DASH Industry Forum

Server-Side Watermarking
DASH-IF End-to-End Architecture

DASH-IF Webinar – February 10, 2023

Presenters: Gwenaël Doërr (Synamedia), Marcelo San Martin (Akamai), Nicolas Weil (AWS), Vladimir Zivkovic (Irdeto), and Laurent Piron (Nagravision)

Contents

Refresher on forensic watermarking

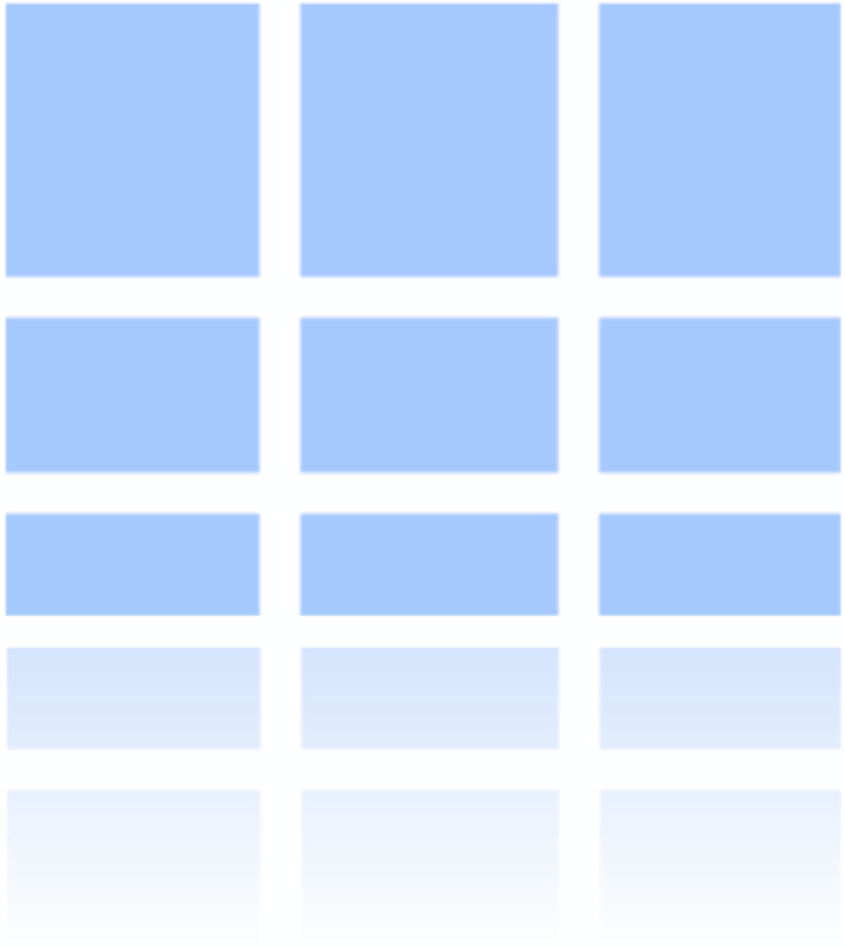
- DASH-IF general architecture
- Feedback of the 1st community review

Review of the modifications

Demo

Conclusion

- 2nd community review
- Next steps



High Level View

Gwenaël Doërr (Synamedia)

Forensic Watermarking

Imperceptible unique identifier embedded within each content copy for the purpose of tracking individual copies to deter unauthorized redistribution

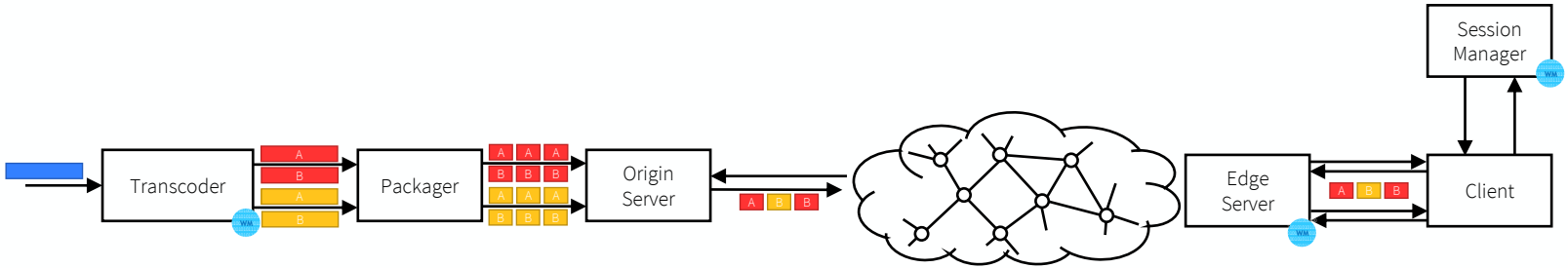
- Forensic watermarking remains in the content after the other content protection features are removed (i.e., after DRM/CAS decryption is applied, after the content has left the Secure Video Path, etc.)

