HbbTV Companion Screen Sync

W3C TPAC 2017

Date of Presentation: 6 November 2017

Chris Needham

BBC | Research & Development

What is HbbTV?

- Open spec implemented by major TV manufacturers for the European market
- Defines an HTML+JS interactive application environment on the TV
- Single app running at any time
- HbbTV 1.x used widely in Europe
- HbbTV 2.0 First deployments in UK

HbbTV features

- A UA with a profile of HTML5 capabilities (CSS 2.1/3, DOM3, etc)
- TV specific functionality (extension of OIPF and CEA2014)
    - Broadcast tuner
    - PVR functions
    - Conditional access modules (DRM)
- Additional features required by broadcasters
    - DASH, TTML subtitles
    - Synchronised media playback: combining broadcast and IP streams
    - Companion screen interaction and synchronisation
    - UHD
    - etc

Companion Screen & Media Synchronisation Features in HbbTV 2.0

Companion screen features

- TV discovers companion
- TV launches companion app

- Companion discovers TV
- Companion launches HbbTV app

- "App to App" communication

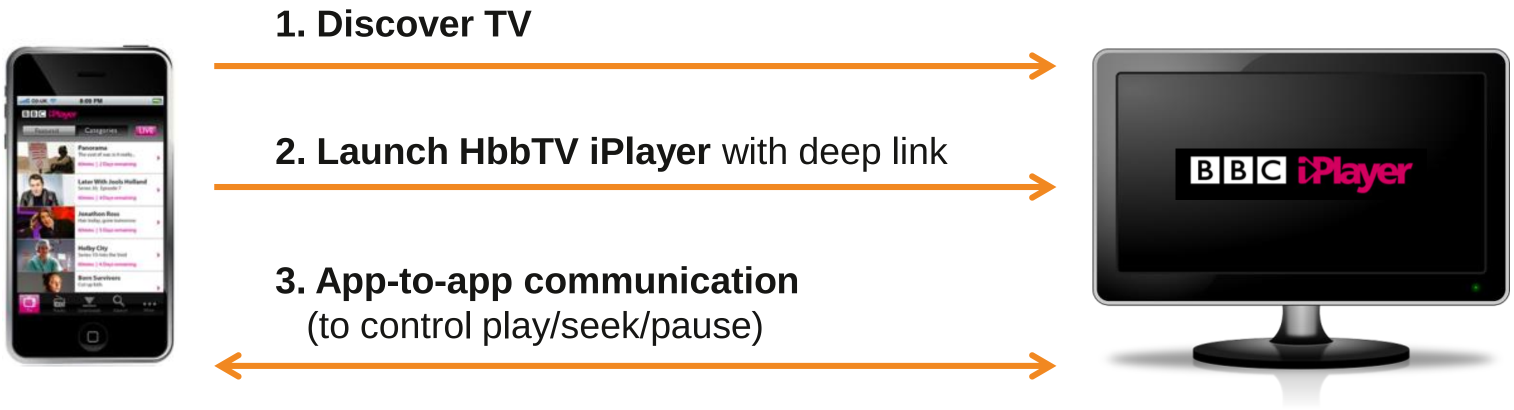
Media Synchronisation features

- Application sync – HbbTV apps synchronise to content (broadcast & IP)
- Inter-device sync – Companion apps synchronise to TV content (broadcast & IP)
- Multi-stream sync – Replace broadcast audio or subtitles with IP content

Example use cases

Casting

User browses programmes on iPlayer mobile app and chooses to "cast" it to watch it on the TV.
The user controls play/pause/seek from the iPlayer mobile app.



1. Discover TV

2. Launch HbbTV iPlayer with deep link

3. App-to-app communication
(to control play/seek/pause)

Example use cases

Personalising accessibility & shared viewing
Audio description / commentary / camera angle … streamed to an app on the phone and frame-accurately synchronised to the TV.

A different experience for everyone in the room. New ways to deliver accessible services.

