

# SCTE-35: In-Band Event Signaling for Live OTT

---

**Alex Zambelli**

Sr. Product Manager  
Hulu

# What Is SCTE?

- Society of Cable Telecommunications Engineers
- SCTE standards
  - Data communications
  - Energy management
  - Equipment & cabling
  - Network operations
  - **Digital video**
    - IP video transport
    - Streaming (DASH)
    - Ad insertion
    - Audio levels & synchronization

# What Is SCTE-35?

- Official name: “Digital Program Insertion Cueing Message for Cable”
  - More intuitive name would be “In-Band Event Signaling for Live Video”
- Standard for signaling splice points, events and content segment boundaries within a live video stream for purposes of
  - Ad insertion
  - Alternate content replacement
  - EPG (live schedule & metadata) synchronization
  - Content identification
- Companion specifications
  - SCTE-67: Recommended Practices for SCTE-35
  - SCTE-224: Channel schedule, metadata and content rights

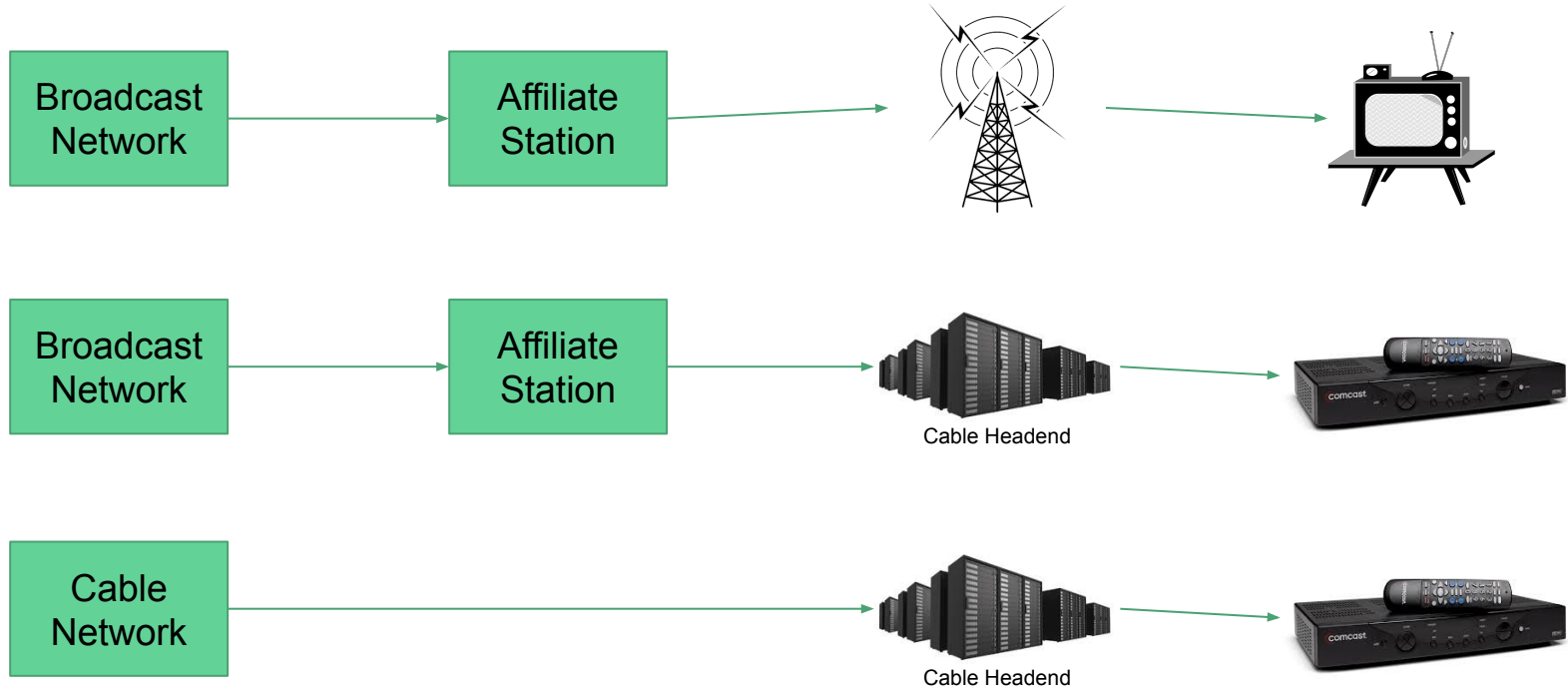
# Terminology Matters

- They're not "SCTE markers"
  - SCTE is an organization, not a standard
  - There are many SCTE standards
- They're not just "ad markers"
  - Ad insertion was the original use case, but the standard has evolved far beyond it