Market drivers

- Live sports
- Premium VOD

Major industry trend = server-side watermarking

- Trade-off cache overhead ↔ watermarking
- No integration on the client side



DASH-IF Watermarking Specification

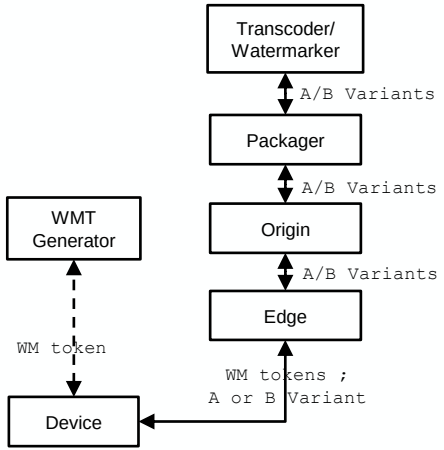
Address the pain points of A/B watermarking with redirection at the edge

- Embedding logic relying on ad-hoc naming conventions
- Brittle synchronization between watermarking components placed at encoder / edge
- Limited support for byte-range requests and different segmentations e.g. HLS vs. DASH

End-to-end architecture that impacts encoder / packager / edge / device

Straightforward A/B routing logic at the Edge

- Watermark token containing a unique A/B pattern transmitted to the Edge by the device
- Explicit metadata stream (`WMPaceInfo` metadata) from the encoder to the edge to signal which WMID bit is associated to a given egress segment



Feedback from the 1st Community Review

Yet another token format

CDN concerns about efficiency

- Size of the `WMPaceInfo` metadata
- Rigid dynamics at the edge

A/B declaration in the ingress manifest is not operational

Others

Watermark Token

Laurent Piron (Nagravision)

Watermark Token

Carries the Unique identifier from the device to the CDN and is required for getting content

Moved to a CWT token (CBOR format)

- Alignment with CTA WAVE CAT (CDN Access Token)
- Minimize the number of formats
- Kept the semantic
  - Identify the WM vendor
  - Direct mode: the token carries the WM pattern that can be encrypted
  - Indirect mode: the token carries parameters to generate the WM pattern
    - Definition of a standard API for integrating the needed extension

DASH-IF created a registry for watermarking vendor
https://dashif.org/identifiers/watermarking/

`WMPaceInfo` and Sidecar File Journey

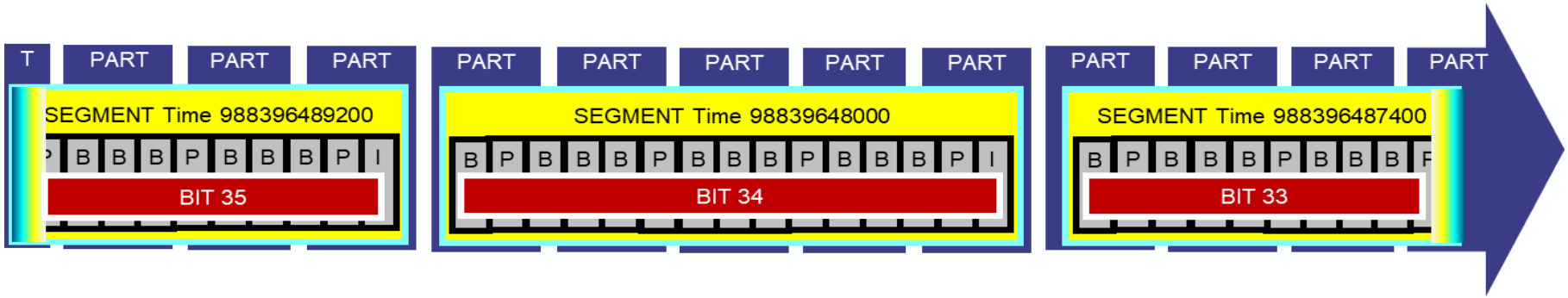
Vladimir Zivkovic (Irdeto)

`WMPaceInfo`: Why, What, Who, Where, How

When a device requests a **Segment**, the edge sequencing logic needs to know which **Bit** in the unique WM pattern to consider for retrieving either A or B **Variant** of the requested **Segment** before delivering it to the device.

`WMPaceInfo` contains this mapping **<Segment, Bit>** in **addition to some data needed for content preparation**.

It is transmitted from the encoder (that is combined with the watermarking pre-processor) to the following servers that may need it (packager, origin, or edge).



`WMPaceInfo` Data

Attribute	Producer	Consumers	Purpose
`variant`	Encoder	Edge	Integration, debugging
`position`	Encoder	Edge	Bit position in the WM pattern
`firstpart`	Encoder	Packager, Origin	Egress packaging
`lastpart`	Encoder	Packager, Origin	Egress packaging

Binary `WMPaceInfo`

```
class WMPaceInfo {
  unsigned int(8)  version;
  unsigned int(8)  variant;
  unsigned int(1)  emulation_1;
  unsigned int(15) position;
  unsigned int(1)  emulation_2;
  unsigned int(1)  firstpart;
  unsigned int(1)  lastpart;
  unsigned int(5)  reserved;
}
```

Conveying `WMPaceInfo`

Ingest protocol	`WMPaceInfo` delivery options
RTMP	SEI
RTP/UDP/RIST/SRT	SEI, TS adaptation field
HLS/TS over HTTP POST	HTTP header, SEI
CMAF-based protocols/formats (HLS/fMP4, DASH) over HTTP POST	HTTP header, ISOBMFF box, SEI
File access protocol	ISOBMFF box, SEI, sidecar file

