

SCTE-104/35 and Beyond: A Look at Ad Insertion in an OTT World

Ciro A. Noronha, Ph.D.

Cobalt Digital

Introduction

Ad Insertion is a very important part of many video delivery systems, because of the monetization aspect – it generates revenue! With Over-The-Top (OTT) video delivery on the Internet, the holy grail of advertisement is finally achievable – it is technically possible to send individual, personalized ads to each viewer. Such systems are based, in part, on the traditional ad insertion workflows that use the SCTE-104 and SCTE-35 standards as their starting point. This paper provides an overview of such systems, showing how a traditional ad-insertion workflow at the programmer side can be used as a basis for an OTT system. We also show some other uses for the ad insertion infrastructure (for program delimitation) and comment on the importance of frame accuracy.

The Programmer Side

The general ad insertion workflow is:

- Start with video feed, containing programs and ads. This is typically a national network feed. Some of the ads are high-priority national ads, and some of the ads are low-priority national ads that could be replaced by local ads down the chain.
- An automation/playout system “decorates” the baseband video feed with markers that delimit the ads. These baseband markers are defined in SCTE-104.
- An encoder converts the baseband video into a compressed bit stream. The baseband markers are translated into compressed stream markers for transmission with the content. The compressed stream markers are defined in SCTE-35.
- Somewhere in the reception chain, before the video is delivered to the consumer, new ads are spliced in the locations indicated by the markers. This is where OTT systems take over.

The Programmer Side is responsible for generating the compressed bit stream with the appropriate ad insertion markers. A general diagram is presented in Figure 1.

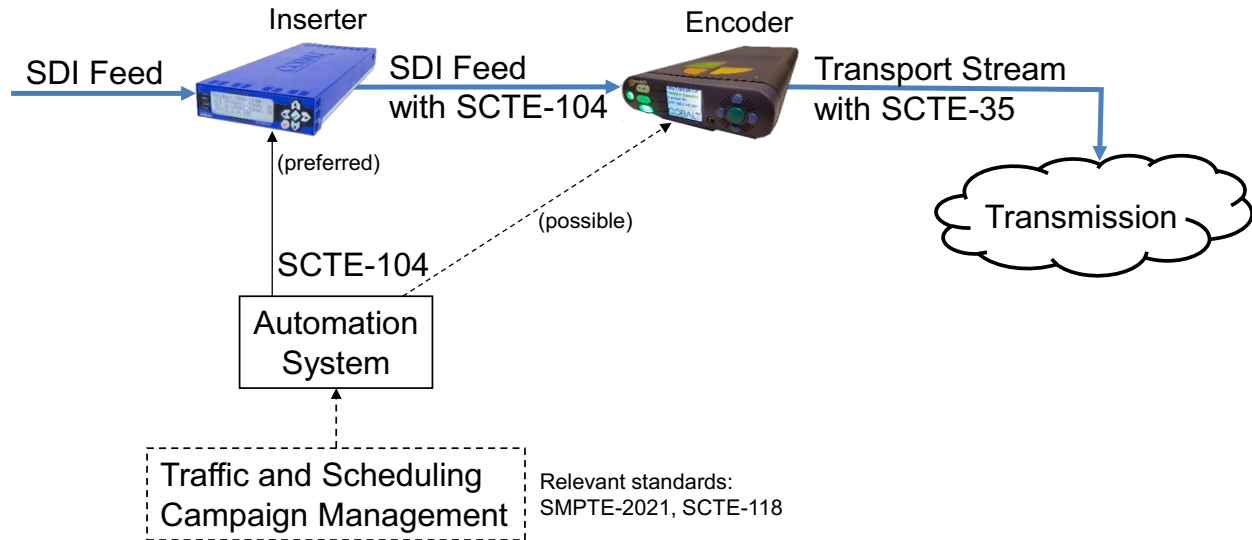


Figure 1: Programmer Side Workflow Diagram

As indicated in Figure 1, the SDI feed from the network goes into an inserter that will add the SCTE-104 markers in the VANC using SMPTE-2010. The inserter does this under the control of an automation/playout system. The interface between the automation system and the inserter is also defined by SCTE-104. The output of the inserter is an SDI signal “decorated” with the SCTE-104 markers. An encoder converts this SDI signal into a compressed bitstream; the SCTE-104 markers are translated into SCTE-35 sections in the bitstream. It is also possible to skip the inserter and have the automation system directly control the encoder (again using the SCTE-104 network interface protocols), but this is discouraged by the standards.

The final output of the programmer side is a transport stream with the compressed content, plus the SCTE-35 markers. This is the feed to the affiliates, and is a good starting point for the OTT ad insertion workflow.

