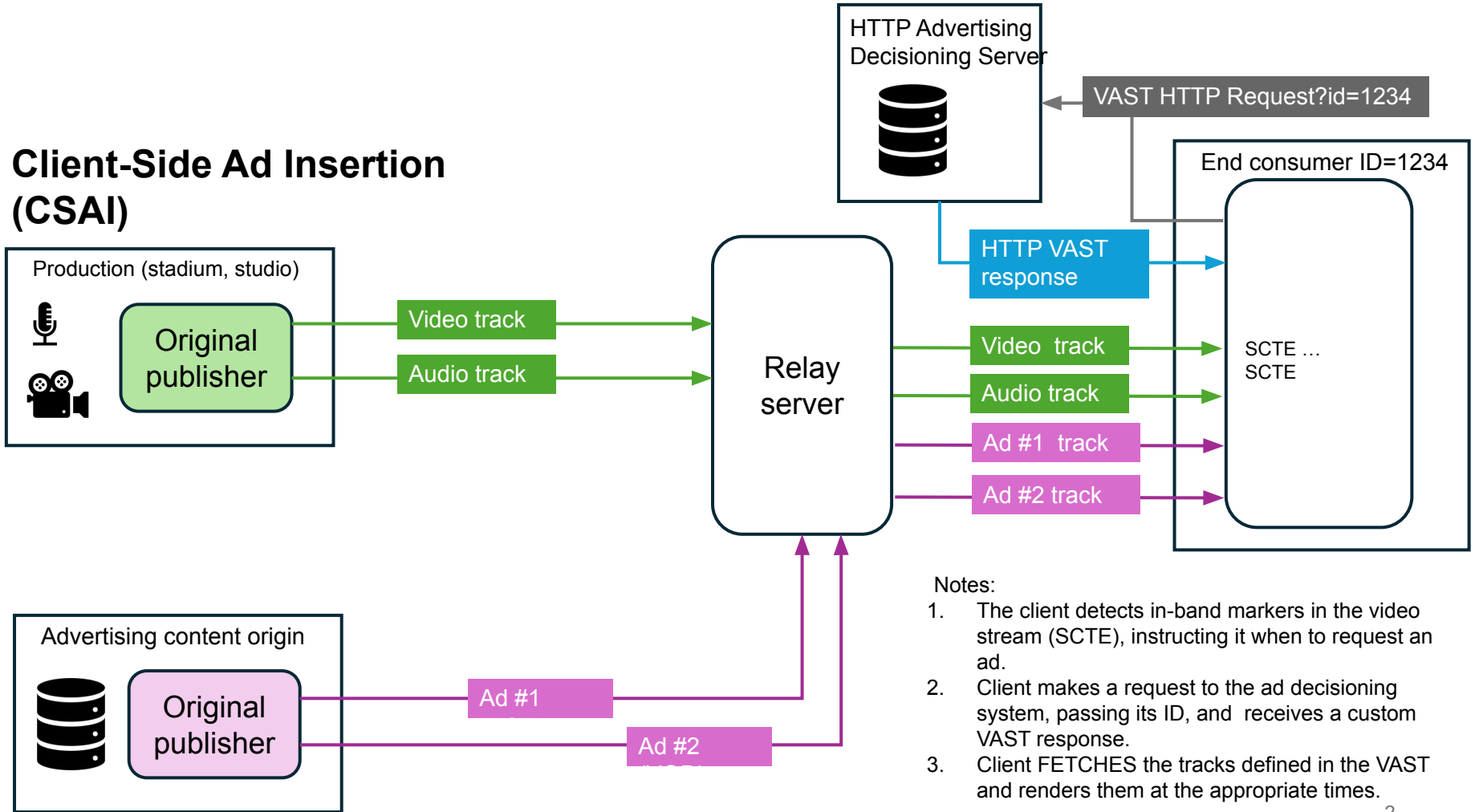


# Advertising with MOQT

Will Law

IETF moq workgroup, Boulder interim, February 2026

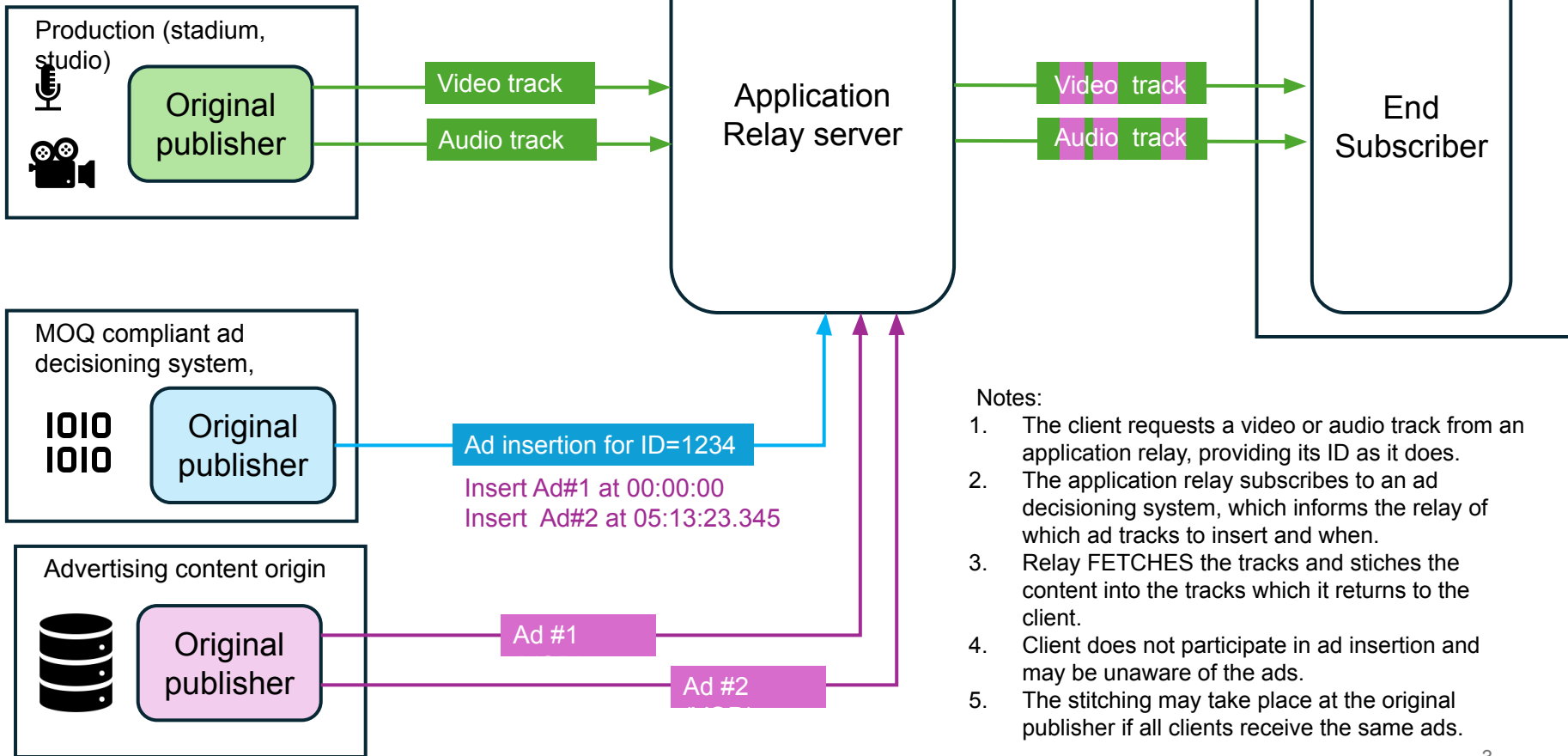
# Client-Side Ad Insertion (CSAI)



## Notes:

1. The client detects in-band markers in the video stream (SCTE), instructing it when to request an ad.
2. Client makes a request to the ad decisioning system, passing its ID, and receives a custom VAST response.
3. Client FETCHES the tracks defined in the VAST and renders them at the appropriate times.

