HbbTV

The HbbTV Targeted Advertising Specification Explained

Contents

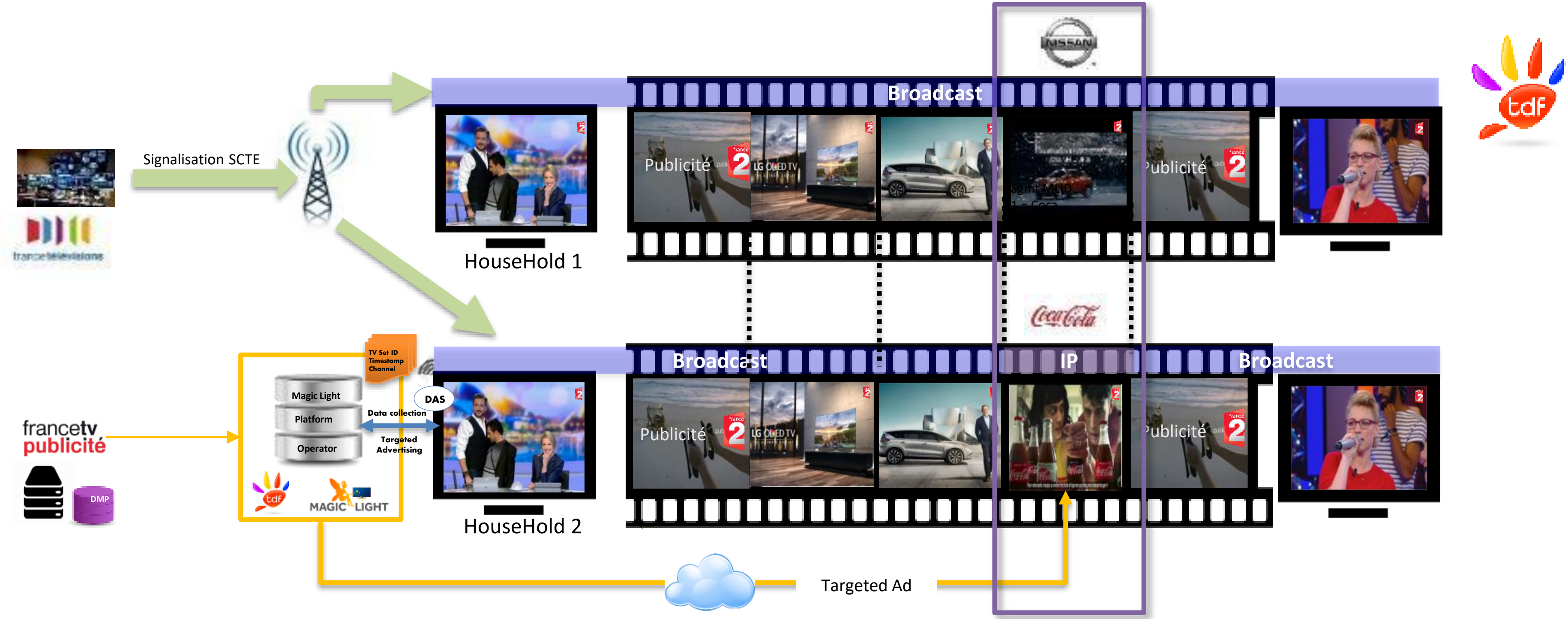
- Overview and use-cases
- Why develop a new spec?
- Step by step walkthrough
- Other considerations and details
- Conclusion

