation to define.

The MSAS sends back Control Timestamps that recommend the timing of presentation of Timed Content. To act upon the recommendation, the SC elementary function of the CSA will adjust the model of the Material Clock, compensating for the time taken to render, buffer and display the Timed Content to the user, and converting back from a Timestamp in terms of the Synchronization to a Timestamp in terms of the Material Timeline.

The sequence just described is illustrated in figure C.7.3.1.2 and is now demonstrated with an example of the calculations involved.

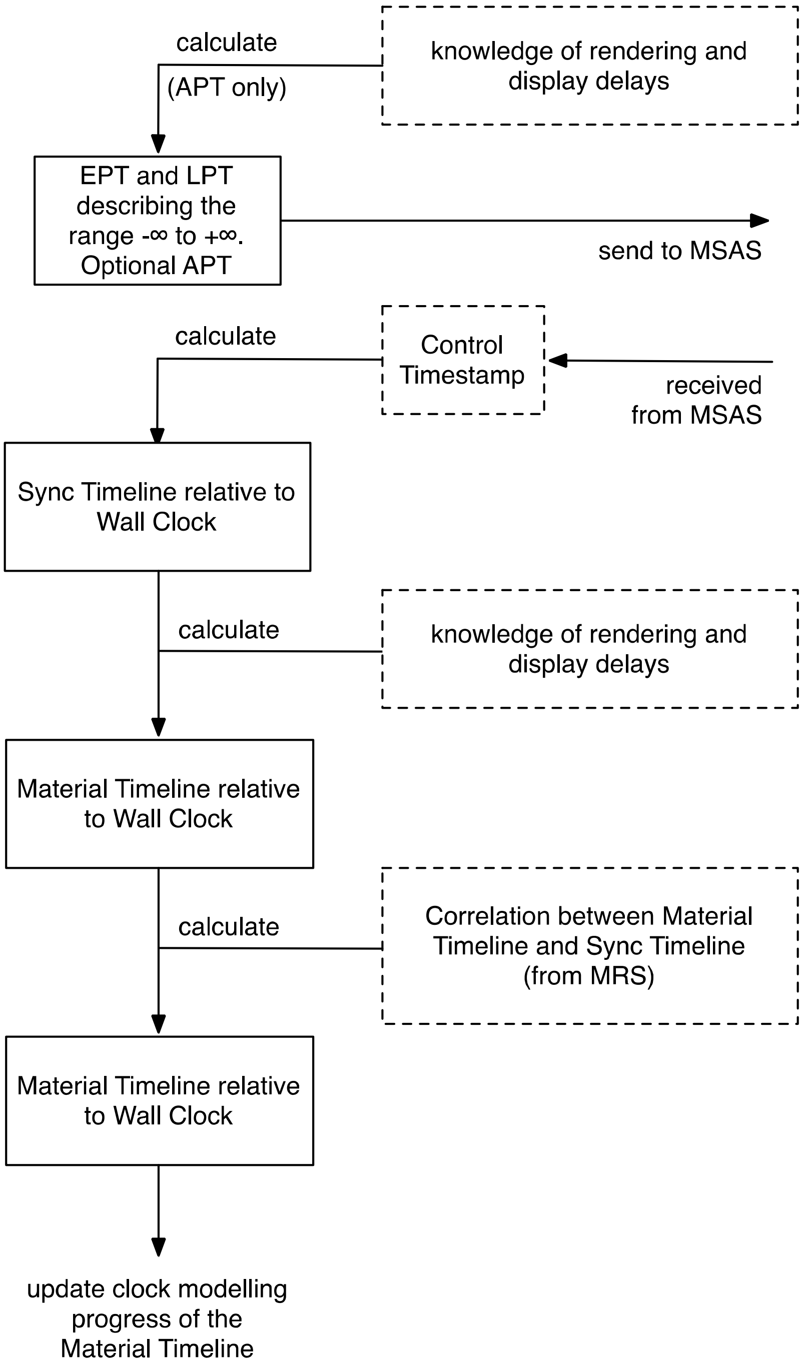


Figure C.7.3.1.2: Example overview of calculation processes for a TV SC

The Earliest and Latest Presentation Timestamps to be sent to the MSAS represent the fact that there are no bounds (in principle) on the timing of presentation by the CSA. This can be represented as follows:

$$EPT = (any\ value, -\infty)$$
$$LPT = (any\ value, +\infty)$$

The reference point for timestamps is the point of light emission from the screen, whereas the clock modelling progress of the Material Timeline should be defined in terms of a reference point corresponding to the moment at which rendering processes is to be triggered. The SC elementary function of the CSA therefore also maintains estimates of:

- F: the rendering and frame buffering delay
- S: the screen delay.

From Material Information, the SC elementary function of the CSA has knowledge of how the Material Timeline correlates with the Synchronization Timeline:

- ($t_{material}$, t_{sync}): point of correlation between time values $t_{material}$ in terms of the Material Timeline and t_{sync} in terms of the Synchronization Timeline
- r_{sync} = (unitsPerSecond$_{sync}$ / unitsPerTick$_{sync}$): the tick rate (ticks per second) of the Synchronization Timeline
- $r_{material}$ = (unitsPerSecond$_{material}$ / unitsPerTick$_{material}$): the tick rate (ticks per second) of the Material Timeline

If the CSA has begun presenting its Timed Content then the SC elementary function of the CSA can observe the current progress of the Material Timeline relative to the Wall Clock:

- ($m_{material}$, m_{wall}): measurement of time values $m_{material}$ in terms of the Material Timeline and m_{wall} in terms of the Wall Clock.

This measurement of a point on the Material Timeline can be calculated in terms of the Synchronization Timeline:

$$m_{sync} = (m_{material} - t_{material})\frac{r_{sync}}{r_{material}} + t_{sync}$$

The Actual Presentation Timestamp to be reported to the MSAS will therefore take the following value:

$$APT = \left(m_{sync}, m_{wall} + F + S\right)$$

Given a Control Timestamp (x_{sync}, x_{wall}) is received back from the MSAS using a Synchronization Timeline with rate of ticks r_{sync}, the SC can calculate a revised Control Timestamp ($x_{material}$, x'_{wall}) that uses the same reference point as the clock and is in terms of the Material Timeline.

$$x_{material} = \text{round}\left(\left(x_{sync} - t_{sync}\right)\frac{r_{material}}{r_{sync}} + t_{material}\right)$$
$$x'_{wall} = x_{wall} - \mathrm{F} - \mathrm{S}$$

The clock modelling progress of the Material Timeline can now be adjusted to correlate the Wall Clock with the Material Timeline according to the Correlation Timestamp ($x_{material}$, x'_{wall}).

The CSA now checks whether it needs to render new application generated Timed Content given that the current position on the Material Timeline may have changed.

C.7.3.2 Calculation Example

The SC elementary function of the CSA sends the Earliest and Latest Presentation Timestamps to the MSAS:

$$EPT = (0, -\infty)$$
$$LPT = (0, +\infty)$$

The Synchronization Timeline has a tick rate of 50 ticks per second. The Material Timeline is defined in terms of milliseconds. The time value 500 210 080 on the Material Timeline corresponds to the time value 75 on the Synchronization Timeline:

- r_{sync} = 50 / 1
- $r_{material}$ = 1 000 / 1
- t_{sync} = 75
- $t_{material}$ = 500 210 080

The SC elementary function of the CSA estimates that rendering takes 0,15 secs and that it will take a further 0,04 secs for the rendered content to be displayed:

- F = 0,15

- S = 0,04

The SC elementary function of the CSA opts to provide an Actual Presentation Timestamp to the MSAS and knows that at Wall Clock Time Value 832 051,50 the Material Timeline was at Time Value 500 210 078:

- $m_{material}$ = 500 210 250
- m_{wall} = 832 051,50

The point on the Material Timeline is recalculated to be in terms of the Synchronization Timeline:

$$m_{sync} = (500\ 210\ 260 - 500\ 210\ 080)\frac{50}{1\ 000} + 75$$
$$= 84$$

The Actual Presentation Timestamp that is sent to the MSAS is therefore:

$$APT = (84;\ 832\ 051{,}50 + 0{,}15 + 0{,}04)$$
$$= (84; 832\ 051{,}69)$$

A Control Timestamp (184; 832 051,80) is received back from the MSAS:

- x_{sync} = 184
- x_{wall} = 832 051,80

Given this information, the clock modelling the Material Timeline is therefore updated by calculating the point $x_{material}$ on the Material Timeline and the point x'_{wall} on the Wall Clock that represent a point of correlation between Material Timeline and Wall Cock:

$$x_{material} = \text{round}\left((184 - 75)\frac{1\ 000}{50} +\ 500\ 210\ 080\right)$$
$$= 500\ 212\ 260$$

$$x'_{wall} = 832\ 051 - 0{,}15 - 0{,}04$$
$$= 832\ 051{,}61$$

C.8 Wall Clock synchronization implementation guidance

C.8.1 General

The CSA synchronizes its Wall Clock with that of the TV Device by regularly exchanging Wall Clock synchronization messages with the TV Device using the protocol described in clause 8. By doing this, the CSA is able to measure the offset between Wall Clocks in the CSA and TV Device and the round trip time of the messages. This is handled by the WC Client elementary function of the CSA and the WC Server elementary function of the TV Device.

Clause C.8.2 provides informative guidance to TV Device implementers regarding WC Server implementations.

Clause C.8.3 provides informative guidance to CSA implementers regarding WC Client implementations. The processes within a WC Client and implementation design decisions are more complex and so are described in more detail:

- Clause C.8.3.1 gives an architectural overview of how a WC Client can be implemented and highlights the differences in requirements when compared to NTP [i.2].
- Clause C.8.3.2 describes how a candidate for an offset is generated from a measurement. This includes the calculation of metrics associated with the candidate (such as dispersion) that may be needed by subsequent filtering and adjustment processes.
- Clause C.8.3.3 describes considerations for the measurement process.

- Clause C.8.3.4 describes approaches to filtering of candidates.
- Clause C.8.3.5 describes approaches to adjusting the CSA Wall Clock.

The CSA can try to smooth out short term variations in measurements, however in practice the achievable degree of synchronization between Wall Clocks in the TV Device and CSA will be primarily limited by the time it takes for a Wall Clock protocol messages to traverse the network (the latency of the network). This manifests as the round trip time. It is recommended that implementers of both TV Devices and CSAs exploit any opportunities that available to them to minimize network latency.

C.8.2 TV Device and WC Server

Although synchronization of Wall Clocks between TV Device and CSA is only needed whilst Timeline Synchronization is being performed, a TV Device may wish to provide the WC Server elementary function at other times too. This will enable a CSA to establish Wall Clock Synchronization to an adequate level of accuracy in advance. Later, when the SC elementary function of the CSA begins the Timeline Synchronization procedure, it will be able to immediately make use of Control Timestamps and provide Earliest and Latest Presentation Timestamps without having to wait for Wall Clock Synchronization to be established.

The WC Server elementary function in the TV Device performs the process illustrated by the following pseudo-code:

```
FOLLOWUP = (can measure more accurate transmit timestamp after sending)

Server_process(IFACE, PORT) {
    for (;;) {
        listen for incoming UDP packet on port number PORT on interface IFACE
        P = received packet payload
        T2 = Wall Clock time value packet was received
        ADDR, PN = IP address and port number of sender of the received packet

        if (FOLLOWUP) {
            P.message_type = 2
        } else {
            P.message_type = 1
        }
        P.precision = log_base_2(local Wall Clock precision)
        P.receive_timestamp = T2
        P.transmit_timestamp = current Wall Clock time value
        send P to address ADDR port number PN

        if (FOLLOWUP) {
            P.transmit_timestamp = more accurate transmit Wall Clock time value of packet
            P.message_type = 3
            send P to address ADDR port number PN
        }
    }
}
```

In practice, it is usually not possible to include an accurate timestamp of the precise time at which a packet is transmitted within the packet payload itself. However there is sometimes hardware and operating system level support for reporting time values for packet transmit times more accurately after the packet is sent. The optional follow-up message provides the mechanism by which the WC Server can convey this more accurate transmit time value to the WC Client.

Precision is included by the WC Server in the responses it sends. It is the measure, in seconds, of the accuracy with which the Wall Clock can be measured. For example, if reading the Wall Clock takes 1 microsecond, then the Wall Clock has a precision of 1 microsecond. Averaging the time taken for multiple successive measurement operations may be used as a way of empirically measuring the precision of a system.

A TV Device needs to have a measurement precision substantially shorter than the maximum tolerance required for the Timeline Synchronization. The WC Server contribution to dispersion and the polling rate with which requests are issued by the WC Client and responded to by the WC Server needs to be sufficiently frequent such that the build up of dispersion (as calculated by the WC Client, see clause C.8.3.2) in between exchanges is also shorter than the maximum tolerance required for the Timeline Synchronization.

It is suggested that the WC Server in a TV Device be able to respond to requests from at least 10 clients simultaneously where each client issues requests at a rate of up to 5 requests per second.

If it is intended to support frame-accurate synchronization of Timed Content at frame rates of between 25 fps and 60 fps then the following minimum conditions are recommended to be met and preferably exceeded:

- The TV Device Wall Clock precision is 1 ms or better.
- The WC Server in the TV Device determines the **receive_timevalue** and **transmit_timevalue** protocol fields with an accuracy 1 ms or better.
- The frequency stability of the Wall Clock in the TV Device is 500 ppm or better.

C.8.3 CSA and WC Client

C.8.3.1 Process Overview

The fundamental principles behind the Wall Clock Synchronization and the algorithms applied by the CSA to align its Wall Clock to that of the TV Device are similar to those for Network Time Protocol (NTP) [i.2]. However, there are important differences:

- NTP clients have many more modes of operation and protocol exchange (e.g. receiving broadcast time messages).
- NTP clients seek to combine measurements from protocol exchanges with multiple servers to improve accuracy. Wall Clock protocol exchanges are with a single server (the TV Device).
- NTP protocol exchanges happen minutes or hours apart. Wall Clock protocol exchanges happen can be much more frequent (e.g. once or more per second).
- NTP clients align clocks gradually over period of seconds or minutes or longer. The Wall Clock needs to be aligned as quickly as possible (e.g. within less than a second).
- NTP clients prioritize monotonicity of the clock. The Wall Clock can tolerate small amounts of jitter (e.g. less than 1/2 the period of a video frame, for frame accurate sync).
- NTP clients align a system-wide Clock available to all applications running on the same device. The Wall Clock is private to the CSA.
- NTP clients control a hardware-based system-wide clock oscillator with hardware support for fine-grained frequency (tick rate) adjustment of the oscillator. The Wall Clock is likely to be derived in software from some existing clock derived from a fixed System Clock oscillator. It is harder to accurately simulate fine grained frequency adjustments in software.

The processes described here are designed to address the requirements of synchronizing the CSAs Wall Clock as quickly as possible to the Wall Clock of the TV Device, accepting that there may be a discontinuity in the tick count at the start of the process and small amounts of jitter throughout. The processes are therefore not identical to those in NTP clients but share many of the same fundamentals.

The processes performed by the WC Client to synchronize the Wall Clock in the CSA to the Wall Clock in the TV Device can be modelled as comprising three main processes. The first process periodically issues requests to the TV Device WC Server and gathers responses. This is a measurement process. The measurements gathered from a single request and response protocol exchange form the basis for a candidate for clock update information. A Candidate is, in effect, the estimate for the offset between the TV Device and CSA Wall Clocks. In practice it may also be accompanied by metadata derived from the measurements - such as the round trip time, the time at which the measurement was made, and the properties of the TV Device Wall Clock that were included in the response message. All of this additional data is useful in the subsequent stages of the process as described below.