The Traditional Delivery System

Figure 2 shows the traditional ad delivery system. The local affiliate receives the transport stream decorated with SCTE-35 markers from the programmer side. This transport stream goes into a splicer, which reads the markers, contacts an Ad Server using the SCTE-30 protocol to request a replacement ad, and splices the replacement ad in the correct place in the transport stream. The output of the splicer is a new transport stream with the national low-priority ad replaced by a local ad. The splicer will also typically filter out the markers (so as not to facilitate the work of “commercial killers”). Back office management systems will interact with the ad server to facilitate the selection of the ad to be played, and to record the fact that the ad has been played for tracking purposes.

The traditional system has a “one-size-fits-all” approach. The resulting output is a fully compliant linear stream, decodable by any set-top box, no special features or functionality needed. This stream typically goes into a cable plant, or a terrestrial transmitter, and all receivers in the service area show exactly the same ad.

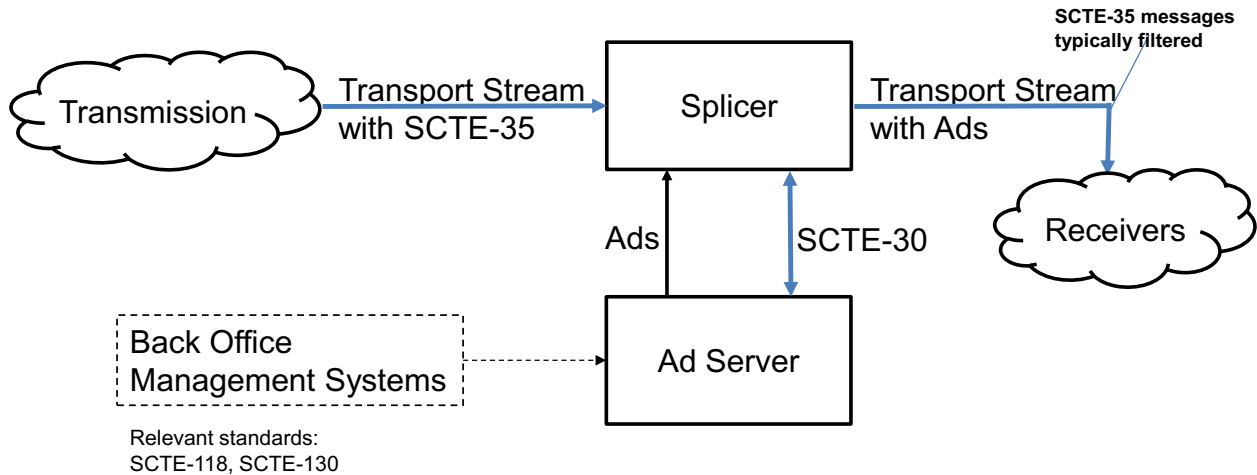


Figure 2: The Traditional Delivery System

A variation of the traditional system has been proposed in the SCTE-138 standard to allow for a small amount of individual ad targeting. In this approach, the transport stream will contain the main program and a certain number of additional ads, carried as separate audio/video PIDs in the same service as the main program. Markers are left in place and processed by the receiving set-top boxes. The set-top box will then decide which one of the small number of available ads to display (or not – it can also leave the ad in the main program). This allows for a very limited amount of ad targeting.

OTT System Basic Operation

The term “Over-The-Top” (OTT) refers content delivery services using the Internet (i.e., “on top” of the network services from the provider). Figure 3 shows the basic OTT operation.