Overview

- Enabling generic ads in a broadcast to be substituted with (targeted) ads delivered via broadband
  - Co-operation with DVB
- Ad substitution opportunities will be announced by existing HbbTV technologies
  - DSM-CC stream events perhaps in combination with the TEMI timeline for better accuracy
  - SCTE 35/104 messages can be translated to stream events (see DVB spec)
- Business logic will re-use web advertising solutions
  - A mixture of JavaScript code in the HbbTV app and servers in the cloud
- Ads will be delivered either via existing DASH / 'progressive download' or new web technologies
  - First appearance of W3C Media Source Extensions in HbbTV
- New 'fast media switch' API' defined to enable optimised switching from broadcast to broadband & back
- 2 performance profiles defined for switching
  - Profile 1 enables replacing ads at the end of an ad break
    - Aimed at implementations where broadcast and ads re-use the same video and audio decoder
  - Profile 2 also enables replacing ads in the middle of an ad break
    - Aimed at implementations that use different video & audio decoders for ads & for broadcast and can keep decoding the broadcast while playing the ads

Dynamic Ad Substitution

(Example by TDF)



Signalisation SCTE

HouseHold 1

HouseHold 2

Broadcast

Broadcast

IP

Broadcast

TV Set ID
Timestamp
Channel

Magic Light

Platform

Operator

DAS

Data collection

Targeted Advertising

francetv publicité

DMP

MAGIC LIGHT

Targeted Ad

Primary Use-Cases

- Replace a single 30s ad in the broadcast
  - Either at the end of an ad break
  - Or in at the start / middle of an ad break
  - More opportunity for a "landing period" at the end of an ad break so timing requirements are more relaxed
- Also variations where replacing consecutive ads is done by combining the replacements into a single stream
  - Can be combined into a single stream in the cloud
  - Can be combined into a single stream by the HbbTV app running in the terminal
  - In both of these only one switch from broadcast to broadband and one switch back again

Secondary Use-Cases

- Replacing multiple consecutive ads without combining them into a single stream
  - See known limitations slide later
- Replacing content longer than an ad
  - e.g. inserting local news into a national broadcast without creating a whole new broadcast
- Replacement content is not played from the beginning
  - e.g. for on-demand content, played from a fixed offset into the content
  - e.g. for live content, played from the live edge

Out of Scope of This Spec

- VOD
  - HbbTV 2 has client–side ad insertion for VOD using 3 video elements
    - Replacing that approach is not a goal of this spec
  - Ad insertion for VOD can also be done server-side
    - Multi Period DASH with some combination of xlink and/or dynamic MPDs
    - HbbTV investigating adding this to the HbbTV-DASH-DRM reference app
- Live OTT
  - More experience needed of live OTT and of ad replacement in broadcast before approaching this
  - Could be considered as a second phase

Why Develop a New Spec?

- TA with HbbTV 1.5 is good enough to start an ecosystem but not good enough to sustain one
  - Some advertisers want a **<u>100%</u>** guarantee that ad playback will **<u>only</u>** start if it's **<u>certain</u>** that it play to the end without stalling, buffering or cutting back to the broadcast ad
    - Can only be achieved if the ad is loaded into RAM & played without any network access
  - TA stresses parts of HbbTV 1.5 that haven't been stressed before
    - Can be variable between different implementations
    - Requires white-listing or at least black-listing
  - On many TVs
    - does not meet user expectations on transitions from broadcast to broadband and (particularly) back to broadcast
    - Does not comply with traditional broadcast quality standards
  - "Landing period" requirements limit where & how often HbbTV 1.5 TA can be used
    - Not possible and/or acceptable to include several seconds of black or sacrificial content everywhere it might be desirable to substitute an ad or to do it several times per hour

Step by Step Walkthrough

1. Broadcaster sends message to app announcing that 'placement opportunity' is near
   - Existing HbbTV mechanisms can be used
     - Some additional testing / clarifications required for reliability
   - Perhaps new mechanism based on SCTE-35
2. App asks ad server for an add that could be played
   - Uses existing web advertising standards ("VAST")
3. App preloads ad
   - Preload may be 100% if there's enough RAM / depending on broadcaster / advertiser requirements
     - 100% preload uses Web "Media Source Extensions" API – new to HbbTV
4. App tells TV /STB when to switch from broadcast to ad
   - New "fast media switch API"
5. App reports back on playback of ad
   - Critical otherwise nobody gets paid
6. App switches back from ad to broadcast
   - Also using new "fast media switch API"

