BATTERIES INCLUDED:
AD INSERTION SUPPORT IN DASH

Alex Giladi, April 2014

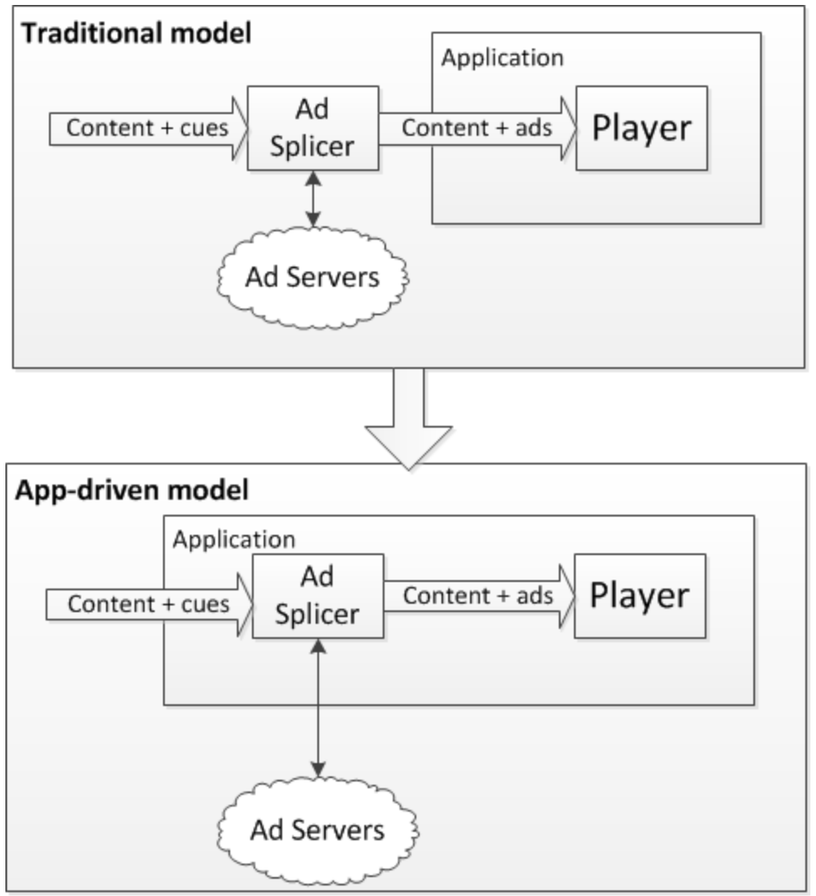
invention | collaboration | contribution

Contents

- App-driven architecture
  - Overview of workflow
  - Overview of user-defined events
- Server-driven architecture
  - Overview of enabling tools in DASH
    - Periods
    - Asset identifiers
    - XLink
    - MPD updates
- Linear workflows
  - Server-driven
  - App-driven

App-driven model: translating existing workflow into DASH

- "Ad Splicer" is a module in the client app
  - Cue messages passed to the client
  - Client-side module acts on cue message
    - Direct communication between client and ad servers
- DASH client is a module in the client app
  - Cues embedded in DASH content
  - DASH client conveys cues to a "splicer" module
- DASH events used to transport cue messages
  - For a DASH client user-defined events are "timed blobs"
    - Can be embedded within segments
    - Can be embedded in MPD (at Period level)\
    - DASH client is not expected to parse user-defined event payloads
    - Application can register a callback for certain event type(s)
  - Cue events are essential
    - DASH client w/o support for a specific cue format cannot play the content