JSON `WMPaceInfoIngest`

```
WMPaceInfoIngest : {
  "version":   version,
  "variant":   variant,
  "position":  position,
  "firstpart": firstpart,
  "lastpart":  lastpart
}
```

Sidecar File – CBOR: What, Why, How

In some cases (VOD, byteranges) the `WMPaceInfo` data can be aggregated in a sidecar file

For large media files (long content) sidecars can be large files, hence:

- **Concise Binary Object Representation (CBOR)** https://www.rfc-editor.org/info/rfc8949

  and

- **Concise Data Definition Language (CDDL)** https://www.rfc-editor.org/info/rfc8610

CBOR sidecar files can be:

- Smaller size-wise when compared to related JSON sidecar files
- Correctly and quickly generated by Packager, using automatically created code from CDDL sidecar specs
- Efficiently transferred from Origin to Edge and cached in Edge for the future use, thanks to the size
- Quickly parsed by Edge, using automatically created code from CDDL sidecar specs

Format	Speed	Size
JSON	628	261262
Flatbuffers (Unchecked)	<1	79424
Flatbuffers (Checked)	206	79424
Cap'n Proto	2	72176
MessagePack	127	46222
ProtoBuf	124	44848
CBOR (String Keys)	457	188626
CBOR (Integer Keys)	119	46326

(Table taken from: https://github.com/Dash-Industry-Forum/Watermarking/issues/1#issuecomment-1155813598)

Sidecar File – CBOR Integer Keys

Sidecar File in CBOR format is specified in accordance to RFC8949's clause 4.2 (i.e., integer keys) and with a reference to Section 5 "CBOR-based protocols".

From [IANA #1261296] Request for Assignment (cbor-tags): "*It is common for particular CBOR based file formats (generally "CBOR-based protocols", that's also the term in RFC8949) to use small integers like the ones currently in the Watermarking document. There doesn't need to be coordination across all of CBOR: These numbers are already conflict-free because the document is of a particular type. The way I read the "multiple providers of encoders and CDN" clause in the description, the intention is to allow different vendors to allocate their own keys. CBOR has no one-size-fits-all answer to that. One solution employed by IETF protocols (such as SenML) is to run an own registry (in IETF protocols, managed by IANA, but other standards bodies have similar provisions) for the key space. In the particular example, values 1-8 would be initially populated, and you might want to reserve the positive integers for further specification and assign negative integers to vendors on a first-come-first-served basis*."

```
;---------------------------------------+
; Maps Integer Keys (Temporary values)
version      = 1
segments     = 2
fileSize     = 3
startRange   = 4
segmentRegex = 5
position     = 6
firstpart    = 7
lastpart     = 8
;---------------------------------------+

discrete-segment  = {
    ?segmentRegex : text,
    position      : int .size 2 .ge -1,
    ?firstpart    : bool,
    ?lastpart     : bool
}

byterange-segment = {
    startRange    : uint .size 8,
    position      : int .size 2 .ge -1
}

sidecar-discrete  = {
    version       : uint .size 1,
    segments      : [+ discrete-segment]
}

sidecar-byterange = {
    version       : uint .size 1,
    fileSize      : uint .size 8,
    segments      : [+ byterange-segment]
}

sidecar = (sidecar-byterange // sidecar-discrete)
```

`WMPaceInfo` vs. Sidecar File

Live/Linear

- **Discrete Segments** match Live Segments => suggestion is to use a WMPaceInfo per each Live Segment:

```
GET /pathname/WMPaceInfo/filename
GET /pathname/${variantPath}filename
```

VOD

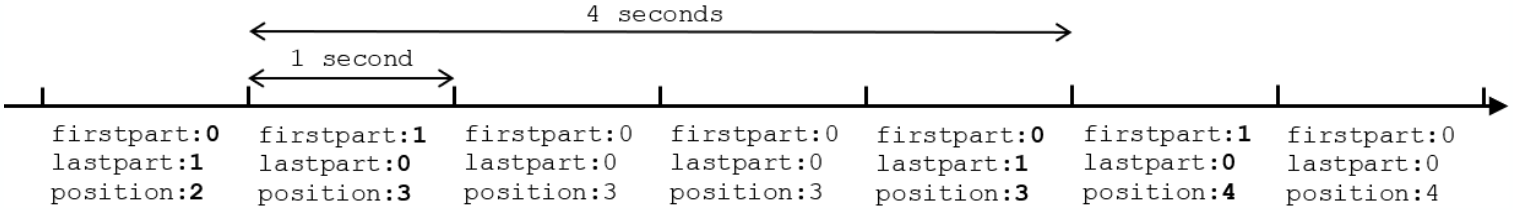
- VOD-like-Live => suggestion is to use **Sidecar File for Discrete Segments**
- VOD-Byteranges => suggestion is to **Sidecar File for Byterange Segments**