  - SCTE-35 messages can be used for content identification, content replacement automation, blackout enforcement, etc - even clock time synchronization!

# Original Use Case: Local Ad Insertion

- Most U.S. network and cable programming is ad sponsored
- Network broadcasters sell national ad time, while MVPDs or local network affiliate stations sell local ad time
- MVPD and/or local station splice in local ads when network partner indicates a placement opportunity (avail)
- Average 16 minutes of advertising per hour of network TV
  - 12 minutes of national (provider) ads
  - 4 minutes of local (distributor) ads
- When re-purposed for OTT delivery, both local and national ad time may be resold depending on contractual rules, DVR, VOD, etc.

# Traditional TV Programming Distribution Models



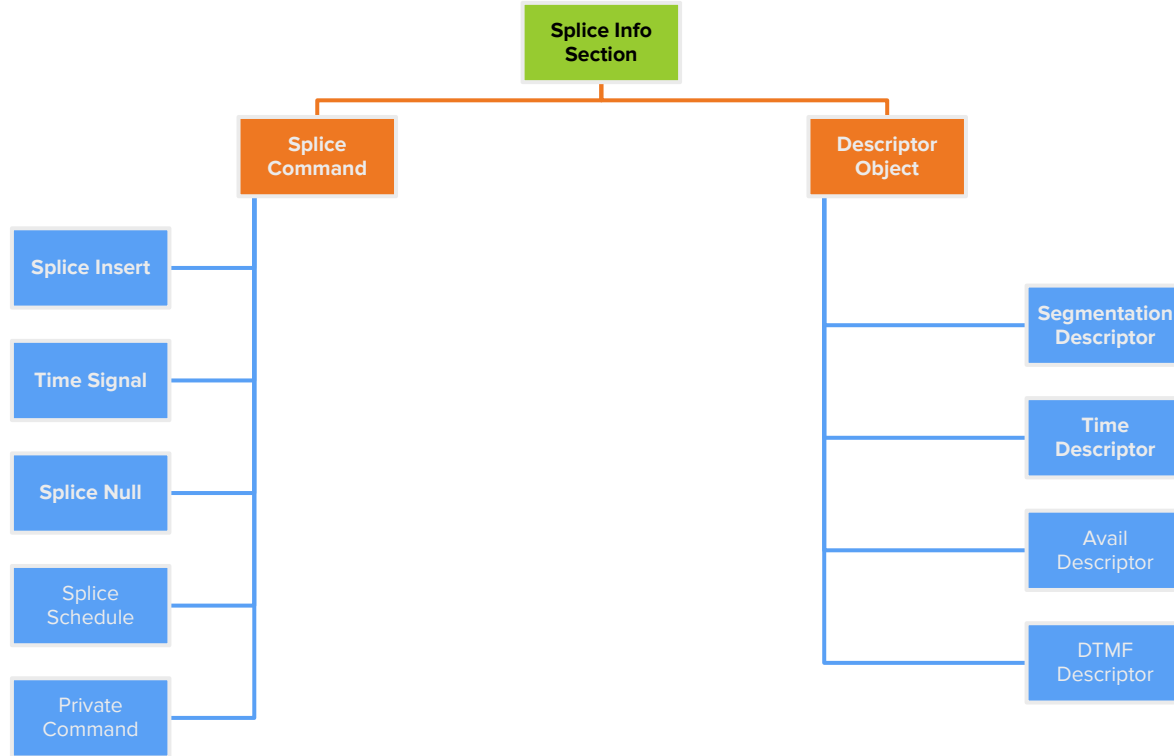
# SCTE-35 Message Formats

- Binary
  - Original message format
  - Sparse messages carried in MPEG-2 TS (on dedicated PID)
    - 30-50 bytes per message
  - Not human readable
  - Non-trivial to parse and decode
  - Converted to Base64 or hex strings when signaled in DASH or HLS
- XML
  - XML schema added to standard in 2013
  - Human readable
  - Still fairly uncommon, even in HLS/DASH manifests, mostly due to size





