YOSPACE®

CLIENT-GUIDED AD MEASUREMENT WITH CMCDv2

BRIDGING THE STANDARDS GAP IN SERVER-CENTRIC AD INSERTION

D. Springall and O. Cortambert

Yospace, England

CONTENTS

INTRODUCTION - THE NEED FOR MEASUREMENT 2

1. VAST AND THE MEASUREMENT MODEL 3

2. MEASUREMENT IN CLIENT-SIDE AD INSERTION (CSAI) 4

3. MEASUREMENT IN SERVER-CENTRIC AD INSERTION MODELS 5

4. INTRODUCING CLIENT-GUIDED AD MEASUREMENT (CGAM) 6

5. INTRODUCTION TO CMCD 8

6. IMPLEMENTING CGAM WITH CMCDv2 9

- 6.1 CMCDv2 as an enabler for CGAM 9
- 6.2 Required telemetry for advertisement measurement 9
- 6.3 Playhead reporting standardisation 10
- 6.4 System architecture and responsibilities 10
- 6.5 Playback workflow example (VOD) 12

7. DISCUSSION: ACCURACY, TRUST, AND LIMITATIONS 12

- 7.1 Accuracy of derived measurement 12
- 7.2 Trust and verification 12
- 7.3 Limitations and edge cases 13

8. CONCLUSION 14

REFERENCES 15

INTRODUCTION - THE NEED FOR MEASUREMENT

Accurate advertisement measurement is fundamental to the commercial operation of streaming media services. Advertisers, broadcasters, and intermediaries rely on measurement data not only to confirm that advertisements were delivered, but also to determine whether they were viewed, completed, and interacted with in ways that justify their value. As streaming delivery has expanded across increasingly fragmented device ecosystems, providing consistent and trustworthy measurement has become a significant technical challenge.

Modern video advertising involves a distributed delivery chain that may include publisher ad servers, supply-side platforms (SSPs), demand-side platforms (DSPs), and advertiser systems. Each participant may require independent verification that advertisement playback milestones have occurred. This has led to widespread adoption of standardised measurement frameworks such as Video Ad Serving Template (VAST) 3.0, IAB (1), and VAST 4.2 (2), IAB Tech Lab (2), which define how advertisement playback events are reported through tracking requests associated with timeline progression and user interaction.

Historically, these measurement models have depended on the playback environment to observe playback state and execute tracking logic. While effective in fully capable client environments, this assumption becomes increasingly problematic across fragmented platforms, constrained devices, and syndicated playback environments where consistent client-side implementation cannot be guaranteed.

This paper explores Client-Guided Ad Measurement (CGAM), a model in which the player reports only primitive playback telemetry while advertisement measurement logic is executed server-side. Early deployments of CGAM, including implementations developed for ITV's Freely platform, Yospace (3), demonstrated the viability of this approach using proprietary telemetry mechanisms. The introduction of Common Media Client Data version 2 (CMCDv2), CTA Wave (4), provides a standardised framework through which the same model can be implemented in an interoperable manner across devices and streaming protocols.

The following sections examine how existing advertisement measurement models operate, the limitations that arise in modern streaming environments, and how CMCDv2 can be used to enable scalable and standardised client-guided advertisement measurement.

1. VAST AND THE MEASUREMENT MODEL

The Interactive Advertising Bureau's VAST standard defines how video advertisements are delivered and measured within digital playback environments. Although newer versions of the specification have introduced enhancements in areas such as verification and server-side support, the underlying measurement model has remained broadly consistent between VAST 3.0 (1) and VAST 4.2 (2), resulting in continued widespread deployment of both versions.

VAST responses define not only advertisement media assets, but also the measurement events associated with playback. These events are typically represented as tracking URLs corresponding to milestones such as impression, quartiles, completion, and user interactions including pause, mute, and fullscreen transitions. The playback environment is expected to invoke these URLs as the relevant events occur.

A key characteristic of the VAST ecosystem is the use of wrapper responses, allowing multiple participants in the advertising supply chain to contribute their own measurement requirements to a single advertisement transaction. A publisher ad server may reference downstream VAST responses provided by SSPs, DSPs, or advertiser systems, each of which can introduce additional tracking requirements.

As a result, a single advertisement may accumulate a substantial number of tracking URLs associated with playback milestones and user interactions. These URLs may themselves be significant in size due to embedded identifiers, verification metadata, and reporting parameters. The playback environment is therefore responsible not only for determining when measurement events occur, but also for executing potentially large volumes of network requests on behalf of multiple independent stakeholders.

This measurement model has proven sufficiently flexible to support complex advertising workflows across a wide range of playback environments. However, it also creates a strong dependency on the capabilities and behaviour of the client. Accurate measurement depends on the player correctly observing playback state, maintaining awareness of advertisement timelines, and reliably executing tracking requests. Any deficiencies in this process can lead to inconsistencies in reported metrics, undermining confidence in the measurement data and reducing the value advertisers are willing to assign to the associated inventory.

2. MEASUREMENT IN CLIENT-SIDE AD INSERTION (CSAI)

Client-Side Ad Insertion (CSAI) represents the original operational model under which VAST 3.0 (1) and its successors were developed. In this model, the playback environment is responsible not only for rendering advertisement media, but also for executing the associated measurement logic. As a result, advertisement insertion and advertisement measurement have historically been treated as implicitly linked concerns within CSAI workflows.