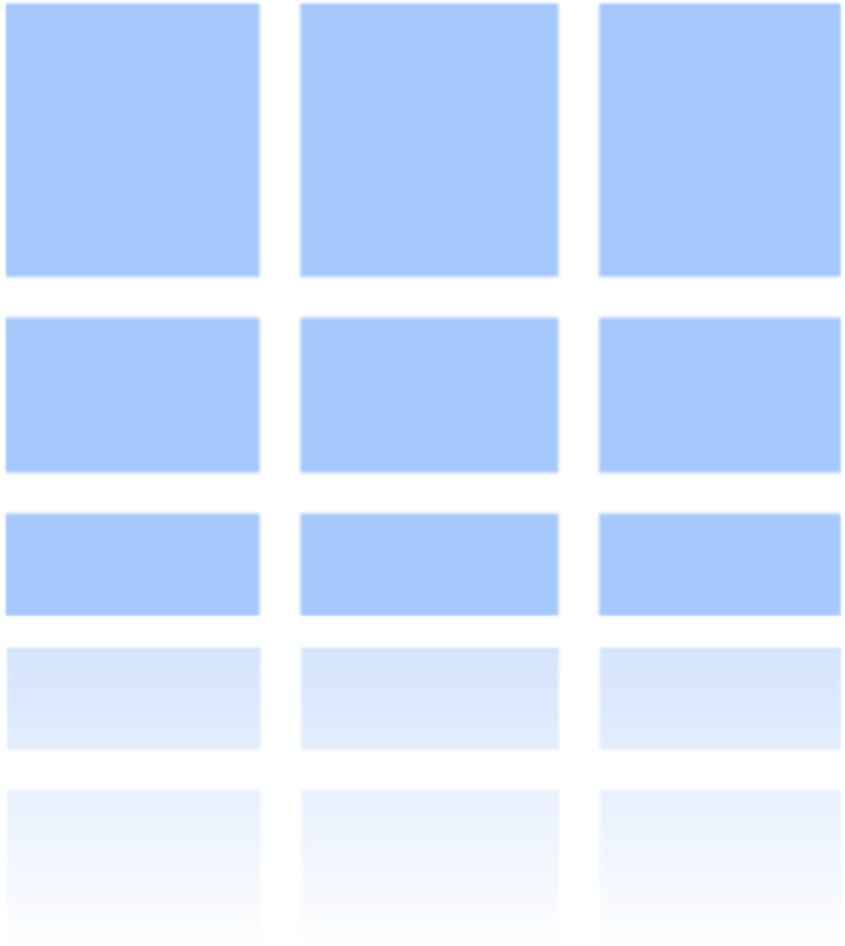
Examples of Sidecars

```
sidecar (
    /version/ 1,
    /segments/ [{/segmentRegex/ "video_segment_ .*?_123.mp4",
                 /position/ 21},
                {/segmentRegex/ "video_segment_ .*?_124.mp4",
                 /position/ 22}])
```

```
sidecar (
    /version/ 1,
    /fileSize/ 262445216,
    /segments/ [{/startRange/ 0,         /position/ -1},
                {/startRange/ 1118,      /position/ 0},
                {/startRange/ 1701212,   /position/ 1},
                 …
                {/startRange/ 261083393, /position/ 118},
                {/startRange/ 262073936, /position/ 119}])
```

Example of `WMPaceInfo` supporting Ingest from Encoder to Packager





DASH Ingest

Nicolas Weil (AWS Elemental)

DASH Ingest (header and SEI approaches)

AdaptationSet for Variant A

content group

variantId

```
<AdaptationSet mimeType="video/mp4" segmentAlignment="true" startWithSAP="1"
subsegmentAlignment="true" subsegmentStartsWithSAP="1" bitstreamSwitching="true">
  <EssentialProperty schemeIdUri="http://dashif.org/guidelines/watermarking_variant#a"
value="tv1"/>
  <SegmentTemplate timescale="60000"
media="a/video_segment_$RepresentationID$_$Time$.mp4"
initialization="a/video_init_$RepresentationID$.mp4" startNumber="10967120"
presentationTimeOffset="903486496960">
    <SegmentTimeline>
      <S t="903487696960" d="240000"/>
      <S t="903487936960" d="186000"/>
    </SegmentTimeline>
  </SegmentTemplate>
  <Representation id="27" width="1920" height="1080" frameRate="30/1" bandwidth="5000000"
codecs="avc1.4D4028"/>
  <Representation id="24" width="1280" height="720" frameRate="30/1" bandwidth="3000000"
codecs="avc1.4D401F"/>
  <Representation id="26" width="640" height="360" frameRate="30/1" bandwidth="1499968"
codecs="avc1.4D401E"/>
</AdaptationSet>
```

content group

variantId

```
<AdaptationSet mimeType="video/mp4" segmentAlignment="true" startWithSAP="1"
subsegmentAlignment="true" subsegmentStartsWithSAP="1" bitstreamSwitching="true">
  <EssentialProperty schemeIdUri="http://dashif.org/guidelines/watermarking_variant#b"
value="tv1"/>
  <SegmentTemplate timescale="60000"
media="b/video_segment_$RepresentationID$_$Time$.mp4"
initialization="b/video_init_$RepresentationID$.mp4" startNumber="10967120"
presentationTimeOffset="903486496960">
    <SegmentTimeline>
      <S t="903487696960" d="240000"/>
      <S t="903487936960" d="186000"/>
    </SegmentTimeline>
  </SegmentTemplate>
  <Representation id="27" width="1920" height="1080" frameRate="30/1" bandwidth="5000000"
codecs="avc1.4D4028"/>
  <Representation id="24" width="1280" height="720" frameRate="30/1" bandwidth="3000000"
codecs="avc1.4D401F"/>
  <Representation id="26" width="640" height="360" frameRate="30/1" bandwidth="1499968"
codecs="avc1.4D401E"/>
</AdaptationSet>
```

