Text

Description automatically generated ISO/IEC JTC 1/SC 29/WG 2 N169

**ISO/IEC JTC 1/SC 29/WG 2  
MPEG Technical requirements   
Convenorship: SFS (Finland)**

**Document type:** Output Document

**Title:** Use Cases and Requirements for Encoder and Packager synchronization

Status: Approved

Date of document: 2021-01-21

Source: ISO/IEC JTC 1/SC 29/WG 2

Expected action: None

Action due date: None

No. of pages: 10 (with cover page)

Email of Convenor: igor.curcio@nokia.com

Committee URL: <https://sd.iso.org/documents/ui/#!/browse/iso/iso-iec-jtc-1/iso-iec-jtc-1-sc-29/iso-iec-jtc-1-sc-29-wg-2>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 02**

**MPEG TECHNICAL REQUIREMENTS**

**ISO/IEC JTC 1/SC 29/WG 02 N0169**

**January 2022, Virtual**

|  |  |
| --- | --- |
| **Title** | **Use Cases and Requirements for Encoder and Packager Synchronization** |
| **Source** | **WG 02, MPEG Technical Requirements** |
| **Status** | **Approved** |
| **Serial Number** | **21299** |

# Introduction and problem description

This document presents draft use cases and requirements defined for encoder and packager synchronisation and related asset storage. The use cases and requirements have been derived from publicly held encoder synch workshop and were further developed in SC 29 WG03 and WG02.

## Introduction

In case of high-value wide-distribution channels and events there is a need for a very high degree of reliability. When reliability is addressed from the adaptive streaming point of view, the link between the origin and the streaming client gets most of the attention. However, this link is not the only source of a failure, and loss of a mezzanine feed or a encoder has catastrophic consequences as it affects all viewers of the channel.

Using geographically distributed encoding facilities is a common mitigation technique. As a US-centric example, a mezzanine source from Tokyo can be sent to a data center location in San Francisco (West Coast) and another one to a data center in New York (East Coast). This makes the service resilient to loss of a encoder or of a complete data center (e.g., due to a power outage).

Additional layers of reliability can be achieved by using different mezzanine sources – e.g., one (primary) data centre is using high-quality high-bitrate lossless mezzanine source over dedicated fiber, while the other is receiving a compressed lower-quality mezzanine source via satellite or the public Internet.

The ultimate goal of such designs is to sustain loss of a single encoder or its network connectivity, a loss or a temporary glitch in a mezzanine source, or even a loss of complete data center without a service disruption. An additional goal may also be load balancing where the client may elect to download a segment produced by either of the redundant encoders.

In order to allow glitch-less switchovers between redundant encoder outputs we need support for encoder synchronization and guidelines for redundant operation. A current way to achieve this is by mandating identical outputs. There is no need for the outputs to be bit-identical – what is needed is complete time alignment. This means that segments output by each encoder have to start at the same frame and have identical earliest presentation time. The encoders may output into multiple origins, and even be accessed through different CDNs. One way how this is supported in MPEG-DASH is by using multiple BaseURL’s in the Media Presentation Description. Another use case is multi-encoder setup. This can happen when a single server chassis is capable of encoding a subset of representations, but not the complete ladder.

Lastly, pure encoder synchronization is insufficient – it is important to have functionally identical MPDs, this packager synchronization, is also in scope of this exploration. This document summarizes the status of the ongoing exploration on these problems.



Figure 1 Synchronized encoder ecosystem with CMAF as an example output format of the encoder and DASH from the origin (HLS or smooth streaming are other possible client protocols, dash is just an example), the orange line indicates some of the scope of the Encoder synchronization standardization for inter encoder communication. The intermediate formats for encoder output are also in scope.

## Problem description

In practice, multiple ingest sources (like encoders or encoder/packager combos) and receiving entities (like packagers or origin servers) are often used for redundancy purposes. This requires that multiple encoders and origins/packagers to work together in a redundant workflow to avoid interruptions when some of the components fail. An example workflow is shown below, where there are three encoders and two packagers providing redundancy and preferably seamless recover from a failover event.

Note that encoder/packager synchronization should be codec and potentially format agnostic. That is, whatever methods are developed and adopted by the standard should work for any modern codec (AVC, HEVC, VVC, etc.) and format (DASH, CMAF, HLS).

Figure 2 shows a common architecture for encoder synchronisation. Encoder A, B and C send their outputs to origins A and B. The output from the encoders A, B and C is synchronised referring to encoder synchronisation. This synchronisation mainly encompasses timestamp/frame and segment boundary alignment, or supplementary information to achieve this synchronisation.

