

HbbTV v2

*Web Specification Usage and Integration
Companion Screen Features*

*Jon Piesing
Chair HbbTV Specification Working Group*

Contents

- » *Background*
- » *Web specification selection*
- » *Web specification integration*
- » *Some practical issues*
- » *Companion screen*
- » *Some strategic issues*

Background - 19 Requirements

- » *The requirements process for HbbTV v2 identified 19 improvements*
- » *Top priority*
 - » New technology
 - » HTML5 and associated technologies
 - » HEVC video
 - » Subtitle support in ISOBMFF content
 - » Companion screens
 - » Launching a companion screen application
 - » Application to application communication
 - » Feature improvements
 - » MPEG DASH
 - » Advert insertion into VoD content
 - » Application and content synchronisation
 - » Non-realtime content delivery via broadcast
 - » Multi-stream synchronisation
- » *Second priority*
 - » New technology
 - » DRM in a CAM
 - » Automated testing
 - » Companion screens
 - » Synchronising applications and content across devices
 - » Remotely launching an HbbTV application
 - » Feature improvements
 - » Support for mouse and keyboard devices
 - » Reconciling key request conflicts
- » *Third priority*
 - » New technology:
 - » User identity management
 - » Feature improvements
 - » Trick mode support for VoD
 - » Caching of object carousels

Items coloured in green are mentioned in this presentation to some extent

Web Specification Selection

- » *Refer to the Open IPTV Forum “Web Standards TV Profile”*
 - » Defines a selection from W3C specifications replacing that formerly defined in CEA-2014
- » *HTML5*
- » *CSS*
 - » CSS 2.1 and a variety of modules of CSS3
 - » Basic UI, Color, Images, Backgrounds and Borders, Selectors, Media Queries, Multi-column Layout, Flexible Box Layout, Fonts, Transforms, Transitions, Animations
- » *DOM*
 - » DOM 3 as required by HTML5 plus legacy keyCode/charCode support
- » *Others*
 - » WOFF, Canvas 2D, XHR, Web Messaging, Web Socket, Web Workers, Server-Sent Events, Web Storage, Media Fragment URI
- » *For some specs only a profile is required*
- » *For specs that are not a Recommendation, using a more recent version is explicitly permitted*

Web Specification Integration

- » *MPEG DASH Integration*
 - » Support DASH MPD as the 'src' of a video element
 - » Expose Adaptation Sets in MPD as VideoTrack and AudioTrack
 - » Define how DASH time line maps to HTML5 media timeline
 - » Expose DASH (2nd edition) event streams as TextTrack and DataCue
- » *Accuracy of time reporting*
 - » Where in the media decoder pipeline is 'currentTime' measured and how accurate does it have to be?
 - » TVs typically do significant media processing on video before combining with graphics and also on the results after combination
- » *Advert insertion using multiple video elements*
 - » Requirement to support >1 video element in a page
 - » Requirement to release hardware decoders from a media element in the paused state if needed for another media element
 - » Requirement for delay between the end of presentation of one media element and starting presentation of another media element to be less than 250ms under specified conditions
- » *Support for TTML subtitles with a native TTML decoder*

Major Issue – the Controls Attribute

- » *From HTML5 ..*
 - » The controls attribute is a boolean attribute. If present, it indicates that the author has not provided a scripted controller and would like the user agent to provide its own set of controls.
 - » If the attribute is present, or if scripting is disabled for the media element, then the user agent should expose a user interface to the user. This user interface should include features to begin playback, pause playback, seek to an arbitrary position in the content (if the content supports arbitrary seeking), change the volume, change the display of closed captions or embedded sign-language tracks, select different audio tracks or turn on audio descriptions, and show the media content in manners more suitable to the user (e.g. full-screen video or in an independent resizable window). Other controls may also be made available.
- » *HbbTV apps typically want to provide play-control UI themselves (e.g. to block ad-skipping) but do not want to provide the rest of this themselves*
 - » No reason to do so – no benefit from doing it
 - » Don't have enough APIs to do this properly (e.g. reading previously set end-user preferences for subtitles, audio description, etc)
 - » Relevant key events not delivered to app (volume up/down, subtitle, audio description,)
 - » Poor user experience from having the rest of this different between broadcast and IP delivered video
- » *This is a pretty serious conceptual disconnect*
- » *See W3C bugzilla #24859, #24860, #24861*

