

# AAMP DASH - T6 linear Client Side DAI Design and Architecture

## 1.0 Overview

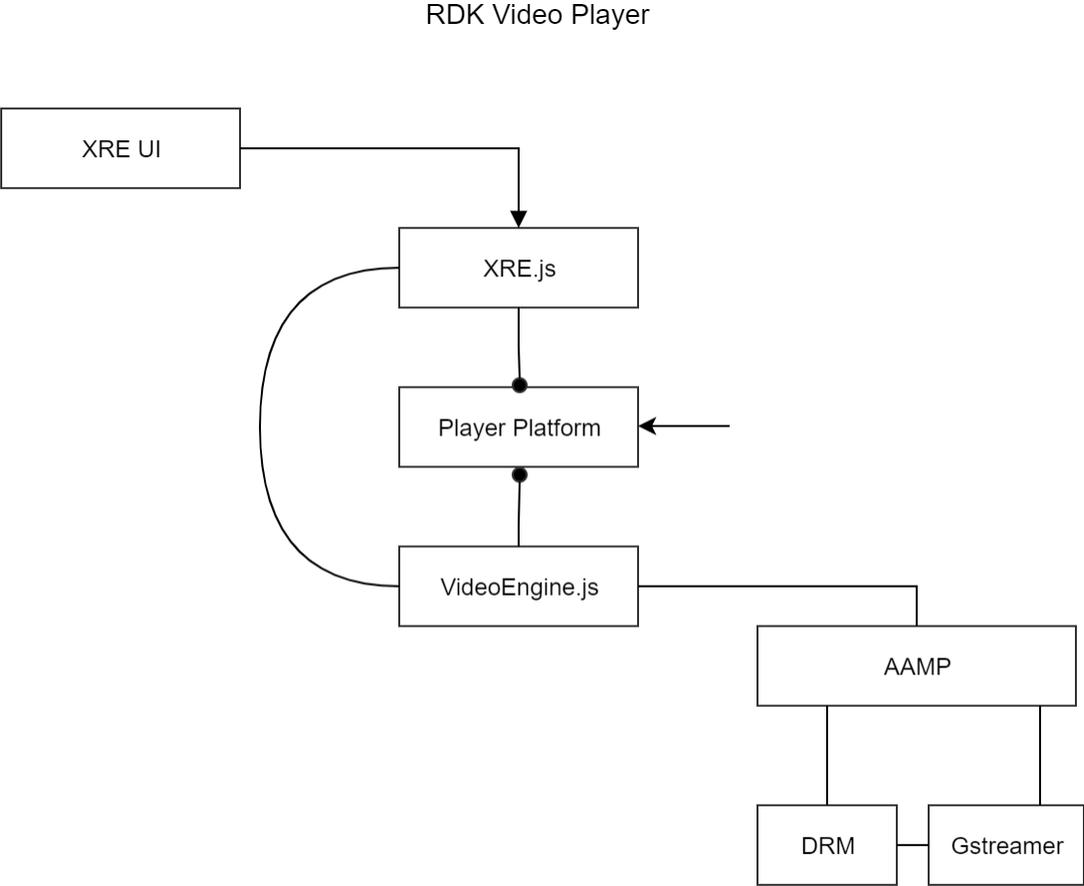
This document captures a high-level requirements and architecture for client-side DAI Implementation in the AAMP Native Engine

## 2.0 Product and Technical Requirements

Req No.	Product Requirements	Status
1.0		
2.0		

## 3.0 Architecture

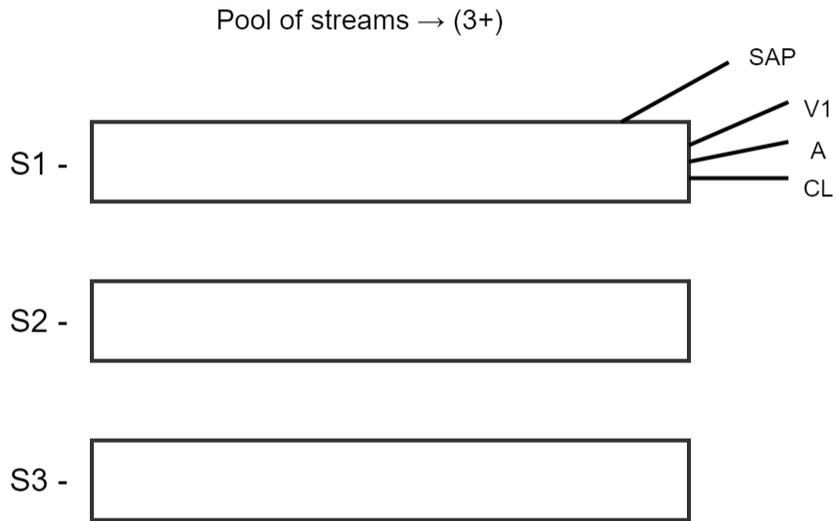
This section provides a high-level architecture