Carrying SCTE 35 cue messages in DASH

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:xml”

value=1001

timescale=90000

presentation_time_delta=540000

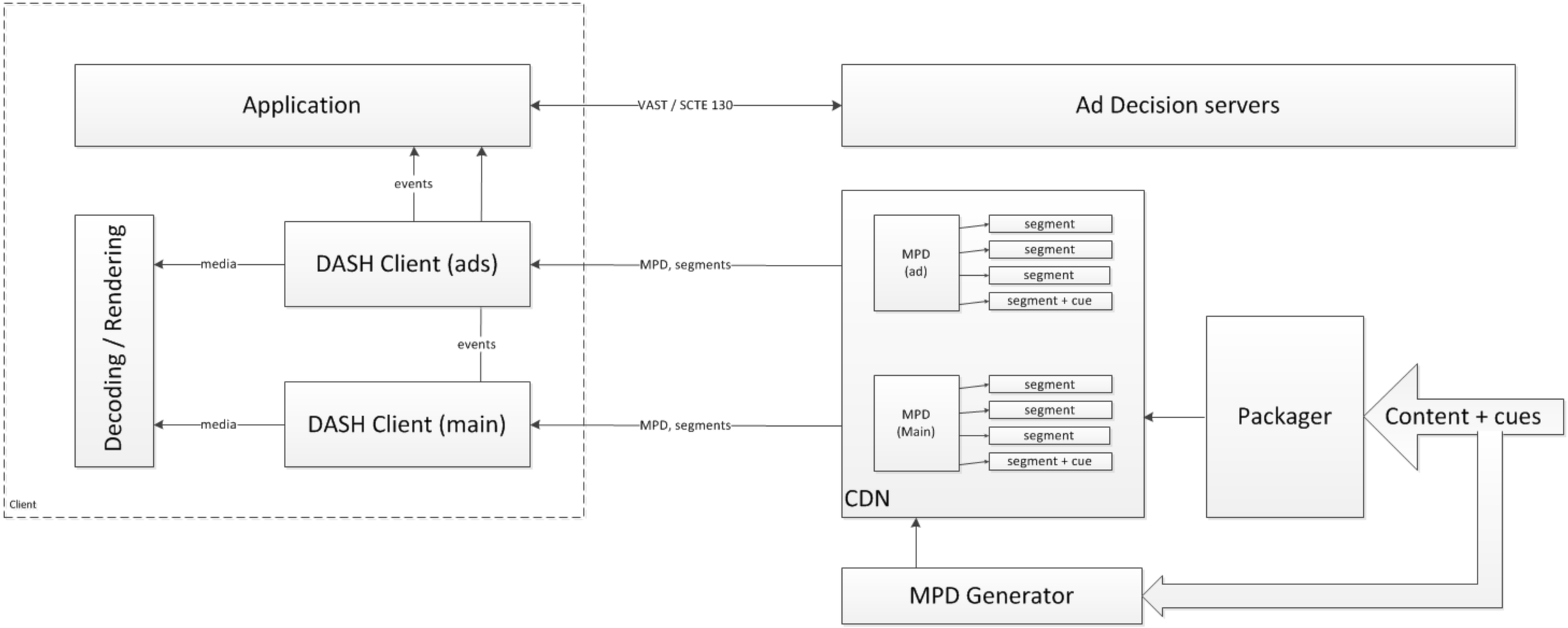
duration=5400000

id=0

message_data[]=

```
<SpliceInfoSection ptsAdjustment="0" scte35:tier="22">
   <SpliceInsert
       spliceEventId="111" spliceEventCancelIndicator="false"
       outOfNetworkIndicator="true" uniqueProgramId="65535"
       availNum="1" availsExpected="2" spliceImmediateFlag="false">
       <Program><SpliceTime ptsTime="122342"/></Program>
       <BreakDuration autoReturn="false" duration="5400000"/>
   </SpliceInsert>
   <AvailDescriptor scte35:providerAvailId="332"/>
</SpliceInfoSection>
```