#1: Broadcaster sends message to app announcing that 'placement opportunity' is near

- Existing HbbTV mechanisms can be used
  - DSM-CC stream events from HbbTV 1.5
  - Access to broadcast timelines from HbbTV 2
    - PTS – simple but may be modified in broadcast distribution network
    - TEMI – work needed for broadcasters to add this but more likely to pass through broadcast distribution network intact
- Stream events on their own not frame accurate
  - Reported to often fire ahead of the video
  - May not matter too much for announcing a placement opportunity is coming in 60-90 seconds
  - Web Crypto API added to HbbTV to enable stream event payloads to be encrypted and/or authenticated
- Announcements far enough in advance can also be sent over the web
  - Several web technologies available
- Direct reception of industry standard SCTE-35 messages not yet included
  - In theory SCTE-35 message payload can be re-packaged into stream events

#2: App Asks Ad Server for an Ad that Could be Played

- Existing web solutions will be used
  - Ad server specific JavaScript / HTTP for the request to the ad server
  - VAST for the response from the ad server
- Care needed to avoid ads having noticeably different quality from the broadcast
  - Video: UHD vs HD vs SD
  - Audio: Stereo vs 5.1 vs other, loudness, language
  - Subtitles?
  - App can query existing HbbTV APIs for information about the broadcast and append this to the query sent to the ad server
- Additional constraints will apply if multiple consecutive ads are to be substituted
  - Same codecs, same number of video & audio tracks, ….
- DVB have made guidelines for this
  - BlueBook A178 - http://dvb.org/wp-content/uploads/2019/12/a178-1_dvb-targeted-advertising-nov_2019.pdf

#3: App Preloads Ad

- Major reasons why preloading the ad is important
  - To minimise switching time from broadcast to ad
  - To minimise risk of ad playback stalling in the middle
  - Load distribution on servers (with HbbTV app adding a small element of randomness)
- HbbTV 2 supports partial preloading
  - HTML5 video element load method does some preloading
  - How much it preloads is invisible to app & device dependent
- HbbTV TA gives app full control & visibility of preloading
  - App loads ad into RAM using XMLHttpRequest (XHR)
    - Care needed to avoid high transient RAM usage – don't just read the entire ad into RAM in one go
  - Ad can be played from RAM using W3C Media Source Extensions (MSE)
    - Reported that MSE works in many HbbTV implementations already & some apps use it
  - Spec requires TA implementations have enough RAM to load a 30s ad
    - Disclaimers apply – no dedicated RAM for preloading ads so apps using lots of RAM for graphics will find there's less for preloading ads
  - MSE also supports playing multiple consecutive ads
    - App just keeps loading each ad in turn via XMLHttpRequest and passing it to MSE
    - Constraints apply on stream composition

#4: App tells TV /STB when to switch from broadcast to ad

- HbbTV TA defines new "fast media switch API"
  - Enables app to request the terminal switch from one media presentation to another
  - Same API for broadcast to ad and ad to broadcast
- Switch from broadcast to ad
  - Switch from a video/broadcast object
  - Switch to a preloaded HTML5 video element
    - Source can be MSE or .mp4 file or DASH
  - Switch at a particular time (PTS or TEMI)
- Knowing about the switch in advance enables optimised implementations
  - Implementations can reject the switch if given <2s notice
- Apps can prioritise between the switch being fast and it happening at all
  - Switching in the middle of an ad break may need to be fast or not happen at all
  - Other kinds of switch may need to happen even if they take longer

#5: App Reports Back on Playback of Ad

- Existing web solutions will be used
  - JavaScript code in the app
  - Querying the HTML5 video element API to report firstQuartile / midpoint / thirdQuartile (etc)
- 2 HbbTV-specific issues need care
  - Really playing an ad from the beginning to the end
    - Some devices may not play the start or the end of the ad
    - APIs may report end of content before the last video and/or audio has actually been output
  - May be relevant to report that the audio has really been output
    - User pressing "mute" on the remote control not visible in HbbTV 1.5/2
    - HbbTV TA spec allows apps to query if overall system audio is muted

