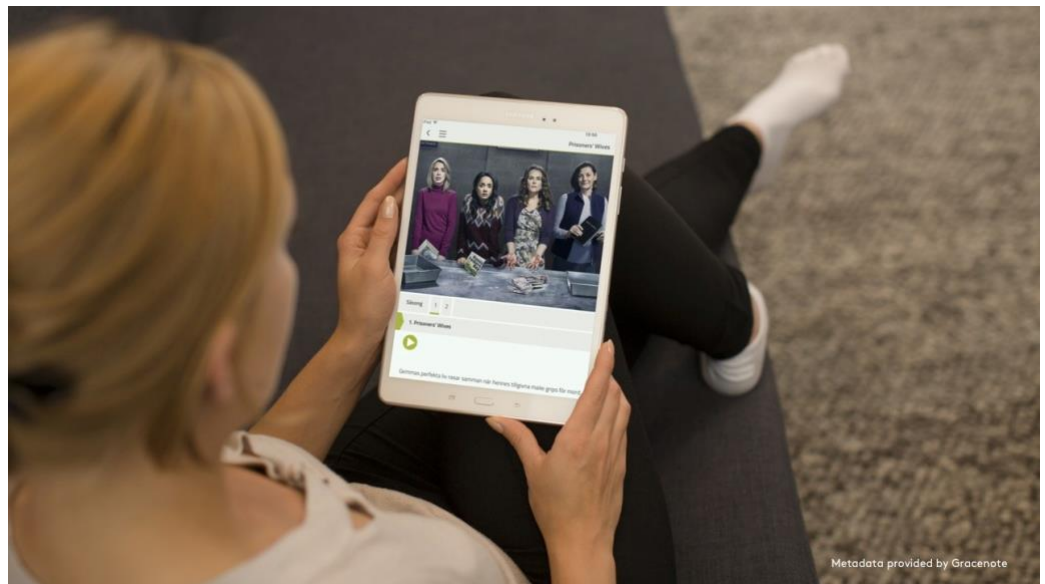


# DIGITAL ADS IN AN OPERATOR'S TVE SERVICE



RevC

Technical specification for ads in a TVE service

SYSTEM DESCRIPTION ..... 2

LOGISTICS AND DISTRIBUTION ..... 3

*REPLACE LINEAR ADS WITH DIGITAL ADS* ..... 3

*INSERTION OF DIGITAL ADS IN "ON-DEMAND" WATCHING* ..... 4

*AD REQUEST* ..... 6

*USER INFORMATION* ..... 7

*ROUTINE FOR HANDLING OBJECTIONS/QUESTIONS FROM USERS* ..... 7

MEASUREMENT AND REPORTING ..... 8

*LINEAR CHANNELS* ..... 9

*"ON DEMAND"* ..... 9

*STRUCTURE OF SDK* ..... 9

*VALIDATION* ..... 11

METADATA ..... 12

APPENDIX A: DYNAMIC AD INSERTION REQUIREMENTS ..... 13

APPENDIX D: VAST TRACKING EVENTS ..... 15

APPENDIX E: AD PROXY REQUEST PARAMETERS ..... 16

# 1 System description

Measurement of ad views takes place from the Operator's TVE client applications to the Broadcaster's ad server or server with a third party as per the information that is in the answer from the ad server (VAST). In addition to ad measurement to Broadcaster, the Operator also sends to MMS measurement of program and ad views. See chapter Measurement and Reporting for more details on tracking.