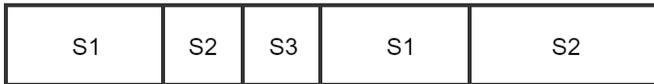
RDKPlayer.js will be hosted by RDK that will manage to release and ensure that north bound interface does not change without proper change control process

## Ad Insertion & blockout use cases

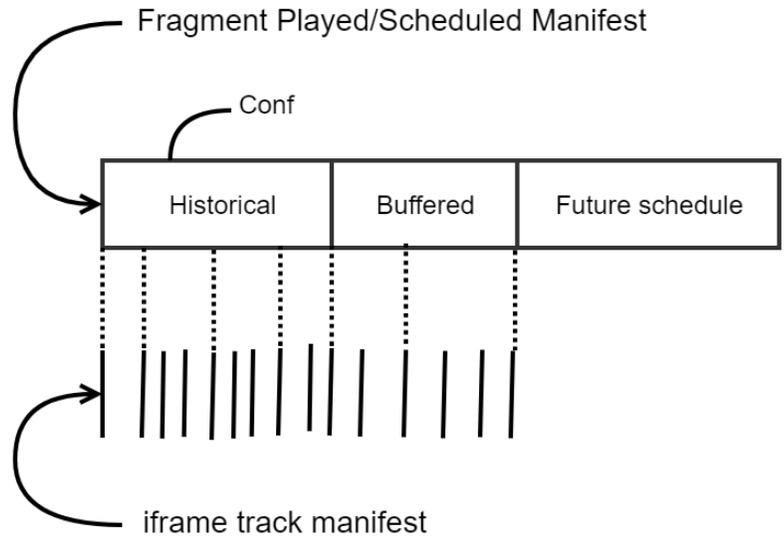
Player uses stream switching construct to support above use cases



Client specific time line & iframe track



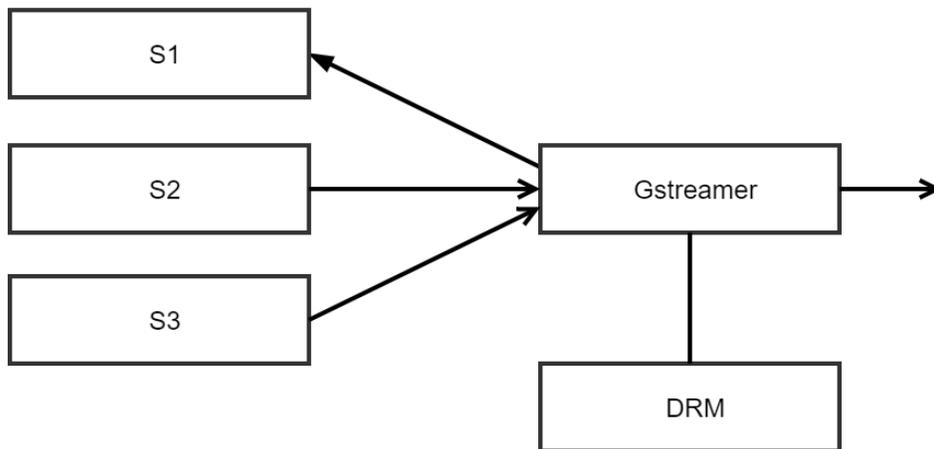
List of fragments played (historical) +  
buffered scheduled fragments(future/best  
guess)



'i' Segments → Fog/Pointed to cloud

i frames → Fog/cloud/locally harvested

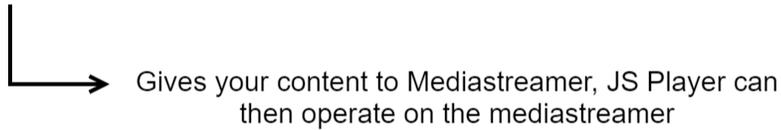
Playback is serialised via Gstreamer



# Top down usage for stream **switch**

JS based model (RDKVideoEngine)

```
MediaStreamS1 = RDKVE.Setsource(URL)
```



MS.play()  
MS.Seek()  
MS.Pause()  
MS.init(JSON value of property\_)  
MS.getproperty()  
MS.setproperty()