The origin/packagers A and B respond to client requests and their output needs to be synchronised as well in order to avoid problems at clients. This is referred as packager synchronisation. The main issue here is producing consistent manifest updates and responses to clients between the packagers/origins.

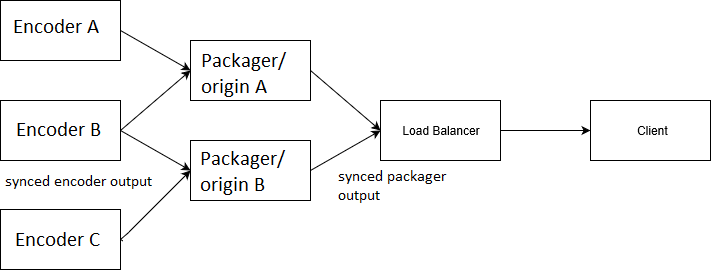


Figure Architecture with dual encoders and orign/packagers followed by a load balancer

## Terminology/Systems Components

In this text an encoder is an entity used to (re-) encode a video stream using a codec, the term may refer to both encoding of an uncompressed mezzanine source, or transcoding from a compressed source into 1 or more different bit-rates.

A packager in this text is assumed to be the entity that applies the (re-) packaging to mp4, fragmented mp4, or other container formats and generation of a streaming manifest. A packager may be integrated in an encoder or in an origin server. The packager may also apply encryption to enable DRM services. Other functions in the packager may include ad insertion or splicing of content with ad content.

An Origin Server in this text refers to the server that can answer requests for media content from clients or intermediate entities such as proxies and content delivery networks. In the common case of ABR streaming it would answer to HTTP GET and HEAD requests. In other cases it may use other protocols. The Origin server is usually separated from the encoder for scalability reasons.

## Scope and Terminology

Requirements that state that the specification shall/should support certain features implies explicit support for these features in the syntax and specification text. Requirements that state that the specification shall/should enable certain use cases or situations imply that based on the specification such can be achieved, but this may not be explicit in the specification text.

# Use cases

## Use cases for encoder synchronization

### Source input of Encoder carries timing

In many cases the source feeds to an encoder carry timing information. In the case of MPEG-2 TS it can be carried using means such as SCTE 35 time\_signal() or CableLabs EBP. In the case of HEVC or AVC mezzanine streams timing can be carried in one or more SEI messages. There are a multitude of ways absolute time can be carried together with MPEG-2 TS. Having established and a precise correspondence between absolute time and frame, can be useful for achieving synchronization and alignment of key frames.

### Identical (MPEG-2 TS) input to encoders

If the input is the same (e.g. an MPEG-2 TS stream), passing through PTS values can establish the same correspondence in the output. Then, the same approach of either creating a PTS-to-segment boundary correspondence may apply. In this case the presentation time stamps from the input may be re-used.

### Synchronized input to encoders

If (a) the same input enters the encoders at the same time (up to a few milliseconds difference), and (b) encoders have clocks synchronized to the same time source, absolute frame acquisition time can be similarly used for coordinating segment boundaries.

### Absence of timing information in the input to encoders

In absence of either timing in input streams or precise clock and input synchronization, boundary synchronization can be achieved by examining the frames. If the frames are bitwise-identical, cryptographic hashes such as MD5 can be used to identify frames. If the frames are different (e.g., one source is high-rate MPEG-2 video off the satellite, the other is JPEG2000 over direct fiber).

### Input over different transmission media

An encoder may get input from both a fiber link and a secondary satellite link. In such a scenario both the compressed signal and the transmission characteristics are different. In this case synchronization of the streams is necessary to use one signal as a backup of the other signal. Thus, one encoder may have delayed input compared to another encoder.

### Partial encoding in distributed encoders

In demanding bit-rate ladders, such as with 4K and 8K resolutions, different distributed encoders may encode part of the total bit-rate ladder. In this case encoder A may encode the 4K and 8K while encoder B encodes the audio, timed text and lower resolution bit-rates. In this case encoders are responsible for encoding only a part of the presentation.

### Redundant operation and failover of synchronised encoders

Encoders may fail and restart again later, or join and/or leave a session gracefully (as indicated by the user). This should not affect the operation and playback operation of clients.

NOTE: There was some discussion on the cases when inter-encoder communication is possible and when this is not possible, both cases should be considered in the different use cases as mandated by the requirements presented later in this document.