# 2 Logistics and distribution

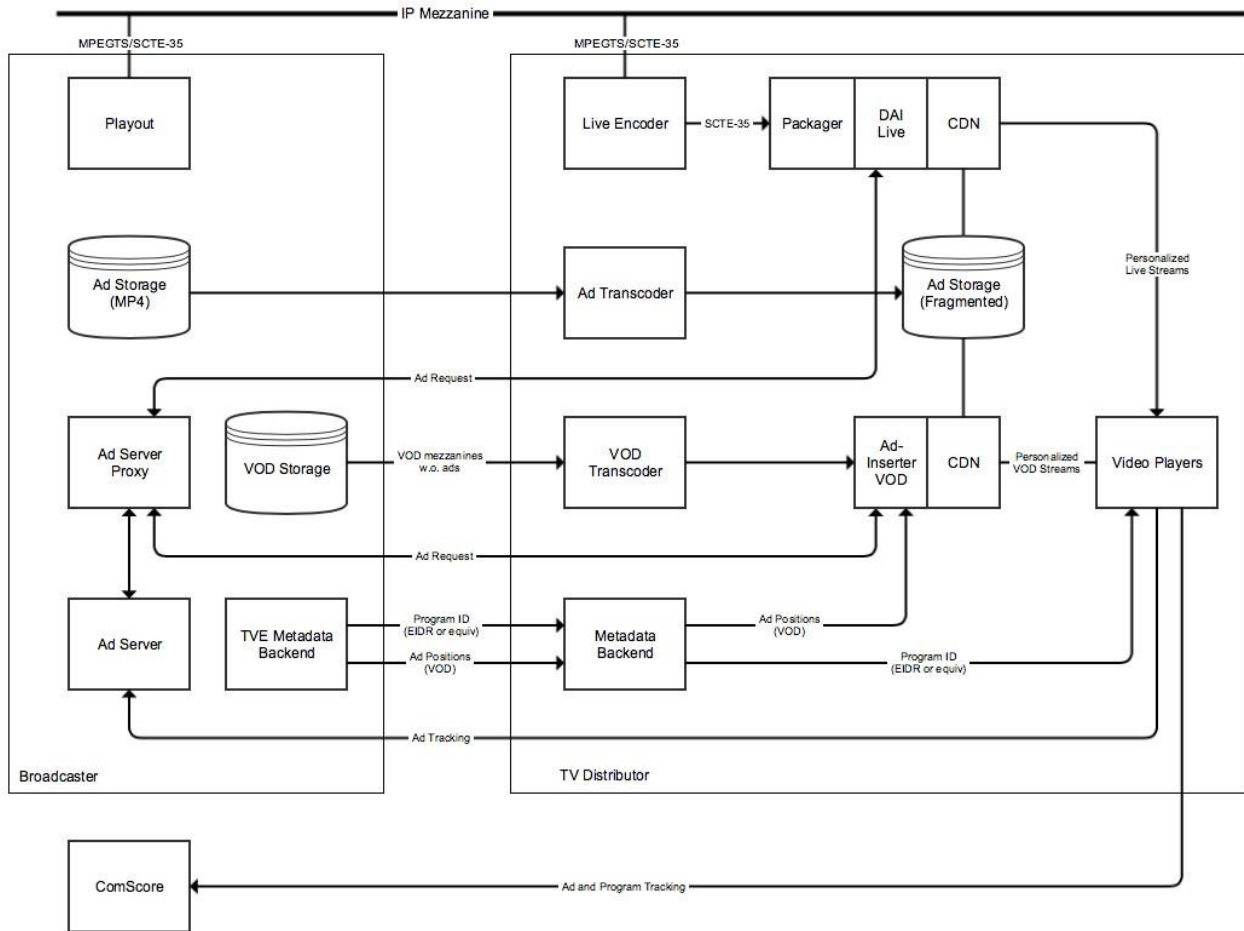


FIGURE 1: LOGISTICS AND DISTRIBUTION (REFERENCE ARCHITECTURE)

## 2.1 REPLACE LINEAR ADS WITH DIGITAL ADS

In the video stream for every channel from the Broadcaster, there are markers according to the SCTE-35 standard. These markers mark when the commercial break begins and how long it is. The Operator transcodes the video stream to ABR streams (adaptive bitrate streams for distribution over the Internet) and makes sure the markers are still present in the various ABR formats the Operator uses. When a user wants to consume a Broadcaster channel, the following essentially happens:

1. The user submits a request to the Operator's servers for an ABR stream for the Broadcaster channel
2. The Operator's DAI (dynamic ad insertion) server creates a unique session and a unique manifest for this user that is based on the ABR stream for the Broadcaster channel
3. The user receives a link to this unique manifest that the DAI server has created
4. The user's video player receives this link and begins to play the video segment that is in this unique manifest. The video segments are distributed over the Operator's distribution network.
5. If the DAI server discovers in the ABR stream that there is a commercial break, the DAI server sends a request to the Ad Server Proxy, which adapts the request and forwards it to the Broadcaster's ad server. The request contains a unique user or session ID and the length of the break.