Candidates are fed into a filtering process. In general, the most accurate understanding will come from the candidate with the lowest dispersion metric (described in clause C.8.3.2) calculated from its accompanying metadata. The filtering process may therefore choose to discard candidates with high dispersion. Surviving candidates are fed into an adjustment process that uses the information in the candidates to predict the offset between the CSA and TV Device Wall Clocks. The adjustment process then adjusts the CSA Wall Clock to compensate for the offset.

Two approaches will now be described that implement these processes. In both approaches, the CSA Wall Clock is assumed to be modelled in software and is based on an existing clock available to the CSA provided by the Companion Device. A simple CSA Wall Clock implementation is a clock that cannot be adjusted in frequency but can adjust its offset relative to the existing clock. At any given moment, the time value reported by the CSA Wall Clock is equal to the time value reported by the existing clock plus an offset. The offset is updated by the adjustment process to keep the CSA Wall Clock aligned with the TV Device Wall Clock. It is possible, but harder, to implement a software-based CSA Wall Clock that can also be varied in frequency.

Figure C.8.3.1.1 illustrates the first approach where the measurement process measures the CSA Wall Clock to generate T_1 and T_4 during the Wall Clock protocol exchange (see clause 8.2.1).

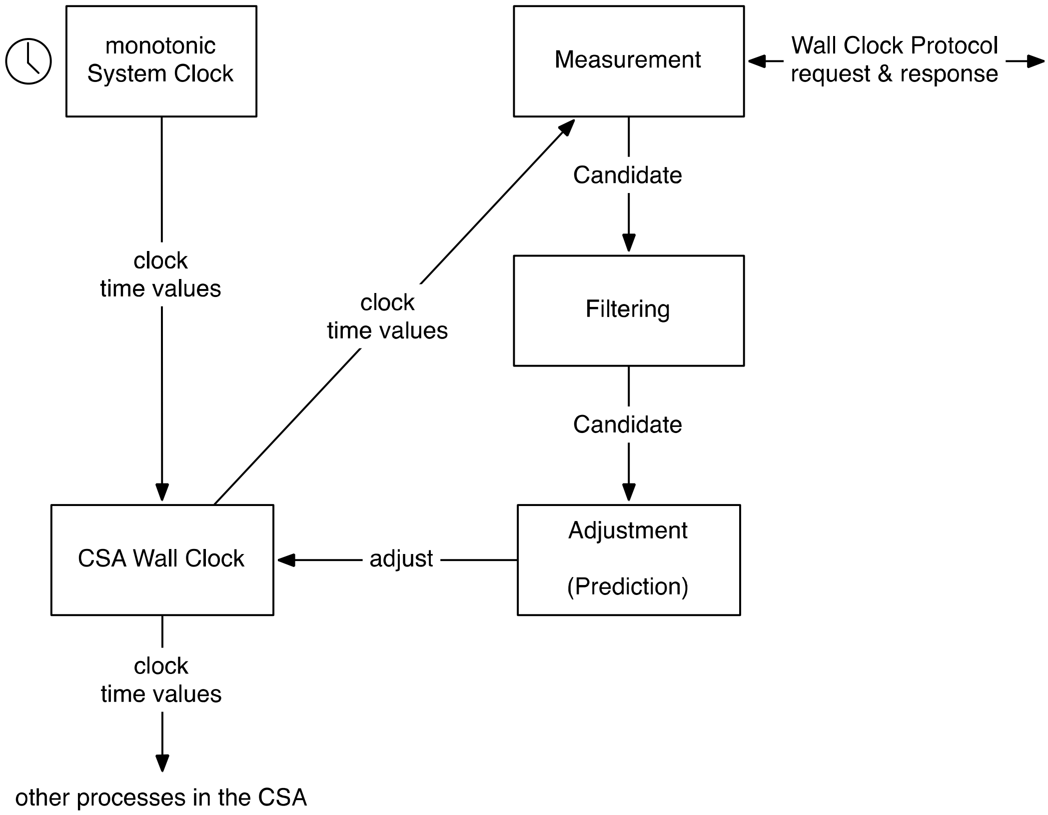


Figure C.8.3.1.1: WC Client processes with a feedback loop

This approach resembles the clock discipline process in NTP and is effectively a feedback loop. Care needs therefore to be taken to avoid instability. There is potential for feedback If the adjustment process retains any information after making an adjustment (such as a candidate offset) that will be used in the calculation of a future adjustment.

A second approach is illustrated in figure C.8.3.1.2 where the measurement process measures a clock from which the Wall Clock is derived to generate T_1 and T_4 during the Wall Clock protocol exchange (see clause 8.2.1).

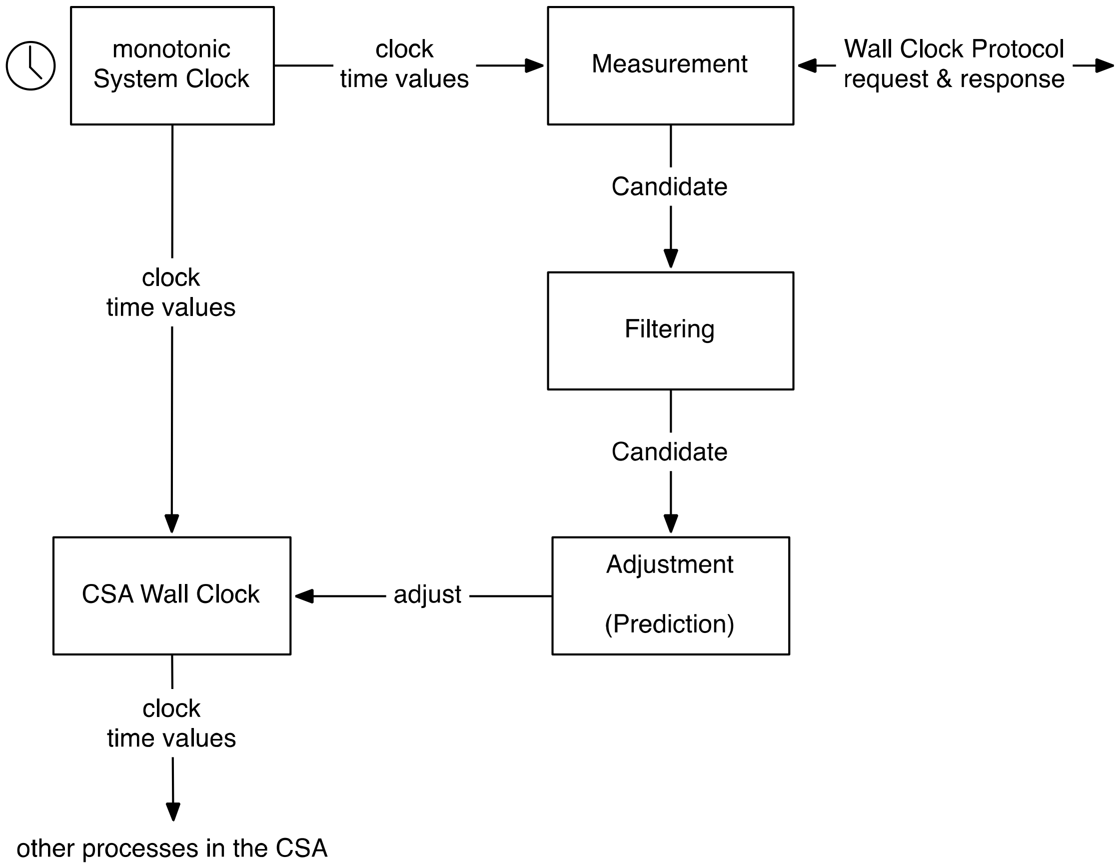


Figure C.8.3.1.2: WC Client processes without a feedback loop

The clock itself is not adjusted. There is therefore no feedback loop. This approach is suitable if historical candidates are used when calculating a new adjustment to apply to the CSA Wall Clock.

The best choice of approach will also depend on the capabilities of the hardware and software of the Companion Device on which the CSA is running.

