# OBS HbbTV 2.0.4 TMs: Partial Implementations

# HbbTV 2.0.4 Test Materials - Implementations

- Partial Implementations for acceleration of test case acceptance
  - Serve as a reference for real device implementations

- OBS has provided partial implementations for:
  - Voice Control
  - Accessibility
  - DVB-I

- The implementations are based on the Open Red Button, open source HbbTV stack, Android reference integration from OBS.
  - This employs the Android system WebView and DTVKit DVB stack

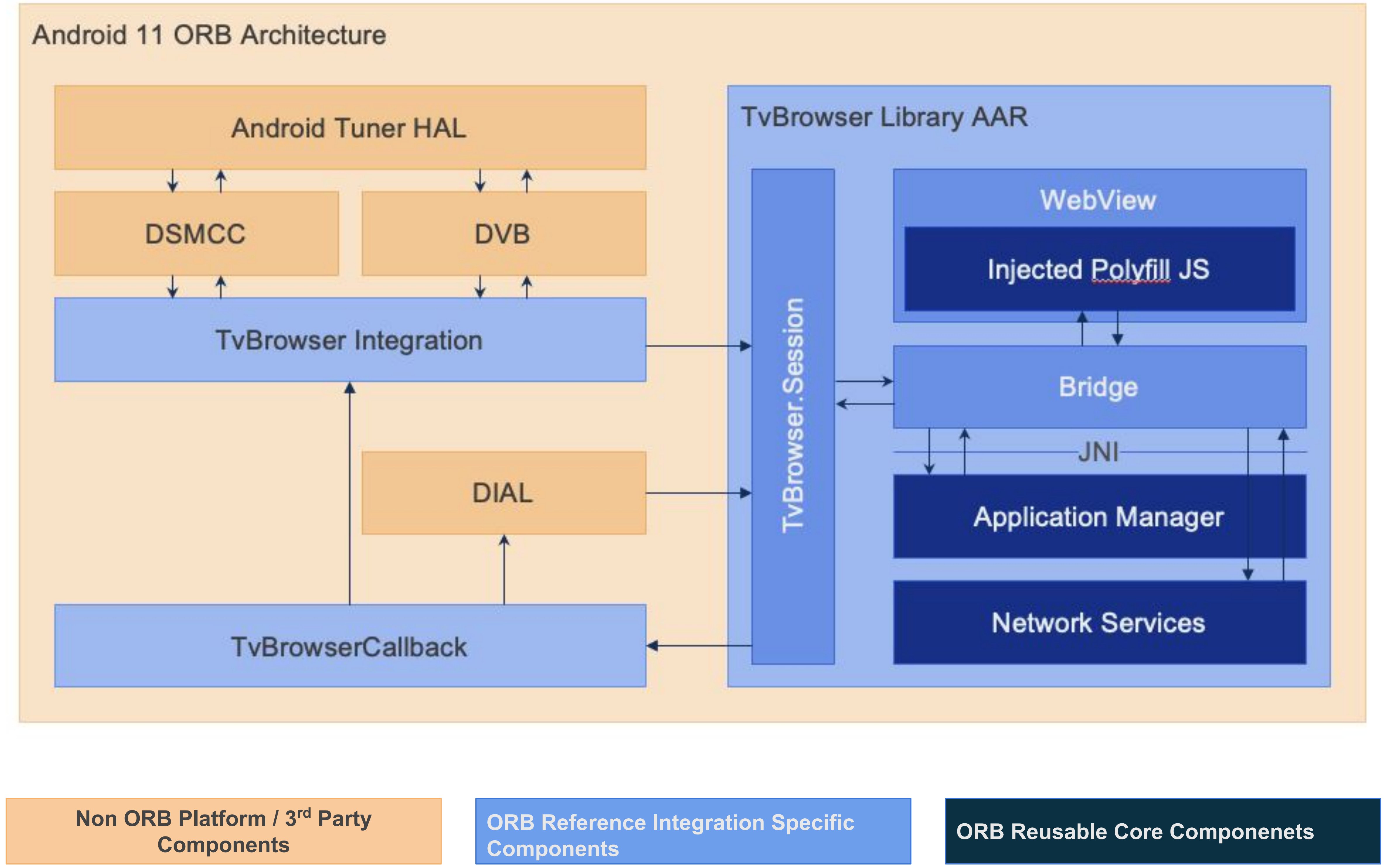
- All code for the HbbTV 2.0.4 enhancements are available in the ORB repository.
  - Additional system components are available in library form and as part of an emulator image.

- The ORB Android reference is executed on both the Android Emulator and reference hardware and used to validated the associated test packages.