6. The Ad Server Proxy adapts the response from the Broadcaster's ad server and gives the DAI server a response according to the VAST protocol (VAST 2.0 or later version). The response contains a list of commercials that are to be shown to the user.
  - a. If a commercial is not adapted to the Operator and the DAI server has not previously served this ad, the commercial is retrieved and transcoded to a compatible format with the ABR stream and no ad replacement occurs.
  - b. If the commercial is adapted to the Operator or already transcoded, the video segments in the ABR stream's manifest are replaced with the commercials' video segments.
7. The user's video play receives this updated manifest, which contains commercial segments and plays the stream as usual and for the user, it is not noticeable that this replacement occurred on the server side.

#### Technical specification for signal distribution

- SCTE-35 markers in IP-mezzanine signal distribution (for example output PID 35)
- SCTE-35 markers are inserted according to SCTE-104 (spliceStart\_normal) with the duration attribute set. This results in an SCTE-35 splice\_insert() with out\_of\_network\_indicator=1, duration\_flag=1 and auto\_return=1

#### Fallback scenarios

To guarantee the best possible customer experience, a fallback scenario is needed when it is not possible to fill an entire commercial break in the stream for various reasons. This may be due to the specific commercial not being transcoded in time.

The Operator's ad insertion solution can expect that the entire commercial break is filled to 100% with commercials from Ad Server Proxy, but there may be scenarios where the ad insertion solution cannot fill it with all the ads. If this scenario arises, the following shall occur:

1. If a commercial from the Ad Server Proxy is not imported and transcoded, the ad insertion solution shall choose a commercial from the extra commercials (fallback) that are included with the response.
2. The ad insertion solution retrieves the missing commercial and transcodes it so that it is in the system for the next time it shows up.
3. If commercials are still missing to fill out the entire break, the ad insertion solution shall have the possibility of adding a "filler" that the Operator has received from the respective Broadcaster. The filler is among the extra commercials and is indicated with the name "filler".
4. If the response from the ad server proxy is timed out or broken, the distributor should let the linear ads to be shown for live, and for VOD the ad break will be ignored if no ads are received.

## 2.2 INSERTION OF DIGITAL ADS IN "ON-DEMAND" WATCHING

In order for the Operator to be able to insert digital ads in content that a viewer can watch afterwards on his/her mobile or tablet, the Operator needs access to an ad-free video file for the program content in advance. Together with the video file, program companies deliver a text file with ad points for the program. When playing the program content, essentially the following happens:

1. Alternative:
  - a. The DAI server sends an ad request via the Ad Server Proxy to the Broadcaster's server with information on all ad points that need to be filled for the entire length of the program

- b. The DAI server sends an ad request via the Ad Server Proxy to the Broadcaster's ad server for every ad point for the program (weak or no support for sequential ads)
2. The Broadcaster's ad server gives the DAI server a response according to the VAST protocol (VAST 2.0 for individual breaks, or VMAP 1.0 for multiple breaks). The response contains a list of commercials that are to be shown to the user.
  - a. If a commercial is not adapted to the Operator and the DAI server has not previously served this ad, the commercial is retrieved and transcoded to a compatible format with the ABR stream and no ad insertion occurs.
  - b. If the commercial is adapted to the Operator or already transcoded, the DAI server puts the commercials' video segments in the ABR stream's manifest at the times for ads.
3. The user's video player receives this updated manifest, which now also includes commercial segments and plays the program as usual.

### Technical specification for file distribution

When files are distributed from the Broadcaster to the Operator to be used for insertion of digital ads, it should be according to the below specification:

- The file format is a high bitrate mezzanine (H.264/MPEG-4/AVC) in an MP4 container file.
- The format should be either HD or SD (HD preferred). Future codecs/formats to handle UHD and HDR will be added in future revisions pending market standards to be settled.
- WebVTT for captions.