In a typical CSAI implementation, a player-integrated advertising Software Development Kit (SDK) retrieves and processes VAST responses, resolves any wrapper chains, selects advertisement media assets, and monitors playback progression during advertisement rendering. As playback milestones are reached, the SDK invokes the corresponding tracking URLs associated with events such as impression, quartiles, completion, pause, mute, and fullscreen transitions.

Because both advertisement playback and measurement execution occur directly within the client environment, CSAI provides highly accurate visibility into playback state and user interaction. The player has direct access to the active playhead position, playback state changes, and user input events, enabling precise triggering of measurement signals in alignment with VAST requirements.

However, this approach also places substantial responsibility on the playback environment. The client must correctly implement VAST processing, manage potentially large numbers of tracking URLs originating from wrapper chains, and reliably execute tracking requests across diverse runtime environments. This creates significant implementation and maintenance complexity, particularly across fragmented device ecosystems where platform capabilities and integration models vary substantially.

The tight coupling between insertion and measurement in CSAI is historically significant because it shaped many of the assumptions embedded within the VAST measurement model itself. Later insertion paradigms such as Server-Side Ad Insertion (SSAI) and Server-Guided Ad Insertion (SGAI) inherited the requirement for advertisement measurement, but unlike CSAI, they do not inherently define how such measurement should be achieved. As a result, while CSAI implicitly combines insertion and measurement responsibilities within the client, newer insertion models require measurement to be treated as a separate architectural concern.

3. MEASUREMENT IN SERVER-CENTRIC AD INSERTION MODELS

Server-centric advertisement insertion models, including SSAI and SGAI, shift some or all advertisement insertion responsibilities away from the client and into upstream infrastructure. In SSAI, advertisements are stitched directly into the media stream before delivery to the player, while SGAI approaches – such as HTTP Live Streaming (HLS) Interstitials, Apple Developer (5), and Motion Pictures Expert Group Dynamic Adaptive Streaming over HTTP (MPEG-DASH) Alternative Media Presentation Description (MPD), Giladi et al (6), – allow the server to coordinate advertisement insertion through signalling mechanisms understood by the player.

Despite their architectural differences, both SSAI and SGAI should be regarded solely as advertisement insertion paradigms. Neither defines how advertisement measurement should be performed, nor do they inherently provide the information required to support the measurement model described by VAST 4.2 (2). Measurement therefore remains a separate concern that must be implemented independently of the insertion mechanism itself.

In live SSAI environments, some implementations attempt to approximate playback progression using delivery-side observations such as manifest updates or segment requests. Where playback occurs close to real time and user interaction with the timeline is limited, this can provide a reasonable approximation of which portions of the stream have likely been consumed. However, this approach remains indirect and does not provide reliable visibility into user interactions or playback state changes such as pause, mute, buffering, or seeking.

In video-on-demand workflows, the limitations are more pronounced. Static manifests provide no inherent indication of how the viewer interacts with the playback timeline, making it necessary to introduce some form of client-side observability if accurate measurement is required. As a result, many SSAI deployments reintroduce player instrumentation to track playback progression and user interaction, either to execute measurement directly or to provide telemetry to a server-side measurement system.

SGAI introduces additional complexity because advertisement playback may occur on timelines separate from the primary content stream. Measurement systems must therefore account for discontinuities between content and advertisement playback, requiring explicit awareness of advertisement boundaries and playback transitions. While standards such as HLS Interstitials (5) and MPEG-DASH Alternate MPD (6) provide standardised mechanisms for coordinating advertisement insertion, they do not standardise the hooks or interfaces required for advertisement measurement. Consistent measurement therefore still depends on player-specific implementations capable of observing playback state and user interactions.

These limitations highlight a broader architectural issue: while modern insertion paradigms reduce the dependency on the client for advertisement delivery, they do not eliminate the

dependency on client-side observability for accurate measurement. Regardless of whether advertisements are inserted client-side, server-side, or through guided insertion models, measurement ultimately depends on access to playback state and user interaction signals originating from the player environment.

4. INTRODUCING CLIENT-GUIDED AD MEASUREMENT (CGAM)

CGAM is an approach to advertisement measurement that reconsiders the distribution of responsibility between the playback device and server-side systems. Rather than requiring the client to execute the full set of measurement operations defined by standards such as VAST 4.2 (2), CGAM instead relies on the client to report a reduced set of primitive playback signals. These signals are then used by a server-side system to derive the higher-level measurement events required by the advertising ecosystem.

At a conceptual level, CGAM separates observation from interpretation. The client is responsible only for observing and reporting its local playback state – such as playhead position, playback rate, and user interactions – while the server is responsible for interpreting these signals in the context of advertisement metadata and generating the corresponding measurement events. This contrasts with traditional client-side approaches, where both observation and interpretation are performed within the playback environment.