Putting it all together: App-driven architecture

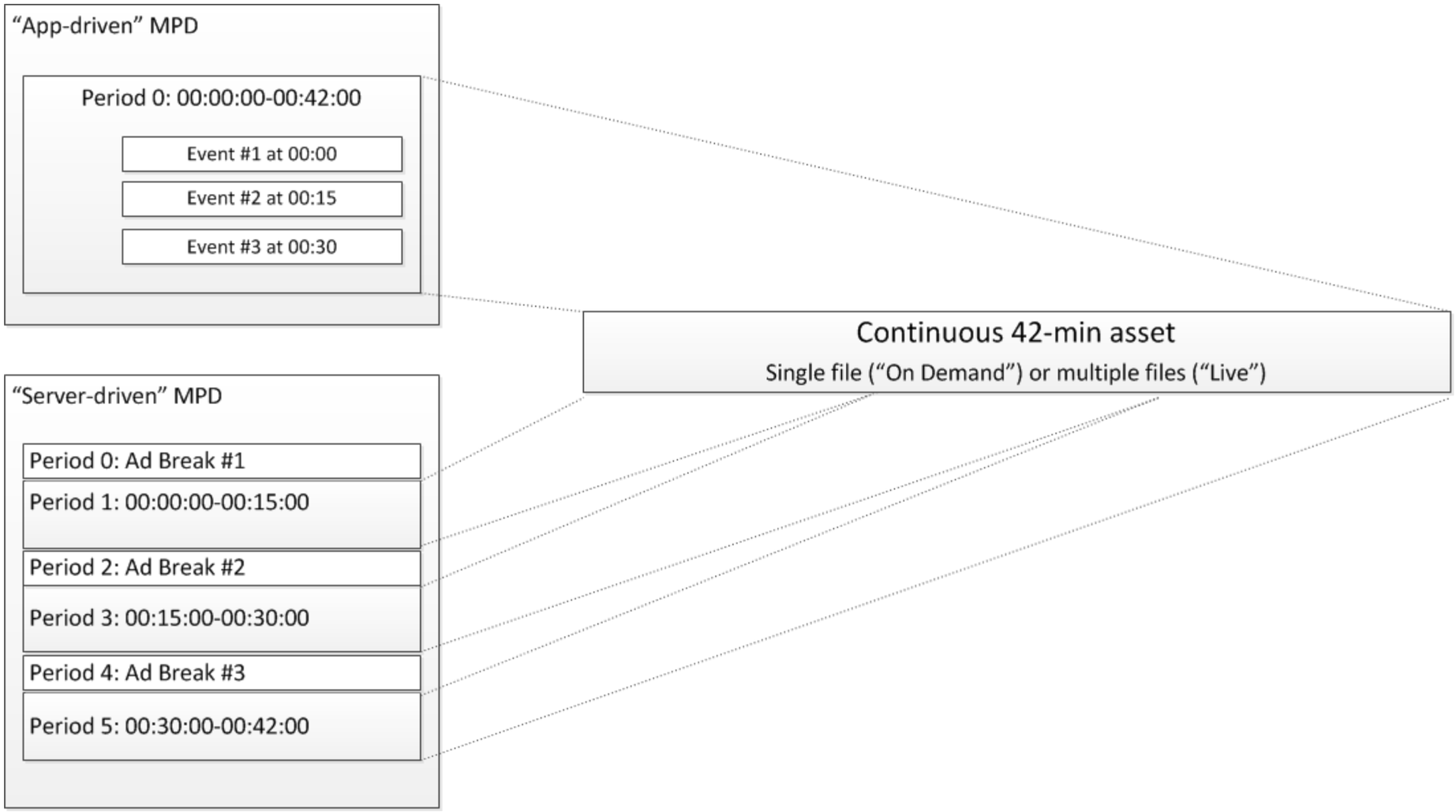


INTERDIGITAL

Server-driven model: single-client native ad insertion

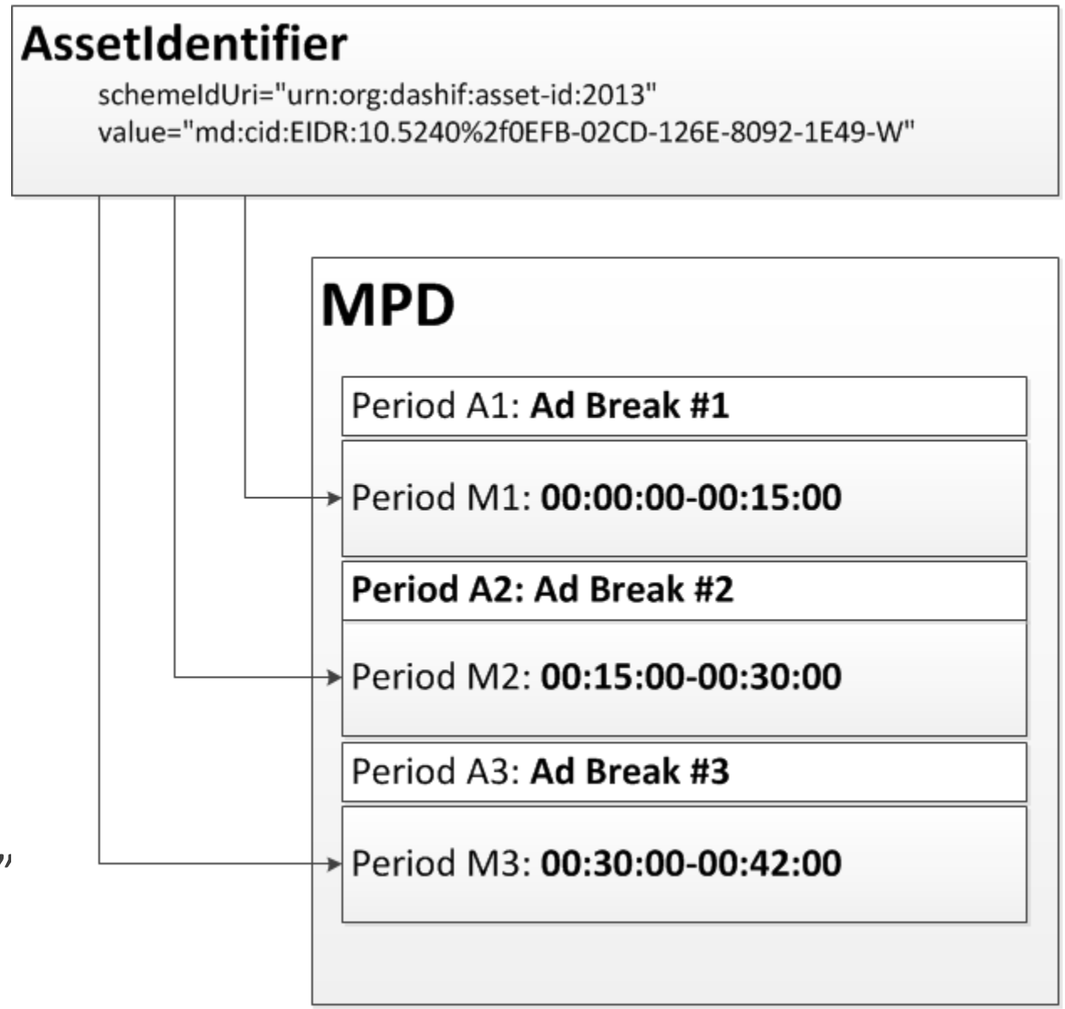
- **App-driven**
  - Non-DASH logic and modules needed
  - Multiple DASH clients are needed
- **Server-driven**
  - Use built-in DASH tools
    - Ad is a period
    - Remote period is a cue
    - MPD updates and XLink for just-in-time functionality
  - Client plays out the MPD it is given
    - Possibly requests ads from an ad proxy using XLink.
  - Ad decision logic solely on server side
    - Ad proxy communicates with ad decision servers on behalf of the client
  - Additional non-DASH logic is possible
    - This logic may not affect client functionality – e.g. tracking

Expressing cues: periods vs user-defined events



Multi-period MPD: identifying assets

- Assets split into multiple periods
  - Periods from the same asset may be separated by inserted content
  - Keeping track of asset time is often needed
    - Progress bar, bookmarks, trick modes, etc.
- Period continuity needs signaling
  - Identical asset identifiers across periods
  - M1, M2, and M3 have same id, thus are continuous
  - Lack of identifiers – no expectation of continuity
- AssetIdentifier is extensible
  - DASH-IF interop point uses MovieLabs URNs
  - It is possible to explicitly indicate “main” vs “inserted”