A simple use case will only need one mediastreamer

# Top down usage for **Player**

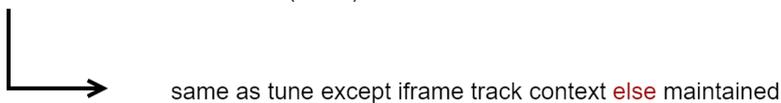
JS based model (RDKVideoEngine)

Abstract out stream switching and only expose one playback stream context

```
MS S1 = RDKVE.SetMediaSource()
```

All Media operation and DAI happens using the same Media stream context

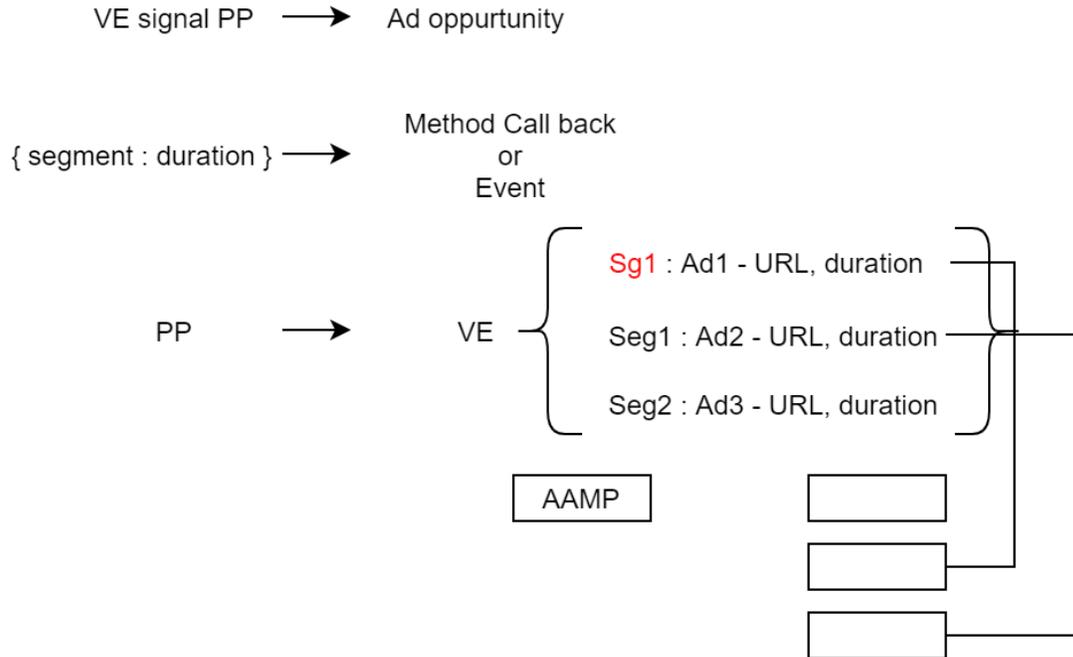
```
MS S1 = RDKVE.Switch(URL)
```



Use case -> Playlisting supports **e.g.** music videos.