In a CGAM model, the player periodically reports telemetry to a measurement endpoint. This telemetry is intentionally lightweight and implementation-agnostic, typically consisting of:

- current playhead position
- playback state (e.g. playing, paused, buffering)
- user interaction events (e.g. mute, unmute, fullscreen)

These signals do not directly correspond to advertising events such as quartiles or completion. Instead, they provide a continuous or near-continuous view of playback behaviour from which such events can be inferred. The server-side measurement system maintains a model of the playback session, combining the reported telemetry with knowledge of ad boundaries, durations, and sequencing derived from the ad insertion workflow. Based on this model, it can determine when conditions for specific VAST events have been met and trigger the appropriate tracking on behalf of the client.

This approach offers several advantages over traditional measurement models. First, by centralising measurement logic on the server, CGAM reduces the complexity required within the client. Players are no longer responsible for parsing VAST responses, managing nested tracking requirements, or precisely timing the firing of tracking URLs. This can be particularly beneficial in constrained environments where code execution is limited, or where maintaining platform-specific SDK integrations is impractical.

Second, because measurement events are generated server-side, CGAM is inherently more resilient to client-side interference. Techniques that rely on suppressing or blocking network requests from the device – such as common ad-blocking approaches – are less effective when the measurement logic is executed within controlled server infrastructure. While CGAM still depends on the integrity of the telemetry reported by the client, it removes the direct dependency on the client to execute third-party tracking requests.

Third, CGAM provides a path toward more consistent measurement across heterogeneous devices. By standardising the format and semantics of the telemetry reported by the player, it becomes possible to apply a uniform measurement model across platforms with widely varying capabilities. This can help reduce discrepancies in reported metrics that arise from differences in SDK implementations or device-specific behaviour.

A further advantage of CGAM is its independence from underlying streaming protocols. Because the model relies on abstracted playback telemetry – such as playhead position and user interactions – rather than protocol-specific signalling, it can be applied consistently across delivery formats including HTTP Live Streaming (HLS) (5) and MPEG-DASH (6). This contrasts with approaches that are more tightly coupled to specific protocol features. For example, while initiatives such as SVTA Ad Signalling define detailed signalling and measurement workflows within the context of MPEG-DASH (6), equivalent standardised mechanisms are not uniformly available across other widely deployed protocols such as HLS (5). As a result, achieving comparable behaviour in HLS (5) environments may require additional, potentially platform-specific implementations. By decoupling measurement from protocol-specific constructs, CGAM enables a more uniform and portable measurement model across diverse streaming ecosystems.

An additional consideration is the potential impact on data governance and privacy. In traditional models, tracking pixels are often invoked directly from the client to multiple third-party endpoints, potentially exposing information about user behaviour across organisational boundaries. In a CGAM model, measurement events can be generated within a controlled server environment, allowing publishers to mediate which data is shared with external parties. While this does not eliminate privacy concerns, it may provide a more structured framework for managing them.

It is important to distinguish CGAM from other emerging approaches to ad signalling and measurement, such as SVTA Ad Signalling. SVTA Ad Signalling defines a rich set of client-side behaviours and interfaces for handling advertisement playback and measurement, placing significant responsibility on the player to interpret and act on signalling data. CGAM, by contrast, deliberately minimises client-side responsibilities, instead relying on server-side systems to perform the bulk of measurement logic. These approaches are not mutually exclusive; rather, they represent different points on a spectrum between client-centric and server-centric measurement models.

In practice, many ecosystems may choose to adopt a combination of both approaches. Where capable client environments are available, richer client-side signalling may enable

more granular measurement and advanced analytics. In more constrained environments – such as third-party applications, embedded players, or devices with limited execution capabilities – CGAM provides a viable alternative that prioritises consistency and deployability over client-side sophistication.

Early implementations of CGAM, including those deployed in production environments such as ITV's Freely (3) platform, have demonstrated that this model can be realised using proprietary mechanisms for reporting playback telemetry. While effective, such implementations require bespoke integration between client and server, limiting their scalability across a broader ecosystem.

The introduction of CMCD version 2 (4) provides a pathway to standardising this approach. Originally developed to support client observability in adaptive streaming, CMCD (4) defines a structured method for players to report playback metrics to network and delivery infrastructure. With the extensions introduced in version 2, these reporting capabilities can be leveraged to support CGAM, enabling the transmission of the primitive telemetry required for server-side measurement without the need for custom client-side implementations. This creates the foundation for a more interoperable and widely deployable model of client-guided measurement, which is explored further in the following sections.

5. INTRODUCTION TO CMCD

The CMCD (Common Media Client Data) standard (4) defines a structured mechanism for reporting playback-related telemetry from a media player to delivery and analytics infrastructure. Standardised through the Consumer Technology Association (CTA) as part of the WAVE (Web Application Video Ecosystem) initiative, CMCD enables players to expose key aspects of playback state in a consistent and interoperable format.

CMCD data is typically transmitted alongside media requests, for example via HTTP headers or query parameters, and consists of a defined set of key-value pairs. These may include information such as buffer level, playback rate, session identifiers, and object characteristics. By standardising how this data is expressed and transported, CMCD allows downstream systems to interpret playback behaviour without requiring proprietary instrumentation or platform-specific integrations.

A key characteristic of CMCD is that it operates at the level of primitive observability. The data it conveys reflects the current state of playback, rather than higher-level application logic or business events. This makes it inherently flexible and applicable across a wide range of devices, players, and streaming protocols.