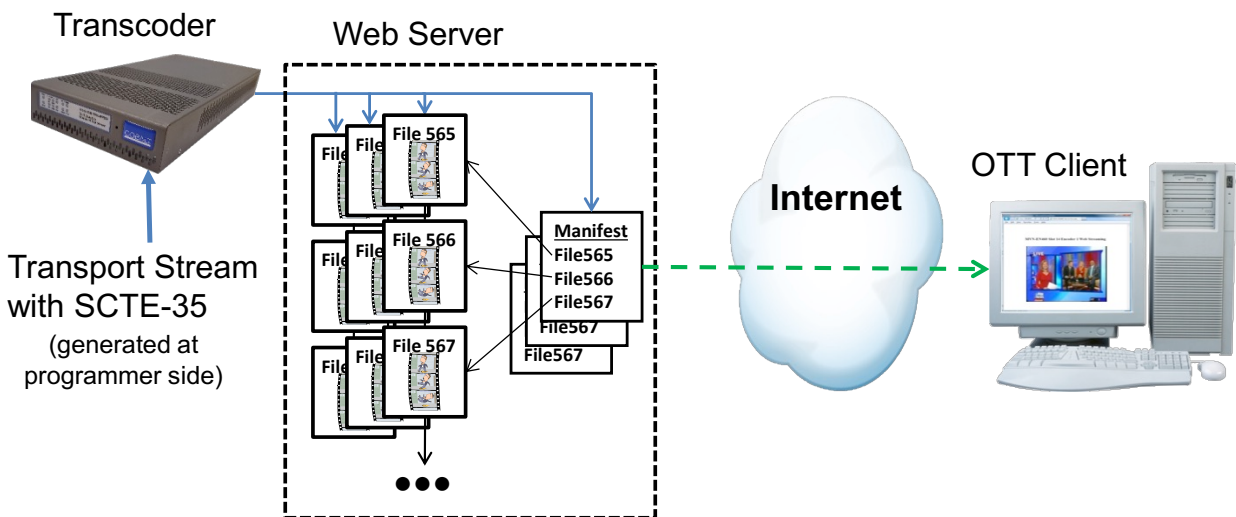


Figure 3: Basic OTT Operation

While there are many protocols used in OTT, they all work as follows (see Figure 3):

- The original transport stream from the programmer side (possibly decorated with SCTE-35 markers) goes to a transcoder device.
- The transcoder device produces several versions of the stream, at different resolutions and bit rates. These versions are called “profiles”.
- Each profile is further divided into individual files, called segments. Each segment is individually decodable – in other words, no data from a previous segment is required to start decoding it, and it can be decoded up to its last frame, with no data required from the next segment. For H.264 streams, the segment starts on an IDR (“Instantaneous Decoding Refresh”) frame, and the last GOP (“Group Of Pictures”) of the segment is closed.
- Segments correspond to a few seconds of video (between 2 and 30 seconds, typically around 5 to 10 seconds).
- Segments are placed in a web server.
- A “manifest” is also placed in the web server. The manifest lists the segments, and there is a top-level manifest that lists the available profiles and their characteristics. Manifests are text files, and their format changes from standard to standard.
- Playback devices will read the top level manifest and learn the available profiles. They will then decide on a profile, read its individual manifest, and start reading decoding the segments. If the network conditions change, the playback device may switch to a higher or lower profile as needed. On a live stream, manifests are frequently updated.

From an ad insertion point of view, OTT delivery has the following very desirable characteristics:

- **OTT is unicast:** each player device establishes its own connection to the server. With appropriate support, ads can be personalized to each specific viewer. Moreover, since these are connections to a web server, the user identification and tracking methodologies developed for the web can be used here as a basis.
- **OTT uses a sequence of short fragments:** since each fragment is individually decodable, splicing becomes straightforward – the player can simply switch to new content at the end of a fragment. This can even be done without the player’s direct cooperation simply by manipulating the manifest to point at the ad.

OTT feeds can carry the original SCTE-35 markers from the programmer side. The general workflow is:

- In general, the video frame identified by the SCTE-35 marker is not aligned with a segment boundary. The transcoder creating the profiles must ensure that a new segment starts at that frame – i.e., it will terminate the previous segment “early”.
- The SCTE-35 marker is added to the manifest. Depending on the OTT standard being used, the way to do that varies as follows:
 - SCTE-67 details how to do this for HLS, DASH and HDS. Some of the same information can also be found in SCTE-35 2016.
 - The DASH implementation is detailed in SCTE-214.

Figure 4 shows an example of an OTT flow with an ad insertion opportunity.