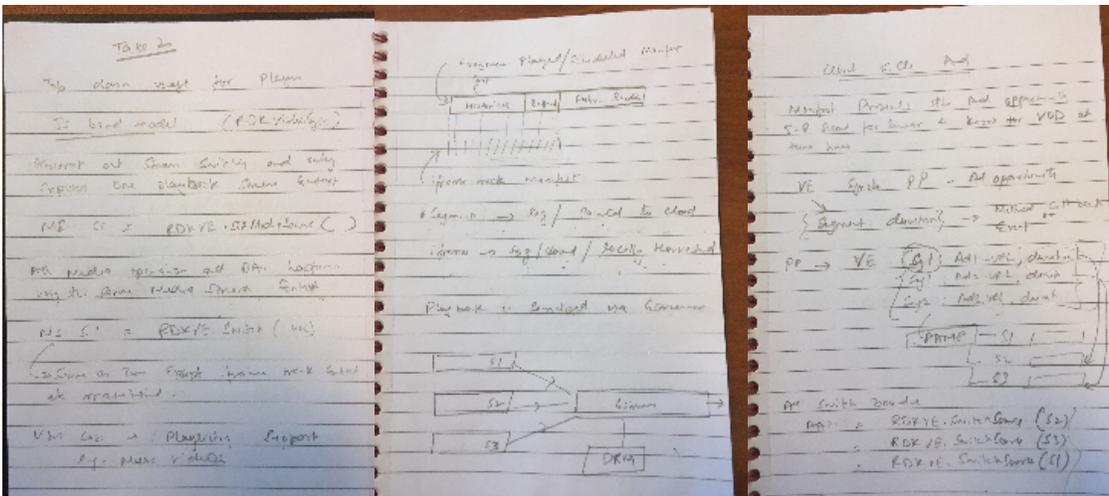
# Client Side Ad

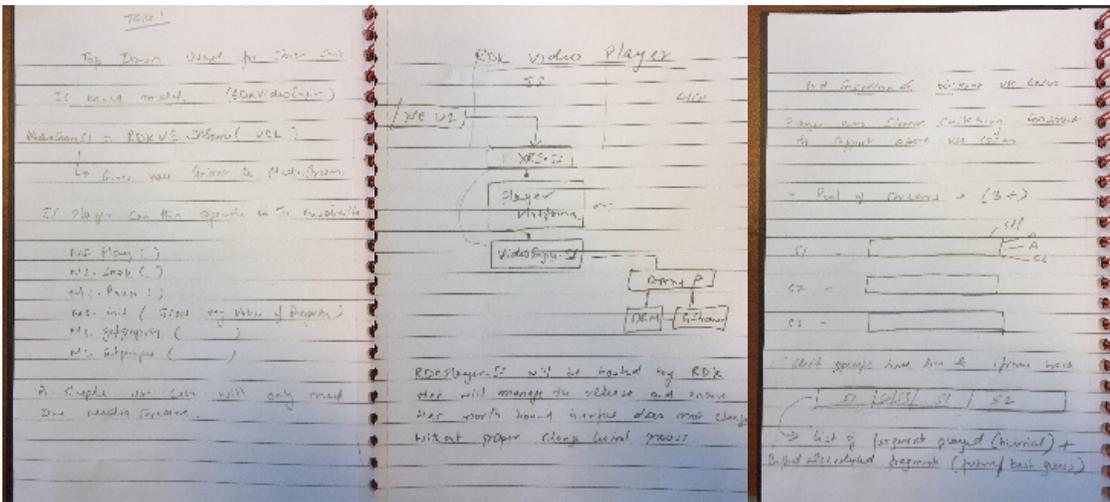
Manifest presents the Ad opportunity j.s second for linear and know for VOD at tune time



At Switch bound

- AA = RDVE.SwitchSource (S2)
- = RDVE.SwitchSource (S3)
- = RDVE.SwitchSource (S1)





### 3.1 Component Overview

#### Process View

## 4.0 Integrated AAMP-FOG

Detaching the AAMP from browser & merging with the FOG's executable gives us the flexibility in caching the dynamic Ads in the local storage. It provides many other advantages and few disadvantages too.

### 4.1 Pros

- a. AAMP Player can be used by other apps too.
- b. FOG & AAMP can be kept running silently in the background . Hence, coming back from an app will quickly play the linear video with TSB available.
- c. Eliminates the requirement of AAMP plugins from the browser. It provides better load time while boot up and app launching/exiting would be faster too.
- d. Can isolate the AAMP bugs from the browser; hence debugging will be easy.
- e. Browser crash won't affect the playback and AAMP crash doesn't need browser reloading too; hence crash recovery will be faster.
- f. No redundant codes for FOG and AAMP; saves development effort, memory and performance.
- g. FOG refreshes the manifest in the memory, which will be readily available to AAMP; No need of explicit refreshing the manifest in the AAMP side.
- h. Overlapping streams (eg: client side DAI) can be cached and played easily.
- i. It eliminates extra operations to send the fragments from FOG to AAMP.
- j. Will be able to run the unified executable in the desktop environment without any code changes.