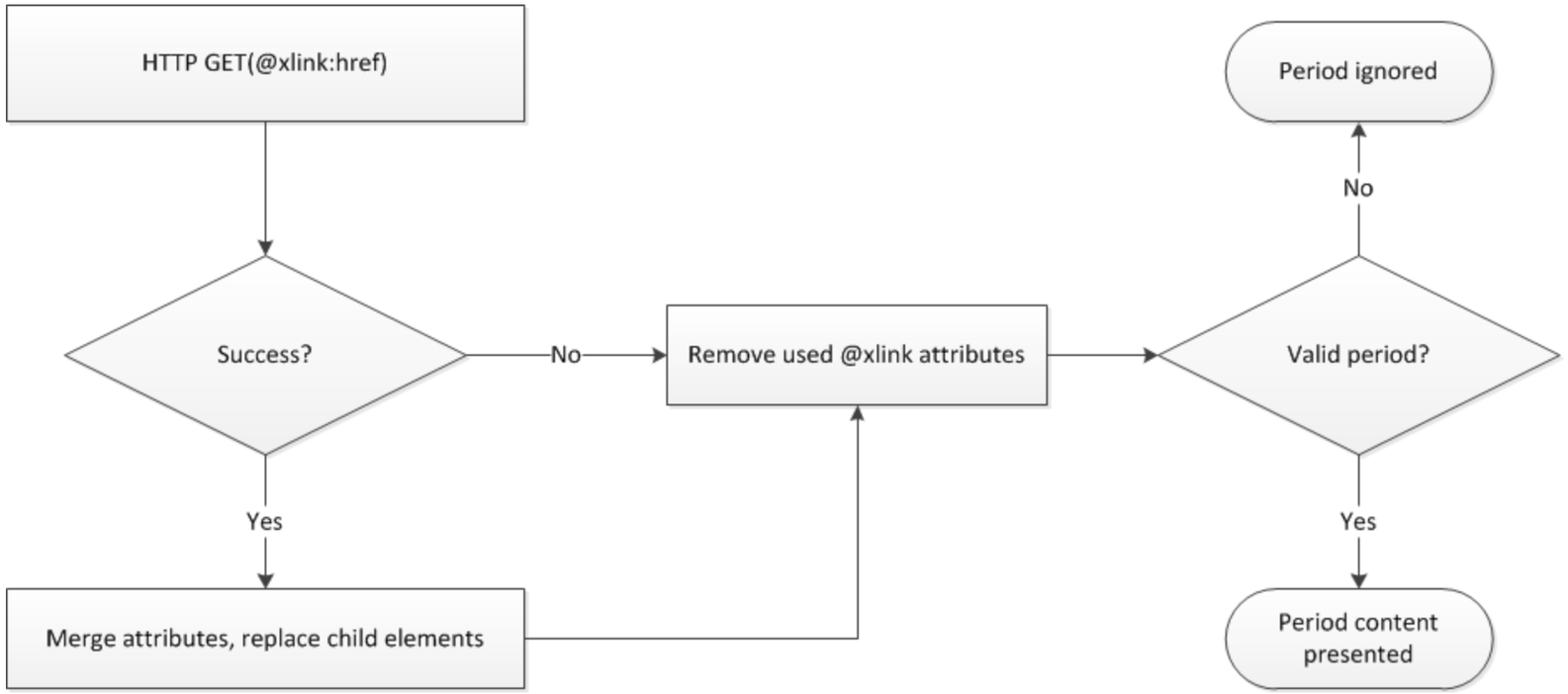


Enter dynamicity

- Periods and asset identifiers provide static ad insertion functionality
  - All ad decisions are made at MPD generation time
    - Trade-off between scalability and targeting
- Insert new ad periods close to their playback time
  - MPD can be updated in real time
    - Regular updates
      - *inefficient for ad insertion – cue message is a rare near-realtime event*
    - DASH MPD Validity Expiration events
  - New ad periods appear in the returned MPDs
- Remote Period elements
  - Remote Period is a period containing **`Period`**`@xlink:href` attribute
  - One or more Period elements can be retrieved from XLink resolver using XLink URL
  - Remote element used as a cue
    - Cue message can be passed in real time to the XLink resolver (making it an ad proxy)
    - XLink resolver conceptually equivalent to the “ad splicer” module in app-driven model