AdaptationSet for Variant B

DASH Ingest (sidecar approach)

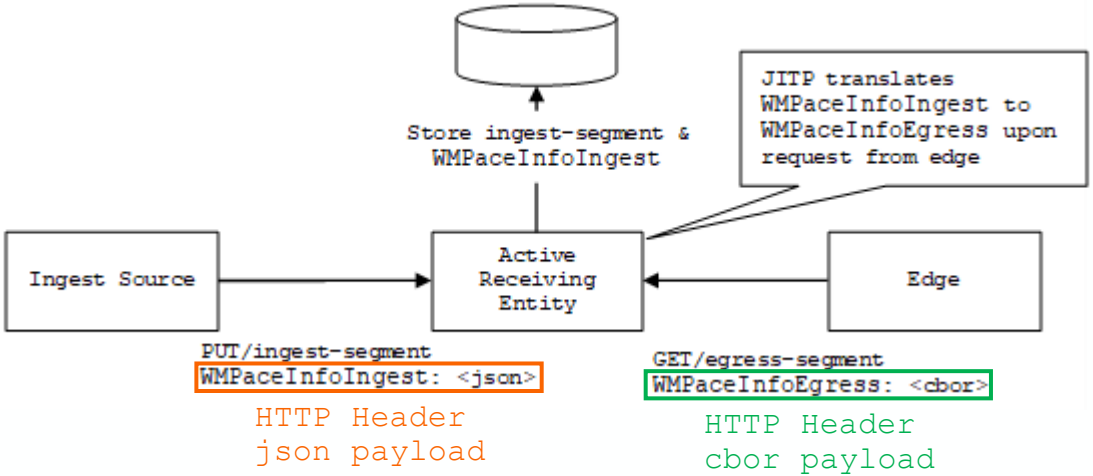
AdaptationSet for Variant A

sidecar file

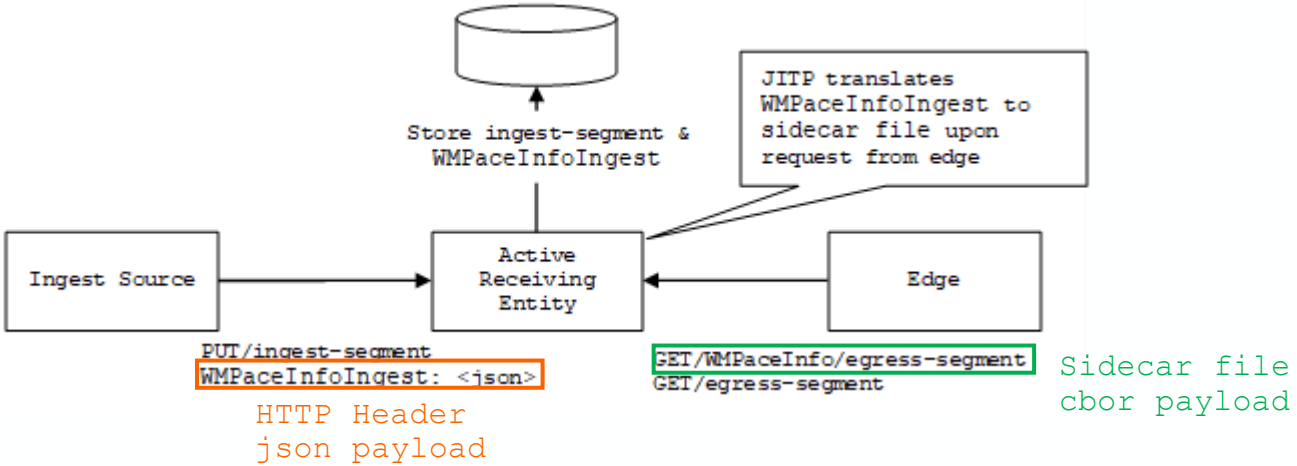
```
  <AdaptationSet mimeType="video/mp4" codecs="avc1.42401E" subsegmentAlignment="true"
subsegmentStartsWithSAP="1" contentType='video' maxWidth="480" maxHeight="360"
maxFrameRate="24" par="4:3">
    <Representation id="2" bandwidth="150000" width="480" height="360" frameRate="24">
      <EssentialProperty
schemeIdUri="http://dashif.org/guidelines/watermarking_variant#a" value="tv1"/>
      <EssentialProperty
schemeIdUri="http://dashif.org/guidelines/watermarking_wmpaceinfo"
value="ElephantsDream_H264BPL30_0100.264.dash_wm_pace_info"/>
      <BaseURL>a/ElephantsDream_H264BPL30_0100.264.dash</BaseURL>
      <SegmentBase indexRange="984-11244">
        <Initialization range="0-983"/>
      </SegmentBase>
    </Representation>
    <Representation id="3" bandwidth="250000" width="480" height="360" frameRate="24">
      <EssentialProperty
schemeIdUri="http://dashif.org/guidelines/watermarking_variant#a" value="tv1"/>
      <EssentialProperty
schemeIdUri="https://dashif.org/guidelines/watermarking_wmpaceinfo"
value="ElephantsDream_H264BPL30_0175.264.dash_wm_pace_info"/>
      <BaseURL>a/ElephantsDream_H264BPL30_0175.264.dash</BaseURL>
      <SegmentBase indexRange="984-11245">
        <Initialization range="0-983"/>
      </SegmentBase>
    </Representation>
  …
  </AdaptationSet>
```