## 4.2 Cons

- a. Extra effort needed for integrating AAMP & FOG.
- b. Complexity and number of threads increases; hence maintenance effort too.
- c. Player platform needs an IPC mechanism to communicate with AAMP - but avoids IPC between AAMP & FOG.
- d. Integrated binary will be heavier than the individual modules of FOG & AAMP; it takes more memory footprint and load time.
- e. Can't unload the AAMP and freeup memory, while it is not in use. Eg: launching an App may not need AAMP.
- f. Can't support the HTML5 video tag directly. (It needs another plugin to interact with the AAMP process; which would result in unnecessary complexities.)

**Integration activity suspended due to the difficulty in supporting the HTML5 video tag.**

## 5.0 AAMP Player APIs

### 5.1 Properties

Name	Data Type	Read or Write	Description
url	String	r/w	video content URL
contentOptions	Object	r/w	hash of k/v pairs required for content playback
autoPlay	Boolean	r/w	when true, video starts playing immediately; when false, playback must be manually started.
audioLanguage	String	r/w	primary audio language. If set to a value that does not exist for the content, value will not be changed.
secondaryAudioLanguage	String	r/w	secondary audio language. If set to a value that does not exist for the content, value will not be changed.
closedCaptionsLanguage	String	r/w	closed captions language. If set to a value that does not exist for the content, value will not be changed.
closedCaptionsEnabled	Boolean	r/w	when true, closed captioning is enabled; when false, closed captioning is disabled
closedCaptionsOptions	Object	r/w	caption options
position	Number	r/w	position of video
speed	Number	r/w	playback speed
volume	Number	r/w	a value from 0 to 100. Defaults to 100.
zoom	String	r/w	"FULL" or "NONE". When FULL, content is stretched to width and height of player. When NONE, content is best fit to width or height of player without distorting the video's aspect ratio.
duration	Number	r	duration of video in milliseconds
availableAudioLanguages	String [ ]	r	array of the available audio languages for this video
availableClosedCaptionsLanguages	String [ ]	r	array of the available captions languages for this video
availableSpeeds	String [ ]	r	array of the available playback speeds for this video
tsbEnabled	Boolean	r/w	when false, TSB will not be available. when true, TSB may be available if video and device support it.
timeline	Object	r	Timeline object used for DAI support (see section 5.0).

### 5.2 Functions

Name	Return Value	Arguments	Description
play	void	none	starts video playback
pause	void	none	pauses video playback. Equivalent to speed = 1
stop	void	none	stops video playback. Video is not expected to resume.
setSpeed	void	speed - Number overshootCorrection - Number (milliseconds)	sets the speed and adjusts the position of the video by the number of milliseconds specified by overshootCorrection

setPositionRelative	void	seconds - Number	Sets the position of the video by adding the given number of seconds. seconds may be positive or negative, but should not cause the position to be less than zero or greater than the duration.
requestStatus	void	none	Requests the onStatus event to be fired
setAdditionalAuth	void	params - Object	provides a set of k/v pairs required for additional authentication and authorization

## 5.3 Events

Name	Payload	Description
onMediaOpened	mediaType - String - has one of the following values: live, liveTSB, recorded  width - Number  height - Number  availableSpeeds - Number  availableAudioLanguages - String [ ]  availableClosedCaptionsLanguages - String [ ]  customProperties - Object  mediaSegments - Object	Fired when video content has been opened
onClosed	none	fired when the video stream is closed
onPlayerInitialized	none	fired when the video player is initialized
onBuffering	none	fired when video starts buffering. playback is not possible at this time.
onPlaying	none	fired when video starts playing for the first time.
onPaused	none	fired when video is paused (or speed is set to 0)
onComplete	none	fired when video reaches its end, VOD or cDVR for example.
onIndividualizing	none	fired when player is individualizing. Playback is not possible. Not all instances will fire this event.
onAcquiringLicense	none	fired when player is acquiring a license. Playback is not yet possible. Not all instances will fire this event.
onProgress	position - Number - current position in milliseconds  duration - Number - length of content in milliseconds (recorded video only)  speed - Number - current playback speed  start - Number - start position of the TSB buffer, -1 when no buffer available  end - Number - end position of the TSB buffer, -1 when no buffer available	fired periodically when player progresses
onWarning	code - Number  description - String	fired when a warning occurs. video playback will likely continue.
onError	code - Number  description - String	fired when an error occurs. video playback will terminate.
onSpeedChange	speed - Number	fired when playback speed changes
onDRMMetadata	props - Object	fired when DRM metadata is acquired. Contains DRM related properties.