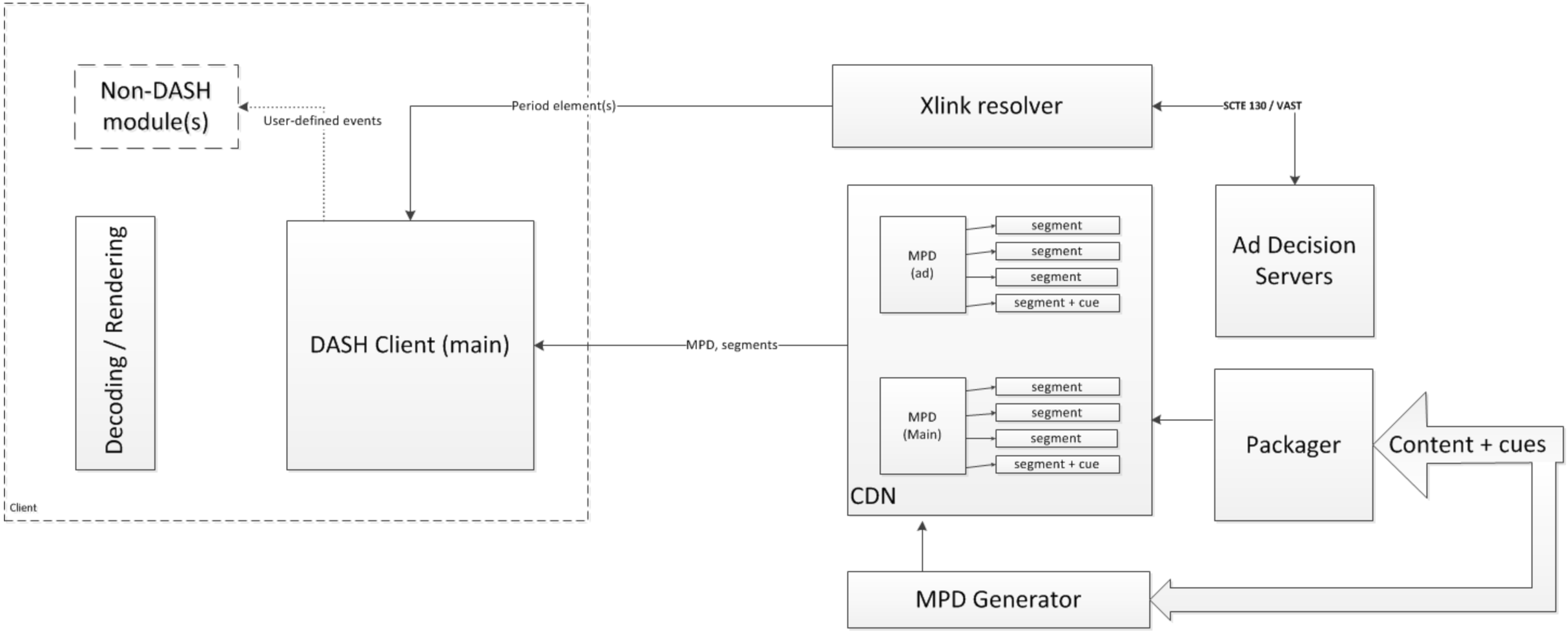
XLink

- Just in time resolution
  - Single period can be resolved into several periods
  - XLink URL can pass cue parameters to resolver
- Default “slate” content
  - Period can have valid “slate” content, replaced in r/t
  - Can be used for ad replacement



- Advanced functionality
  - Delayed resolution: resolution starts only close to playback time and not at MPD parse time
    - Useful for real-time functionality when cue is known long time before splice time

