BEST PRACTICES FOR OTT DYNAMIC AD INSERTION

Matt Smith

Anvato Inc.

ABSTRACT

HTTP adaptive streaming (HAS) has become the de facto mode of delivering content for over-the-top (OTT) video services , both live and on-demand. As these services grow and mature, so does the need for viable, robust capabilities for monetization and personalization. These capabilities (and more) are driving many programmers and broadcasters to look to DAI to help drive additional revenue streams on the plethora of devices now capable of streaming. This paper and presentation will explore some of the key considerations related to OTT DAI.

First, what are the the fundamentals of ad stitching and the differences between legacy client-side ad insertion (CSAI) and today's server-side ad insertion (SSAI)? How can those who deploy OTT services defeat the very real concern posed by ad blockers and reach the audiences that seek their content?

This paper will also look at IAB standards (e.g., VAST/VPAID) and how OTT adverts can now deliver national/local/regional payloads within ad pods, essentially mimicking the broadcast world and making true monetization a reality for content creators who have long looked at OTT and broadcast as separate worlds.

Finally, we will discuss the implications of personalization and true user targeting, which are now a reality for OTT services using a variety of data sources (GPS, postal code, IP tables). These capabilities represent a great advance for OTT and the granular experiences it can provide. This means that service providers can deliver adverts that are more relevant to the viewer, OTT is more impactful as an overall experience, and will likely drive ad rates (and revenue) higher over time. This will also include interactive overlays, where advertisers can now provide viewers on connected devices with tangible elements (coupons) they can redeem. These new capabilities help to cement relevant experiences for the viewer and audience and increase revenue opportunities for OTT providers.

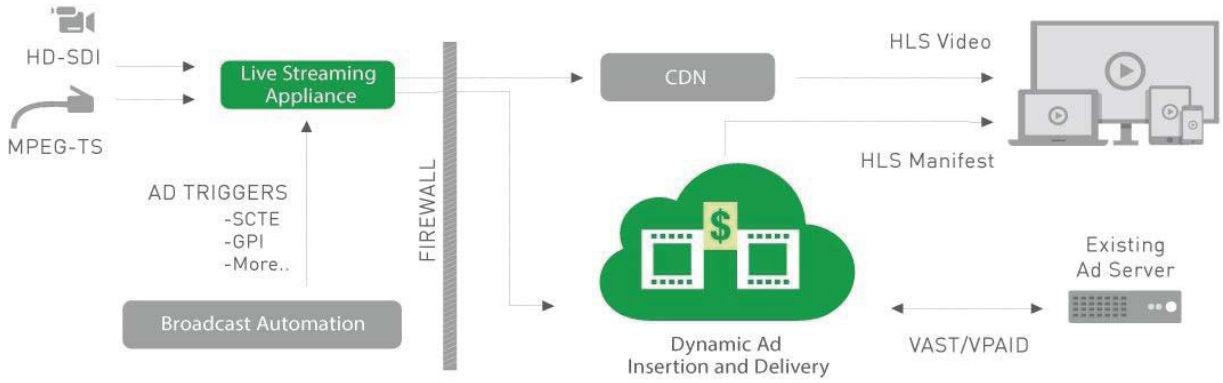
FUNDAMENTALS OF AD STITCHING

With the streaming format wars in the rear view mirror, HAS, specifically Apple's HTTP Live Streaming (HLS) and the Dynamic Adaptive Streaming over HTTP (MPEG DASH) specifications now allow for the efficient, scalable delivery of media content globally from conventional HTTP servers. True to the name, this content adapts to changing bandwidth rates across fixed line and wireless networks, delivering the best experience relative to the available throughput and screen size. These standards are enabling high definition quality video (and

soon, 4K) content to be consumed globally, on any capable device. Now, these streams are moving from being a cost center to a profit center, as programmers, service providers and broadcasters seek to monetize these experiences in the same way they have been able to for their traditional services for years.

Now, OTT delivery technology is being relied upon to not only bring TV-like experiences to all screens with ad experiences that have seamless transitions between content and ads with no buffering across all screens, but also adding elements through interactivity, personalization and more. The actual function of inserting the ads is reliant on some elements inherent to traditional broadcast.