Mezzanine formats:

- HD: MP4 container H.264 High@L4.1 1920x1080 25fps progressive at 15 Mbps (audio AAC LC2.0 48 KHz at 256 Kbps or AAC LC5.1 [6ch] 48 KHz at 450 Kbps)
- SD 16x9: MP4 container H.264 High@L3.1 1024x576 25fps progressive at 6 Mbps (audio AAC LC2.0 48 KHz at 256 Kbps or AAC LC5.1 [6ch] 48 KHz at 450 Kbps)
- SD 4x3: MP4 container H.264 High@L3.1 768x576 25fps progressive at 6 Mbps (audio AAC LC2.0 48 KHz at 256)

The Broadcaster uploads the above files to an FTP delivery point at the Operator. The goal is for the files to be uploaded before the program is shown on TV, but it cannot be guaranteed so it is important that the Operator can handle this case. The Operator is then also not expected to put ads into the catch-up version until they have received this late file delivery. Except for the programs broadcast live and program episode premiers, the Broadcaster sends the files to the Operator at least three days before the program is broadcast linearly.

### VMAP for VOD

Ad markers and ad placements in on demand content (VOD) are retrieved through the ad proxy according to VMAP specification<sup>1</sup>. In this case, on an ad request, the ad proxy responds with information on where the ad breaks are and what ads to place (alternative 1 a).

<sup>1</sup> <https://www.iab.com/wp-content/uploads/2015/06/VMAP.pdf>

In the scenario where the VOD is a recorded live stream (catchup / non-linear start-over) an ad request is sent by the Operator when a cue point is found in the recorded stream (alternative 1b) but it is still the Broadcaster's ad server (via the ad proxy) that defines the length of the ad break.

Both methods mentioned for handling on demand content are accepted by Broadcasters however the VOD alternative is the preferred method by the Broadcasters.

## 2.3 AD REQUEST

The ad request from the Operator's DAI server to the Broadcaster's ad server shall include the following information:

1. Unique ID for specific users (pseudonymized) [userId]
2. Program ID [programId]
3. Channel ID [channelId]
4. Device type [deviceType]:
  - a. iphone
  - b. ipad
  - c. android\_phone
  - d. android\_tablet
  - e. desktop\_mac
  - f. desktop\_windows
  - g. desktop\_linux
5. Screen size (e.g. 1920x1080) [screenSize]
6. Location (geographic location based on IP address, where the three last chars is anonymized) [geolocation].
7. Duration of ad break (only for live) [duration]

See Appendix A for a detailed requirement specification on the DAI server and appendix E for ad proxy request parameters.

In accordance with GDPR, an opt-out possibility could exist and in such a case, only parameter (2), (3) and (7) are sent with. The user then receives randomly selected ads.

The above parameters are sent in the HTTP request to the ad server proxy at the Broadcaster in the form of HTTP query parameters, such as:

<http://<adserverproxy.example.com>/ads?duration=120&userId=23412313&channelId=TV4&deviceType=ipad&os=ios&screenSize=1920x1080&geolocation=193.14.163.194>

Example when using VMAP for VOD:

<http://<adserverproxy.example.com>/vod?userId=23412313&contentId=1341155&deviceType=ipad&os=ios&screenSize=1920x1080&geolocation=193.14.163.194>

Example of opt-out (GDPR):

<http://<adserverproxy.example.com>/ads?duration=120&channelId=TV4>

## 2.4 USER INFORMATION

Information regarding personal data processing (related to ad request, ad tracker and measurement) shall be displayed in the Operator's TVE service and such user information shall be produced and provided by Broadcaster.

## 2.5 ROUTINE FOR HANDLING OBJECTIONS/QUESTIONS FROM USERS

User's first point of contact regarding personal data processing related to the ad request, ad measurement and ad tracking should be the Broadcaster. Thus, Operator shall provide relevant contact information to the users within the TVE service.

In case of objections to Broadcaster's personal data processing, Broadcaster shall swiftly provide user with a response explaining the legal ground for the processing. Should a user continue to claim opt out from Broadcaster's personal data processing after receiving the response or if other questions (or comments) appear from user as a reaction to Broadcaster's response, the Broadcaster shall handle the communication.

Note: Operator and Broadcaster shall always act swiftly when handling requests or questions related to personal data processing hereunder.



# 3 Measurement and reporting

Measurement of ad views takes place from the Operator's TVE client applications to the Broadcaster's ad server or server with a third party as per the information that is in the answer from the ad server (VAST). Besides ad measurements, the Operator also sends to MMS measurement of program views.

For the Broadcaster's need to, according to the contract, be able to capitalize on the stitched ads, a correct measurement is necessary according to both MMS, with technical help from comScore (or other MMS technology partner), and the ad supplier that the Broadcaster uses for delivery of VAST to the Operator's DAI server.