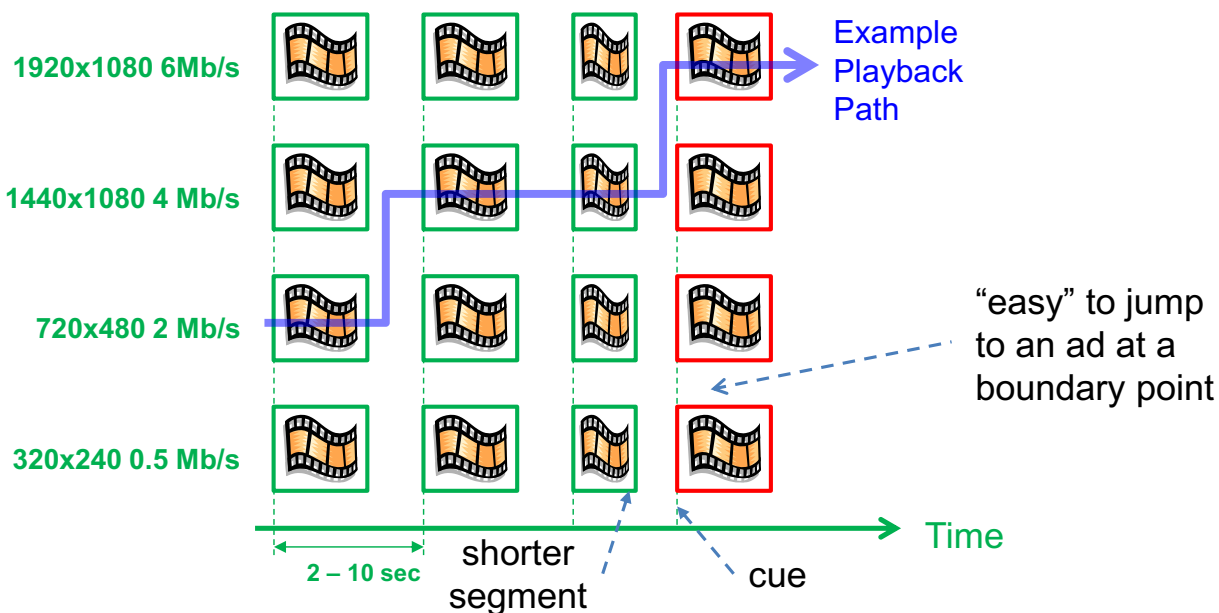


Figure 4: Example OTT Playback Path

OTT Ad Insertion System Control

In some situations, additional control is required for OTT systems, due to the following:

- Geographical constraints on where content can or cannot be presented.
- Time constraints on when content can or cannot be displayed.
- Differences in how the SCTE-35 messages are interpreted between different providers may require them to be translated.
- Choices between how the ad is inserted – it be left up to the playback device, or will it be done by manifest manipulation?

This functionality is provided by the CableLabs **E**vent **S**ignaling **A**nd **M**anagement API (ESAM). This is a set of interfaces between a transcoder, segment packager, and a control element. It provides the following functionality:

- Marker signal conditioning:
 - Validity, start/stop times, duration, additional data to be inserted
 - Content restrictions (blackouts)
- Manifest conditioning:
 - Instructions on how to manipulate the HLS manifest

The standard uses HTTP RESTful transport with XML or JSON payloads. Figure 5 shows the overall ESAM system diagram.

An alternate standard for a subset of this functionality is SCTE-224, the Event Scheduling and Notification Interface (ESNI). This standard, however, is designed primarily for blackout scenarios.

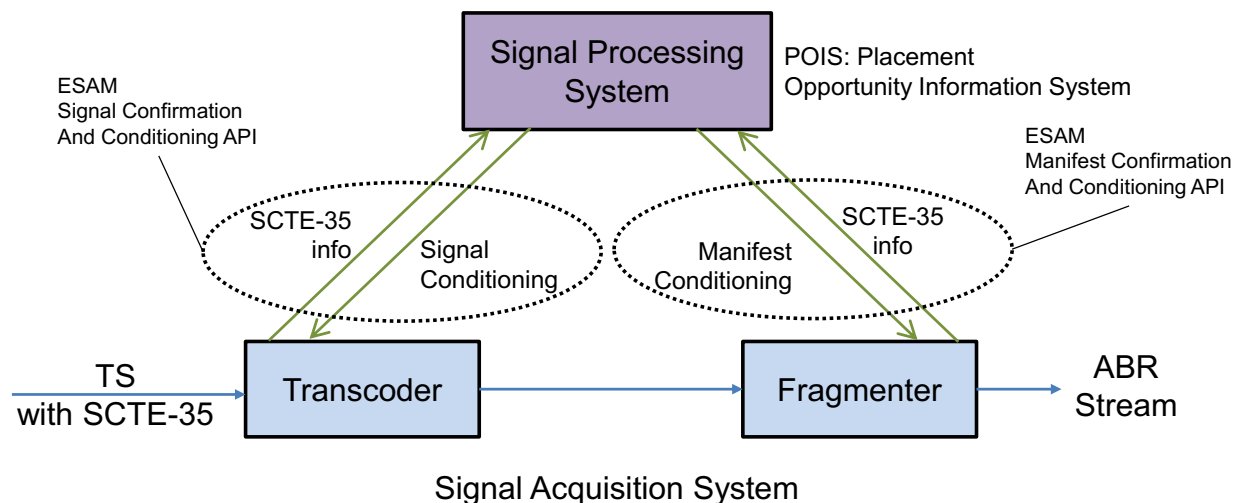


Figure 5: ESAM Diagram

OTT Ad Insertion

After the program is transcoded, converted to OTT profiles, and possibly conditioned, the actual ad insertion can happen. There are two possibilities for the insertion:

- **Client-Side Ad Insertion:**
 - Client processes the manifest and learns of an ad insertion opportunity
 - Client reaches out to a server and gets an ad to play
- **Server-Side Ad Insertion:**
 - Server manipulates the manifest provided to client, to point at the ad at the correct points in time
 - Client is unaware of this – just goes on playing

Note that both possibilities support the notion of individually targeted ads.