C.8.3.2 Calculation of candidates and metrics from measurements

As described in clause 8.2.1, the measurement process results in a candidate estimate for the offset between TV Device and CSA Wall Clocks and the total round trip time of the protocol exchange:

$$offset\ \theta = \frac{(T_3+T_2)-(T_4+T_1)}{2}$$

$$round\_trip\_time\ \delta = (T_4 - T_1) - (T_3 - T_2)$$

The TV Device also provides estimates of its measurement precision and maximum frequency offset, as described in clauses 8.2.2 and 8.2.3.

The clock being measured by the CSA will also have a measurement precision and maximum frequency error characteristics. Averaging the time taken for multiple successive measurement operations may be used as a way of empirically determining the measurement precision. Maximum frequency error may be harder for a WC Client to determine, as it may not have detailed knowledge of the hardware characteristics of the Companion Device in the same way that a WC Server has knowledge of a TV Device. Instead it may need to make an educated estimate.

The candidate offset is only an estimate that is only accurate if:

- the request and response messages contributed equally to the round trip time;
- measurements by WC Client and WC Server are perfectly precise;

- and the hardware oscillators form which both TV Device and CSA Wall Clocks are derived match perfectly in frequency.

In practice none of these are true, and they all contribute uncertainty to the candidate offset. This uncertainty can be quantified by a dispersion metric that quantifies the maximum amount by which the true offset may deviate from the candidate offset. As time passes and a candidate becomes older, dispersion of that candidate gradually increases. This is illustrated in figure C.8.3.2.1.

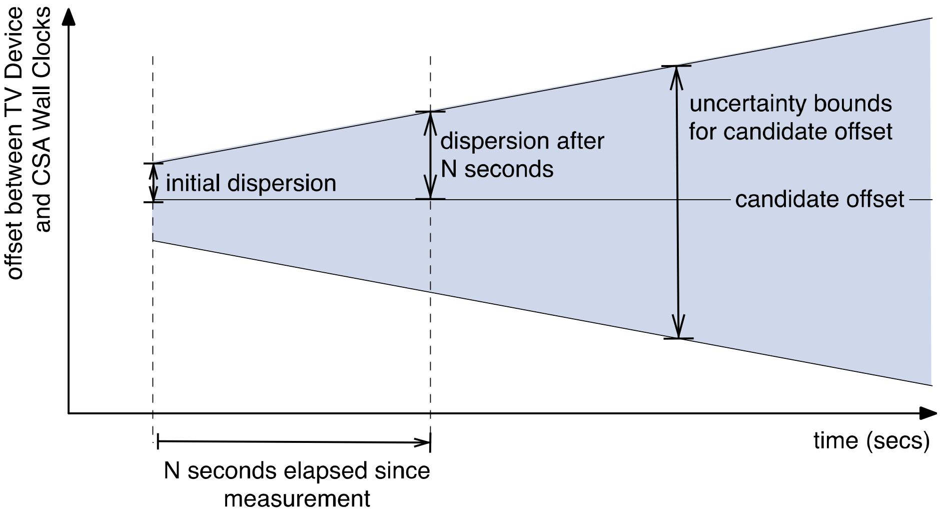


Figure C.8.3.2.1: How dispersion quantifies the bounds of certainty for a candidate offset over time

Calculating dispersion at any given moment gives a measure of how accurate a given estimate of the TV Device Wall Clock is if based on the corresponding candidate. This is a useful input to the assessment of how accurately a CSA can synchronize its presentation of Timed Content to the presentation of Timed Content by the TV Device and determine whether it is within an acceptable tolerance.

Dispersion is the combination of the following possible causes of inaccuracy:

- The offset θ is calculated assuming that the flight time of the request message equals the flight time of the response message in the protocol exchange. However, in reality one may be fast and the other may be slow. In the worst case the assumed reference point for the offset calculation could be in inaccurate by $\pm {}^{\delta}/_{2}$ where δ is the total round trip time in seconds.

- The measurement precision ρ_s of the WC Server and $\rho_c$of the WC Client may mean a measurement by the WC Server is inaccurate by up to ρ_c seconds or a measurement by the WC Client is inaccurate by up to ρ_c seconds.

- The intervals $(T_4 - T_1)$ and $(T_3 - T_2)$ are used in the calculation of the offset where T_1, T_2, T_3 and T_4 are in units of seconds. The potential for a frequency error of up to φ_s ppm in the Wall Clock of the Server or ϕ_c ppm in the clock of the WC Client means the calculated offset may be inaccurate by up to $\phi_c(T_4 - T_1) + \phi_s(T_3 - T_2)$ microseconds.

- If τ seconds have passed since the measurement was made, then the potential for a maximum frequency error of up to ϕ_s ppm in the Wall Clock of the Server and ϕ_c ppm in the clock of the WC Client means the calculated offset may be inaccurate by an additional $(\phi_s + \phi_c)\tau$ microseconds compared to when the measurement was made.

Combining these together, the dispersion ε (in units of seconds) is calculated as according to the following equation:

$$\varepsilon = \frac{\delta}{2} + \rho_s + \rho_c + \frac{\phi_c(T_4 - T_1) + \phi_s(T_3 - T_2) + (\phi_s + \phi_c)\tau}{1\,000\,000}$$

The true offset θ_0 can therefore be described as being a value somewhere within the bounds:

$$\theta - \varepsilon \leq \theta_0 \leq \theta + \varepsilon$$

Or, equivalently, the candidate offset can be described as being equal to the true offset plus an unknown error θ_{ε}:

$$\theta = \theta_0 + \theta_{\varepsilon} \; where \; -\epsilon \leq \theta_{\varepsilon} \leq \varepsilon$$

A CSA without knowledge of its measurement precision ρ_c or its maximum frequency error $\phi_c$still perform these calculations while assuming these two values are fixed (e.g. at zero). The resulting dispersion metric will still allow useful relative comparisons of inaccuracy between candidate offsets but will not correctly quantify the potential error between the TV Device and CSA Wall Clocks.

C.8.3.3 Measurement process

The measurement process in the WC Client regularly sends a Wall Clock protocol request message to the WC Server of the TV Device. How regularly this needs to happen can be determined from dispersion metrics and knowledge of the desired accuracy of synchronization of presentation of Timed Content between the TV Device and CSA.

If the adjustment uses a particular candidate offset, then its associated dispersion metric quantifies the degree of accuracy that can currently be guaranteed for Wall Clock Synchronization. The measurement process therefore can decide when it needs to carry out new measurements by checking current dispersion against the accuracy requirement. If dispersion exceeds required accuracy then new measurements need to be taken until the adjustment process is able to use a candidate offset with sufficiently low dispersion.

If dispersion currently does not exceed the accuracy requirement, then the measurement process can calculate how much time will elapse before dispersion exceeds the accuracy requirement. The measurement process can plan to make another measurement before that point in time is reached.

Given the following information:

- The dispersion was initially ε_0 seconds at the time immediately after the measurement was made.
- The maximum frequency errors (measured in ppm) of the CSA and TV Device are $\phi_c$and $\phi_s$respectively.
- The maximum tolerable dispersion (the required accuracy of synchronization) is ε_{max}seconds.

Then the number of seconds T after which the dispersion will exceed the maximum amount can be calculated as:

$$T \leq (\varepsilon_{max} - \varepsilon_0)\frac{1\,000\,000}{(\phi_s + \phi_c)}$$

EXAMPLE: A CSA wishes to achieve Wall Clock synchronization with 5 ms accuracy. The dispersion ε_0 immediately after the measurement is made is 2 ms and the sum of the maximum frequency errors of both the CSA and the TV Device totals 1 000 ppm.

The dispersion will increase by 1ms per second. Therefore, after 3 secs the dispersion will exceed 5 ms, unless the WC Client obtains a new candidate offset with lower dispersion before that time has elapsed.

In practice this means that a WC Client makes frequent requests when the measurement process first starts in order to lower dispersion as quickly as possible. After that, the time between measurements tends to increase. The lower the maximum frequency error of the CSA and TV Device clocks, and the lower the round trip time, then the greater the time between measurements will be.