Take-away viewing
Phone/tablet shows same content as the TV, synchronised frame-accurately to it.

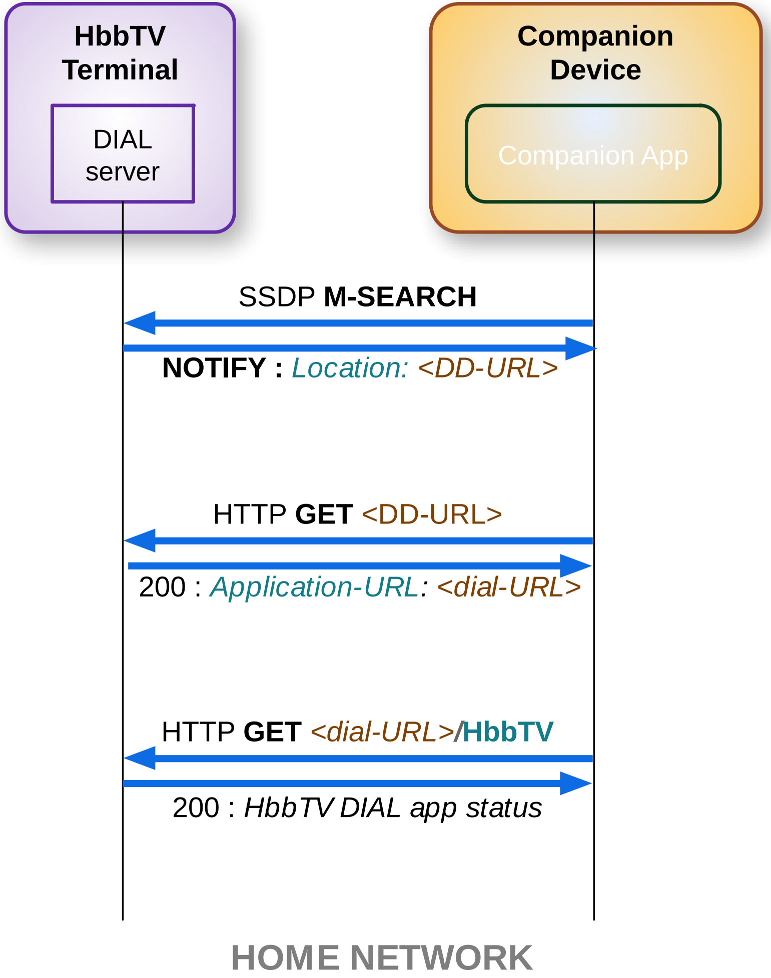
Take viewing temporarily with you out of the room, then rejoin seamlessly.

HbbTV application lifecycle

- Launched:
    - from the home screen on connected TVs
    - by AIT signalling in a DVB broadcast
    - by companion device via DIAL HbbTV app
- At any time an app is either:
    - Broadcast-related
        - Mixed with broadcast audio/video
        - Must be listed as allowed in transport stream signalling (DVB AIT)
    - Broadcast-independent
        - No broadcast audio/video (but can play IP streams)
        - Can become broadcast related if listed in AIT
- User exits app, or app is terminated by system

Discovering an HbbTV terminal

- **DIAL** is for discovering a TV and launching TV apps
  - Uses SSDP from UPnP for discovery
  - M-SEARCH reply returns UPnP Device Description URL in response "Location" header
- HbbTV engine appears as a DIAL app
- Companion fetches status of HbbTV DIAL app. Response includes:
  - URL for app-to-app communication
  - URL for inter-device synchronisation (DVB CSS-CII)
  - User Agent string of the HbbTV engine