#6: App Switches Back From Ad to Broadcast

- Same API as switch from broadcast to ad with different parameters
  - Switch from an HTML5 video element
  - Switch to a video/broadcast object
  - Switch either at a particular time or at the end of the ad
- Timing very dependent on implementation and broadcast stream properties
  - See later slides for discussions on
    - Single vs dual decoder implementation scenarios
    - Landing periods

Implementation Considerations (1)
Single Decoder Implementations

- Broadcast playback and ad playback use the same video and audio decoder
  - Preloading an ad only means loading the data is in memory
- Switching from broadcast to ad requires disconnecting the decoders, re-initialising the decoders and reconnecting the decoders
  - This will take time reported to be anywhere between 250ms and over 1s in HbbTV 2 products
- Switching from ad to broadcast requires all the above & additionally requires waiting for an access point (aka GOP) in the encoded broadcast stream
  - Interval between GOPs could be 1.5s to 2s
  - Ideally broadcast encoders could add more GOPs at the end of ads that could be substituted but none do this today & there are practical issues
- Summary
  - Single decoder implementations will struggle to switch from broadcast to broadband and back again fast enough to fit within reasonable expectations of a landing period
    - May be possible for silicon providers to optimise this but could be time consuming and bring a real risk of breaking things that currently work

Implementation Considerations (2)
Dual Decoder Implementation

- Broadcast playback and ad playback would use different video and audio decoders
  - Preloading the ad can also include initialising the video & audio decoder to use for the ad as this won't disturb the broadcast
- Switching from broadcast to ad could be almost instant – perhaps 1 frame time?
- Decoder used for broadcast can keep decoding broadcast while ad is being played
- Switching from ad to broadcast can be almost instant
  - No need to wait for GOP as decoder has kept up to date with the broadcast
- Dual UHD decoding requires a lot of memory and a lot of memory bandwidth
  - May not be any products that support this
  - HbbTV TA spec excludes UHD from performance requirements
- Summary
  - Believed that many UHD-capable TVs and STBs would be capable of dual HD decoding
    - Confidentiality makes quantifying this very hard
    - Connecting dual HD decoding to HbbTV for TA may be a lot less work than optimising switching time for a single decoder implementation
  - TVs and STBs supporting dual HD decoding can fall back to single decoder mode for UHD

Broadcast Considerations - Landing Periods

- How to compensate for the time to switch from broadcast to broadband & back?
  - Allowing for single decoder implementations will need at least a GOP length – perhaps 1.5s to 2s
  - If switching duration was 240ms & accuracy was 40ms then 1.5+2x(240+40) = 2.06s
- Use a "landing period" in the broadcast
  - Content where it doesn’t matter if the viewer doesn't see it – black, logo, ….
- May already exist, may need to be added
  - Some broadcasters include 6 frames of black at every content change (content/ad, ad/ad, ad/content)
  - Some countries may have regulatory requirements for transition zones between content and ads
  - Otherwise would need to be added at the expense of something else
- No technical limit on landing periods but …
  - Multiple seconds of black several times an hour is a poor user experience
  - Landing periods of (say) 4 seconds and having 30 placement opportunities per day would mean 4 complete ads lost
    - Not possible during early phases of TA introduction

Implementation Considerations (3)
2 Profiles for Switching

	Profile #1	Profile #2
Target implementation	Single decoder	Dual decoder
Target use-case	Replace ad(s) at the end of an ad break	Replace ad(s) anywhere in an ad break
Broadcast to broadband switching duration	<= 240ms	<=120ms
Broadband to broadcast switching duration	<= 240ms	<=120ms
Switching accuracy	40ms	40ms
Decoding broadcast while ad is playing	No	Yes
Typical landing period(with 1.5s GOP in the broadcast)	2.06s = 1.5s + 2*(240ms+40ms)	320ms = 2*(120+40)