The next step in the workflow is to decide which ad to play. In the traditional system, a splicer simply uses SCTE-30 to request an ad from the ad server; in the simplest system, the ad server has a list of ads and provides them in sequence. However, if the final objective is to individually target the ad, a more sophisticated system is required. Two standards cover this functionality:

- The Interactive Advertising Bureau (IAB) created a standard called VAST (Video Ad Serving Template) for a client to request an ad from a server.
 - VAST is a layer on top of browser technology – HTTP is used for the player to server interactions.
- Another option is the SCTE-130 set of standards.

VAST is typically used as an interface to get ads from third parties, while SCTE-130 is used when only one party controls (or owns) the whole infrastructure.

VAST Operation

VAST supports both client-side and server-side ad serving. Figure 6 illustrates the client side ad serving.

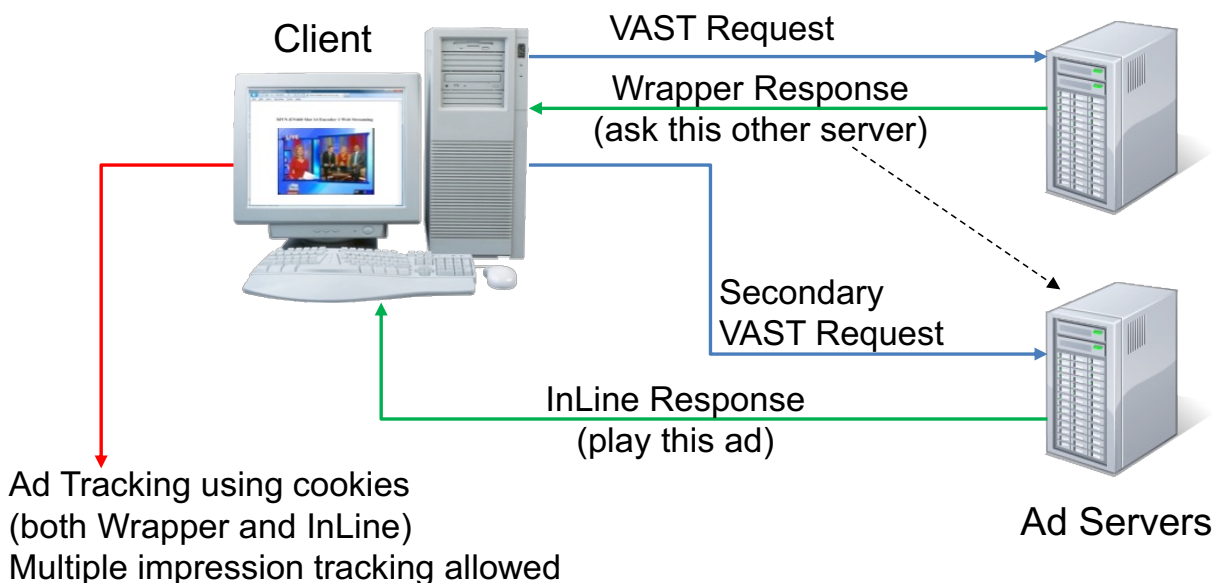


Figure 6: VAST Client-Side Ad Serving

The steps shown in Figure 6 are as follows:

1. In response to a marker, the client sends a VAST ad request to an Ad Server.
2. The Ad Server may return the ad, or may point the client to another server. In the figure, the first ad server sends a Wrapper Response to the client, directing it to contact another server.
3. The client sends a secondary VAST request to this other server.
4. In the figure, the second server provides an InLine Response with the ad to be played.
5. The client plays the ad. Ad tracking happens at the client, using standard web cookies.

Figure 7 illustrates the server-side ad serving in VAST. The mode of operation is sometimes preferred because it is less dependent on the client capabilities, and everything is managed at the server side. A standard OTT client can be used.

The steps illustrated in Figure 7 are:

1. In response to a marker, the splicer sends a VAST request to the server. The splicer itself can trigger this request by inspecting the manifest and finding the marker, or the request may come from the client.
2. The server will send a VAST response, which will include the ad in a mezzanine file format. This is a high bit rate format with good quality that will need to be transcoded prior to transmission to the client.
3. The splicer will contact a transcoder for the ad. The transcoder may have a cached version of that ad ready to go; in this case, it is provided to the splicer, which will “stitch” the ad in the right place and send to the client. The standard has provisions for the case where the transcoder does not have a cached version of the ad – in this case, another lower priority ad is served, while the transcoder prepares the new ad. If this happens, this insertion opportunity is lost, but the next time this ad is requested, it will be ready.