# SCTE-35 Commands and Descriptors



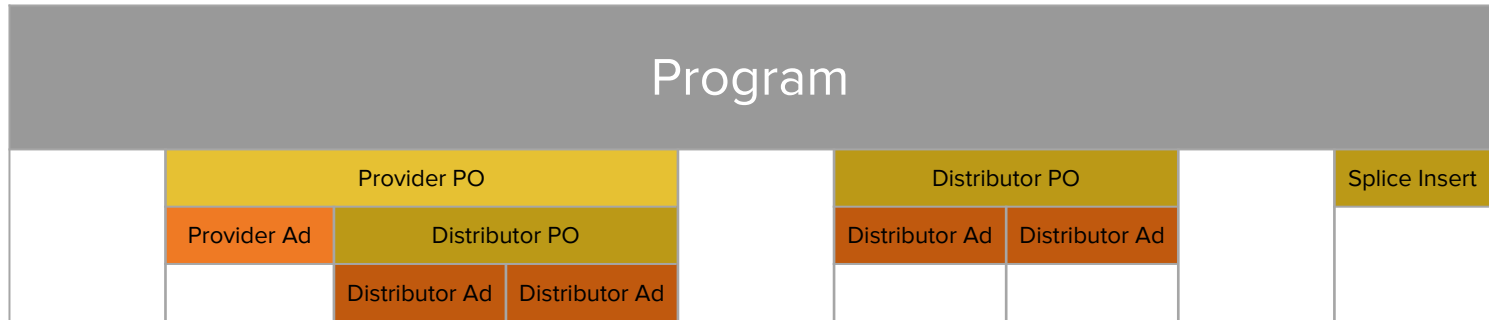
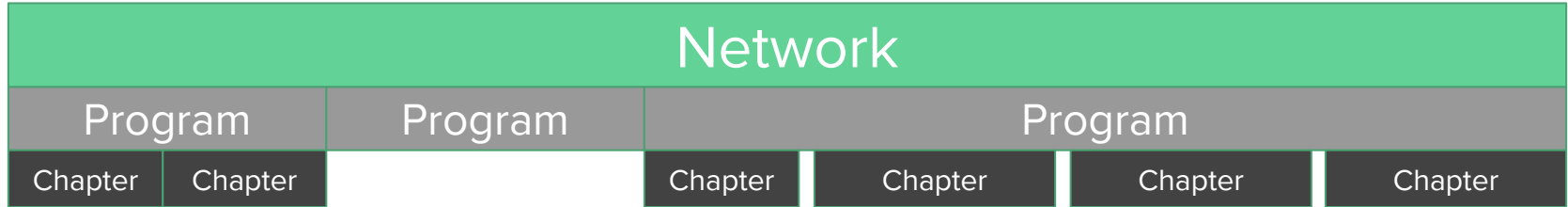
# Common Commands

- **Splice\_insert**
  - Historically used to signal local (distributor) ad placement opportunities, aka avails
  - Typically only includes splice timestamp and (optionally) planned duration
  - Limited in syntax but still heavily used for backwards compatibility with legacy systems
- **Time\_signal**
  - Generic command for signaling timestamp-synchronized data
  - Useful only in conjunction with **segmentation\_descriptor** object
  - Used in nearly all use cases that involve more than just local avail signaling

# Segmentation Descriptors

- Used to describe events or segments within a linear timeline (not to be confused with HLS/DASH segments)
  - Networks
  - Programs
  - Chapters
  - Breaks, placement opportunities, advertisements
  - Unscheduled events
- Segment boundaries marked with Start/End message pairs
- Segments identified by UPID (Unique Program ID)
  - 15 standardized UPID types, including extensible private
  - Common UPID types: EIDR, TMS ID, ADI, ISAN, Ad-ID