onSegmentStarted	segmentType - String duration - Number segmentId - String segment - Object	fired when a new segment is started.
onSegmentCompleted	segmentType - String duration - Number segmentId - String segment - Object	fired when a segment is has completed
onSegmentWatched	segmentType - String duration - Number segmentId - String segment - Object	fired when a segment has been started and completed.
onBufferWarning	warningType - String - one of BUFFER_UNDERFLOW or BUFFER_OVERFLOW bufferSize - Number - total size of buffer bufferFillSize - Number - current filled size of buffer	
onPlaybackSpeedsChanged	availableSpeeds - String [ ]	fired when playback speeds have changed. This may happen when the video switches from one segment to the next.
onAdditionalAuthRequired	locator - String eventId - String	fired when video needs additional auth to continue playback

## 6.0 AAMP Timeline APIs

### 6.1 DAI support: Timeline, AdBreak, Ad

Based on the timedMeata events, AAMP will construct a Timeline object that contains a list of AdBreak and Ad.

AAMP.timeline;

-- Get a reference (read-only) to the current Timeline object.

```
interface Timeline {
  readonly attribute AdBreak adBreaks[];
  readonly attribute AdOpportunity adOpportunities[];
  readonly attribute Number start;
  readonly attribute Number duration;
  readonly attribute Number position;
  readonly attribute TrickModeRestrictions restrictions;
  addEventListener(eventType, listener);
  removeEventListener(eventType, listener);
};

interface AdBreak {
  const unsigned short ADBREAK_TYPE_INSERT = 0;
  const unsigned short ADBREAK_TYPE_REPLACE = 1;
  readonly attribute unsigned short type;
  readonly attribute String id;
  readonly attribute Number start;
  readonly attribute Number duration;
  readonly attribute Ad ads[];
```

```

};

interface Ad {

readonly attribute String id;
readonly attribute String url;
readonly attribute Number duration;
readonly attribute Object metadata;
readonly attribute TrickModeRestrictions restrictions;

};

interface TrickModeRestrictions {

readonly attribute Number pause;
readonly attribute Number rewind;
readonly attribute Number fastForward;

};

interface AdOpportunity {

const unsigned short AD_PLACEMENT_TYPE_PREROLL = 0;
const unsigned short AD_PLACEMENT_TYPE_MIDROLL = 1;
const unsigned short AD_PLACEMENT_TYPE_POSTROLL = 2;

const unsigned short AD_PLACEMENT_MODE_INSERT = 0;
const unsigned short AD_PLACEMENT_MODE_REPLACE = 1;

readonly attribute unsigned short placement;
readonly attribute unsigned short mode;
readonly attribute String id;
readonly attribute Number start;
readonly attribute Number duration;

readonly attribute Object metadata;

};

```

## 6.2 Timeline Properties

Name	Data Type	Read or Write	Description
adBreaks	Object[]	r	Array of AdBreak objects (sorted by time). Contains replaced and inserted Ad placed into the timeline.
adOpportunities	Object[]	r	Array of AdOpportunity objects (sorted by time). Contains replaced and inserted Ad opportunities available for client-side Ad insertion.
duration	Number	r	Duration of the entire timeline in milliseconds (includes content and placed Ads).
position	Number	r	Current playback position in milliseconds.
start	Number	r	Earliest seekable position in milliseconds
timedMetadata	Object[]	r	Array of TimedMetadata objects (sorted by time). Contains metadata associated with HLS subscribed tags and embedded SCTE35 data.

## 6.3 Timeline Functions

Name	Return Value	Arguments	Description
addEventListener	void	eventType - String, listener - Function	Register an event listener for the specified eventType.
removeEventListener	void	eventType - String, listener - Function	De-register the specified event listener for the specified eventType.
placeAdBreak	bool	adBreak - Object	Places the specified AdBreak into the timeline. Return true if successful.
subscribedTimedMetadata	void	tags - String []	Sets collection of HLS tags to monitor during parsing, or embedded data. Fire "timedMetadata" event when subscribed tags are parsed.