Most broadcast workflows and the accompanying ad insertion workflow rely on SCTE 35 cue messages passed through the MPEG-2 Transport Stream, or SCTE 104 in SDI (baseband) video, marking the placement of an ad pod. This pod usually is comprised of national ad payload, and also an ad replacement opportunity for a local broadcast affiliate. The video workflow components read this cue message and translate it to an ad marker format specific to HLS or DASH. The ad marker is read and the player client relays the ad break and user metadata to a VAST-compliant ad decisioning network. The ad decisioning network provisions the replacement ad, which is then seamlessly inserted into the video stream either server or



client-side. Figure 1 shows the typical elements in a live DAI workflow.

Figure 1. Live dynamic ad insertion (DAI) workflow

To be clear, ad insertion has been a part of the streaming vernacular for the past few years. In early iterations, the request for an ad has taken place at the client, or player. This approach is referred to as client side ad insertion or CSAI. More recently, a move toward server side ad insertion (SSAI) has been an observed trend.

CSAI VERSUS SSAI: STITCHING COMES OF AGE

Ad stitching became a popular solution for monetization in the industry when device fragmentation became a problem. At the time, web video, cable/IP television, set-top boxes, game consoles, and mobile handsets all lacked client-side capabilities, so developers on those

platforms came to rely on stitching to eliminate the problem. In other cases, server-side insertion was the only available option, as was the case with the first generation Apple TV.

These issues have since been resolved, with industry factions either selecting client-side support outright or spending some engineering cycles to customize the client side during initial development. Others choose to use SSAI, while some on both sides continue to hope for an all-purpose solution that simplifies device fragmentation challenges.

Stitching is an excellent solution for most use cases, but certainly not all. The essence of the problem is that most current ad technologies, to include programmatic trading, are built on client-side methodologies. These client-side elements measure things like viewability, interactivity and clickability in the video experience, where SSAI specifications have been slower to address these measurements.

When an ad is stitched on the server side, developers must often implement client-side code in order to add current ad technologies, increasing complexity and maintenance. The new IAB VAST 4 standard will address some of these challenges, but won't cover every use case.

Despite its ability now to more effectively measure some metrics, CSAI is not well equipped to deal with a massively growing element of OTT services: the live event or broadcast. Live broadcasts are very challenging for CSAI because of the latency it introduces. CSAI technology functions by dynamically fetching an ad immediately once a user invokes playback, creating a lag between call and response. This latency can create errors in frame accuracy and cause an ad to be served too late or too early. Server-side ad insertion, by contrast, perfectly "stitches" the ad and the video into a single piece of content, mimicking the television experience.

One distinct challenge with ad stitching is that it prevents publishers from serving popular ad formats. For example, stitching doesn't work with Video Player-Ad Interface Definition (VPAID), an ad format that represents more than half of all video pre-rolls served in Europe. This shortcoming and inability to serve VPAID generally means lost advertising revenue. VPAID tags are relied upon to measure the viewability and verification mentioned above, and in some cases interactivity, which has led some agencies and the advertisers they serve to insist on VPAID, rather than VAST tags in order to support this.

But change is in the works. VAST 4.0 allows verification and interactivity to be addressed, where they are loaded as part of the stream-stitching model, providing a path to support verification and interactivity. This is critical when it comes to very large scale live streamed events offered by major broadcasters— arguably the most valuable inventory in the video ecosystem. VAST 4.0 also makes non-stitched desktop ads load quicker by splitting out verification/viewability calls and eliminating fourth-party served ads.

Expect to see server side ad stitching to gain more traction and mitigate the aforementioned VPAID gaps as the VAST 4.0 spec is more broadly adopted. SSAI can deliver the consistent quality of user experience with live feeds and help avoid viewer drop-offs because its single-stream feed eliminates pauses between content and ads. Further, it is the only way to accommodate the growing audiences who are streaming more and more live content. CSAI

won't die a quick death by any stretch, but it is becoming an at-risk approach because of the new 500 pound gorilla in the video space - the ad blocker.