# Segment Topology



# Signaling SCTE-35 in DASH

- [SCTE-214 Part 1](#) standardizes MPD constraints for SCTE use cases
- SCTE-67 suggests two possible SCTE-35 message signaling methods
  - Align MPD periods with SCTE-35 segments
    - Simple but assumes clean segment delineation with no gaps or overlaps  
For example:  
`Program.Chapter → Provider Ad → Distributor PO → Program.Chapter → Program.Chapter...`
  - Signal events in the MPD or in Media Segments as Event Message Box ('emsg') structures
    - `<EventStream>` with `@schemeIdUri` set to either `"urn:scte:scte35:2013:xml"` or `"urn:scte:scte35:2014:xml+bin"`

# SCTE-35 in DASH: Event Stream Example

## MPD Event

```
<Period>
...
<EventStream schemeIdUri="urn:scte:scte35:2013:xml">
  <Event timescale="90000"
    presentationTime="54054000"
    duration="5400000" id="1">
    <scte35:SpliceInfoSection scte35:ptsAdjustment="0"
      scte35:tier="22">
      <scte35:SpliceInsert
        scte35:spliceEventId="111"
        scte35:spliceEventCancelIndicator="false"
        scte35:outOfNetworkIndicator="true"
        scte35:uniqueProgramId="65535"
        scte35:availNum="1"
        scte35:availsExpected="2"
        scte35:spliceImmediateFlag="false">
        <scte35:Program>
          <scte35:SpliceTime
            scte35:ptsTime="122342"/>
        </scte35:Program>
        <scte35:BreakDuration
          scte35:autoReturn="false"
          scte35:duration="5400000"/>
        </scte35:SpliceInsert>
        <scte35:AvailDescriptor
          scte35:providerAvailId="332"/>
        </scte35:SpliceInfoSection>
      </Event>
    </EventStream>
  ...
</Period>
```

## Inband Event (`emsg`)

scheme_id_uri="urn:scte:scte35:2013:bin"
value=1001
timescale=90000
presentation_time_delta=540000
duration=5400000
id=0
0xFC 0x30 0x08 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x10 0x00 0x06 0x7F 0x23 0x45 0x67 0x89 0x00 0x10 0x02 0x00 0x43 0x55 0x45 0x49 0x40 0x00 0x00 0x00 0x7F 0x9C 0x00 0x00 0x00 0x00

Binary SCTE 35 cue message

# Signaling SCTE-35 in HLS

- Binary messages carried in tags in media (variant) playlists
- SCTE preferred syntax: **#EXT-X-SCTE35**
  - Required attribute: **CUE**
    - Value is SCTE-35 binary message as Base64 string
  - Optional attributes: ID, DURATION, ELAPSED, TIME, TYPE, UPID, BLACKOUT, CUE-OUT, CUE-IN
- Apple preferred: **#EXT-X-DATERANGE**
  - Generic HLS method for signaling timeline ranges
  - Required attributes for SCTE-35 signaling: SCTE35-CMD, SCTE35-OUT, SCTE35-IN
    - Value is SCTE-35 binary message as hex string
- Both do the job but...
  - For timing, EXT-X-SCTE35 relies on position in playlist; EXT-X-DATERANGE relies on UTC
  - EXT-X-DATERANGE has native AVPlayer API support (no SCTE-35 awareness though)