ε_0 will tend to be lower if the measured round trip time is as accurate as possible. In practice, it is usually not possible to include an accurate time value T_1 of the precise time at which a packet is transmitted within the packet payload itself. Instead it is generated some time before the packet is sent, resulting in slightly higher calculated round trip time. However, there is sometimes hardware and operating system level support for reporting time values for packet transmit times more accurately after the packet is sent. If this facility is available, a WC Client can use it to gain a more accurate measurement of the transmit time value for a request message and can store this information for use when the corresponding response message is received.

For the same reason described above, the WC Server in the TV Device may choose to send follow-up responses as part of the Wall Clock protocol exchange (see clause 8.2.1). A WC Client that receives a response message that indicates a follow-up should be expected can ignore the initial response message and wait for the follow-up. It is advisable for a WC Client to not speculatively use both the initial response and the follow-up. The follow-up is more accurate and should supersede the initial response. The initial response will just contribute inaccuracy to the adjustment process.

C.8.3.4 Filtering process

The measurement process produces candidate offsets and associated dispersion metrics. Candidate offsets with high dispersion are potentially less accurate than a previous candidate with lower dispersion.

There will always be a more recent candidate offset with a lower dispersion than a previous candidate offset because dispersion grows over time as a measurement from which a candidate was derived becomes more out of date.

A WC Client can employ a filtering process to select the best candidate offset to put forward for the adjustment process by simply selecting the candidate with the lowest dispersion at the present time. If the assumption is made that contributory parameters such as measurement precision and maximum frequency error are constant, then dispersion will grow at the same rate for all candidate offsets. A simple algorithm only needs to remember the candidate offset with the lowest dispersion found so far, for example:

```
bestCandidate = get_candidate_from_measurement_process()
send_to_adjustment_process(bestCandidate)
do forever {
    newCandidate = get_candidate_from_measurement_process()
    bestDispersion = calculate_dispersion(bestCandidate, time_now())
    newDispersion = calculate_dispersion(newCandidate, time_now())
    if (newDispersion <= bestDispersion) {
        bestCandidate = newCandidate
        send_to_adjustment_process(bestCandidate)
    }
}
```

C.8.3.5 Adjustment process

Using the candidate offsets that survived filtering, the WC Client can apply an adjustment process to adjust the CSA Wall Clock.

A simple approach is to add the candidate offset to the CSA Wall Clock whenever a new candidate offset is provided by the filtering process.

Any given candidate offset contains an amount of error (bounded by the dispersion ε) and so using the candidate offset to update the CSA Wall Clock will result in jitter. The amount of jitter will be bounded by $\pm\varepsilon$. This means that the Wall Clock is no longer monotonic at small time scales equal to the magnitude of the jitter. Therefore, a CSA wishing to ensure smooth presentation of Timed Content to the user may wish to apply hysteresis when measuring the CSA Wall Clock to supress small jumps backwards in time due to the jitter.

Jitter can also be reduced and general accuracy improved, by combining several recent candidate offsets. A simple approach is to perform a simple equally weighted average across the most recent *n* candidate offsets. An improved approach is to calculate a weighted average offset $\bar{\theta}$, where the weight for each candidate offset θ_i is the reciprocal of its dispersion ε_i at the time the calculation is performed:

$$\bar{\theta} = \frac{\sum_{i=1}^{n} w_i \theta_i}{\sum_{i=1}^{n} w_i} \; where \; w_i = \frac{1}{\varepsilon_i}$$

Again, if presentation of Timed Content is closely timed to the Wall Clock, then this may improve presentation smoothness.

In a home network environment where there is significant network congestion, the measurement process may struggle to obtain low dispersion candidate offsets as frequently as desired. If the time between useful candidates is significant, then dispersion will eventually be dominated by the contribution due to frequency error between the TV Device Wall Clock and CSA Clock. In this situation it may be useful to try to apply frequency adjustments to the CSA Wall Clock to compensate.

A frequency error will manifest as a trend of increasing or decreasing candidate offset over time. Oscillator frequency varies slowly over time due to effects such as temperature and therefore can be approximated as a linear relationship over short periods. Standard statistical techniques such as those for linear regression estimation can be applied to estimate the function relating time to offset.

However, the error in the candidate offsets will begin to dominate if the measurements being considered are not spread over a long enough period of time or if initial dispersion immediately after a measurement is taken is comparable to or greater than the amount by which offset is varying from one measurement to the next. In those situations, the slope may be estimated with substantial error. In the worst cases this may mean that the frequency error is compounded, therefore increasing jitter and reducing the accuracy of the CSA Wall Clock as time progresses between measurements and adjustments. This is illustrated in figure C.8.3.5.1.

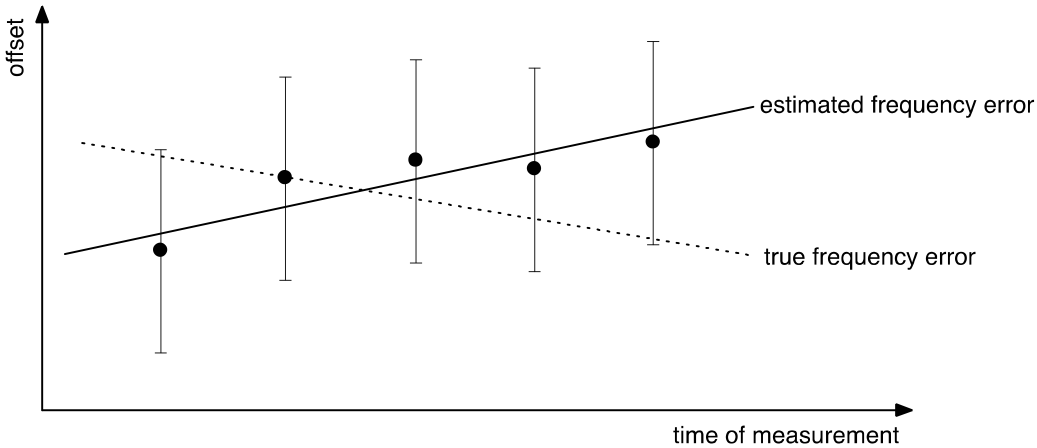


Figure C.8.3.5.1: Potential for inaccuracy when estimating frequency error

C.9 Status of Presentation and behaviour of TV Device

C.9.1 Primary aspect of status of presentation and behaviour of TV Device

Table C.9.1.1 describes examples of possible situations for the status of presentation of a TV Device and recommends an appropriate Primary aspect of status of presentation to be reported in each case as part of Content Identification and other Information (clause 5.6) that is provided by the TV Device through the Content Identification and other Information protocol (clause 6).

Table C.9.1.1: Recommendations on reporting primary aspect of status of presentations

Situation on TV Device	Primary aspect of status of presentation
Presenting a broadcast service.	okay
Presenting a broadcast service but currently paused by the user.	okay
Presenting a broadcast service but not showing audio or video due to loss of broadcast signal.	fault
In process of tuning to a broadcast service, not yet showing audio or video from the broadcast service.	transitioning
Presenting a broadcast service, with current and next programme guide information overlaid in a portion of the screen.	okay
Presenting streamed media, currently showing audio or video to the user from the media stream.	okay
Presenting stream media but currently paused, jumping to another point, fast-forwarding, rewinding or playing back in slow motion.	okay
Waiting for streamed media to fill a buffer before beginning presentation. Not yet presenting any audio or video to the user from the media stream.	transitioning

Situation on TV Device	Primary aspect of status of presentation
Temporarily paused while waiting for a buffer to fill before continuing presentation. Currently presenting a frozen video frame.	okay
Stopped presenting streamed media because of problems with streaming or rendering.	fault
Presenting some other application or service that does not have any of: content identity, a Timeline, the possibility of Trigger Event notification.	other (see note)
Displaying a full screen EPG and not showing the Timed Content.	epg (see note)
NOTE: This value for the primary aspect of status is an example of defining implementation specific values where none of "okay", "transitioning" or "fault" are applicable. Implementers can decide upon any value in these situations.	

This table is not an exhaustive list of situations. For example: it is likely that there are other situations in which it is appropriate to report values other than "okay", "transitioning" or "fault".

