HBBTV2.0 CS AND MEDIA SYNCHRONIZATION IMPLEMENTATION LESSONS LEARNED

Louay Bassbouss, HbbTV Symposium 2016, Madrid

Fraunhofer FOKUS

Engineering a Connected World

AGENDA

1. HbbTV 2.0 CS Features Overview
2. HbbTV 2.0 CS Open Source Implementation
3. Lessons Learned
4. Use Cases and Demos
5. Fallback solution for HbbTV 1.x Terminals
6. HbbTV 2.0 CS & W3C Presentation API

HBBTV 2.0 CS FEATURES OVERVIEW

- Launch HbbTV App from CS App
  - Technologies: DIAL (SSDP Discovery)
- Communication between HbbTV and CS Apps
  - Technologies: WebSockets
- Launch CS App from HbbTV App
  - Technologies: Proprietary
- Multi Device Synchronization
  - Technologies: DVB-CSS

USAGE OF HBBTV 2.0 CS APIS

HbbTV App

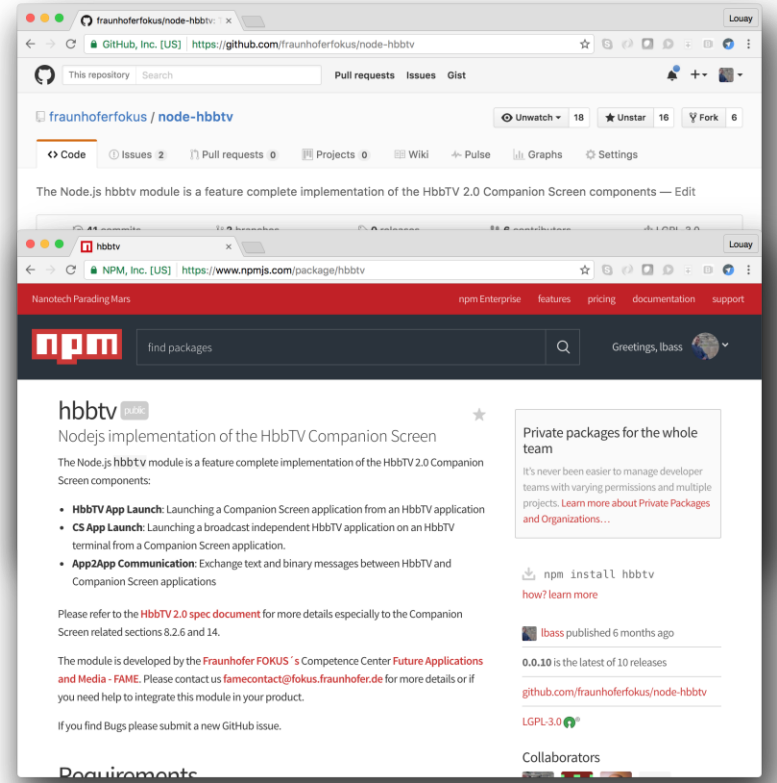
- Use *HbbTVCSManager* embedded object to discover CS Launchers, other HbbTV Terminals or launch CS App
  - *discoverCSLaunchers(function onCSDiscovery)*
  - *discoverTerminals(function onTerminalDiscovery)*
  - *launchCSApp(Number enum_id, String payload, function onCSLaunch)*
- Use *MediaSynchroniser* embedded object for Multi-stream or Multi-Device Synchronization.
  - *initMediaSynchroniser(video, timeline)* → Master
  - *initSlaveMediaSynchroniser(cssCiiUrl)* → Slave
  - *enableInterDeviceSync()*
- Use W3C WebSockets API for App2App Com.
  - *new WebSocket(app2appLocalUrl);*