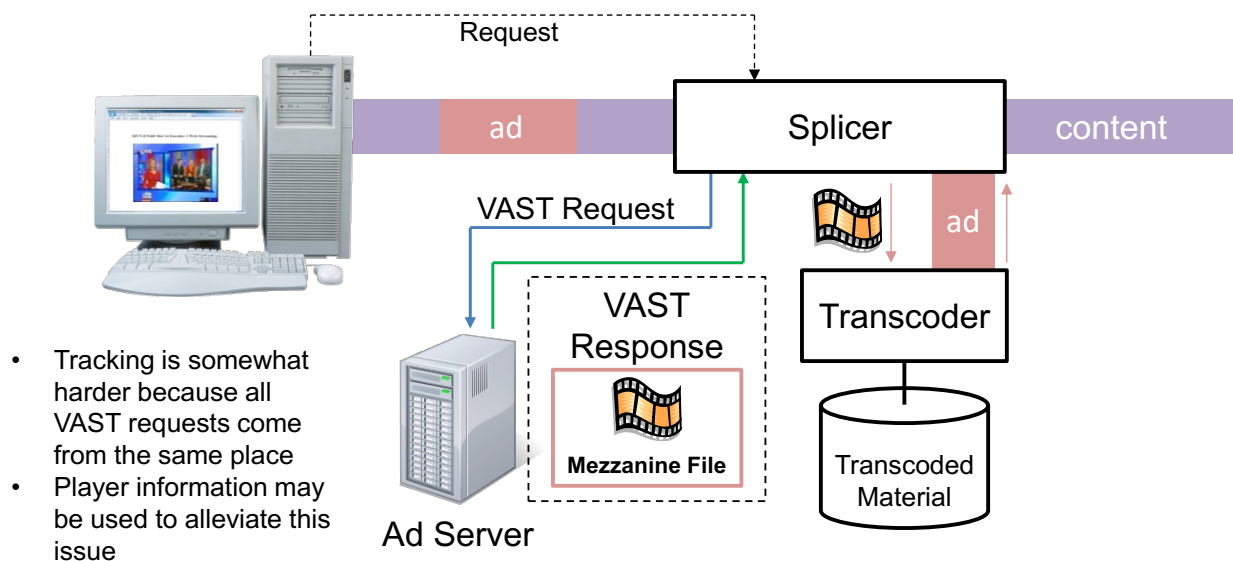


Figure 7: VAST Server-Side Ad Serving

One of the issues with server-side ad serving is the tracking of impressions. From the point of view of the ad server, all requests come from the same device (the splicer), so this may appear similar to faked traffic fraud. However, the standard allows for additional headers to identify the client.

Interactive Ads with VAST

The Interactive Advertising Bureau has defined a protocol called “Video Player-Ad Interface Definition” (VPAID) to support interactive ads. This is an extension of the client-side ad serving case.

VPAID is layered on top of VAST. As part of the VAST response, the server may provide a VPAID ad unit. This is an executable “app” that remains in contact with the appropriate server, providing interactivity and possibly impression reporting. Executable ads can be written in ActionScript 3, Silverlight, or JavaScript.

A Quick Look at SCTE-130

SCTE-130 defines logical functions and interfaces for managing advertisement systems, and is similar to some aspects of VAST. The logical functions provided are:

- Ad Management Service (ADM)
- Ad Decision Service (ADS)
- General Information Services (GIS)
 - Content Information Service (CIS)
 - Placement Opportunity Information Service (POIS)
 - Subscriber Information Service (SIS)

SCTE-130 uses XML-based data interfaces, defined in part 2 of the standard. The network transport is SOAP, defined in part 7. Figure 8 shows a diagram of the basic blocks in the SCTE-130 set of standards.