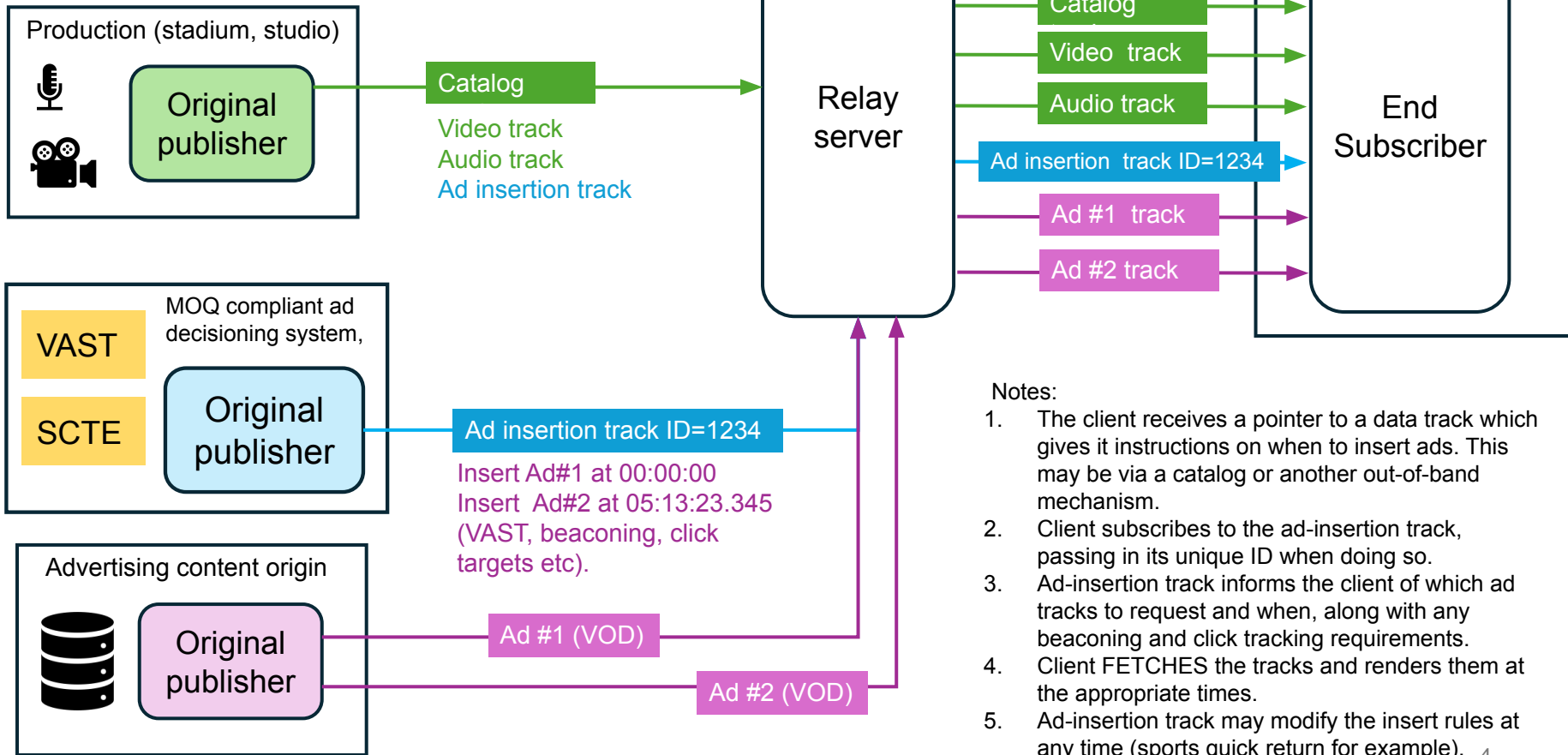
# Server-Side Ad Insertion (SSAI) (traditional)



## Notes:

1. The client requests a video or audio track from an application relay, providing its ID as it does.
2. The application relay subscribes to an ad decisioning system, which informs the relay of which ad tracks to insert and when.
3. Relay FETCHES the tracks and stitches the content into the tracks which it returns to the client.
4. Client does not participate in ad insertion and may be unaware of the ads.
5. The stitching may take place at the original publisher if all clients receive the same ads.