CS App

- Native App:
  - Use *DIAL* Client Lib for target Platform (Android, iOS, …) and launch App with identifier "*HbbTV*" (*POST* request contains *XML AIT* describing the target HbbTV App)
  - Use *WebSocket* Lib for target platform and connect to *app2appRemoteUrl*
  - Use DVB-CSS Client Lib for target platform and connect to *cssCiiUrl* of master Terminal
- Web App:
  - DIAL not supported (Discovery requires *UDP*)
  - Use *W3C WebSockets API* for App2App com.
  - *DVB-WC* (Wall Clock) requires also UDP

HBBTV 2.0 CS OPEN SOURCE IMPLEMENTATION (TERMINAL)

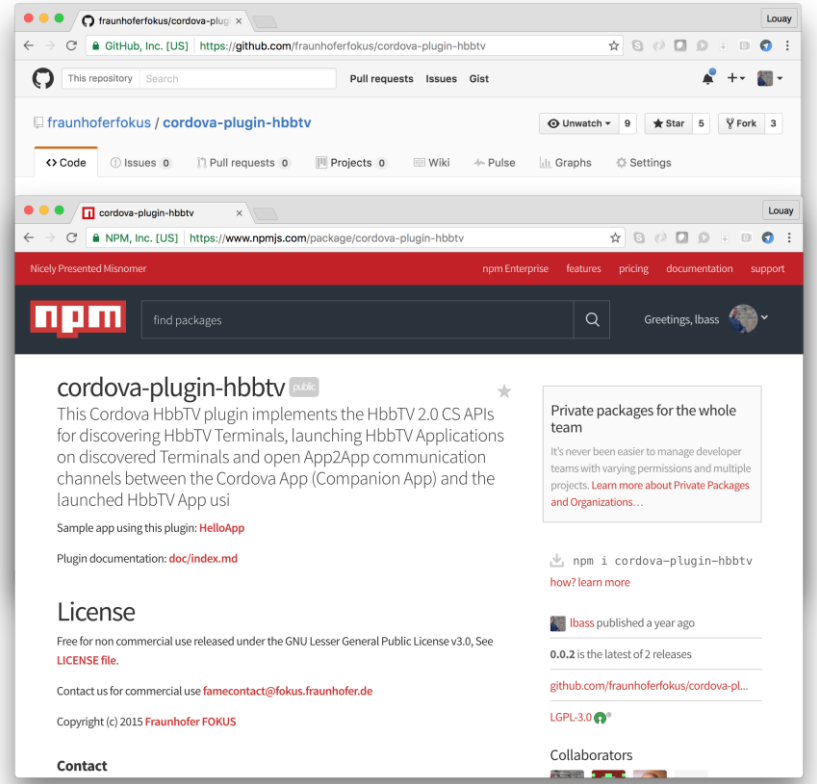
- HbbTV 2.0 CS Implementation as Node.js Module:
  - github: https://github.com/fraunhoferfokus/node-hbbtv
  - npm: https://www.npmjs.com/package/hbbtv
- Uses Fraunhofer Open Source DIAL and SSDP Node Modules:
  - DIAL: https://github.com/fraunhoferfokus/peer-dial
  - SSDP: https://github.com/fraunhoferfokus/peer-ssdp
- CLI with two modes:
  - Terminal mode: `$ hbbtv -m terminal -p 8080`
  - Terminal-mode components: DIAL Server/Client, App2App Server, DVB-CSS Server/Client (Client for Slave Terminal)
  - CS mode: `$ hbbtv -m cs -p 8090`
  - CS-mode components: CSLauncher, DIAL Client, DVB-CSS Client
- Can be used with any User Agent (e.g. Firefox with FireHbbTV Add-On)



Fraunhofer FOKUS

HBBTV 2.0 CS OPEN SOURCE IMPLEMENTATION (CORDOVA)

- HbbTV 2.0 CS Implementation of CS components as Cordova Plugin:
  - github: https://github.com/fraunhoferfokus/cordova-plugin-hbbtv
  - npm: https://www.npmjs.com/package/cordova-plugin-hbbtv
- Supported Platforms: Android, iOS (coming soon)
- Components:
  - DIAL Client (includes SSDP Client)
  - Helpers for XML AIT POST requests
  - DVB-CSS Client (coming soon)