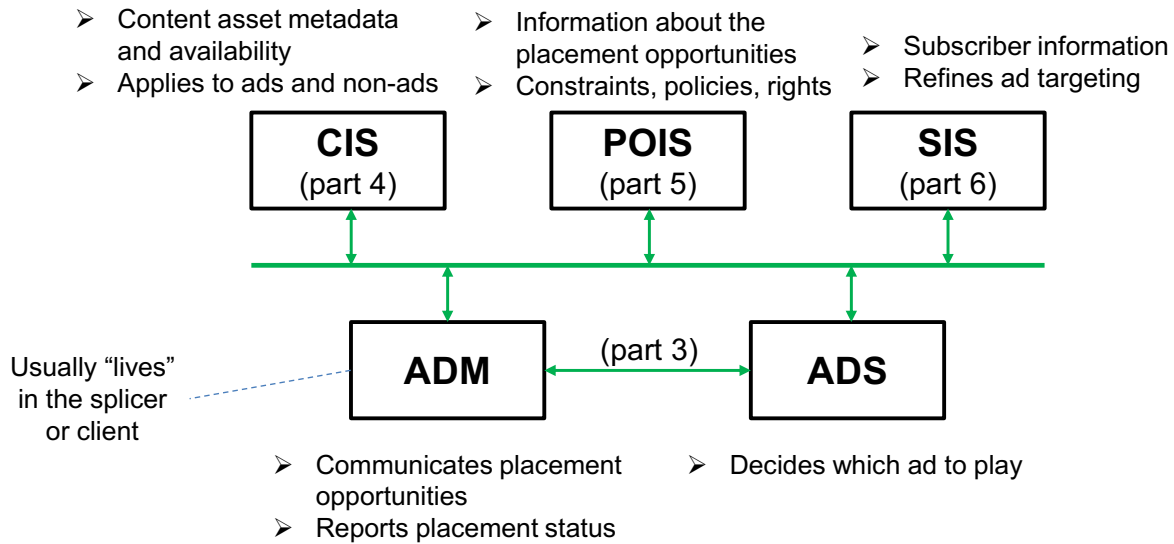


Figure 8: SCTE-130 Basic Diagram

The blocks indicated in Figure 8 are:

- The Ad Management System (ADM) is a module that is typically implemented in the client or the splicer. It has access to the SCTE-35 markers. When it receives a marker, it contacts the Ad Decision Service (ADS) to figure out which ad to play. SCTE-130 part 3 defines this interface. A basic ad insertion system only needs these two modules. However, in order to achieve targeted ad insertion, a number of other systems are required.
- The Content Information Service (CIS) has knowledge of what is available to be played. It can manage both ads and programs. The interface to the CIS is in SCTE-130 part 4.
- The Placement Opportunity Information System (POIS) manages policies, rights and constraints. The interface to the POIS is in SCTE-130 part 5. Note that ESAM can also be used here.
- Finally, the Subscriber Information System (SIS) may have knowledge of individual subscribers, and may help refine ad targeting. The interface to the SIS is in SCTE-130 part 6.

Putting It All Together

Figure 9 shows a high-level block diagram of the main ad insertion modules described in this paper.

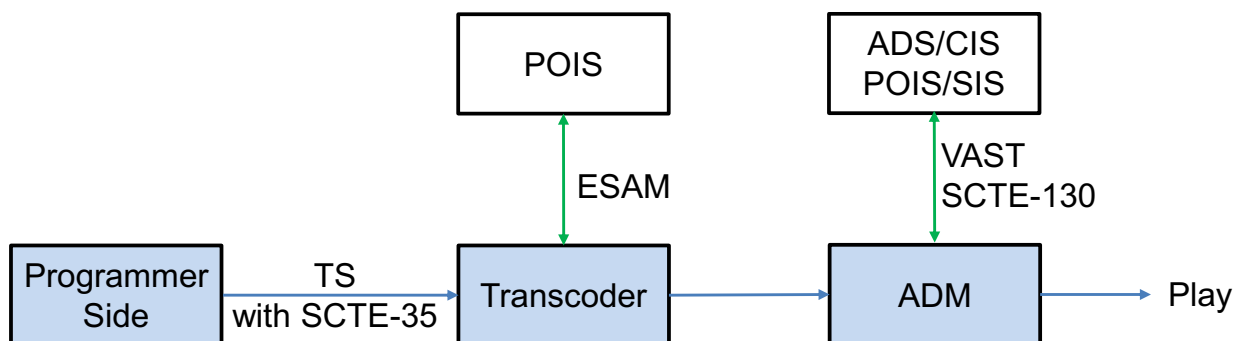


Figure 9: Overall Ad Insertion System Diagram

A vastly more detailed block diagram can be found [in the SCTE DVS site](#), and is reproduced in Figure 10 below.