This abstraction is particularly relevant in the context of CGAM. As described in the previous section, CGAM relies on the availability of basic playback telemetry – such as playhead position and user interactions – from which advertisement measurement events can be derived. CMCD provides a standardised mechanism through which such telemetry can be communicated, removing the need for bespoke client-server integrations.

With the enhancements introduced in CMCDv2 (4), this capability can be extended to support more sophisticated measurement use cases. In particular, CMCDv2 enables the consistent reporting of playback state in a form that can be used by server-side systems to reconstruct viewing behaviour and derive events aligned with VAST 4.2 (2). The following section examines how these capabilities can be applied in practice to implement CGAM without requiring custom client-side logic.

6. IMPLEMENTING CGAM WITH CMCDv2

6.1 CMCDv2 as an enabler for CGAM

The introduction of Event Mode in version 2 of the CMCD (4) enables playback telemetry to be reported independently of media segment requests. This represents a fundamental shift from earlier approaches, where observability was inherently tied to the cadence of content delivery.

In this model, the player is responsible only for emitting structured telemetry, while the interpretation of that telemetry – specifically, the derivation of advertisement measurement events aligned with VAST 4.2 (2) – is performed server-side. This removes the need for a full client-side ad measurement SDK, while still enabling the generation of standardised measurement signals. The inclusion of the necessary telemetry in CMCDv2 (4) to support the full VAST tracking requirements has been by design.

6.2 Required telemetry for advertisement measurement

To support CGAM, the telemetry emitted by the player must provide sufficient information for a server-side system to reconstruct playback behaviour and determine when advertisement-related events occur. This can be broadly categorised into three types of signal:

- Playback progression, including playhead position and playback state
- Session correlation, enabling events to be associated with a single playback instance
- User interaction events, capturing actions such as mute, skip, or player state changes

CMCDv2 (4) defines a set of keys that, when used together, provide this information in a standardised format. The table in Figure 4 outlines the key fields required to enable robust advertisement measurement.

Together, these fields provide a sufficient basis for reconstructing both the temporal progression of playback and the contextual events that influence advertisement measurement. See Figure 1 for a visual representation of a simple ad-inserted VOD playback.

6.3 Playhead reporting standardisation

A critical aspect of this model is the standardisation of the time base associated with playback position reporting, ensuring consistent interpretation of playback progression across all player endpoints.

The pt (playhead time) field is normatively defined in CTA-5004-B, CTA Wave (7), as follows: for video-on-demand content, it MUST be expressed as the number of milliseconds offset from the beginning of the media asset; for live streams with a playhead date-time, it MUST be expressed as the number of milliseconds elapsed since the Unix epoch (1 January 1970, 00:00:00 UTC), excluding leap seconds. In the case of an HLS Interstitial (5) or MPEG-DASH Alternative MPD (6), the player must indicate to the endpoint that it has switched to the Alternative Playlist by providing its content-id (cid). In the case when the switch is made to advertising content, the Alternative Playlist is a VOD Asset (or a list of VOD Assets) which requires the player to report the pt as the number of milliseconds offset from the beginning of that playlist.

6.4 System architecture and responsibilities

In a CGAM implementation using CMCDv2 (4), responsibility for advertisement measurement is explicitly divided between the player and a server-side telemetry endpoint.

Figure 1 provides, for illustrative purposes, a hypothetical architecture that incorporates CGAM for measurement.

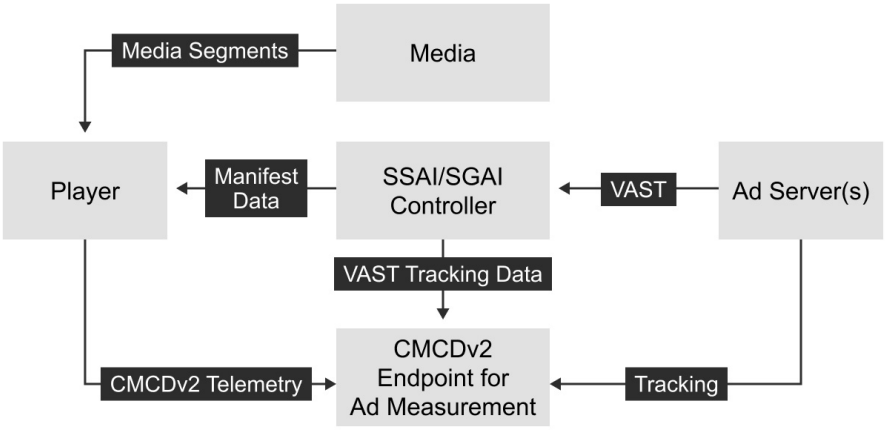


Figure 1 – Hypothetical Server-Centric Ad Insertion incorporating CGAM

Player responsibilities

The player is responsible for emitting CMCD telemetry in Event Mode, configured with the following parameters:

- CMCD Version: 2
- Mode: Event
- Reporting Endpoint: A configured URL to which telemetry is sent
- Reporting Interval: Application-defined (e.g. 1000 ms for playhead updates)
- Transmission Method: Typically HTTP query parameters or POST payloads
- Reported Keys: A defined subset of CMCDv2 fields required for measurement

The player does not perform VAST parsing or tracking execution. Instead, it emits a continuous stream of primitive playback signals, including periodic updates (e.g. playhead position) and discrete events (e.g. mute, seek, ad start).