AdaptationSet for Variant B

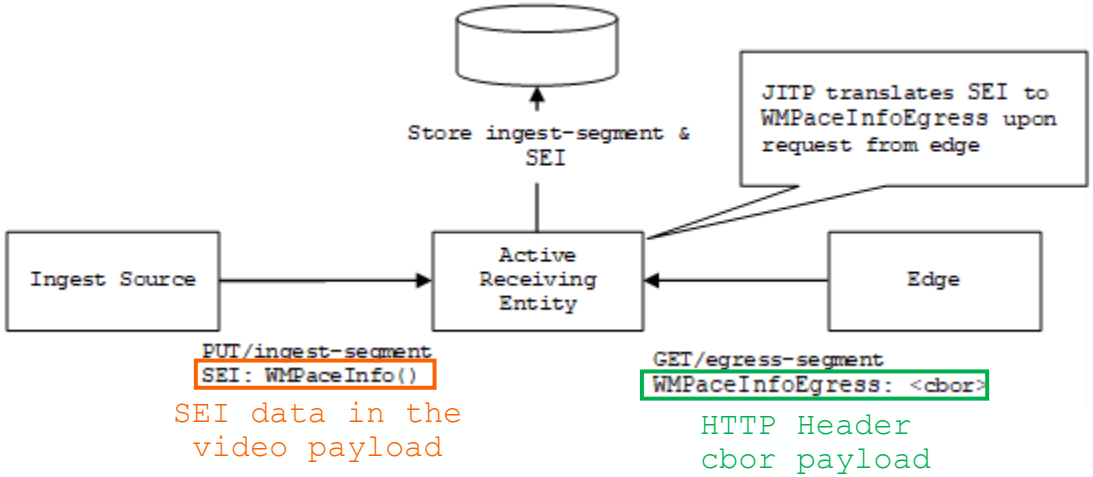
WMPaceInfo in Live Workflows
Scenario 1: JITP header / header



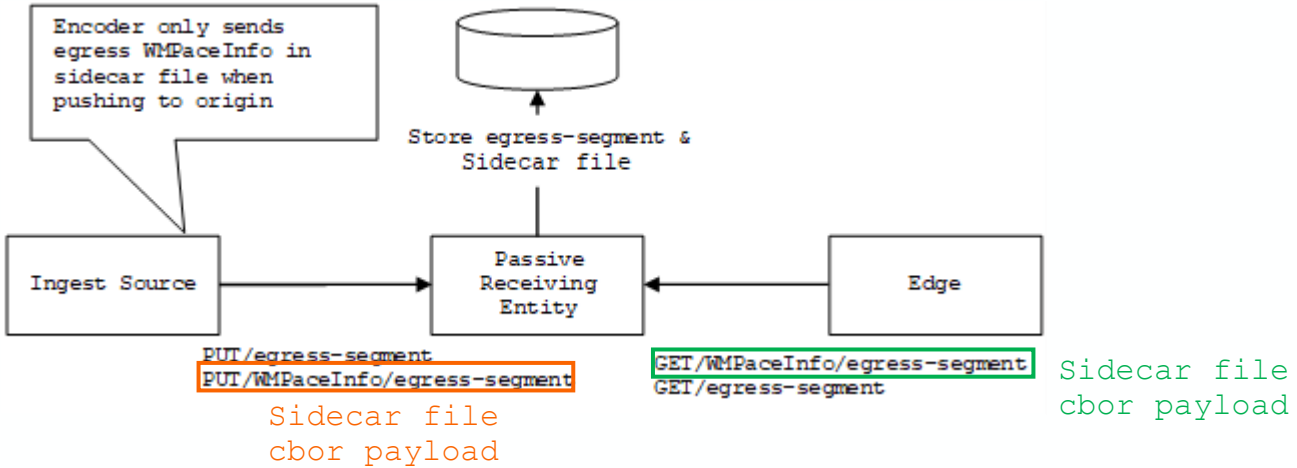
WMPaceInfo in Live Workflows
Scenario 2: JITP header / sidecar