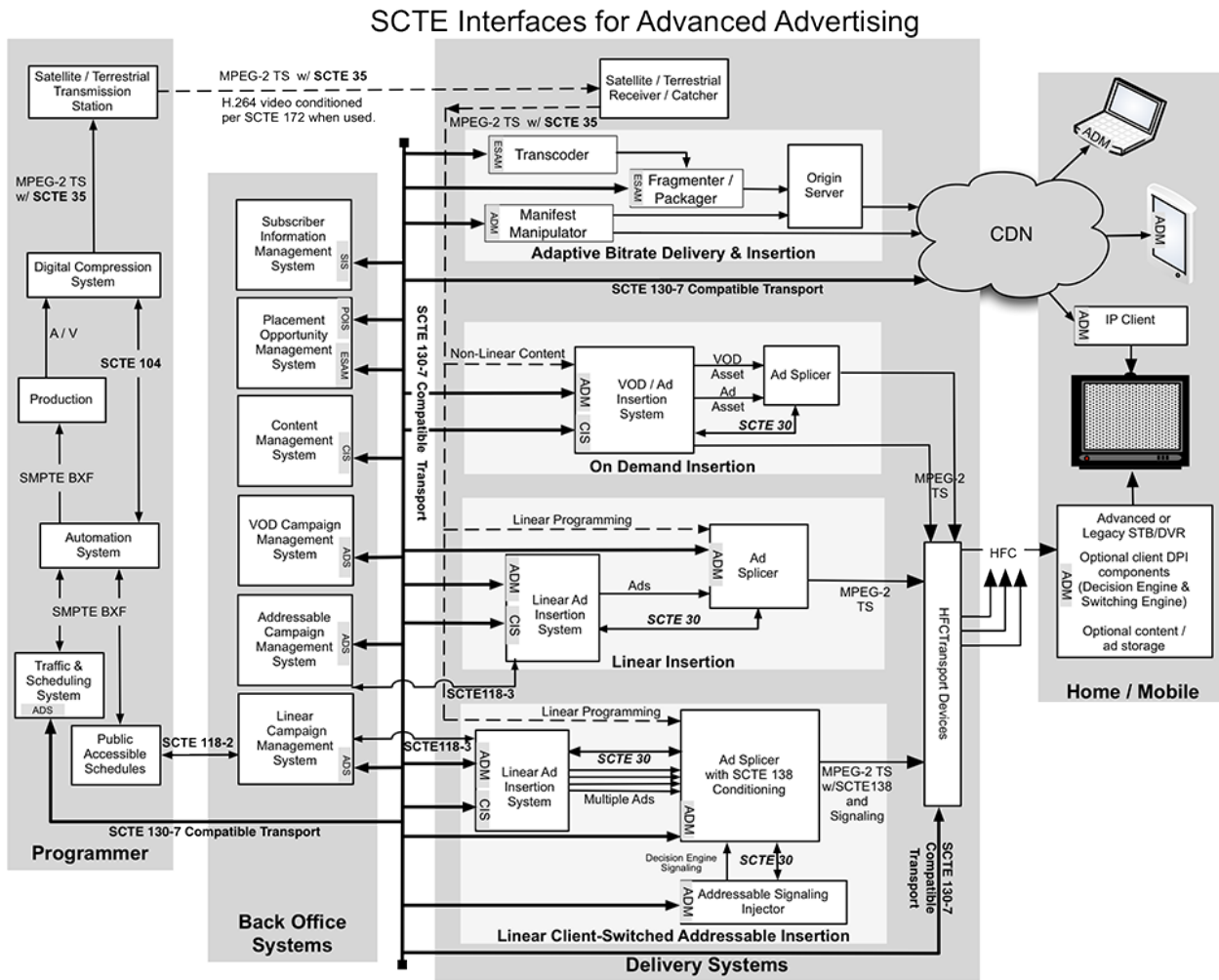


Figure 10: Detailed Block Diagram (Source: SCTE DVS)

The figure indicates the main interfaces and protocols previously discussed in this paper.

Other Applications for SCTE-104/35 Markers

At the most basic level, SCTE-104/35 markers delimit points in time for the program. Besides signaling ad avails, the markers can also delimit programs and other regions of interest. The following applications are using this infrastructure today:

- Delimiting programs for DVR recording:
 - EPGs are usually not very precise – recording may miss head or tail
 - Programs may not be exactly aligned at the scheduled time
 - Triggering the DVR by SCTE-35 messages ensures that the recording does not miss the beginning or end of the program.

- Enforcing DRM restrictions on specific programs and regions
- Whole program replacement:
 - Due to content rights, sometimes an entire program cannot be sent to a given region or transmitted over the Internet.
 - SCTE-35 markers can be used to replace it with a slate or another program.
- Trim program regions for submission to Nielsen ratings
- Use the markers to create VOD content:
 - Live content may be recorded into VOD servers for later access.
 - SCTE-35 markers can be used to segment these recordings and automatically create separate programs in the VOD server.
- Use SCTE-35 metadata for other applications
 - Example: MLB stats

Frame Accuracy

At the inserter in the programmer side (refer to Figure 1), the SCTE-104 markers are associated with very specific, well-defined video frames. These markers are inserted in the VANC for a specific frame, so they are completely frame accurate. When the baseband video is compressed, the corresponding SCTE-35 markers make a reference to the Presentation Time Stamp (PTS) of a specific frame in the bitstream. Therefore, with the right equipment, frame accuracy can be maintained end-to-end.

Applications can benefit from this as follows:

- Frame accurate DVR recording: start and end at the right places
- Frame accurate VOD content creation from live sources
- Exact ad placement