To support tracking in Server Guided Ad Insertion, the player must also provide the HLS Interstitial ID (5) (ID of the Interstitial EXT-X-DATERANGE) or the DASH Event ID (6) (of the event containing the Alternative Representation). If the player use a two decoders approach (where one decoder is used to play the content and another decoder to play the ad), it also need to send the nr key to inform if one of the decoder is playing in the background (typically content will be in the background when an Ad is played).

Telemetry endpoint responsibilities

The telemetry endpoint is responsible for executing advertisement measurement logic. This includes:

- Maintaining a stateful representation of each playback session
- Reconstructing the playback timeline, including transitions between content and advertisement segments
- Interpreting telemetry to determine when conditions for measurement events are satisfied
- Executing tracking requests in accordance with VAST 4.2 (2), including:
  - impression
  - quartiles
  - completion
  - interaction events (e.g. mute, skip)

To ensure accurate and compliant measurement, the endpoint must account for real-world playback behaviours such as buffering, seeking, and partial ad playback. For example, quartile events should only be fired if the advertisement is played sequentially from its start, and should be suppressed if the user seeks beyond the ad or abandons playback.

This architecture represents a clear shift from traditional models: the client acts as a source of observability, while the server assumes responsibility for measurement execution.

6.5 Playback workflow example (VOD)

Figure 2 illustrates how the CMCDv2 telemetry elements operate in a standard ad-inserted VOD playback.

7. DISCUSSION: ACCURACY, TRUST, AND LIMITATIONS

The CGAM model introduces a different set of trade-offs compared to traditional client-side approaches. By shifting the execution of advertisement measurement logic from the player to a server-side system, it changes how accuracy is achieved, how trust is established, and where limitations may arise.

7.1 Accuracy of derived measurement

In CGAM, measurement accuracy is a function of both the fidelity of the telemetry reported by the client and the correctness of the server-side interpretation of that data. Unlike client-side SDKs, which can trigger tracking events at the exact moment a playback milestone is reached, CGAM relies on periodic and event-driven telemetry to reconstruct playback behaviour.

The reporting interval of telemetry is therefore a key parameter. A shorter interval provides higher temporal resolution, enabling more precise detection of quartile boundaries and playback transitions, at the cost of increased network overhead and server-side computing resources. Conversely, longer intervals reduce overhead but introduce coarser granularity, requiring the server to interpolate between reported positions.

The use of a standardised playhead time definition, as specified in CTA-5004-B, is fundamental to maintaining accuracy. Because the pt field is expressed in a consistent time base across both VOD and live playback, the telemetry endpoint can reliably map reported positions to advertisement timelines and determine when measurement thresholds are crossed. This consistency enables accurate derivation of quartile and completion events, even in the presence of buffering, pauses, or other playback state changes.

7.2 Trust and verification

A key consideration in any measurement model is the degree to which reported events can be trusted by participants in the advertising ecosystem. In traditional CSAI implementations, trust is derived from the direct execution of tracking URLs within the client, some-

times supplemented by third-party verification scripts. CGAM replaces this with a model in which measurement events are generated server-side based on client-reported telemetry.

This shift in trust boundary also creates opportunities for stronger server-to-server verification models. Rather than relying exclusively on client-executed tracking requests, CGAM enables measurement interactions to occur between infrastructure components whose identities can be independently established. The digital advertising industry has already adopted comparable trust frameworks through initiatives such as ads.txt, IAB Tech Lab (8), and sellers.json, IAB Tech Lab (9), which use publicly verifiable domain-associated metadata to establish authorised relationships between advertising systems. A similar approach could potentially be applied to advertisement measurement workflows, allowing telemetry and derived measurement events to be exchanged between trusted server systems with stronger guarantees of provenance and authenticity.

At the same time, CGAM can strengthen certain aspects of trust. Because tracking requests are executed server-side, the system can ensure that measurement logic is applied consistently and in accordance with specification requirements such as VAST 4.2 (2). It also reduces exposure to client-side interference, including the suppression of tracking requests, which can lead to under-reporting in traditional models.

7.3 Limitations and edge cases

Despite its advantages, CGAM has inherent limitations that must be considered. The model depends on the availability and continuity of telemetry from the client. Loss of connectivity, delayed reporting, or dropped events can introduce gaps in the reconstructed playback timeline, potentially affecting measurement accuracy. While server-side logic can mitigate some of these issues – for example, by interpolating between known states – there are limits to what can be inferred in the absence of reliable input.

Live streaming scenarios introduce further complexity, particularly in the presence of latency, drift, or variable playback speeds. Aligning telemetry with advertisement boundaries requires accurate synchronisation between the client's reported playhead position and the server's understanding of the media timeline. The standardised time base defined in CMCDv2 (4) helps address this, but implementation care is still required to handle edge cases such as clock skew or discontinuities in the live window.

Finally, while CGAM reduces the need for complex client-side SDKs, it does not eliminate the requirement for client instrumentation entirely. Players must still be capable of emitting accurate and timely telemetry, including user interaction events. In environments where even this level of instrumentation is constrained, measurement capabilities may still be limited.