Putting it all together: Server-driven architecture



INTERDIGITAL

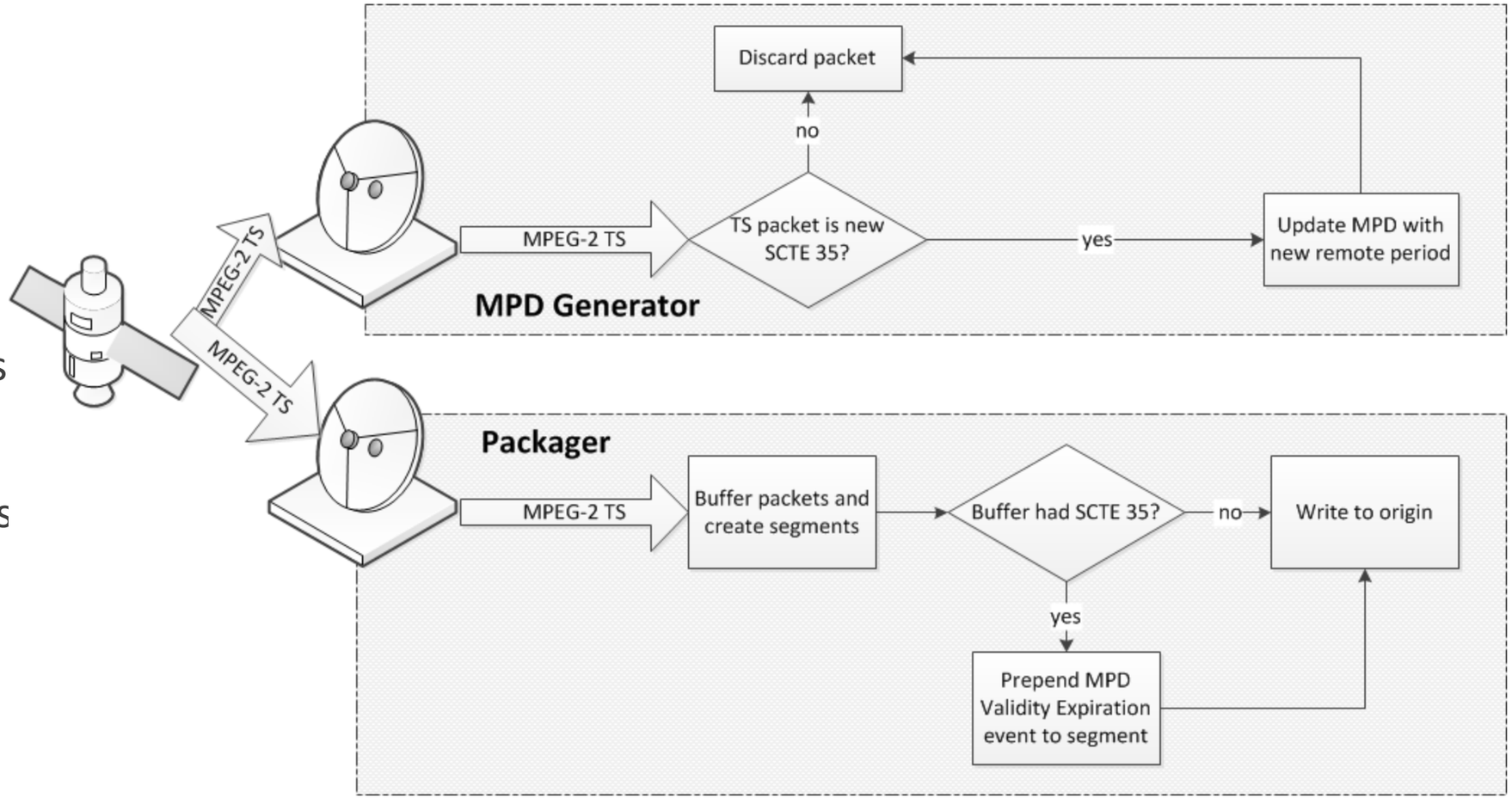
Summary

- DASH has enough native tools to implement interoperable ad insertion
  - “Batteries included”
  - Both HLS-style and HDS/Smooth-style app-driven functionality can be supported
  - Easy to integrate with existing back-end systems
- DASH-IF is working on recommendations and interoperability points
  - In sync with standards bodies – DVB, SCTE
  - Document and test vectors available in Q3 2014

THANK YOU!