## Use cases for packager and/or origin server synchronization

### Multiple encoders transmitting a full presentation to one or more origins or packagers

In this use case, more than one encoder(s) are pushing full presentations to one or more origin or packager servers. Packagers may be in different domains and not aware of each other. Manifests and segments output by the packager must be inter-changeable, i.e. not breaking or retro-active changes. Packagers may fail and re-join later. Also, the case should be handled by packagers when encoders fail and re-join later. In this case encoder synchronisation is assumed.

Regarding origin synchronization, the case when a segment is missing on one origin but not the other should be addressed and considered in the solution.

Architecturally, preference for solutions that do not affect clients too much or need advanced features are preferred. This is because it is preferred by operators to target as many devices as possible and achieve synchronised and redundant setups without the need for new client device updates.

All packagers should produce inter-changeable manifests that would not break client playback when interchanged in a follow up request in an MPD update for example. This means for example not retro-actively changing the segment timeline (e.g. removing segments from the manifest). In addition, a static manifest converted from a dynamic manifest is possible, but not the other way around. In case a dynamic manifest is converted to a static manifest, this should be the latest manifest published by different packagers.

### Multiple encoders pushing partial presentations to one or more origins or packagers

In this use case, more than one encoder is pushing full presentations to one or more origin or packager servers. Origins/packagers may be in different domains and not aware of each other. Packagers produce manifests and segments that are inter-changeable., i.e., no retro-active changes. Origins and packagers may fail and re-join later. Encoders may fail and re-join later. In this case encoder synchronisation is assumed.

### Packager synchronisation without encoder synchronisation

In this use case packagers are responsible for the final alignment of the content segments, fixing some possible misalignments introduced at the encoder. For the rest similar conditions as in the previous use cases apply.

### Packager synchronisation with support for ad-insertion solutions

In this use case packagers are also responsible for doing ad-insertion and communicating with ad insertion solutions. Examples could be period replacement or specific interpretation of ad-insertion signalling. The signalling and ad-insertion operations in this use case do not break the packager synchronisation and the timeline generated.

# Requirements

## General Specification requirements

1. The specification shall be a self-contained document standardized in ISO/IEC containing both guidelines and normative statements to achieve the requirements in this document.
2. The specification shall emphasize and focus on the use of existing MPEG standards
3. The specification shall not emphasize and focus on specific protocol related aspects not related to MPEG standards. The assumption is that organisations outside of MPEG will develop such protocol specific aspects.

## Encoder Synchronisation requirements

1. The specification shall support cases when there are more than one redundant encoder to synchronize.
2. The specification shall enable the case when sources to encoders may be delayed relative to each other (e.g., direct fiber vs transcoded satellite feed).
3. The specification shall enable the case when the sources to encoders may be differently encoded. For example, the primary sources may be compressed with a lossless codec such as JPEG2000 or JPEG-XS while the secondary source can be a high-rate HEVC video
4. The specification shall enable the case when there is not any time signalling present in the (mezzanine) input signal to the encoder
5. The specification shall support the case when encoders may fail and restart (re-join) later again, or encoders join or leave a session again.
6. The specification shall not require proprietary boxes in the ISO Base Media file format.
7. The specification shall enable using multiple redundant inputs without the same presentation time stamp for corresponding frames and/or different presentation times at the segment boundaries.
8. The specification shall provide methods for achieving synchronisation of the output of the encoder.
9. The specification shall support distributed encoding of parts of a bit-rate ladder or parts of a presentation.
10. The specification shall enable the case when segments produced by different encoders for the same interval are not bit-identical.
11. The specification shall define methods for identification of segments covering interchangeable content/encodings that are not bit-identical.
12. The specification shall define guidelines for synchronisation of audio and video segments, taking into account specific aspects of audio such as support for pre-roll and encoding from different samples introducing different audio encoder/decoder state.
13. The specification shall not require exact system clock synchronisation (< 1 ms) of the systems/hardware of the encoder.
14. The specification shall enable encoder and packager synchronisation support when the system clock synchronisation is within reasonable bounds; bounds up to at least including +-100ms shall be supported (as this can be achieved in most practical (cloud) setups).
15. The specification shall support leap seconds.
16. The specification shall define how the media presentation times, based on internal media clocks are calculated with the goal of synchronisation of different tracks and video frames.
17. The specification shall support encoder synchronisation on output with frames synchronised based on presentation time stamps and aligned segment boundaries.
18. The specification shall support at least a subset of input formats like SDI, HD-SDI, SMPTE 2110, MPEG-2 TS and other such as RTMP, RIST.
19. The specification shall enable the case when signals in the input to the encoder arrive from different media and with a different latency (e.g. primary over fiber, secondary over satellite etc...) and the secondary signal is to be used as a backup.
20. The specification shall enable both Video on Demand and Live use cases
21. The specification shall enable the recording and storing of encoder output for a live use case
22. The specification shall define/support the option to target optimization for the Video on Demand case that do not apply to live.
23. The specification shall support the case where the communication between encoders is enabled.
24. The specification shall support the case when there is no communication between encoders.
25. The specification shall support additional functionality for ad insertion solutions, such as the insertion of splice points, IDR frames and/or segment boundaries. These shall not break the solution for encoder synchronisation.
26. The specification shall support segment boundary alignment of different tracks at the output.
27. The specification shall support synchronised timed metadata carriage.