The measurement of ads to the ad server takes place through tracking links defined in the VAST file on the positions these link elements define, e.g. "impression". All tracking events in the TrackingEvents element should be tracked.

Excerpt from an VAST xml:

```
<AdTitle>S1UYAR2011</AdTitle>
<Error>URL</Error>
<Impression>URL</Impression>
<Impression>URL</Impression>
<Creatives>
  <Creative id="video">
    <Linear>
      <TrackingEvents>
        <Tracking event="fullscreen">URL</Tracking>
        <Tracking event="firstQuartile">URL</Tracking>
        <Tracking event="start">URL</Tracking>
        <Tracking event="thirdQuartile">URL</Tracking>
        <Tracking event="midpoint">URL</Tracking>
        <Tracking event="mute">URL</Tracking>
        <Tracking event="unmute">URL</Tracking>
        <Tracking event="complete">URL</Tracking>
        <Tracking event="pause">URL</Tracking>
      </TrackingEvents>
    </Linear>
  </Creative>
</Creatives>
```

The impression URLs are tracked with an HTTP GET when the first frame of the ad is displayed, and for video ads the tracking event URLs are tracked on each of the different event types. For example, the tracking event Start URL is tracked when the video ad starts and tracking event firstQuartile when 25% of the video ad has played. A complete list of events that publisher may track is in Appendix D, though distributor's player may not support each interaction type. Any interaction supported must be tracked by triggering URL if tracking event is provided in VAST.

For comScore, measurement occurs during both content and ads on MMS- and comScore-defined and required locations. This is facilitated through an SDK supplied by MMS.

While the ad server's measurements only require the initial arguments, defined on the previous page, in the ad request, MMS and comScore require multiple well-defined arguments to be counted as valid. These are defined in the documentation provided by comScore<sup>2</sup>.

The measurement of the linear channels and On Demand are largely similar, but there are a number of parameters that differ by nature between these two measurements. The way the SDK handles these is addressed below.

Both HTTP and HTTPS can be used for reporting.

### 3.1 LINEAR CHANNELS

By reporting to the SDK that one is running a linear channel and possibly timeshift in a DVR window, we can handle correct reporting of the parameters suitable to the purpose on to comScore. In that the consumption occurs on a linear channel, we will not change the parameter values with the program one is watching, but rather the reporting takes place as consumption of that particular channel and with a time value that further in the statistics can be traced to a specific program. When it comes to ads, however, every individual ad shall be reported with its unique values to measure the consumption. This is because every stream gets unique stitching and a time point cannot be traced to a specific ad.

### 3.2 "ON DEMAND"

In consumption of VOD material, the reporting becomes more specific around the current program one is actually consuming in accordance with the parameters specified by the implementation documentation provided by MMS.

### 3.3 STRUCTURE OF SDK

The final structure may be adjusted and adapted by delivery, but the conditions, ambition and vision is the same as below.

MMS supplies a library that is to be implemented in the Operator's clients according to the implementation documentation that will be supplied with the SDK.

When starting the client, it is recommended to initiate the measurement class with the player's basic parameters, which will then be used in all calls during the current session.

- comScore Client Number
- Name of Media Player
- Version of Media Player
- User Agent
- Device Type

When starting the video playback, the client reports a function in the measurement class with the content's basic parameters, which will also be used in all subsequent events during the consumption of the specific content.

- Content ID (ns\_st\_ci)
- Live, true or false (ns\_st\_li)
- Content Length (ns\_st\_cl)
- Title (for troubleshooting purposes)

<sup>2</sup> For latest comScore implementation documentation please advise your MMS contact.

- Program name (ns\_st\_pr)
- Type (ns\_st\_ty)

Thereafter, the player calls the functions available for the purpose for play, pause, end of content and ads with the parameters below in every call.

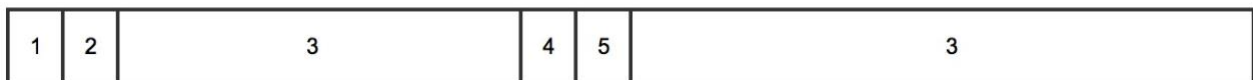
- Clip number in playlist (see explanatory sketches below)
- Position, milliseconds (ns\_st\_po)

And for ads, the unique ad ID as well.