# SCTE-35 in HLS: Media Playlist Example

```
#EXT-X-TARGETDURATION:4
#EXT-X-MEDIA-SEQUENCE:63260
#EXT-X-PROGRAM-DATE-TIME:2017-09-22T17:13:46.816+00:00
#EXT-X-DATERANGE:ID="1207959552",START-DATE="2017-09-22T17:00:14+00:00",PLANNED-DURATION=32399.998,SCTE35-OUT=0xFC302F00000000000000000700506FE1D26AD4F001902174355454948
#EXTINF:4.004,
VIDEO_1_5428000/1505849102_set_31/TBSW_VIDEO_1_5428000_63326.ts
.....
#EXT-X-DATERANGE:ID="1207959602",START-DATE="2017-09-22T17:14:34+00:00",PLANNED-DURATION=196.066,SCTE35-OUT=0xFC303400000000000000000800506FE21C4E367001E021C435545494800
#EXT-X-SCTE35:CUE="/DA0AAAAAAAAACABQb+IcTjZwAeAhxDVUVVJSAAMn/PAENQbYICAAAAAAsArNAIC7B65SQ==",ID="1207959602"
#EXTINF:1.602,
VIDEO_1_5428000/1505849102_set_31/TBSW_VIDEO_1_5428000_63338.ts
#EXTINF:4.004,
VIDEO_1_5428000/1505849102_set_31/TBSW_VIDEO_1_5428000_63339.ts
.....
#EXTINF:4.004,
VIDEO_1_5428000/1505849102_set_31/TBSW_VIDEO_1_5428000_63356.ts
#EXT-X-DATERANGE:ID="1342177264",START-DATE="2017-09-22T17:15:48+00:00",PLANNED-DURATION=60.326,SCTE35-OUT=0xFC302F000000000000000001014054FFFFFFFF07FEFFE222CB8A5FE0052D8E
#EXT-X-SCTE35:CUE="/DAvAAAAAAAAAQFAVP//wF+/+Iiy4pf4AUtiwAAAAAAKAhDVUVVJAAABNRtSaDA=",ID="1342177264"
#EXTINF:3.537,
VIDEO_1_5428000/1505849102_set_31/TBSW_VIDEO_1_5428000_63357.ts
#EXTINF:4.004,
VIDEO_1_5428000/1505849102_set_31/TBSW_VIDEO_1_5428000_63358.ts
.....
#EXTINF:4.004,
VIDEO_1_5428000/1505849102_set_31/TBSW_VIDEO_1_5428000_63371.ts
#EXT-X-DATERANGE:ID="1342177264",START-DATE="2017-09-22T17:15:48+00:00",END-DATE="2017-09-22T17:16:48.112+00:00",SCTE35-IN=0xFC3020000000000000000000FFF00F054FFFFFFFF07F4FEE2
#EXT-X-SCTE35:CUE="/DAgAAAAAAAAAP/wDwVp//wF0/+In+rVAAAAAAAConIsA=",ID="1342177264"
#EXTINF:4.271,
VIDEO_1_5428000/1505849102_set_31/TBSW_VIDEO_1_5428000_63372.ts
.....
#EXTINF:4.004,
VIDEO_1_5428000/1505849102_set_31/TBSW_VIDEO_1_5428000_63386.ts
#EXT-X-DATERANGE:ID="1207959602",START-DATE="2017-09-22T17:14:34+00:00",END-DATE="2017-09-22T17:17:48.441+00:00",SCTE35-IN=0xFC302F000000000000000000FFF00506FE22D2251C00190
#EXT-X-SCTE35:CUE="/DAVAAAAAAAAAP/wQB+ItIlHAAZAhDhDVUVVJSAAMGcACAGAAAAAKwRAKzUBAScSDJ0=",ID="1207959602"
#EXTINF:4.071,
VIDEO_1_5428000/1505849102_set_31/TBSW_VIDEO_1_5428000_63387.ts
#EXTINF:4.004,
VIDEO_1_5428000/1505849102_set_31/TBSW_VIDEO_1_5428000_63388.ts
```





# SCTE-35 Challenges In OTT Applications

- Not all SCTE-35 is created equal
  - Lots of ambiguity and room for interpretation - every network does it slightly differently
  - Industry could benefit from SCTE-35 interoperability points or conformance profiles
- Non-advertising use cases often require cross-referencing content owner's CMS or EPG (e.g. SCTE-224) for content identification
- Very few SCTE-35 open-source tools, makes troubleshooting difficult
- Signaling SCTE-35 messages in HLS/DASH manifests is more complicated than simply passing through input signals
  - State machine required to keep track of S35 segment topology and open/closed segments
- SCTE-35 parsing and segment normalization logic too complex for most players

# Developer Resources

- SCTE-35 binary decoders
  - Javascript: <https://github.com/hmanikkothu/SCTE35-Parser>
  - Rust: <https://github.com/m2amedia/scte35dump>
  - Go: <https://godoc.org/github.com/Comcast/gots/scte35>
  - Python: <https://gist.github.com/use-sparingly/6517a8b94a52746af028>
  - Java: <https://github.com/nfl/scte35>