## Packager Synchronisation requirements

1. The specification shall enable packager synchronisation for geo-distributed setups, where the output manifest and segments of the packagers does not introduce inconsistencies in the timeline or retro-active changes.
2. The specification shall support Packager synchronisation when encoder synchronisation is applied. Such simpler packagers that need not rewrite the timestamps are typically more robust.
3. The specification shall enable Packager synchronisation when encoder synchronisation is not (fully) applied, and the packager may apply some changes to the timeline.
4. The specification shall describe the need for atomic read/write operations and trade-off between consistency, availability and tolerance to partition. Most packagers need to index segments using a segment index or a database, the specification shall describe in detail how such operations may affect packager synchronisation.
5. The specification shall enable packager synchronisation for integer and non-integer (e.g., fractional) frame rates.
6. The specification shall not preclude additional communication with ad insertion solutions and shall not break client playout of the output formats that target the player in such cases.
7. The specification shall support synchronisation and segment boundary alignment of timed metadata and timed text as to provide timed text and timed metadata to clients.
8. The specification shall not require strict system clock synchronisation of the system running the packager. Systems shall at least support system clock synchronisation boundaries up to +-100 ms.
9. The specification shall support the case where SCTE-35 metadata is packaged, and segments or timelines may need to be altered based on SCTE-35 based information
10. The specification shall not require the use of client-side solutions for mitigation upstream failures (e.g., multiple base URL client steering), as these are not implemented consistently throughout devices.
11. The specification shall describe specific aspects related to audio/video sync for different audio codecs. This includes sample rate, edit list, pre-roll audio encoder state that may work slightly different as when compared to the video coder.
12. The specification shall at least support popular output streaming formats to players such as MPEG-DASH and HTTP Live Streaming.
13. The specification shall support cases when inter-packager communication is possible.
14. The specification shall support cases when there is no inter-packager communication possible.
15. The specification shall support the case when different packagers or origins may have a missing segment in one receiver but not in the other.
16. The specification shall support the case when packagers may fail and restart (re-join) later again, or packagers join or leave a session again.
17. The specification shall be agnostic to encryption or DRM systems, details on how this is achieved shall be provided in the specification text.

## Requirements on communication between Encoder and packager/origin

1. The specification shall support ingest/communication between an encoder and packager/origin based on well-defined existing industry specifications targeting such applications.
2. The specification shall enable push-based transmission of media data segment from the encoder to the origin/packager.

## Requirements for Content storage

The specification shall also meet the following requirements for content storage.

1. The specification shall support describing content stored for either a VoD or Live presentation using an external file, e.g., a specific manifest
2. The specification shall support describing content stored for either a VoD or Live presentation using an internal identifier, e.g., a stream or group identifier
3. The specification shall enable both single track per file and single segment per file storage
4. The specification shall support sequences of presentations. The specification shall enable linking these tracks explicitly or implicitly.
5. The specification shall support directory structure\ organisation of the storage archive.
6. The specification shall support storage of timed metadata (both inband emsg and timed metadata track).
7. The specification shall support/define the option of signalling or fixing gaps, if content was received with gaps those gaps may either be repaired or explicitly signalled.
8. The specification shall support/define the option of repairing, appending, replacing parts of content.
9. The specification shall describe in guidelines how support in typical cloud storage systems may work and how the solution fits with existing APIs for cloud-based storage solutions
10. The specification shall support efficient and fast indexing of segments and content, for live, VoD and Live-to-VoD cases.
11. The specification shall support the recording and storing of encoder/packager output for a live use case.
12. The specification for content storage shall be based on existing MPEG specifications to the largest extent possible