The SDK will itself also during the playing of content have an ongoing so-called heartbeat, a continuous data collection at a set interval, which will require an available function that exposes the data that the event requires as per above.

### Explanation clip number in playlist

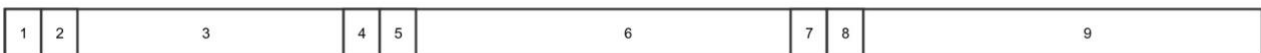
One of the more complex parameters to understand about the comScore measurement is the terminology around clips and playlists. This is also of significance in various cases, e.g. VOD or Live on a linear channel.



In consumption of VOD, the asset one is watching, together with the commercials stitched in, is considered to become a playlist. In this case, what is important is that the same content keeps its clip number in the playlist, according to the sketch above.

- 1 = [First preroll ad](#)
- 2 = [Second preroll ad](#)
- 3 = VOD asset one chose to watch, e.g. Leila Bakar
- 4 = [First midroll ad](#)
- 5 = [Second midroll ad](#)
- 3 = Continuation of the VOD asset one chose to watch, e.g. Leila Bakar

Every unique clip is given its clip number, and retains it when it continues regardless of the stitched ad.



When consuming a linear channel, or another playlist of different content, every new asset in the playlist gets its own identification number.

- 1 = [First preroll ad](#)
- 2 = [Second preroll ad](#)
- 3 = First program, e.g. Leila Bakar
- 4 = [First midroll ad](#)
- 5 = [Second midroll ad](#)
- 6 = Second program, e.g. Sommar med Ernst
- 7 = [Third midroll ad](#)
- 8 = [Fourth midroll ad](#)
- 9 = Third program, e.g. Idol

If an ad comes up in the middle of a program, the program shall of course keep its clip number even after the ad, according to the VOD example above.

### 3.4 VALIDATION

In order for a Broadcaster to be able to claim and capitalize on the ad measurement, MMS should validate the client's reporting implementation. This should take place before going live with an application and usually occurs by sharing with MMS the beta versions of the app and the web clients one intends to go live with, and they can track the reporting by connecting this up to a proxy. One is then provided a validation protocol with the results whether or not one has been approved or given points to resolve before being able to go live.

Validation with MMS is done for major changes or at least four (4) times a year. Prior to validation, the Operator provides the Broadcaster and MMS beta versions of the apps so that validation can be done before they reach the end customers.

## 4 Metadata

For the Operator to be able to request ad and for MMS/comScore reporting a Broadcaster needs to provide the Operator with the following metadata (but not limited to).

- Content ID (program or channel) to be used to request ad and reporting to MMS/comScore
- Metadata that comScore requires
- Position of ad breaks for a program (for the file distribution)

In the metadata the Broadcaster sends to the Operator, there is a unique ID number<sup>3</sup> for each Broadcaster program. This ID number needs to be preserved when the metadata passes a metadata broker such as Gracenote so that this ID number can reach the client application. With this ID number, the client retrieves through a standard format defined by the program companies the comScore parameters that need to be send on to MMS/comScore in reporting. This unique ID is owned and issued by the Broadcaster or a global registry such as EIDR.

---

<sup>3</sup> Public Schedule XML: <ns2:contentId>se.Programbolag.content.5913494</ns2:contentId>

# 5 Appendix A: Dynamic Ad Insertion Requirements

Dynamic Ad Inserter component is referred to as DAI.

ID	PRIO	Requirement
DAI1.1	MUST	The DAI <b>must</b> provide a per-user unique LIVE stream where ads are inserted just-in-time based on the response from the ad server.
DAI1.2	MUST	The DAI <b>must</b> provide a per-user unique VOD stream where ads are inserted based on the response from the ad server
DAI2	MUST	The DAI <b>must</b> ingest and transcode the ad videos in the response from the ad server. Ingested and transcoded ads must be cached by the DAI.
DAI3	MUST	The DAI <b>must</b> support that before an ad break in a live channel a request to the ad server is sent as specified in ad request section.
DAI4	MUST	The DAI <b>must</b> support and handle ad responses according to VAST.
DAI5	MUST	The DAI <b>must</b> handle that the VAST response can include interactive ads and other ad formats that cannot be inserted in the stream.
DAI6	MUST	The DAI <b>must</b> support VAST wrapper ads.
DAI7	SHOULD	The DAI <b>should</b> support VAST passback and if the DAI does not support VAST passback it <b>must</b> support that an ad request to the ad server can take up to 5000 ms to process.