Whether it is appropriate, in such other situations, to report a Content Identifier or make Timelines available via the Timeline Synchronization protocol or notify of Trigger Events via the Trigger Event Notification Protocol is not discussed in the present document.

For the primary aspects "okay", "transitioning" and "fault" it is appropriate for the TV Device to report the Content Identifier of the Timed Content that it is presenting (or transitioning to present, or failing to present).

Where the primary aspect is "okay", it is usually appropriate for a Timeline to be available to any CSA that selects it via the Timeline Synchronization protocol (clause 9) and for Trigger Event Notifications to be provided. A media stream that is temporarily stopped while the TV Device is waiting to fill a buffer can report a Timeline as being available but paused.

If the primary aspect is "transitioning" or "fault" then it is not necessarily appropriate or practical because the TV Device is not yet necessarily receiving and decoding any signalling that conveys the Timeline or triggering of Trigger Events.

However, a possible exception is a TV Device that was receiving broadcast signals but then loses the signal and therefore reports the primary aspect as "fault". In this situation, the TV Device could still make a Timeline available by extrapolating an estimate of its progress (because the broadcast is still progressing even though the TV Device is not currently receiving it). Because the primary aspect is "fault", the CSA is expected to treat any Control Timestamps and Live Event notifications provided by the TV Device as best effort estimates that are unreliable.

C.9.2 Examples of extended aspects of status of presentation

Extended aspects of status of presentation provide an opportunity for the TV Device to provide more nuanced status information to a CSA. Here is an example:

EXAMPLE 1: The TV Device is presenting a broadcast service with subtitles enabled and audio muted. It reports the following as the status of presentation:
"okay subtitles muted".

EXAMPLE 2: The TV Device is presenting a broadcast service but only as a picture-in-picture rectangle in the corner of the screen while the majority of the screen is showing the electronic programme guide (epg).
The TV Device reports the following status of presentation that describes which is the main and the picture-in-picture parts:
"okay pip.main:epg pip.sub:presentation".

C.10 Trigger Event Presentation Time (informative)

C.10.0 Trigger Event Presentation Time Introduction

In a Trigger Event Notification message, the TV Device informs the CSA as to the anticipated time (in the very near future) at which the event is due to start.

The reference point for presentation of a given moment of Timed Content, or a Trigger Event associated with the Timed Content, is the point at which light and sound emerge from the TV Device.

The time is expressed in terms of the Wall Clock and ignores the playback speed of the Timed Content at the TV Device, instead assuming that is a normal speed of x1. This is suitable for the most common viewing situations, where presentation will almost always be proceeding at normal speed.

C.10.1 Calculation of presentation time by the TV Device

In the following explanation the Timed Content being presented by the TV Device is assumed to have a notional Timeline. The scale of that Timeline is assumed to be 1 unit equalling 1 second for convenience.

The TV Device has knowledge of the current playback speed and presentation timing relationship between Wall Clock and Timeline time. This knowledge can conceptually be represented by a Control Timestamp:

- w_{CT} is equivalent to wallClockTime property from the Control Timestamp
- c_{CT} is equivalent to contentTime property from the Control Timestamp
- m is equivalent to timelineSpeedMultiplier from the Control Timestamp
- 1 000 000 000 is the number of ticks of the Wall Clock per tick of the Timeline

The signalling for a Trigger Event is detected in the media stream for the Timed Content. This signalling is located at (or associated with) a point corresponding to t_{CALC} on the Timeline of the Timed Content. The signalling indicates that the Trigger Event will start d units of the Timeline after the point t_{CALC}.

NOTE 1: For some types of Trigger Event signalling (e.g. DSM-CC "do it now" stream events), the location on the Timeline of the Timed Content is not accurately signalled. In these situations the location can be approximated by the TV Device.

The TV Device first translates t_{CALC} to a Wall Clock time w_{CALC}:

$$w_{CALC} = (t_{CALC} - c_{CT})\frac{1\ 000\ 000\ 000}{m} + w_{CT}$$

This calculated Wall Clock time is the calculation reference point that is included in the Trigger Event Notification message. This takes into account the current playback speed, as represented by the timelineSpeedMultiplier.

NOTE 2: If playback speed is not taken into account, then the CSA cannot perform the reverse calculation unless it knows which instance of a Control Timestamp was used to calculate w_{CALC}.

The TV Device can now also calculate the presentation time w_{TEN} for the Trigger Event in terms of the Wall Clock. This is relative to the calculation reference point and assumes a playback speed of x1.

$$w_{TEN} = w_{CALC} + 1\ 000\ 000\ 000\ d$$

If signalling of Trigger Events is delivered separately from the actual media stream (e.g. within an MPD for an MPEG DASH presentation) then the TV Device is free to choose an arbitrary value of t_{CALC}. It is recommended to choose a value for t_{CALC} that minimizes d. The calculations described above remain the same and d still represents the difference between the signalled time for the event and the chosen value of t_{CALC} in terms of the Timeline for the Timed Content.

C.10.2 Calculation by the CSA to adjust for playback speed

If a CSA is also using the CSS-TS interface, then it is possible for the CSA to calculate the time that the Trigger Event is anticipated to start in terms of the Synchronization Timeline. In doing so, it will be able to compensate for when the playback speed is not x1 - as indicated by the timelineSpeedMultiplier property of Control Timestamp messages.

The procedure a CSA to calculate the time in terms of the Synchronization Timeline is explained below. In the equations shown it is assumed that Wall Clock time is measured in units of nanoseconds (1 tick = 1 nanosecond).

The CSA has recently received a Control Timestamp message via the CSS-TS interface from the TV Device, and knows the scale (tick rate) of the Synchronization Timeline:

- w_{CT} = wallClockTime property from the most recent Control Timestamp message
- c_{CT} = contentTime property from the most recent Control Timestamp message
- m = timelineSpeedMultiplier from the most recent Control Timestamp message
- unitsPerSecond and unitsPerTick describe the Synchronization Timeline

For readability, a multiplier is defined that equals the number of ticks of the Synchronization Timeline per tick of the Wall Clock when the timelineSpeedMultiplier is 1:

$$r = \frac{1}{1\,000\,000\,000} \frac{unitsPerSecond}{unitsPerTick}$$

The CSA now receives a Trigger Event Notification message:

- w_{TEN} = presentationWallClockTime from the Trigger Event Notification message
- w_{CALC} = calculationWallClockTime from the Trigger Event Notification message

The CSA first calculates the delta between the presentationWallClockTime (the anticipated start of the Trigger Event) and the calculationWallClockTime (the time at which the TV Device made this observation). This is calculated in terms of the Wall Clock:

$$w_{delta} = w_{TEN} - w_{CALC}$$

The delta can then be translated to be in terms of the Synchronization Timeline while making the assumption that the playback speed is x1:

$$\begin{aligned} c_{delta} &= w_{delta} \times r \\ &= (w_{TEN} - w_{CALC})r \end{aligned}$$

The CSA next translates the calculationWallClockTime to the Synchronization Timeline using the most recent Control Timestamp and knowledge of the rate of the Synchronization Timeline. This time the true speed of presentation, as represented by the timelineSpeedMultiplier, is taken into account:

$$c_{CALC} = (w_{CALC} - w_{CT})(r \times m) + c_{CT}$$

The CSA then adds the previously calculated delta to determine the point on the Synchronization Timeline at which the Trigger Event is anticipated to start. This gives the anticipated start time of the Trigger Event in terms of the Synchronization Timeline:

$$c_{TEN} = c_{CALC} + c_{delta}$$

The complete calculation described above can be summarized in the following single equation:

$$c_{TEN} = (w_{CALC} - w_{CT})(r \times m) + c_{CT} + (w_{TEN} - w_{CALC})r$$

If the CSA receives a new Control Timestamp between now and the time at which the Trigger Event starts, then the CSA will be able to recalculate the Wall Clock time at which the Trigger Event is due to start. It can do this because it now knows where the anticipated start of the Trigger Event lies on the Synchronization Timeline.

Annex D (informative): Data model diagrams description

D.1 General

This informative annex describes the UML-based structure diagrams used in the present document. This clause provides an informative explanation of the diagrams used.