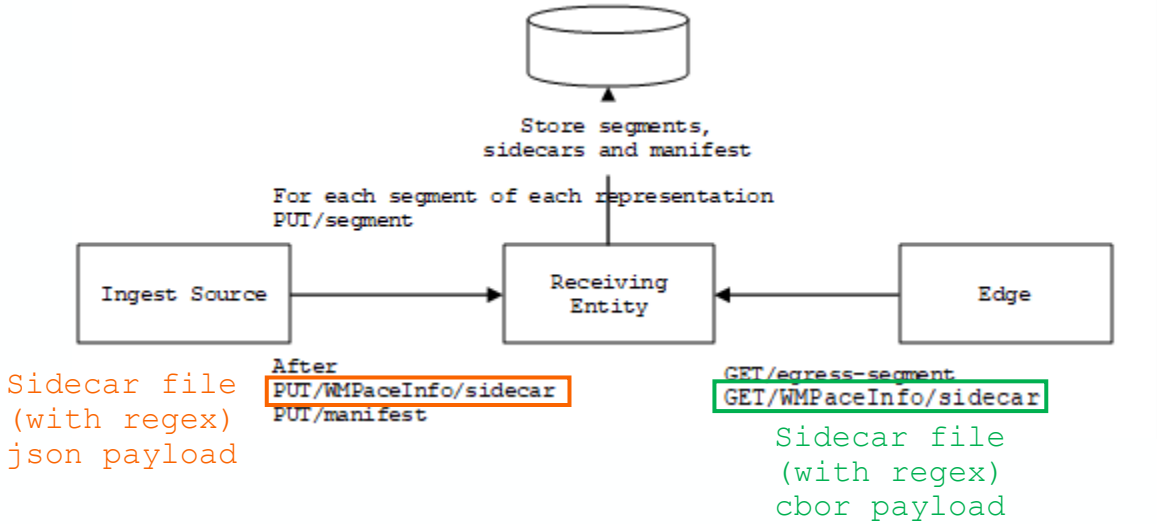
`WMPaceInfo` in Live Workflows
Scenario 3: JITP SEI / header)



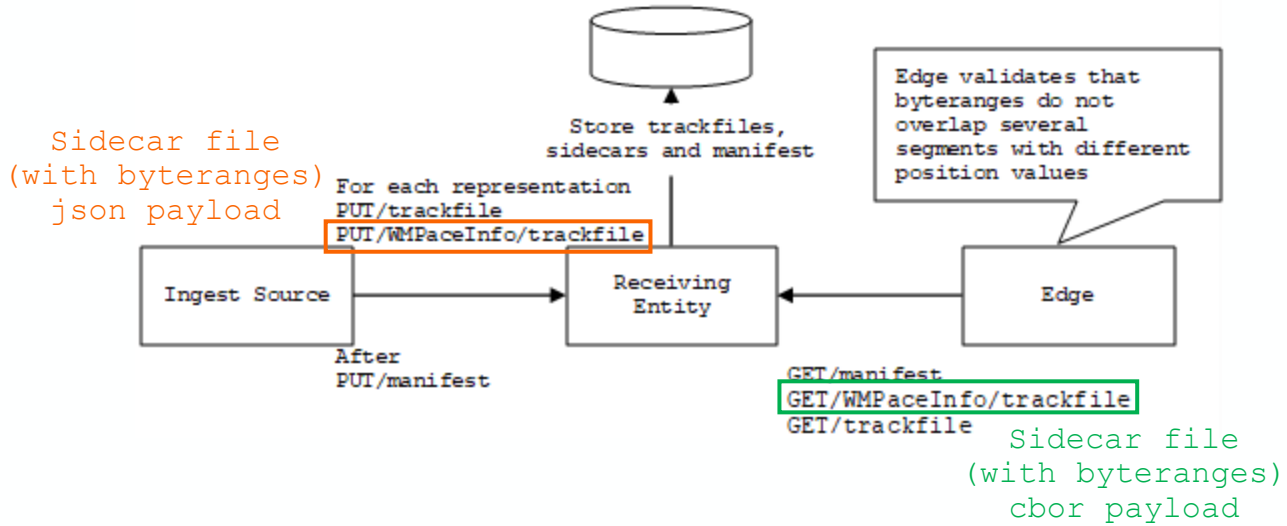
WMPaceInfo in Live Workflows
Scenario 4: Passthrough origination



WMPaceInfo in VOD Workflows
Live profile: JITP sidecar / sidecar



WMPaceInfo in VOD Workflows
on-demand profile: JITP sidecar / sidecar



CDN configuration

Enabling or disabling the edge sequencing logic is set through the **CDN configuration**

When sequencing is disabled, the edge shall pull segments from the **origin endpoint for Variant A**. If this endpoint is not working properly, the origin shall deliver any available Variant on this endpoint.

Watermarked objects names shall include a **pattern** that the CDN can match to differentiate these objects from non-watermarked objects (initialization segments, subtitles, trickplay images)

watermarked naming pattern

```
<SegmentTemplate timescale="60000" media="video_segment_$RepresentationID$_$Time$.mp4"
initialization="video_init_$RepresentationID$.mp4" startNumber="10967120"
presentationTimeOffset="903486496960">
```

non-watermarked naming pattern

Origin pulls: `variantId` shall be **translated** into `variantPath` as `variantPath = ${variantId}` followed by `/`

if `${variantId}` is `a` or `0` then `${variantPath}` may be empty

Changes Summary

Nicolas Weil (AWS Elemental)

Changes Recapped

Theme	1st Community Review	2nd Community Review
System configuration	WMPaceinfo.iswm==false	Edge configuration
WM token	JWT format	CWT format Streamline the format for direct vs indirect mode options
`WMPaceInfo`	Rigid structure	Adapted to the interface. Not all servers need all data
Sidecar file	JSON file	CBOR file
DASH ingest	Single AdaptationSet, 2 EssentialProperty	2 AdaptationSets, Single EssentialProperty
`WMpaceInfo` transport/acquisition	No guidance	4 clearly identified approaches for live workflows 2 clearly identified approaches for VOD workflows