AD BLOCKING AND CIRCUMVENTION

Some say that the digital ad process is broken and that publishers and advertisers need to work together on a common mindshare and ultimately, a fix. Others say that ad blocking is a technology problem in search of a solution. Regardless of position, ad blockers represent a real and present threat to publishers' content and ad revenue today.

Some video producers estimate that up to one-third of their streams never reach their audience today. Analysts indicate that ad blockers could cost publishers $25 billion in 2016 for both display and video advertising. Globally, over 200 million devices will use ad blockers, an estimated 40% growth year over year.

Video ad businesses that rely on pre-roll ads are hit the hardest, because ad blockers either skip the pre-rolls or prevent video playback all together. Enter stream stitching,a technology that while not new is helping to circumvent ad blocking technology today. Stitching isn't specific to the advert either, and enables per-user content replacement when needed.

To understand why ad stitching is important, you must first understand how video ads are currently served. Videos and video adverts are coming from different places — videos typically from a content delivery network (CDN) and ads from a third-party ad decisioning network (ADN), such as Doubleclick or Freewheel. These elements are then assembled on the fly when a viewer begins watching content. With server-side ad stitching, that combination of video and advertising happens in the cloud. In this scenario, the ad is sometimes adjusted by a process called just in time (JIT) transcoding, for scenarios when the ad payload isn't of the same quality as the program stream.

Ad blockers function by checking web pages and then using filter lists to block domains and addresses that lead to third-party ad servers. The ad blockers can then prevent ads from being served while also letting publisher content through. In the case of stream and server-side ad stitching, though, since the stream and ad "look" like one contiguous stream, if the ad doesn't get through, the video itself can't either.

Server-side ad insertion was actually created to solve other problems with video advertising. Combining video and advertising assets into a single stream avoids, for example, issues where adverts around the video load but the videos don't. Likewise, server-side ad insertion also keeps the quality of the elements consistent, which means better video quality and little to no buffering. SSAI also provides for broadcast-like fallback scenarios. For example, if a lower quality ad is fetched, there is no payload present, a promo or house ad can play for a small swath of users while the system communicates with the ADN for the appropriate ad elements.

INDIVIDUALIZED TARGETING AND HYPER LOCAL ADS

Server-side stitching has applications beyond advertising that enable publishers to insert certain kinds of video content on the fly. In-market sports rights - where a viewer is or isn't permitted to

watch a particular game or match, or where broadcast rights, but not streaming rights have been provisioned - a case for on-the-fly stream stitching is made. In this scenario, the publisher can replace the game or program with an alternative show that is provisioned for that particular viewer. Further, this technology could be used to support hyper local content. So, when a viewer is watching a nationally delivered weather channel, for example, the local weather forecasts and commercial breaks can be targeted down to the postal code, providing them with ads and content most relevant to them.

Given that each viewer in these scenarios maintains a unique connection to the cloud, and in turn, server, delivery platforms are able to deliver highly relevant, individualized content to them. Targeting the individually connected viewer can happen using a variety of means - from postal codes to IP address tables and GPS data. This means that the publisher (and in turn, advertisers) has the ability to group viewers in places and at times that are relevant to those buying audience. For example, Starbucks may seek 50,000 viewers in any London post code each morning between 7-9am. Using the data aforementioned, they can target mobile users, and even offer them a savable coupon where the viewer can scan the device in-store for a discount on their beverage. This is one example of the highly targeted, extremely relevant ad workflows that both publishers and advertisers can now utilize to increase viewer engagement and monetization.

CONCLUSION

Change is a constant in the video delivery space. As this change extends to OTT, it has enabled HTTP adaptive streaming, and now as ad replacement is evolving, SSAI is improving monetization strategies and enabling streaming delivery services to mimic (and even improve upon) the broadcast workflows compliment. Today, these OTT workflows are increasingly being managed more efficiently and no longer are looked at as separate and disconnected from their broadcast brethren.

Despite their past differences, technologists, publishers and advertisers must refrain from engaging in a religious war pitting CSAI and SSAI. Look to the VAST 4.0 spec to reduce feature and functionality gaps a great deal and enable broader monetization efforts. The benefits from circumvention of ad blockers alone is well worth the investment. In turn, revenues from streaming will increase by orders of magnitude as a result of being “better than broadcast”.