D.2 Objects

An object is the basic building block of the data model. As used in the present document, the name of the object is shown in bold at the top and the attributes of the object are shown in normal text below. An example is shown in figure D.2.1, where an object called "Content" is defined, that has the attributes "Title", "Kind" and "Date of Production".

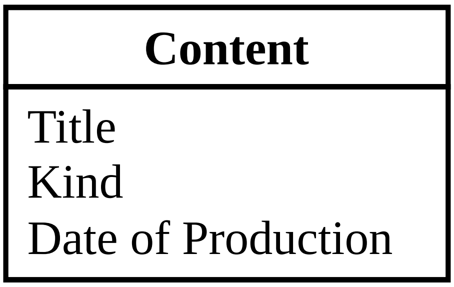


Figure D.2.1: Example Object

D.3 Links and associations

An association is a relationship between two objects, and is shown by a line linking the two entities. The ordinality of the relationship is indicated by the optional number at each end of the line. If no number is given, then an arbitrary ordinality is inferred. An example is shown in figure D.3.1, where one "Schedule" is associated with an arbitrary count of "Content", including the possibility that a "Schedule" may be associated with no "Content" (an ordinality of zero).

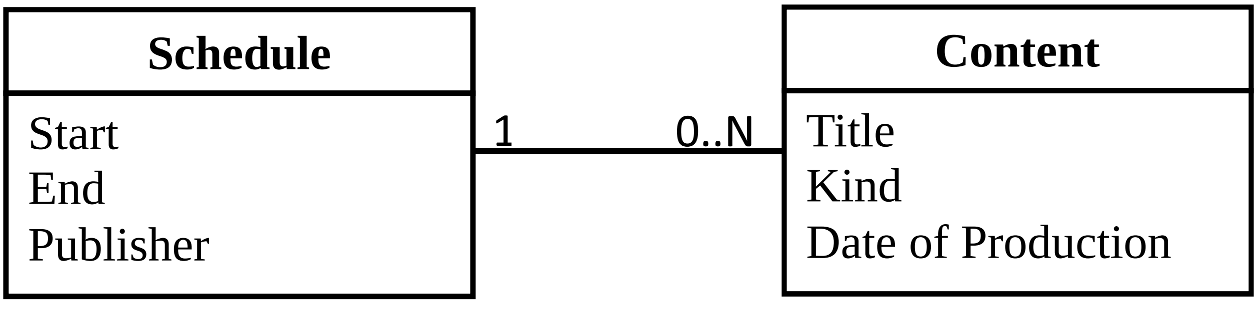


Figure D.3.1: Example association between objects

D.4 Aggregation

Aggregation represents a "part of" relationship between two objects. One entity is shown to be aggregated within another entity by a line linking the two entities with a hollow diamond at the end touching the entity that is aggregating the other entity within it. An example is shown in figure D.4.1, where one "Car" aggregates an "Engine" because an engine is considered to be a part of a car.

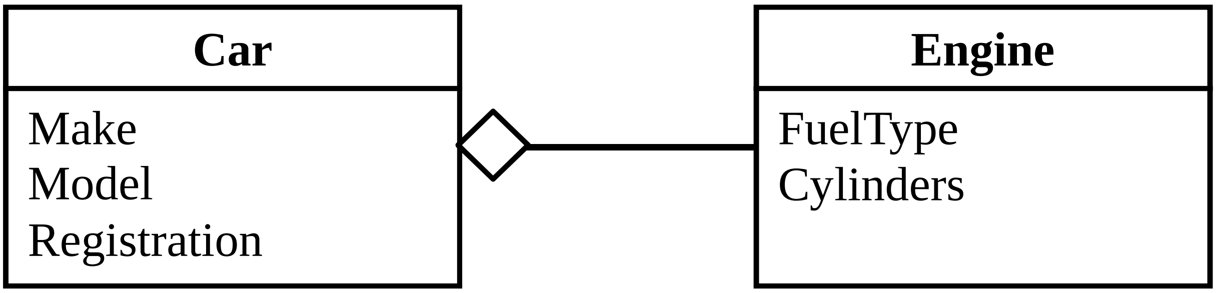


Figure D.4.1: Example aggregation of one object within another

Aggregation is used in the present document to indicate that one entity is conveyed nested within another entity for the purposes of structuring and representing data in a particular format.

Annex E (normative): Signalling of MRS and CI ancillary data in DASH MPDs

E.1 General

The present document makes use of the carriage of the MRS URL and optionally ancillary content identification information carried in the content delivery system. For DVB defined transport stream delivered content and IP carried content clauses 5.2.3 and 5.6 make reference to other specifications ETSI EN 300 468 [13] and ETSI TS 102 034 [5] for the carriage of this information in the content delivery system. For DASH delivered content, the carriage of this information is defined in this annex.

The definition of the carriage of this information in this clause:

- specifies the additional elements that may be carried in the MPD schema;
- the format of the data they carry;
- the locations where these additional elements may occur;
- the namespace with which these elements are associated.

E.2 DASH MPD Schema Extensions

E.2.1 XML Element Namespace

The added elements described below shall use the namespace "urn:dvb:css:dash:2014". This refers to the schema defined in clause E.2.4. The XML excerpts below assume that the identifier "dvb" is associated with this namespace.

E.2.2 MRS URL

Where an MRS URL is to be carried for DASH delivered content, it shall be carried as the element mrsUrl, with a type mrsUrlType as a child of the MPD element. The MRS element shall use the namespace indicated in clause E.2.1. Any value carried shall comply with all constraints detailed in the present document and in clause 6.4.14 of ETSI EN 300 468 [13].

```
<element name="mrsUrl" type="dvb:mrsUrlType"/>
```

E.2.3 CI Ancillary Data

Where ancillary data is to be provided, it shall be carried as the element ciAncillaryData with a type ciAncillaryDataType as a child of either the MPD or the Period element. The ciAncillaryData element shall use the namespace as indicated in clause E.2.1.

NOTE: This is equivalent to the ancillary data as defined in clause 6.4.1 (CI ancillary data descriptor) of ETSI EN 300 468 [13] and as used in clause 5.2.3.5.

```
<element name="ciAncillaryData" type=" dvb:ciAncillaryDataType"/>
```