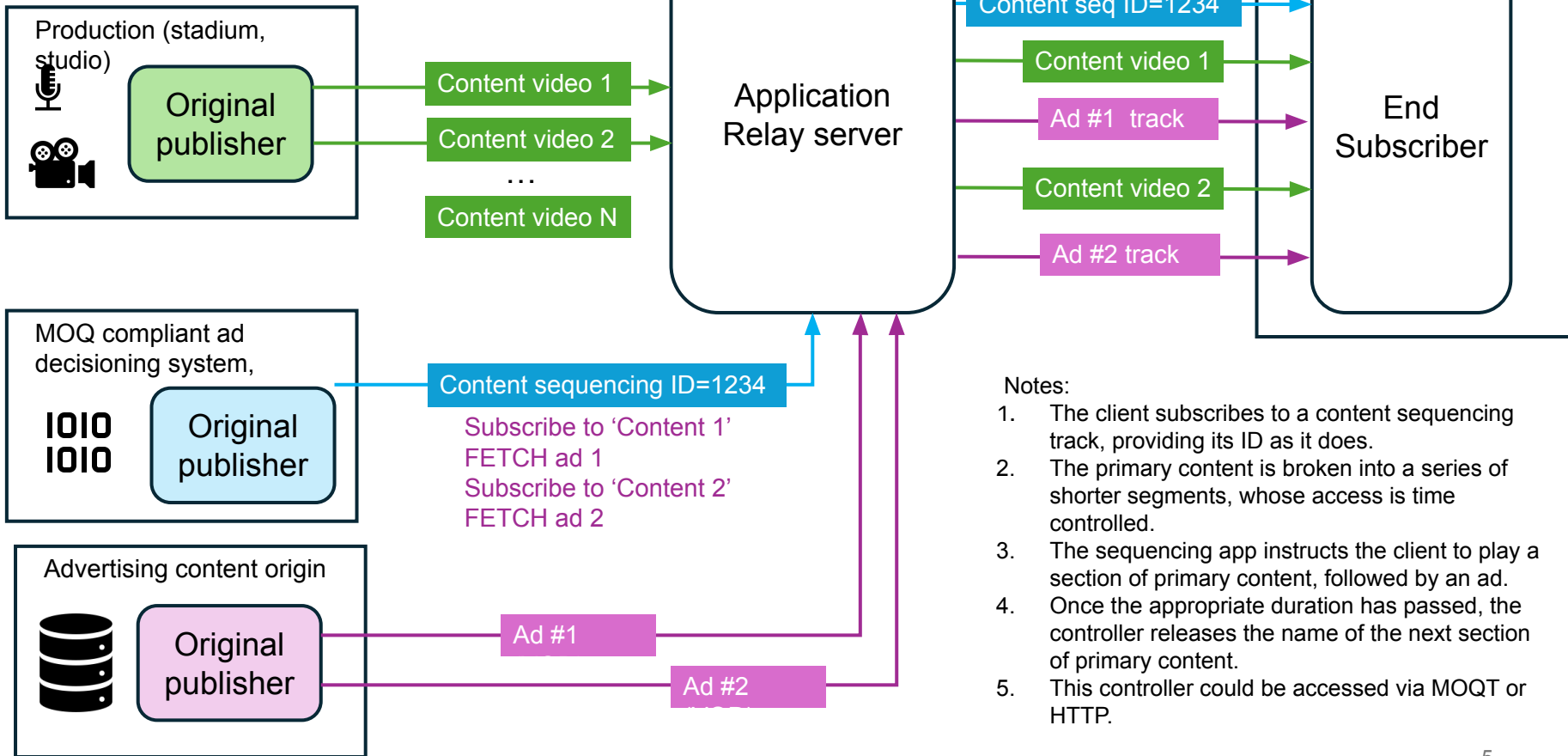
# Server-Guided Ad Insertion (SGAI)



## Notes:

1. The client receives a pointer to a data track which gives it instructions on when to insert ads. This may be via a catalog or another out-of-band mechanism.
2. Client subscribes to the ad-insertion track, passing in its unique ID when doing so.
3. Ad-insertion track informs the client of which ad tracks to request and when, along with any beaconing and click tracking requirements.
4. Client FETCHES the tracks and renders them at the appropriate times.
5. Ad-insertion track may modify the insert rules at any time (sports quick return for example).

# Server-Side Ad Insertion (SSAI) (Luke sequence)



# Ad insertion options with MOQT

Feature	Client-Side Ad Insertion (CSAI)	Server-Guided Ad Insertion (SGAI)	Server-Side Ad Insertion (SSAI)
Where Ad is Stitched	Client (Video Player)	Client (Video Player)	Relay (Ad Stitcher)
Stream Delivery	Two separate tracks : Content track (containing ad markers) and an Ad Track	Two separate tracks referenced in a catalog (or equivalent)	Single, continuous track (content + ad) or a sequence of primary tracks separated by ads.
Ad Blocker Resistance	Low. Easier to detect and block the separate ad FETCHes	Medium/High. The ad is called via a dedicated playlist entry, making it harder for simple blockers to detect.	High for stitched. The ad is physically integrated into the content stream, making it difficult to block without blocking the entire video. Medium for sequenced.
Playback Seamlessness	Medium. Transitions may involve a visible pause, a black screen, or a "hiccup" as the player switches between content and ad players.	High. The server maintains timing control, and the client player is designed to seamlessly switch between the main content track and the ad track referenced in the manifest.	High for stitched. The experience is seamless because the content and ads are delivered as one track. Medium for sequenced, since the client must make transitions.
Targeting/Personalization	High. The player sends real-time user data directly to the ad server.	High. The client still requests the specific ad at the last minute, allowing for real-time targeting.	High if targeting is done at the edge relay, low if targeting is done at the original publisher.

# What new features do we need in MOQT to support these modes?

**Client-side ad insertion** – none

**Server-guided ad insertion** – none

**Server-side ad insertion**

- Add the notion of an “Application relay” – one that does modify track content based upon external configuration
- Add the capability to configure this relay to substitute Group N of track A with the same group of track B. For example (relay API):

**SUBSTITUTE (FromTrack, ToTrack, Filter, Token)**

Thank you for your time.