8. CONCLUSION

The evolution of streaming advertisement insertion mechanisms has exposed a growing separation between the standardisation of advertisement delivery and the standardisation of advertisement measurement. While client-side insertion models historically combined both responsibilities within the playback environment, newer server-centric approaches such as SSAI and SGAI define mechanisms for advertisement insertion without defining interoperable approaches to measurement. As a result, accurate and trustworthy advertisement measurement continues to depend on fragmented, platform-specific implementations.

This paper introduced CGAM as a server-centric measurement model in which the player reports primitive playback telemetry while advertisement measurement logic is executed server-side. By separating playback observability from measurement interpretation, CGAM reduces the dependency on complex client-side measurement implementations while preserving compatibility with established measurement frameworks such as VAST 4.2 (2).

The paper further demonstrated how version 2 of the CMCD provides a standards-based mechanism through which CGAM can be implemented across heterogeneous playback environments. In particular, CMCDv2 Event Mode and its standardised playhead time semantics provide the telemetry foundation required to reconstruct playback progression and derive advertisement measurement events consistently across both VOD and live streaming workflows.

By leveraging CMCDv2 as an interoperable telemetry layer, CGAM helps bridge the standards gap that currently exists between server-centric advertisement insertion and advertisement measurement. This creates a pathway toward more consistent, protocol-agnostic, and device-agnostic measurement architectures that can operate across increasingly fragmented streaming ecosystems while reducing reliance on bespoke client-side integrations.

REFERENCES

1. IAB Tech Lab, 2012. Video Ad Serving Template (VAST) Version 3.0. Available at: https://iabtechlab.com/wp-content/uploads/2016/04/VASTv3_0.pdf

2. IAB Tech Lab, 2019. Video Ad Serving Template (VAST) Version 4.2. Available at: https://iabtechlab.com/wp-content/uploads/2019/06/VAST_4.2_final_june26.pdf

3. Yospace, 2026. ITV Selects Yospace for Advanced Ad Measurement and Monetisation on Freely. Available at: https://www.yospace.com/news/itv-selects-yospace-for-advanced-ad-measurement-and-monetisation-on-freely

4. CTA WAVE, 2026. Common Media Client Data (CMCD) Specification CTA-5004-B. Available at: https://cta-wave.github.io/Resources/common-media-client-data--cta-5004-b.html

5. Apple Developer, 2021. Getting Started with HLS Interstitials. Available at: https://developer.apple.com/streaming/GettingStartedWithHLSInterstitials.pdf

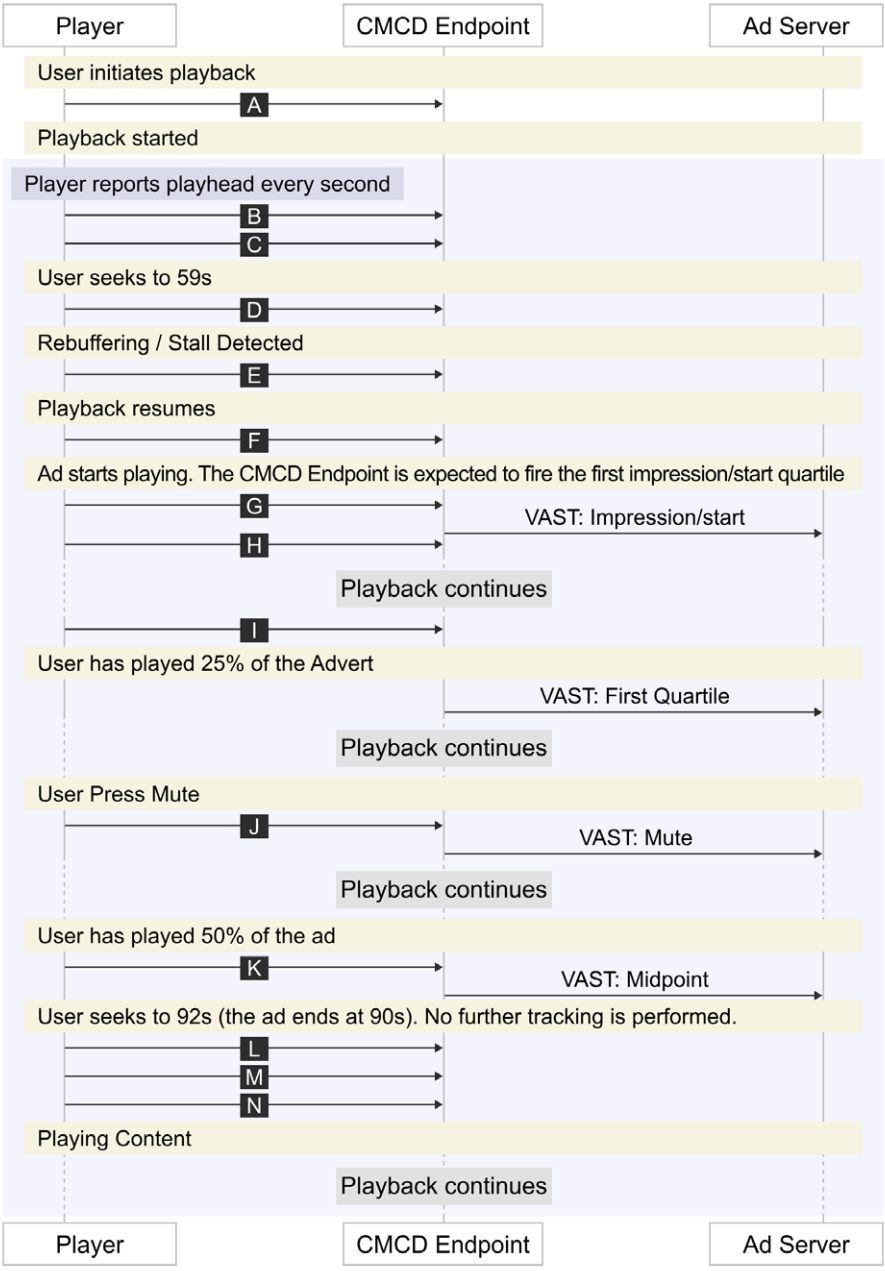
6. Giladi, A., Levy, N. and Griebel, T., 2024. Alternative MPD Support for MPEG-DASH Streaming. Available at: https://dl.acm.org/doi/pdf/10.1145/3789239.3793274