DAI8	MUST	The DAI <b>must</b> support that the requested duration not always can be fully filled with ads by the ad server and that a filler loop is inserted for the remainder of the ad break.
DAI9	MUST	The DAI <b>must</b> provide a way for the video player to get the VAST payload that was the base for the inserted ads. This is to enable clientside tracking.
DAI10	MUST	The DAI <b>must</b> provide a way to signal to the video player that an ad is being played. This is to make it possible for the client application to disable scrub and pause, but also for client-side tracking.
DAI11	SHOULD	The DAI <b>should</b> support server side ad-tracking.
DAI12	SHOULD	The DAI <b>should</b> support server side ad-inserted pre-rolls.

# 6 Appendix D: VAST Tracking Events

As per IAB VAST 3.0 specifications, [https://www.iab.com/wp-content/uploads/2015/06/VASTv3\\_0.pdf](https://www.iab.com/wp-content/uploads/2015/06/VASTv3_0.pdf), section 2.3.1.7

- **creativeView**: not to be confused with an impression, this event indicates that an individual creative portion of the ad was viewed. An impression indicates the first frame of the ad was displayed; however an ad may be composed of multiple creative, or creative that only play on some platforms and not others. This event enables ad servers to track which ad creative are viewed, and therefore, which platforms are more common.
- **start**: this event is used to indicate that an individual creative within the ad was loaded and playback began. As with creativeView, this event is another way of tracking creative playback.
- **firstQuartile**: the creative played for at least 25% of the total duration.
- **midpoint**: the creative played for at least 50% of the total duration.
- **thirdQuartile**: the creative played for at least 75% of the duration.
- **complete**: The creative was played to the end at normal speed.
- **mute**: the user activated the mute control and muted the creative.
- **unmute**: the user activated the mute control and unmuted the creative.
- **pause**: the user clicked the pause control and stopped the creative.
- **rewind**: the user activated the rewind control to access a previous point in the creative timeline.
- **resume**: the user activated the resume control after the creative had been stopped or paused.
- **fullscreen**: the user activated a control to extend the video player to the edges of the viewer's screen.
- **exitFullscreen**: the user activated the control to reduce video player size to original dimensions.
- **expand**: the user activated a control to expand the creative.
- **collapse**: the user activated a control to reduce the creative to its original dimensions.
- **acceptInvitationLinear**: the user activated a control that launched an additional portion of the creative. The name of this event distinguishes it from the existing "acceptInvitation" event described in the 2008 IAB Digital Video In-Stream Ad Metrics Definitions, which defines the "acceptInvitation" metric as applying to non-linear ads only. The "acceptInvitationLinear" event extends the metric for use in Linear creative.
- **closeLinear**: the user clicked the close button on the creative. The name of this event distinguishes it from the existing "close" event described in the 2008 IAB Digital Video In-Stream Ad Metrics Definitions, which defines the "close" metric as applying to non-linear ads only. The "closeLinear" event extends the "close" event for use in Linear creative.
- **skip**: the user activated a skip control to skip the creative, which is a different control than the one used to close the creative.
- **progress**: the creative played for a duration at normal speed that is equal to or greater than the value provided in an additional attribute for offset. Offset values can be time in the format HH:MM:SS or HH:MM:SS.mmm or a percentage value in the format n%. Multiple progress events with different values can be used to track multiple progress points in the Linear creative timeline.



# 7 Appendix E: Ad Proxy Request Parameters

Field	Request parameter	Format
<b>Desired duration</b>	duration	[0-9]+ (seconds)
<b>Unique User ID</b>	userId	String
<b>Program ID</b>	contentId	String
<b>Channel ID</b>	channelId	String
<b>Device Type</b>	deviceType	String
<b>Screen Size</b>	screenSize	[0-9]+x[0-9]+ (width X height)
<b>Location</b>	geolocation	[0-9]+.[0-9]+.[0-9]+.[0-9]+ (IP address)

Example:

<http://<adserverproxy.example.com>/ads?duration=120&userId=23412313&channelId=TV4&deviceType=ipad&os=ios&screenSize=1920x1080&geolocation=193.14.163.194>