HbbTV Terminal

DIAL server

Companion Device

Companion App

SSDP **M-SEARCH**

NOTIFY : *Location: <DD-URL>*

HTTP **GET** <DD-URL>

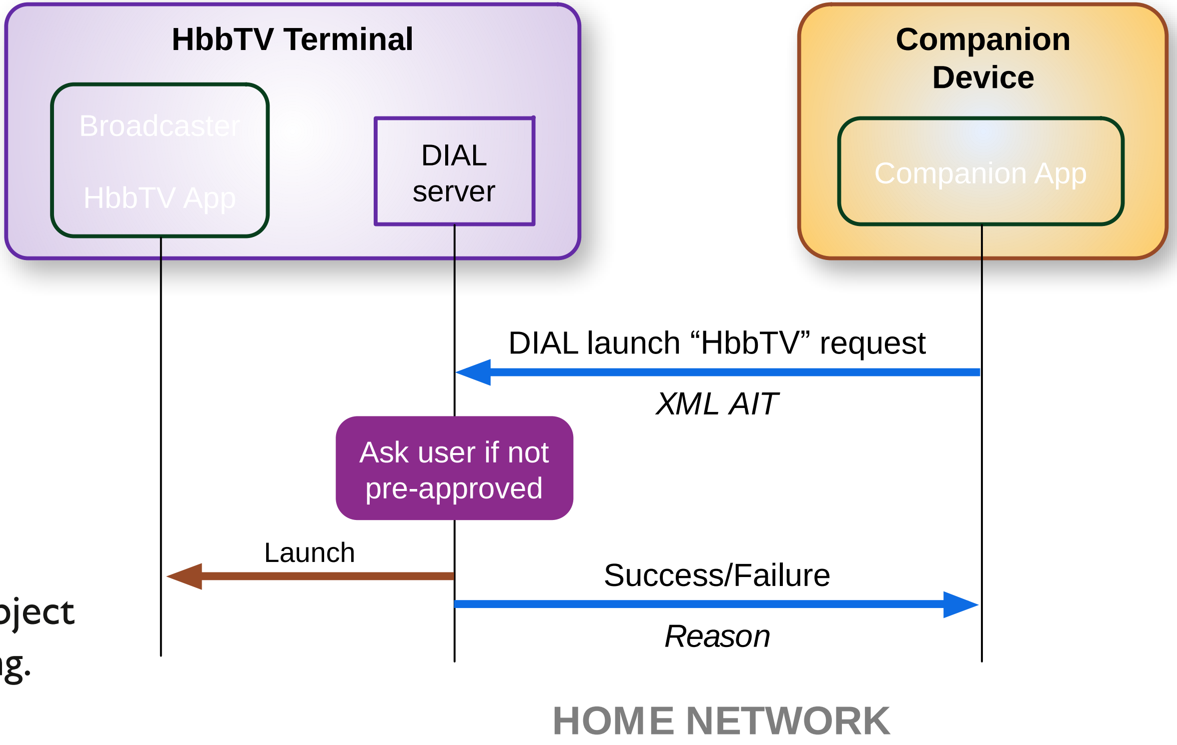
200 : *Application-URL: <dial-URL>*

HTTP **GET** *<dial-URL>*/**HbbTV**

200 : *HbbTV DIAL app status*

HOME NETWORK

Launching an HbbTV Application



This mechanism launches a **broadcast independent** HbbTV app.

The app can convert itself to **broadcast related** later, subject to suitable broadcast signalling.

Launching an HbbTV Application – security

User security considerations:

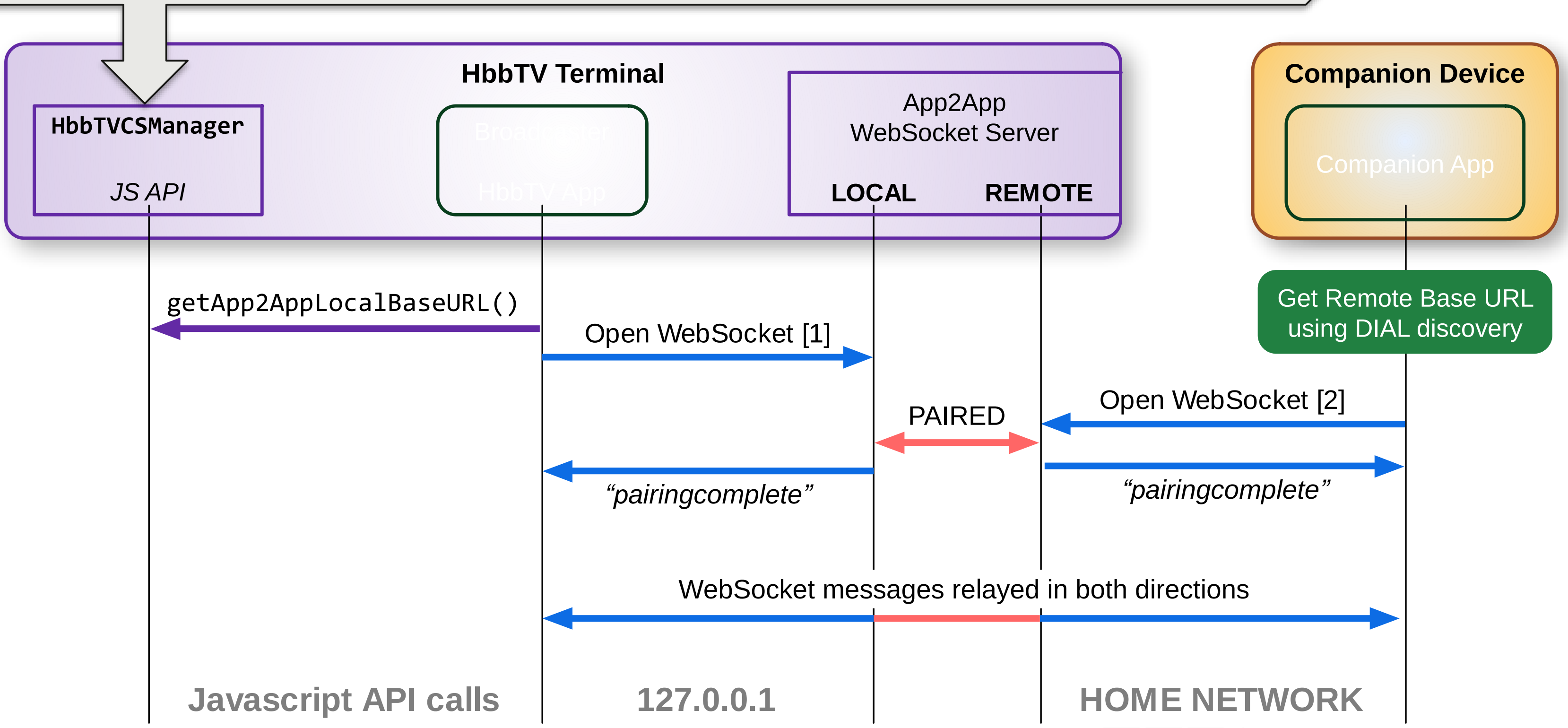
“Did you just request an application to be launched on this TV?”

User will be prompted to approve/deny
(unless HbbTV app is in whitelist by manufacturer or network operator)

HbbTV app is identified by its URL, as conveyed in
<applicationTransport> and <applicationLocation> elements in the XML AIT
(not including any ‘?’ query or ‘#’ fragment suffixes)

App to app communication

```
<object type="application/hbbtvCSManager" id="csMgr">
```



BBC | Research & Development

App to app communication

For pairing, the ***app-endpoint*** suffix must match

It does not matter who connects first

Once paired, the connection is transparent
Message content is application defined

Connections are 1-to-1, but multiple connections supported. The HbbTV app must create a separate connection for each companion

Connections are not secure

```
localBaseUrl = csMgr.getApp2AppLocalBaseURL();
appEndpoint = "uk.co.bbc.myapp";

ws = new WebSocket(localBaseUrl + appEndpoint);

ws.onmessage = function(evt) {
    if (evt.data == "pairingcompleted") {
        // can now send/receive msgs as normal
    }
});
```

Mapping to Presentation API

Opening a presentation

```
    const req = new PresentationRequest(url, params);
```

Or:

```
    const req = new PresentationRequest(url);
    req.start(params);
```

HbbTV `applicationTransport` and `applicationLocation` could be obtained from the presentation URL.