7. CTA WAVE, 2026. Common Media Client Data (CMCD) Specification CTA-5004-B. Available at: https://cta-wave.github.io/Resources/common-media-client-data--cta-5004-b.html

8. IAB Tech Lab, 2022. Ads.txt Specification Version 1.1. Available at: https://iabtechlab.com/wp-content/uploads/2022/04/Ads.txt-1.1.pdf

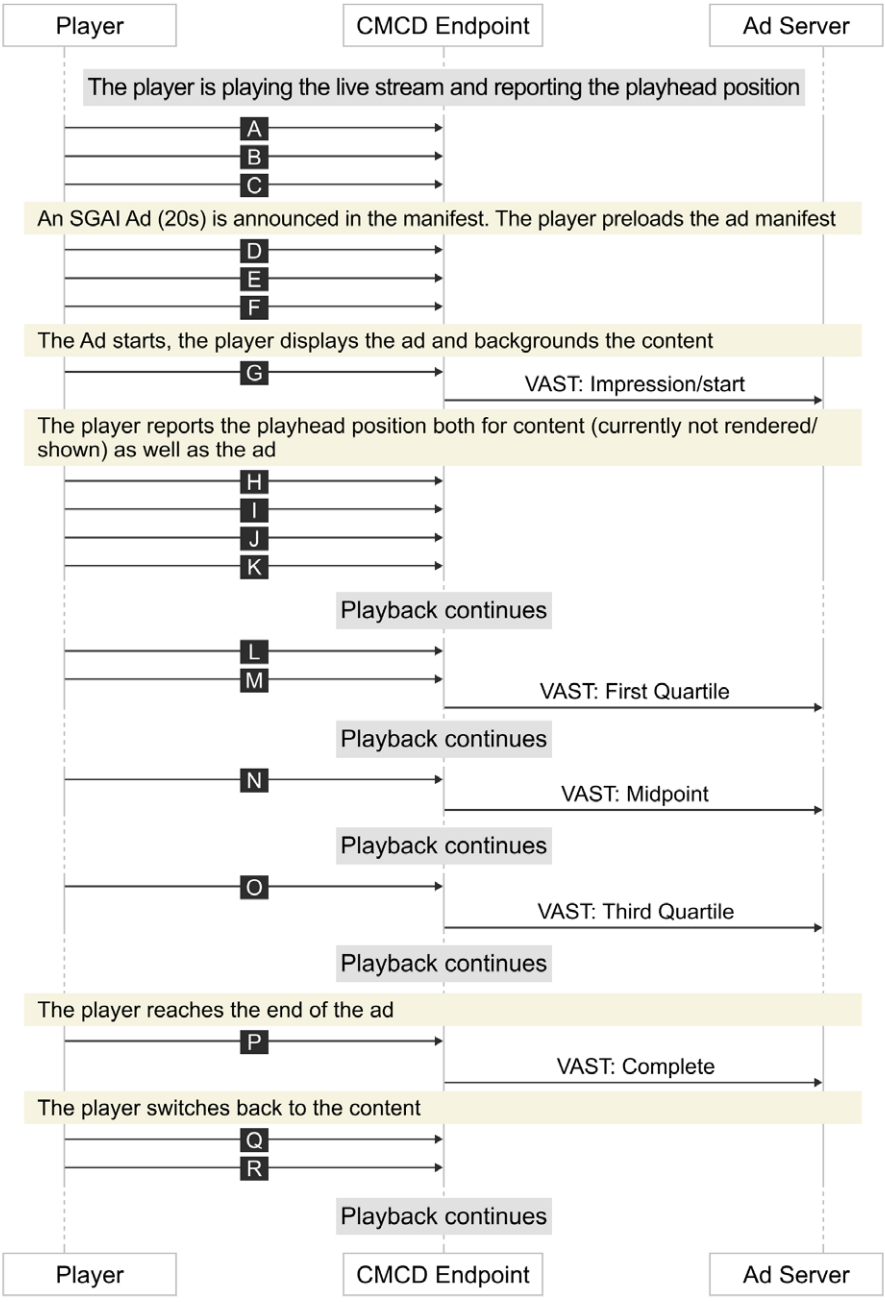
9. IAB Tech Lab, 2019. sellers.json Specification. Available at: https://iabtechlab.com/wp-content/uploads/2019/07/Sellers.json_Final.pdf