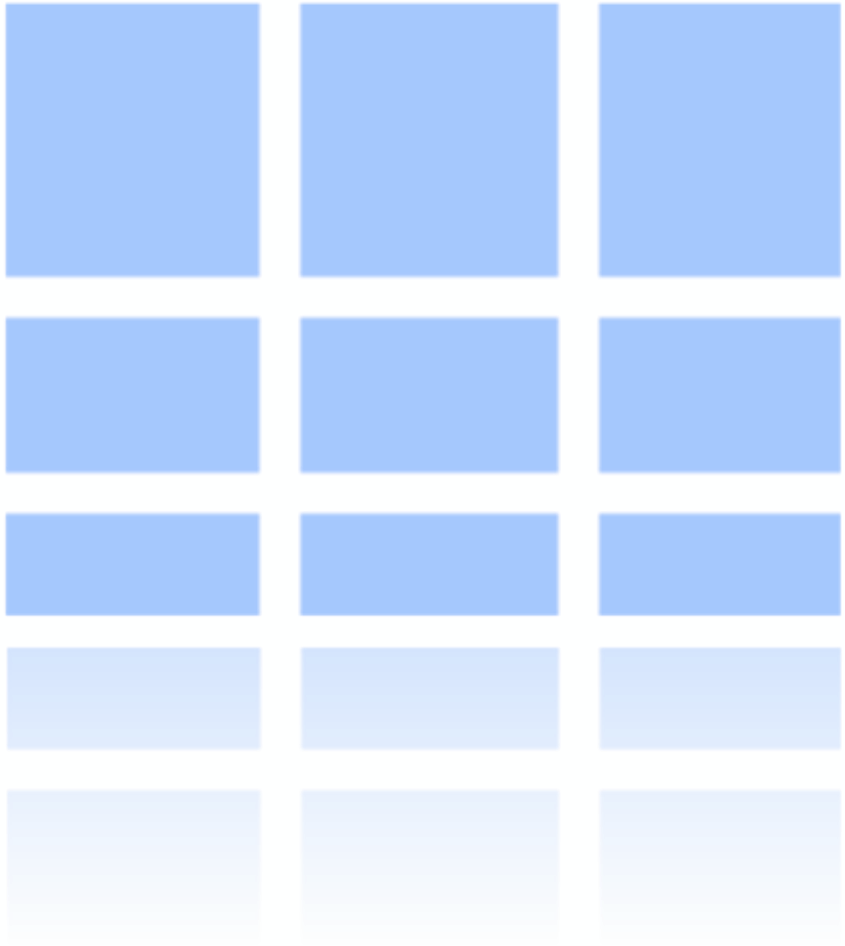
Conclusion

Community Review until end of February 2023

- Document available at https://dashif.org/docs/IOP-Guidelines/DASH-IF-CTS-00XX-AB-Watermarking-0.9.pdf
- Anyone interested can propose corrections, changes https://github.com/Dash-Industry-Forum/Watermarking/issues

Next steps

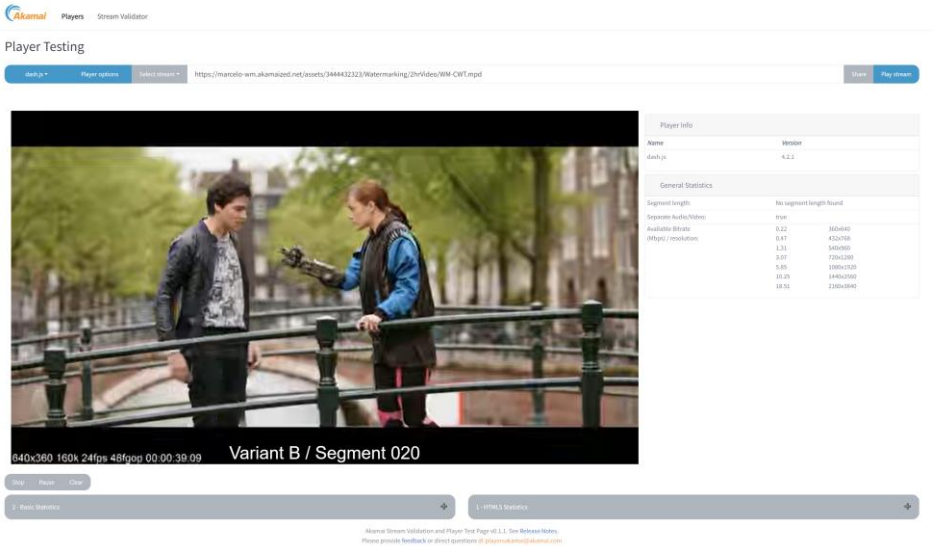
- Need feedback from all stakeholders (encoder, packager, CDN) and finalize specification incl. identifiers registration
- Liaise with UHDF to request an update of their encoder specification
- Liaise with CTA WAVE CAT to merge the watermarking and authentication tokens



DASH IF in Action

Marcelo San Martin (Akamai)

VOD Demo (courtesy Akamai)



Demo

- Player dash.js
- CDN Akamai - EdgeWorkers
- Origin

Checklist

☑ A/B sequencing

☑ Sub-range request ok

☑ Overlapping range requests denied