Querying Device Capabilities

- Apps are able to query static and dynamic multi-stream decoding capabilities
  - "static" meaning the maximum possible device capabilities
    - 2 new JavaScript methods
      - oipfCapabilities.broadbandCapabilities(decoder) and
      - oipfCapabilities.broadcastCapabilities(decoder)
    - Results use same format as existing HbbTV XML capabilities
  - "dynamic" meaning what capabilities are available at that time
    - HbbTV 1.5 JavaScript properties extraSDVideoDecodes, extraHDVideoDecodes clarified & extended with a matching extraUHDVideoDecodes
    - Example

Content being decoded	UHD	HD	SD
extraUHDVideoDecodes	0	0	0
extraHDVideoDecodes	0	1	1
extraSDVideoDecodes	0	2	3

- HbbTV 1.5 / 2 device XML capabilities extended to indicate version of TA spec supported

HbbTV TA Spec Contents – Part #1

Section	Title	Summary
4	Overview	Conceptual model of switching times and accuracies
5	User experience	Explanation of what users will see
6	Service and application model	Integration of MSE into core HbbTV service & app model
7	Formats and protocols	Requirement on broadcasters re URLs
8	Browser application environment	Definition of the "fast media switch API
9	System integration	Integration of MSE, reliability and resilience, limitations & restrictions
10	Capabilities	Various device capabilities and capability querying
11	Security	Security considerations
12	Privacy	Privacy considerations
13	Media synchronisation	Integration of MSE into core HbbTV media sync feature
14	Companion screens	Integration of MSE into core HbbTV companion screen feature
A	OIPF specification profile	Additional web and OIPF APIs
B	Timings and terminology	More on conceptual model of switching times
C	Advert data flow, buffering and memory management	Considerations on how data for ads is loaded using XMLHttpRequest and played with MSE
D	Advert reporting	Considerations on reporting ad playback
E	Video and audio decoder resource allocation	Considerations on video and audio decoder resource allocation

HbbTV TA Spec Contents – Part #2

- Part 2 contains non-functional requirements
  - Bitrate (etc) defining the 30s ad HbbTV TA terminals must be able to load
    - HD AVC video at 8 MBit/sec
    - HE-AAC audio at 128 kBit/sec
  - Definitions of profile #1 and #2 for broadcast / broadband switching time and accuracy
    - See previous slide

Independent Specification

- HbbTV TA is an "independent specifcation"
  - Not (yet) integrated into the core spec
- HbbTV has independent specs to allow for more experimental or controversial features
  - Anything that may not be appropriate to include in all retail TV sets
  - Other examples include operator applications, IPTV and application discovery over broadband
- Commercial aspects around TA have been controversial and are not yet resolved
  - More time, thought & market evolution needed before TA can be considered for the core spec
  - Parts of the TA spec may be integrated earlier – e.g. W3C MSE

Next Steps

Commission, review & accept the unit tests

Known Limitations

- Use-case "Replacing multiple consecutive ads without combining them into a single stream" has issues
  - Even on dual decoder implementations, the decoders may have different capabilities
    - Only one may be connected to the broadcast
    - Only one may be connected to DRM
    - Other obscure restrictions may apply
  - May need to fall back to (un-optimised) single decoder mode so the correct decoder is used for the correct content
- See also previous comments about substituting a UHD ad in UHD content
  - Switching times of (say) 1s + 1.5s GOP length would mean a 3.5s landing period
    - Well beyond what can be justified with limited numbers of TA devices at the start

Conclusions

- The TA spec is done
- Significantly smaller & simpler than other independent specs
- Focussed on 2 key primary use-cases but can be used for other secondary ones too
- Meets most commercial requirements
- Key implementation responsibilities lie with silicon providers & less with TV integrator or software suppliers
- Widespread deployment needs remaining commercial aspects to be resolved