## 6.4 Timeline Events

Name	Payload	Description
timelineUpdated	seekableRangeChanged - bool adBreaks - Object[] - array of modified AdBreaks adOpportunities - Object[] - array of inserted AdOpportunities duration - Number - duration of entire timeline (milliseconds) position - Number - current position in the timeline (milliseconds) start - Number - earliest seekable position in the timeline (milliseconds)	Fired timeline is updated.  An update occurs when the timeline start/duration changes, or new AdOpportunity has been inserted, or an AdBreak / Ad has been placed.
timedMetadata	timedMetadata - Object	Fired when new TimedMetadata has been parsed.
adBreakStart	adBreak - Object - the AdBreak being started speed - Number - current playback rate seenCount - Number - number of times Ad break was played (in full).	Fired when player starts playing an AdBreak.
adBreakComplete	adBreak - Object - the Adbreak being finished progress - Number - percentage of Ads played ads - Object[] - array of Ad including played progress of each Ad.	Fired when player finishes playing an AdBreak.
adBreakSkipped	adBreak - Object - the Adbreak being skipped or exited progress - Number - percentage of Ads played	Fired when player skips over an AdBreak.
adStart	ad - Object - the Ad being started speed - Number - current playback rate seenCount - Number - number of times Ad was played (in full).	Fired when player start playing an Ad.
adProgress	ad - Object - the Ad being played progress - Number - percentage of Ad played	Reports the player's progress as it plays an Ad.
adComplete	ad - Object - the Ad that finished begin played progress - Number - percentage of Ad played	Fired when player finishes playing an Ad.

## 6.5.AdBreak Properties

Name	Data Type	Read or Write	Description
ADBREAK_TYPE_INSERT	0	static const	Indicates adBreak was inserted.
ADBREAK_TYPE_REPLACED	1	static const	Indicates adBreak was replaced.
type	Number	r	Specified the abBreak type, and indicates if adds are inserted or replace existing content.
id	String	r	Unique identifier associated with the ad break.
start	Number	r	Starting position (milliseconds) of the ad break in the timeline.
duration	Number	r	Duration (milliseconds) of the ad break.

ads	Object[]	r	Array of Ad objects (sorted by time). Contains the ads that will be played during the ad break.
-----	----------	---	--

## 6.6 AdBreak Functions

Name	Return Value	Arguments	Description
placeAds	bool	position - Number, ads - Object[]	Place the specified Ad objects in the AdBreak. Return true if successful.

## 6.7 Ad Properties

Name	Data Type	Read or Write	Description
id	String	r	Unique identifier associated with the ad.
url	String	r	URL specifying the location of the ad's manifest.
duration	Number	r	Duration (milliseconds) of the ad.
metadata	Object	r	Additional metadata associated with the ad.

## 6.8 AdOpportunity Properties

Name	Data Type	Read or Write	Description
AD_PLACEMENT_TYPE_PREROLL	0	static const	Indicates opportunity places ad before the main content.
AD_PLACEMENT_TYPE_MIDROLL	1	static const	Indicates opportunity places ad in the main content.
AD_PLACEMENT_TYPE_POSTROLL	2	static const	Indicates opportunity places ad after the main content.
AD_PLACEMENT_MODE_INSERT	0	static const	Indicates opportunity inserts ad inside content.
AD_PLACEMENT_MODE_REPLACE	1	static const	Indicates opportunity replaces content.
id	String	r	Unique identifier associated with the ad opportunity.
placement	Number	r	Indicates placement type: preroll vs. midroll vs. postroll.
mode	Number	r	Indicates placement mode: insertion vs. replacement.
start	Number	r	Starting position (milliseconds) of the ad opportunity.
duration	Number	r	Duration (milliseconds) of the ad.
metadata	Object	r	Additional metadata associated with the ad opportunity.

## 6.9 TimedMetadata Properties

Name	Data Type	Read or Write	Description
METADATA_TYPE_TAG	0	static const	Indicates metadata is from the manifest.
METADATA_TYPE_ID3	1	static const	Indicates metadata was embedded in the content.
type	Number	r	Specified the metadata type: manifest vs. embedded.
time	Number	r	Time (in milliseconds) of the metadata.
name	String	r	Name of the metadata. E.g., #EXT-X-CUE, #EXT-X-SCTE35.
content	String	r	Value of the metadata.
id	String	r	Unique identifier associated with the metadata.
metadata	Object	r	Additional name / value pairs obtained from the metadata content string.

## 6.10 TrickModeRestrictions Properties

Name	Data Type	Read or Write	Description
pause	Number	r	Specifies if pause is restricted during Ad playback.

rewind	Number	r	Specifies if rewind is restricted during Ad playback.
fastForward	Number	r	Specified if fastForward is restricted during Ad playback.