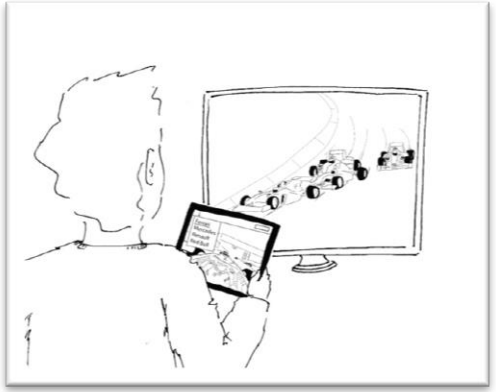
Fraunhofer FOKUS

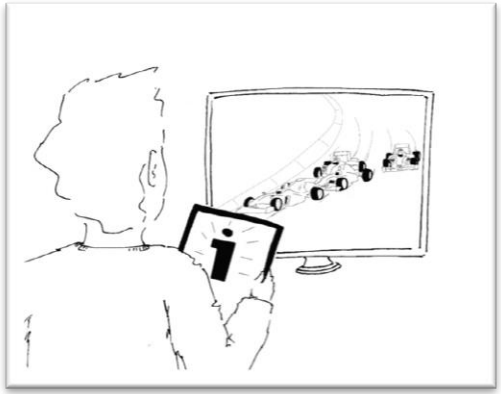
LESSONS LEARNED

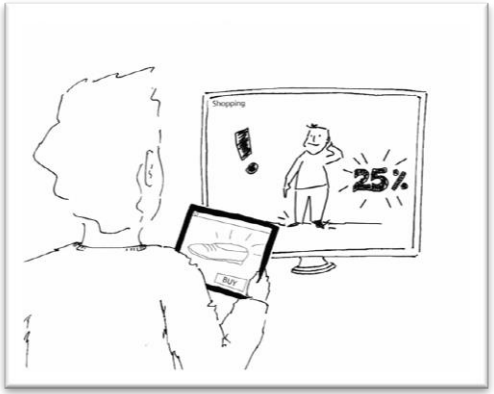
- Feature-complete: any multiscreen scenario can be realized using the HbbTV2.0 CS specification at least with native CS Apps
- Addressing existing technologies: WebSockets, DIAL, …. → Existing Implementations can be reused.
- Best practices and guidlines for developing HbbTV Apps with CS support will help to improve usability. e.g. use same RC button for launching CS App (similar to Red Button).
- Not clear how to use HTMLVideoElement on CS for Multi-device synchronization → W3C Multi-Device Timing could be relevant
- Not clear how to launch HbbTV from a Web App on CS → W3C Presentation API could be relevant (Already discussed in W3C Second Screen WG in this GitHub issue https://github.com/w3c/presentation-api/issues/67).

USE CASES

- Cast Media to Broadcaster's HbbTV Application
- Show additional Content on Companion Screens
- Multiple Camera Perspectives
- Personalized Audio Tracks on Companion Screens
- Multi-User Support
  - Multiplayer Games
  - Quizzes
  - …
- Multiscreen Advertisement