Additional parameters: `orgId` and `appId` (for broadcast-related presentations, or UA could supply for broadcast-independent).

Same-origin policy for mixed broadcast / Web content?

Mapping to Presentation API

Messaging between controlling and receiving pages

```
    connection.send(channel, message);

    connection.onmessage(channel, (message) => { ... });
```

HbbTV uses an app-endpoint to allow routing of messages between WebSocket connections.

Must be known to both controlling and receiving pages.

One WebSocket connection per client

Add a `channel` parameter to the `send` and `onmessage` methods?

Media Synchronisation

Application sync

- Extract current time position for broadcast or IP streamed content

Inter-device sync:

- Allow companions to sync to the content being presented on the TV
- TV implements protocol server defined by DVB CSS spec (ETSI TS 103 286-2)

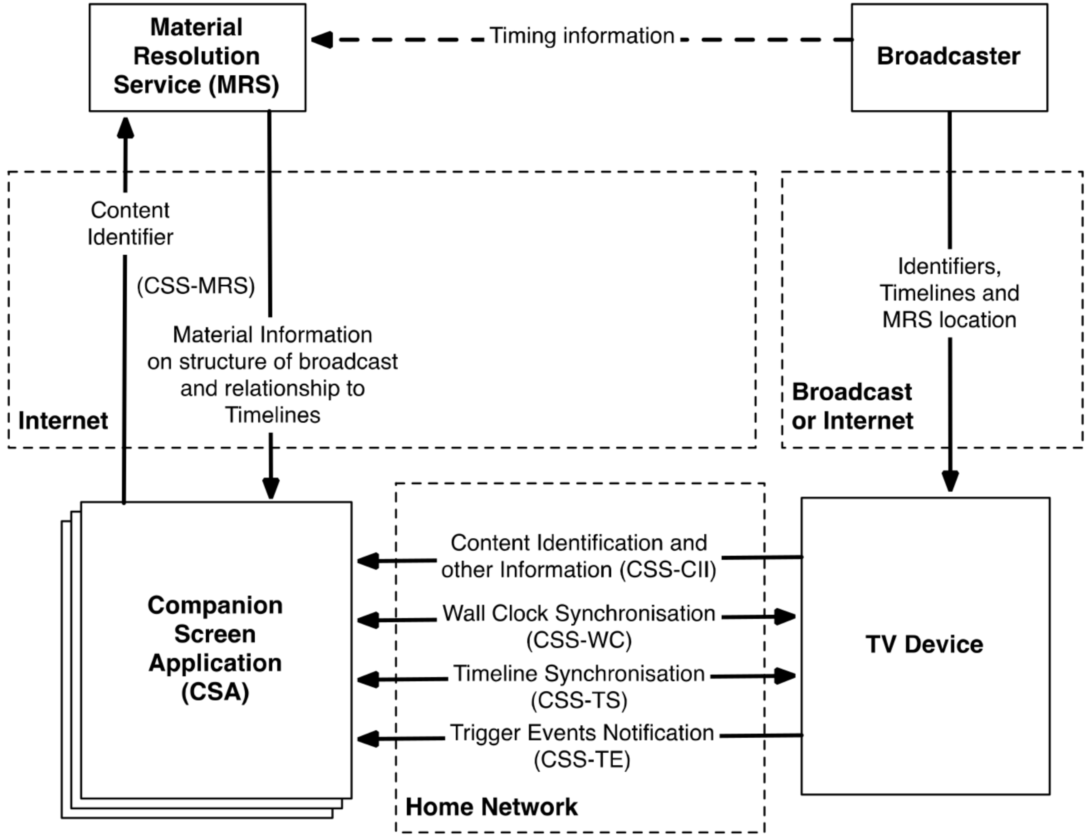
Multi-stream sync:

- Sync a DASH stream (and optionally EBU-TT-D subtitle document) to broadcast

Related:

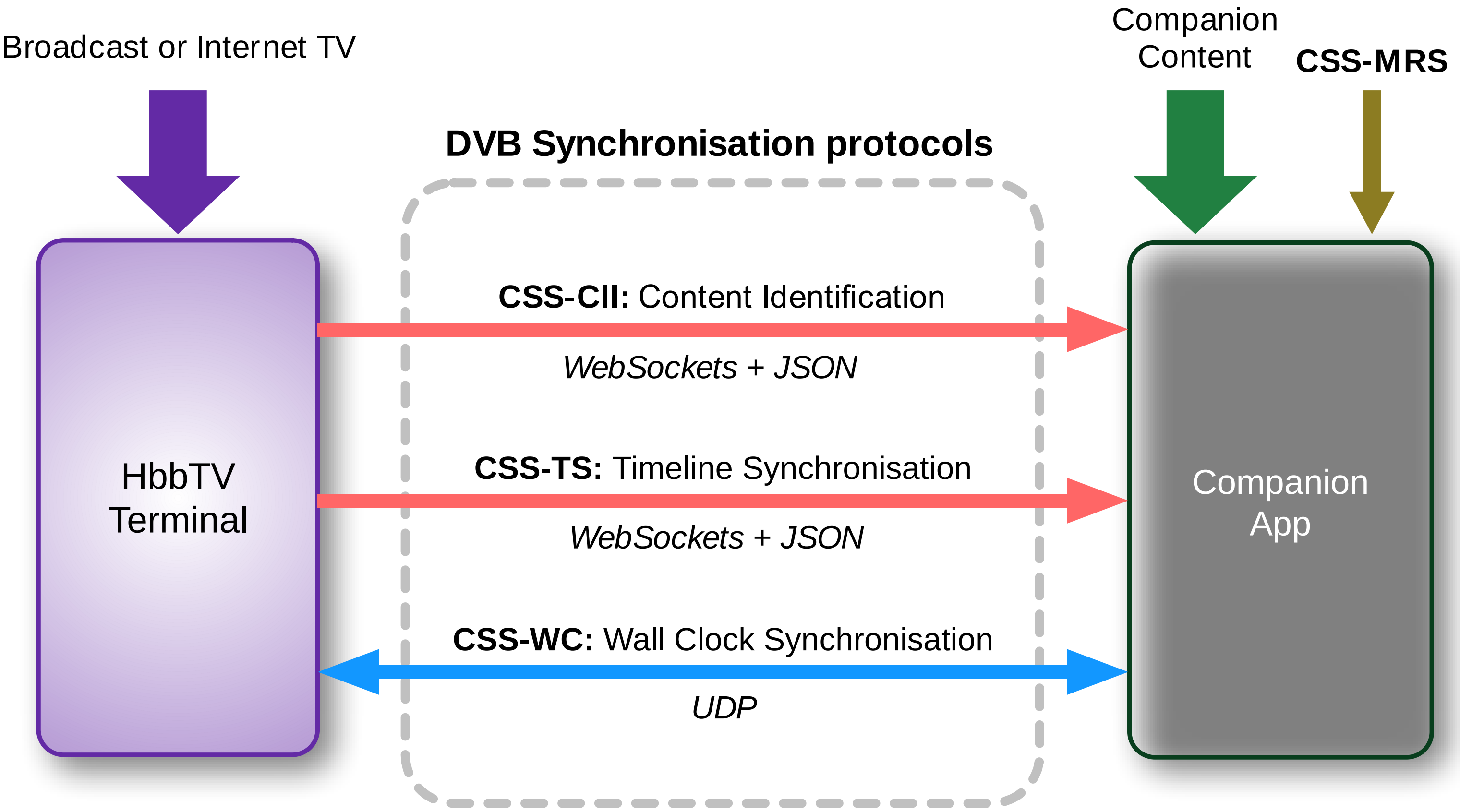
- W3C Timing Object spec (Multi-Device Timing CG)