E.2.4 Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:dvb:css:dash:2014" xmlns:dvb="urn:dvb:css:dash:2014"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="mrsUrl" type="dvb:mrsUrlType"/>
    <xs:element name="ciAncillaryData" type="dvb:ciAncillaryDataType"/>
    <xs:complexType name="mrsUrlType">
        <xs:simpleContent>
            <xs:extension base="xs:anyURI">
                <xs:attribute name="minOccurs" fixed="0"/>
                <xs:attribute name="maxOccurs" fixed="1"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="ciAncillaryDataType">
        <xs:simpleContent>
            <xs:extension base="xs:base64Binary">
                <xs:attribute name="minOccurs" fixed="0"/>
                <xs:attribute name="maxOccurs" fixed="1"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
```

E.3 Example MPD

The following example MPD is based on the example of annex G.1 of ISO/IEC 23009-1:2014 [17]. This shows the presence of the mrsUrl element and a ciAncillaryData element as part of the Period.

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:mpeg:dash:schema:mpd:2011"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd" xmlns:dvb="urn:dvb:css:dash:2014"
type="static" mediaPresentationDuration="PT3256S" minBufferTime="PT1.2S"
profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">
    <BaseURL>http://cdn1.example.com/</BaseURL>
    <BaseURL>http://cdn2.example.com/</BaseURL>
    <Period>
        <!-- English Audio -->
        <AdaptationSet mimeType="audio/mp4" codecs="mp4a.0x40" lang="en" subsegmentAlignment="true"
subsegmentStartsWithSAP="1">
            <ContentProtection schemeIdUri="urn:uuid:706D6953-656C-5244-4D48-656164657221"/>
            <Representation id="1" bandwidth="64000">
                <BaseURL>7657412348.mp4</BaseURL>
            </Representation>
            <Representation id="2" bandwidth="32000">
                <BaseURL>3463646346.mp4</BaseURL>
            </Representation>
        </AdaptationSet>
        <!-- French Audio -->
        <AdaptationSet mimeType="audio/mp4" codecs="mp4a.40.2" lang="fr" subsegmentAlignment="true"
subsegmentStartsWithSAP="1">
            <ContentProtection schemeIdUri="urn:uuid:706D6953-656C-5244-4D48-656164657221"/>
            <Role schemeIdUri="urn:mpeg:dash:role:2011" value="dub"/>
            <Representation id="3" bandwidth="64000">
                <BaseURL>3463275477.mp4</BaseURL>
            </Representation>
            <Representation id="4" bandwidth="32000">
                <BaseURL>5685763463.mp4</BaseURL>
            </Representation>
        </AdaptationSet>
        <!-- Timed text -->
        <AdaptationSet mimeType="application/ttml+xml" lang="de">
            <Role schemeIdUri="urn:mpeg:dash:role" value="subtitle"/>
            <Representation id="5" bandwidth="256">
                <BaseURL>796735657.xml</BaseURL>
            </Representation>
        </AdaptationSet>
        <!-- Video -->
        <AdaptationSet mimeType="video/mp4" codecs="avc1.4d0228" subsegmentAlignment="true"
subsegmentStartsWithSAP="2">
            <ContentProtection schemeIdUri="urn:uuid:706D6953-656C-5244-4D48-656164657221"/>
            <Representation id="6" bandwidth="256000" width="320" height="240">
                <BaseURL>8563456473.mp4</BaseURL>
```

```
            </Representation>
            <Representation id="7" bandwidth="512000" width="320" height="240">
                <BaseURL>56363634.mp4</BaseURL>
            </Representation>
            <Representation id="8" bandwidth="1024000" width="640" height="480">
                <BaseURL>562465736.mp4</BaseURL>
            </Representation>
            <Representation id="9" bandwidth="1384000" width="640" height="480">
                <BaseURL>41325645.mp4</BaseURL>
            </Representation>
            <Representation id="A" bandwidth="1536000" width="1280" height="720">
                <BaseURL>89045625.mp4</BaseURL>
            </Representation>
            <Representation id="B" bandwidth="2048000" width="1280" height="720">
                <BaseURL>23536745734.mp4</BaseURL>
            </Representation>
        </AdaptationSet>
        <dvb:ciAncillaryData >TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQK</dvb:ciAncillaryData>
    </Period>
    <dvb:mrsUrl xmlns="urn:dvb:css:dash:2014">http://broadcaster.com/mrsservice</dvb:mrsUrl>
</MPD>
```

Annex F (normative): Signalling of MRS and CI Ancillary data in DVB IPTV Service Discovery and Selection XML

F.1 General

This annex provides details of how the MRS location and CI ancillary data are signalled in DVB IPTV systems that use the Service Discovery and Selection XML as defined in ETSI TS 102 034 [5]. It is expected that a future version of ETSI TS 102 034 [5] (version 1.6.1) will include the mechanisms defined in this annex and associated attachments. The functionality defined in future versions is expected to be compatible with that defined in this annex. When the future version of ETSI TS 102 034 [5] is available that includes this functionality, the mechanisms in this annex shall be deprecated in favour of those defined in ETSI TS 102 034 [5].

F.2 XML Namespace

The added elements described below shall use the namespace "urn:dvb:metadata:iptv:sdns:2014-1". This refers to the schema referred to in clause F.5.

F.3 MRS (URILinkage)

The MRS URL is a URL that provides the location of the MRS server, as defined in clause 7. Where a MRS URL is to be carried for SDnS described content, it shall be carried as the element URILinkage, with a type URILinkageType as defined in the schema in clause F.5 and in the locations defined in the schema in clause F.5. Any value carried shall comply with all constraints detailed in the present document and in clause 6.4.14 of ETSI EN 300 468 [13].

F.4 CI Ancillary Data

The CI Ancillary Data is a base 64 encoded string that defines the CI ancillary data, as defined for use in clause 5.2.3. Where CI Ancillary data is to be carried for SDnS described content, it shall be carried as the element ciAncillaryData, with a type ciAncillaryDataType as defined in the schema in clause F.5 and in the locations defined in the schema in clause F.5.

NOTE: This is equivalent to the ci_ancillary_data as defined in clause 6.4.1 (CI ancillary data descriptor) of ETSI EN 300 468 [13] and as used in clause 5.2.3.5.

F.5 Schema

The schema for the Service Discovery and Selection XML is contained in ts_10328602v010201p0.zip which accompanies the present document.

Annex G (informative): Change History

Date	Version	Information about changes
May 2015	1.1.1	First publication of the TS.
May 2017	1.2.1	The reference to "Delivery of Timeline for External Data" (TEMI) has been updated. Incorrect use of upper case fixed in content IDs in examples in clauses 5.2.2 and 7.5. A cross reference in clause 5.5.9.4 has been corrected to point to clause 5.5.9.6. Descriptor parsing requirements in clause 5.2.3.1 and clause 5.2.3.5 have been relaxed: **ci_ancillary_data** descriptors in the BAT do not need to be parsed unless the TV Device installed the current service from the BAT; and **content identifier descriptors** do not have to be parsed unless platform requirements state that CRIDs are signalled. A clarifications has been made in clause 5.2.3.5 that key-value pairs (in the query part of a content ID URI) are to be included even if the value is zero-length. In clause 5.2.4, the terminology when referring to MPD anchors has been corrected and the inclusion of a Period ID in the content identifier for DASH presentations has been clarified. This includes explicit handling of the situation where a Period has no ID. TSAP timelines have been deprecated and removed by removing the **timeline selector** from table 5.3.3.1, by removing reference to it from clause 11.1 and by voiding clause 11.2. References to TSAP have been replaced with TEMI in example JSON in clause A.2.2.1 and in diagrams and calculations in clauses C.5.2 and C.6.2. A requirement has been added to clause 5.3.4 defining handling of PTS wrapping. In clause 5.3.5, the ISOBMFF box names from which **timescale** elements are extracted have been corrected and clarified. The flow diagram in figure 5.5.8.1 has been corrected. The JSON format of an MRS response has been adjusted. The **index** property has been removed from the JSON for a Material in clause 5.5.9.2 and the **materials** property defined in clause 7.4 has been changed from an array to an object where the **index** becomes the object property names. References to this in clause 5.5.9.6 and the schema defined in clause A.1.3 have also been adjusted to match. Example JSON in clauses A.2.2.1, A.2.2.2 and B.2.2.3 have also been updated. The text of clause 5.6.1 and title of clause 5.6.6 have been corrected to refer to the list of timelines as being a list of **timeline options**. Clause 5.6.2 has been rewritten to clarify MRS URL carriage in DVB services: scoping rules are now defined for the **uri_linkage_descriptor** when carrying MRS URL; and it is no longer required to search the BAT for the **uri_linkage_descriptor** unless the TV Device installed the current service from the BAT. Property names have been corrected in clause 7 and annex A: from **repollInterval** to **repollingInterval**; from **updatedTimelineSync** to **updateTimelineSync**; and from **updateMaterials** to **updateMaterial**. A cross reference to clause 5.6.2 has been added to clause 7.3.2. Clause 8.3 has been clarified to better describe the permitted values for the **originate_timevalue** field and the units for all timevalue fields have been corrected from seconds to nanoseconds. In clause 11.3.3 the field name **tick_rate** has been corrected to **timescale** and additions have been made to define behaviour of TEMI timeline when the pause field set and when PTS values wrap. Clause 11.3.4 and clause 11.3.5 have been added to define recommendations for behaviour when determining if TEMI timeline signalling is present or not present.

Date	Version	Information about changes
		Missing "id" property names have been added to the JSON example in clause B.2.2.3. The use of mathematical symbols in text in clause C.8.3.2 and clause C.8.3.3 have been corrected to match those used in equations.

History

Document history		
V1.1.1	May 2015	Publication
V1.2.1	August 2017	Publication