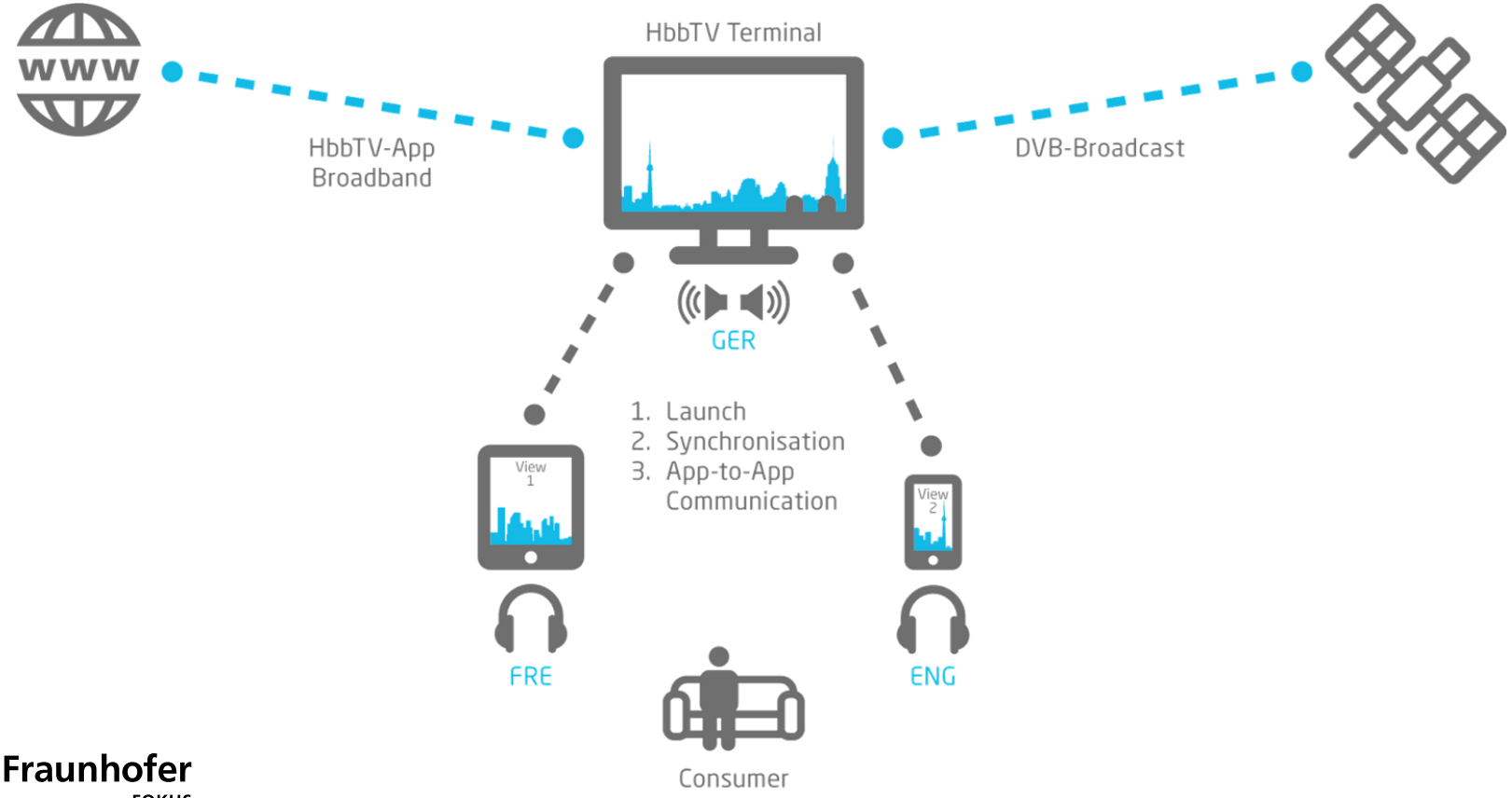






Fraunhofer FOKUS

DEMO: PERSONALIZED VIDEO AND AUDIO STREAMS ON CS

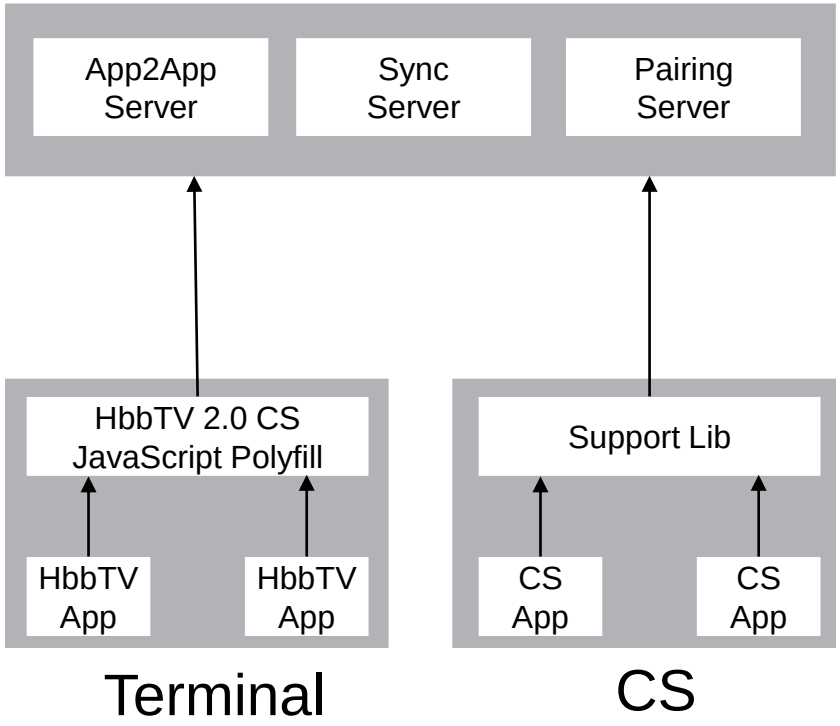


DEMO: MAIN CAMERA ON TV IN SYNC WITH 360° VIDEO ON CS

DEMO: SILENT VIDEO ON TV IN SYNC WITH AUDIO ON CS

FALLBACK SOLUTION FOR HBBTV 1.X TERMINALS

- HbbTV 2.0 CS JavaScript Polyfill on Terminal → support same APIs as in HbbTV 2.0
- App2App Server in the Cloud. Fallback to XHR+Polling if WebSocket is not supported (WebSocket is activated on most new TV models)
- Sync Server in the Cloud → Synchronization may not be accurate as for HbbTV 2.0 Terminals
- Manual pairing (QR-code, PIN, …) instead of Discovery. Pairing token can be stored on the Terminal as cookie.
- During pairing, the Server may also store a Push Notification Token (e.g. APN, GCM) that can be used to launch the CS App



HBBTV 2.0 CS & W3C PRESENTATION API

- The W3C Second Screen WG works on a specification “Presentation API” with similar features as in HbbTV 2.0 CS → https://www.w3.org/TR/presentation-api/
- Goal: defines an API to enable web content to access external presentation-type displays and use them for presenting web content.
- W3C Candidate Recommendation 14 July 2016 → expect Proposed Recommendation by end of Q1 2017
- Two Implementations: Chrome (Desktop and Android) & Firefox (Desktop and Android)
- Different terminologies: First Screen is mobile device, Second Screen is Presentation Display.
- Specifies only a Browser API but not the underling protocols. Browser vendors may implement the API on top of different protocols (e.g. Chrome implementation supports Google Cast).
- Support of HbbTV 2.0 CS is already discussed and is labeled as v2 Feature → https://github.com/w3c/presentation-api/issues/67

MORE DEMOS

- Please visit our booth:
  - HbbTV 2.0 CS and Media Synchronization
    https://www.fokus.fraunhofer.de/go/hbbtv
  - Cloud-based 360° Video Playout for HbbTV
    https://www.fokus.fraunhofer.de/go/360
  - HbbTV Application Toolkit
    https://www.fokus.fraunhofer.de/go/hat

Fraunhofer FOKUS

CONTACT

Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany

www.fokus.fraunhofer.de

Louay Bassbouss

Senior Project Manager R&D

louay.bassbouss@fokus.fraunhofer.de

Tel. +49 (0)30 3463-7275