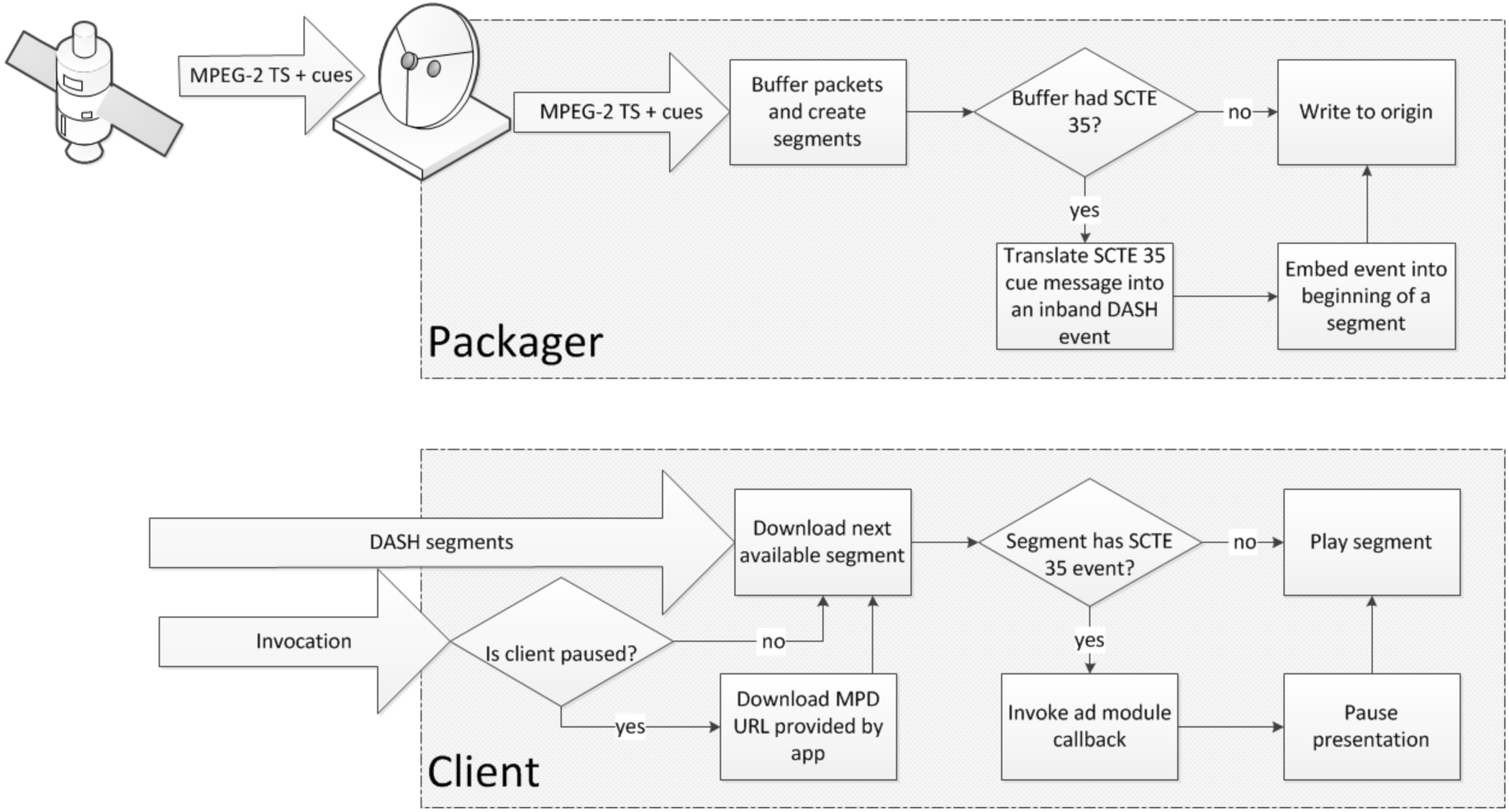
Linear workflow, server-driven case

- Packager
  - Inserts MPD validity expiration events
- MPD Generator
  - Translates cues into remote Period elements
- XLink Generator
  - Resolves remote Periods into complete periods
  - Is allowed to fail



INTERDIGITAL

Linear workflow, app-driven case



INTERDIGITAL

Client operation with XLink

- Client
  - Resolves XLink URL at parse or playback
- Resolver
  - Uses state (cookie) and cue (URL) information
  - Requests content from ADS
  - Returns Period(s)