Figure 2 - Playback workflow example for VOD



A	v=2,sid="s-01",st=v,sta=s,pt=0,ts=1764752400000,e=ps
B	v=2,sid="s-01",st=v,sta=p,pt=0,ts=1764752401000,e=ps
C	v=2,sid="s-01",st=v,sta=p,pt=1000,ts=1764752402000,e=t
D	v=2,sid="s-01",st=v,sta=k,pt=59000,ts=1764752402500,e=ps
E	v=2,sid="s-01",bs,st=v,sta=r,pt=59000,ts=1764752403000,e=ps
F	v=2,sid="s-01",st=v,sta=p,pt=59000,ts=1764752404000,e=ps
G	v=2,sid="s-01",st=v,sta=p,pt=60000,ts=1764752405000,e=t
H	v=2,sid="s-01",st=v,sta=p,pt=61000,ts=1764752406000,e=t
I	v=2,sid="s-01",st=v,sta=p,pt=67500,ts=1764752413000,e=t
J	v=2,sid="s-01",st=v,sta=p,pt=70500,ts=1764752415500,e=m
K	v=2,sid="s-01",st=v,sta=p,pt=75000,ts=1764752420000,e=t
L	v=2,sid="s-01",st=v,sta=k,pt=92000,ts=1764752423500,e=ps
M	v=2,sid="s-01",bs,st=v,sta=r,pt=92000,ts=1764752424000,e=t
N	v=2,sid="s-01",st=v,sta=p,pt=92000,ts=1764752424200,e=ps

Figure 3 - Dual Decoder Workflow Example



A	v=2,sid="s-sgai-01",cid="content-1", st=l,sta=p,pt=1778060998000,ts=1778061011000,e=t
B	v=2,sid="s-sgai-01",cid="content-1", st=l,sta=p,pt=1778060999000,ts=1778061012000,e=t
C	v=2,sid="s-sgai-01",cid="content-1", st=l,sta=p,pt=1778061000000,ts=1778061013000,e=t
D	v=2,sid="s-sgai-01",cid="interstitial-ad-1", nr,st=v,sta=d,ts=1778061013500,e=ps
E	v=2,sid="s-sgai-01",cid="content-1", st=l,sta=p,pt=1778061001000,ts=1778061014000,e=t
F	v=2,sid="s-sgai-01",cid="content-1", st=l,sta=p,pt=1778061002000,ts=1778061015000,e=t
G	v=2,sid="s-sgai-01",cid="interstitial-ad-1", st=v,sta=p,pt=0,ts=1778061015500,e=ps
H	v=2,sid="s-sgai-01",cid="content-1", nr,st=l,sta=p,pt=1778061003000,ts=1778061016000,e=t
I	v=2,sid="s-sgai-01",cid="interstitial-ad-1", st=v,sta=p,pt=1,ts=1778061016500,e=t
J	v=2,sid="s-sgai-01",cid="content-1", nr,st=l,sta=p,pt=1778061004000,ts=1778061017000,e=t
K	v=2,sid="s-sgai-01",cid="interstitial-ad-1", st=v,sta=p,pt=2,ts=1778061017500,e=t
L	v=2,sid="s-sgai-01",cid="content-1", nr,st=l,sta=p,pt=1778061007000,ts=1778061020000,e=t
M	v=2,sid="s-sgai-01",cid="interstitial-ad-1", st=v,sta=p,pt=5,ts=1778061020500,e=t
N	v=2,sid="s-sgai-01",cid="interstitial-ad-1", st=v,sta=p,pt=10,ts=1778061025500,e=t
O	v=2,sid="s-sgai-01",cid="interstitial-ad-1", st=v,sta=p,pt=15,ts=1778061030500,e=t
P	v=2,sid="s-sgai-01",cid="interstitial-ad-1", st=v,sta=e,pt=20,ts=1778061035500,e=ps
Q	v=2,sid="s-sgai-01",cid="content-1", st=l,sta=p,pt=1778061023000,ts=1778061036000,e=t
R	v=2,sid="s-sgai-01",cid="content-1", st=l,sta=p,pt=1778061024000,ts=1778061037000,e=t

ABOUT YOSPACE

Yospace has been at the forefront of the digital video revolution for 25 years. From the early days of mobile delivery, to the mass consumption and monetisation of premium streaming, we have a history of driving business success for the world's biggest telcos, broadcasters, and media companies.

Our pioneering dynamic ad insertion technology has helped monetise seven Olympic Games, five FIFA World Cups, and forty Grand Slam tennis tournaments. We have over 2,000 live channels under management and stitch 10 billion advertisements per month.

Customers include DIRECTV, FOX, and Tubi in the USA; Allente, Channel 4, ITV, RTL, Telenor, Telia and TV4 in Europe; ABEMA, Nine, and Seven West Media in APAC.

sales@yospace.com

YOSPACE