Companion Screen App Synchronisation


Material Resolution Service (MRS)
Timing information
Broadcaster
Content Identifier
(CSS-MRS)
Material Information on structure of broadcast and relationship to Timelines
Internet
Identifiers, Timelines and MRS location
Broadcast or Internet
Companion Screen Application (CSA)
Content Identification and other Information (CSS-CII)
Wall Clock Synchronisation (CSS-WC)
Timeline Synchronisation (CSS-TS)
Trigger Events Notification (CSS-TE)
Home Network
TV Device

BBC | Research & Development

Inter-device synchronisation – DVB CSS protocols



Broadcast or Internet TV

Companion Content

CSS-MRS

DVB Synchronisation protocols

HbbTV Terminal

CSS-CII: Content Identification

WebSockets + JSON

CSS-TS: Timeline Synchronisation

WebSockets + JSON

CSS-WC: Wall Clock Synchronisation

UDP

Companion App

BBC | Research & Development

Media Synchronisation Terminology

A **Media object** is the DOM element for the (broadcast or streamed) media being presented

Master media is the media (e.g. broadcast) that we are synchronising to

Other media is played back in sync with **master media**

A **Timeline Selector** specifies how to derive the timeline for a piece of media (broadcast, DASH, ISOBMFF, …)

`urn:dvb:css:timeline:temi:11:1`	MPEG TEMI broadcast timeline carried in component 11
`urn:dvb:css:timeline:mpd:period:rel:1000`	Time since start of MPEG DASH stream, measured in milliseconds

A **Correlation** describes how to align **other media** to **master media**

API lifecycle

1. Create and initialise a **Media Synchroniser** object:
    - Select a media object (e.g. video/broadcast object) as the **master**
    - **Select the timeline** to use from the master media

2. Use as required, <u>in any combination</u>:
    - Query the **current time** (application sync)
    - Enable/disable **inter-device sync**
    - Start/stop synchronising **other media** to the master (multi-stream sync)

3. Change of **master** media (switch between broadcast & IP, or changed IP stream)
    - Discard existing Media Synchroniser and initialise a new one

Create and initialise to nominate **master** media

```
<object type="video/broadcast" id="vb" />
<object type="application/hbbtvMediaSynchroniser" id="ms" />
```

Ensure the video/broadcast object is *bound* and ready in the “presenting” state:

```
vb.bindToCurrentChannel();
vb.onPlayStateChange = function() {
    if (vb.playState == 2) { ...
```

… then initialise, setting broadcast video as the **master media**, using a TEMI timeline:

```
ms.initMediaSynchroniser(vb, "urn:dvb:css:timeline:temi:11:1");
```

Application synchronisation

Get the current timeline position of the **master media**, in seconds:

```
    var secs = ms.currentTime;

    console.log("Current TEMI timeline position:", secs);
```

Note: This is the current time on the timeline specified by the **Timeline Selector** for the **master media.**

It is not the same as the `currentTime` property on a `<video>` or `<audio>` or A/V `<object>`

Inter-device synchronisation

Enable/disable synchronisation for companion devices:

```
    ms.enableInterDeviceSync(function() { console.log("enabled!"); });
    ms.disableInterDeviceSync(function() { console.log("disabled!"); });
```

Starts/stops the service in the TV that implement the DVB CSS protocols for synchronisation

References

- HbbTV 2.0.1 (ETSI TS 102 796 V1.4.1)

  http://hbbtv.org/resources

- DVB Companion Screens and Streams; Part 2: Content Identification and Media Synchronization
  (ETSI TS 103 286-2 V1.2.1)

  https://www.dvb.org/standards/dvb_css

- Timing Object

  http://webtiming.github.io/timingobject/

Thank you

bbc.co.uk/rd

Email:

chris.needham@rd.bbc.co.uk

BBC | Research & Development