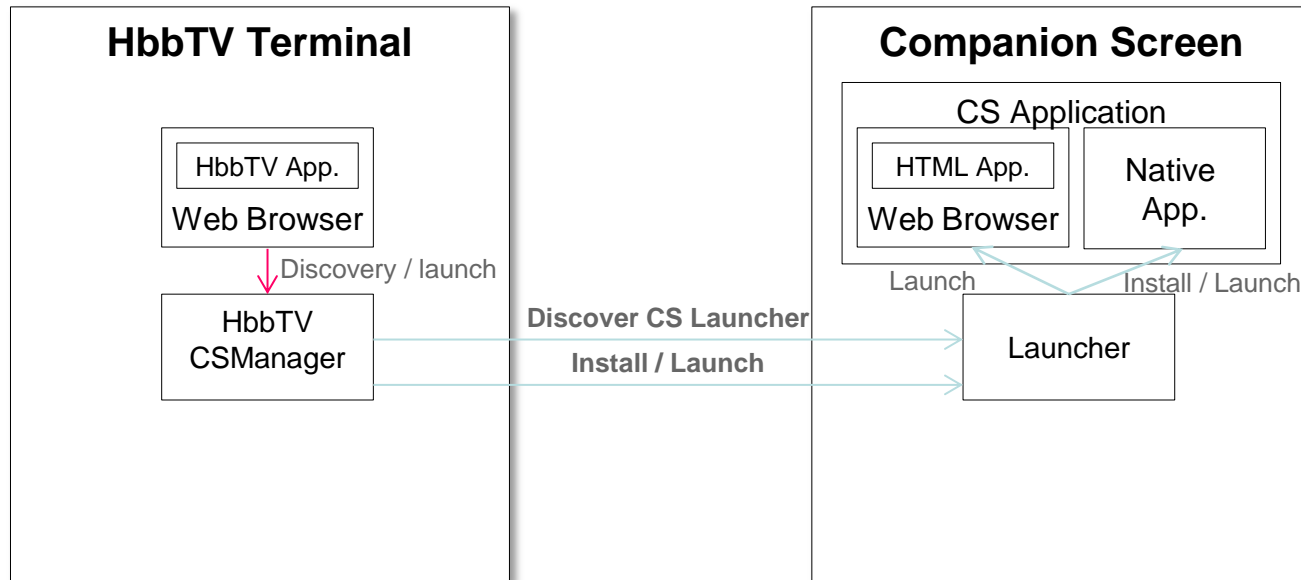
Some Smaller Practical Issues

- » *HTML5 MediaError not detailed enough*
 - » No easy way to distinguish failure due to DRM from ...
 - » Failure due to parental access control or from ...
 - » Failure due to other reasons
- » *HTML5 VideoTrack and AudioTrack support too limited a set of metadata*
 - » See W3C bugzilla #24863
 - » Do not expose codec, number of audio channels,
- » *Missing short-lived (instantaneous) cues*
 - » TV often uses instantaneous events to notify an app that something happened
 - » Mapping these to TextTracks and Cues is obvious but there's a risk of them being missed
 - » See W3C bugzilla #24161
- » *Unclear in spec if DOM FocusEvents are generated when the UA gains and loses focus*
 - » In practice they are but the spec is unclear & asking for it to be clarified just got it re-written but no clearer
 - » See W3C bugzilla #23271

Companion Screen

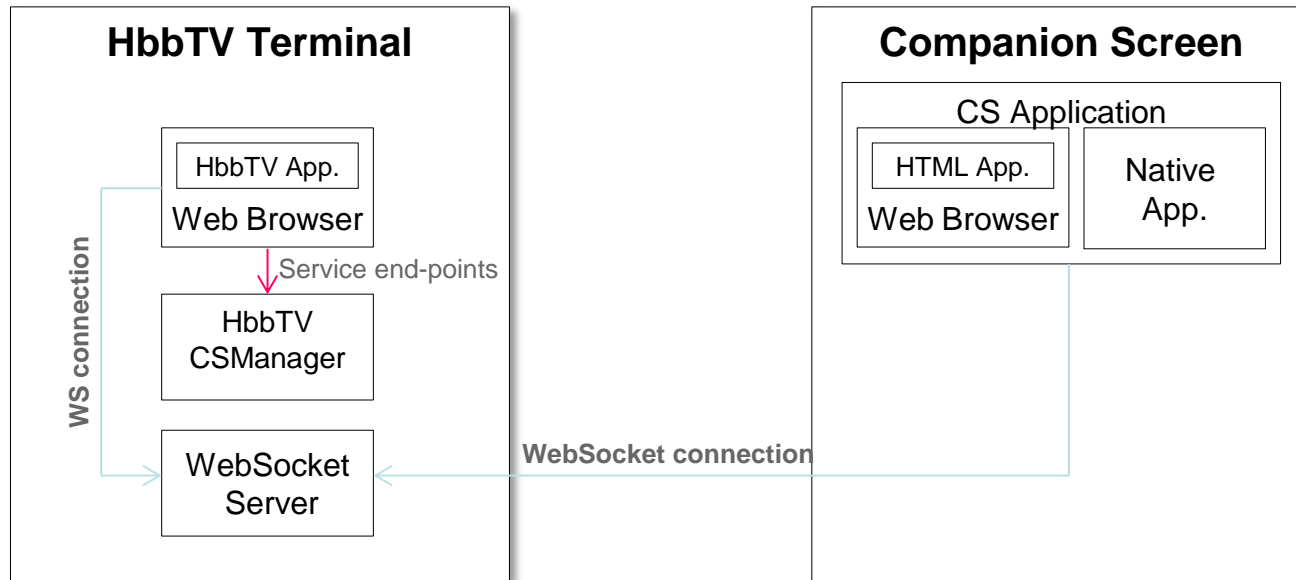
- » *3 companion screen features are covered here*
 - » Launching a CS application from the TV
 - » App2App communication
 - » Remotely Launching an HbbTV application
 - » “Synchronising applications and content across devices” is omitted due to lack of time and the solution not being web centric
- » *Some things are common to all of these*
 - » Discovery based on UPnP (SSDP)
 - » HbbTVCSManager object in the TV
- » *Companion screen apps can obtain URLs for service endpoints on TV either by*
 - » UPnP (SSDP) or .
 - » App on TV obtaining URL for remote endpoint from HbbTVCSManager and then passing that either by
 - » App2App communication or
 - » Launching a CS application from the TV

Launching CS App. From the TV



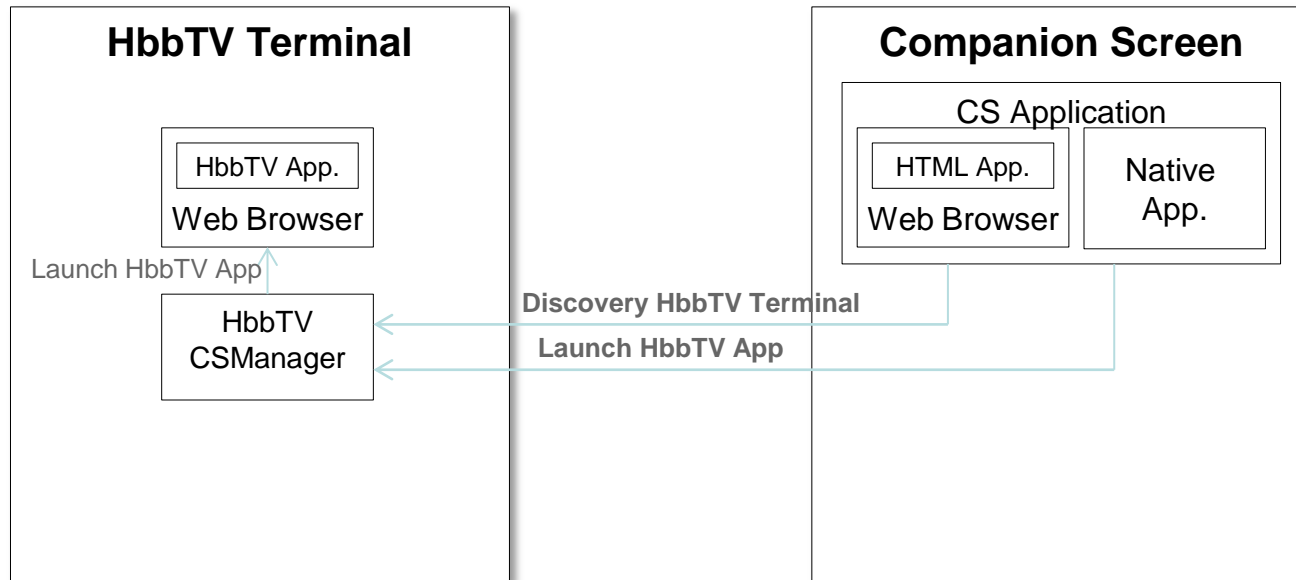
- App asks HbbTVCSManager for list of currently running launchers
- App asks chooses a device with a running launcher
- App asks launcher to (install and) launch app
- Protocol between terminal and launcher is proprietary but payload for describing what app to launch (from what app store) is specified

App2App Communication



- TV runs websocket server
- App asks HbbTVCSManager local end point of websocket server
- App makes websocket connection to local end point
- Companion screen app obtains remote end point of websocket server
- Companion screen app makes websocket connection to remote end point
- WebSocket server pairs local and remote connections

Remotely Launching HbbTV App.



- Similar in concept to DIAL but does not refer to that
- App on companion screen discovers HbbTV terminal & makes HTTP POST request to the end point to launch an app
- Payload of HTTP POST is an XML fragment used elsewhere in HbbTV to define an app
- App launching can be refused for various possible reasons listed in the spec

Some Strategic Issues

- » *Gaps in HTML5 for IP delivered video*
- » *Enabling the video element to be used for classical broadcast video*
- » *How far can/should the W3C go in specifying integration with TV*
 - » W3C specs are global but TV is not when you look at the details
 - » Content metadata, parental rating, (etc) are all regional
 - » Many global specs are localised at a regional and again at a country or even operator level
 - » W3C has been uncomfortable about leaving provision for market specific extensions but what other options are practical?
 - » TV uses codecs and other specs which are not royalty free