# ORB HbbTV Solution

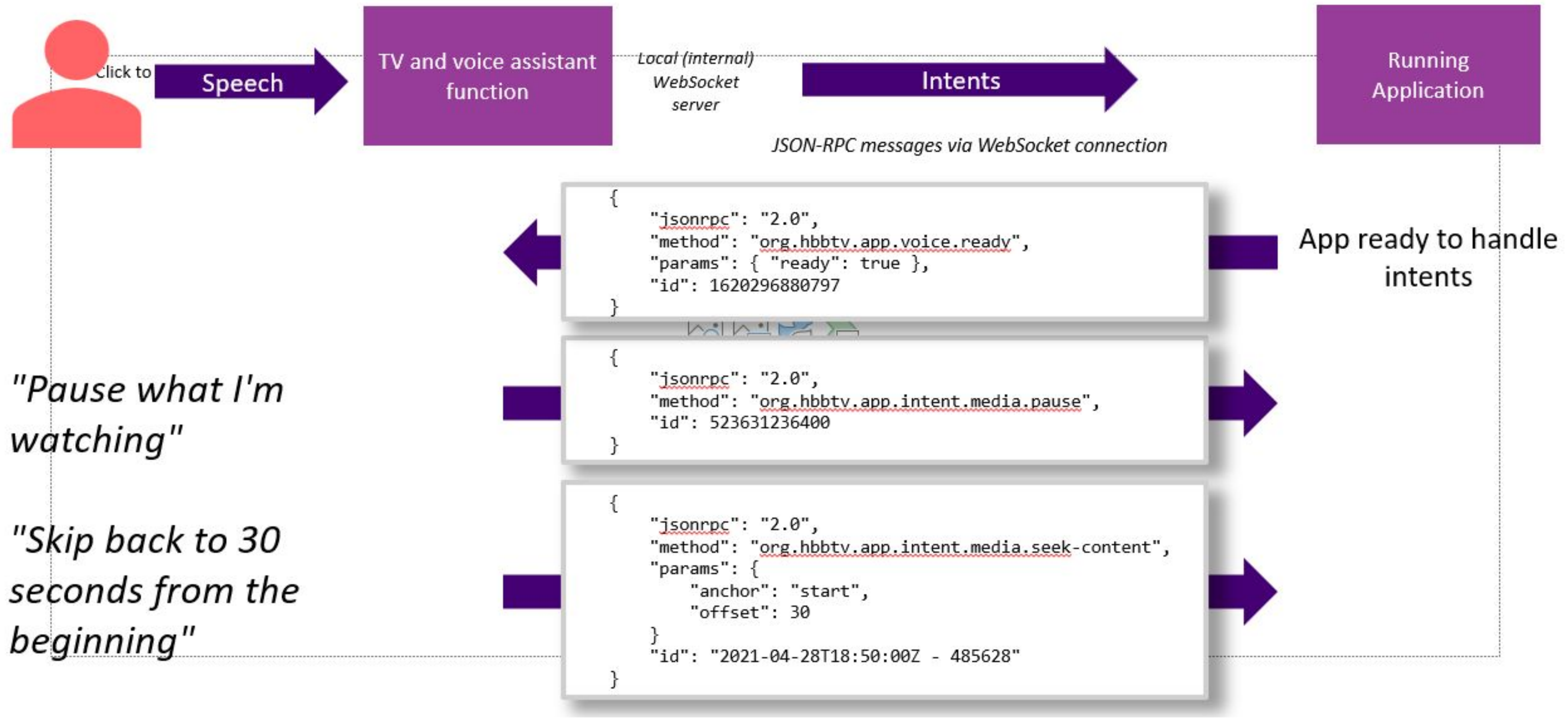
## Cross Platform Resuable Core

- Injected JavaScript polyfill extending the browser frame for HbbTV
- Default JavaScript behaviour overridden where needed
- HbbTV video object polyfilled to use the browsers native HTML5 Video object for a fast start – can easily be replaced
- Other features fully in JavaScript include TTML subtitle decoder
- Small number of reusable C++ components (Application manager, parts of media synchroniser)



# Voice Control

- Extended ORB with
  - ORB was extended with websocket communication API to enable communication with the terminal
  - Used AWS for voice recognition but could be changed for an alternative
  - Text to speech was not implemented but output is echoed to the console.


Click to
Speech
TV and voice assistant function
Local (internal) WebSocket server
Intents
Running Application
JSON-RPC messages via WebSocket connection
App ready to handle intents

```
{
    "jsonrpc": "2.0",
    "method": "org.hbbtv.app.voice.ready",
    "params": { "ready": true },
    "id": 1620296880797
}
```

"Pause what I'm watching"

```
{
    "jsonrpc": "2.0",
    "method": "org.hbbtv.app.intent.media.pause",
    "id": 523631236400
}
```

"Skip back to 30 seconds from the beginning"

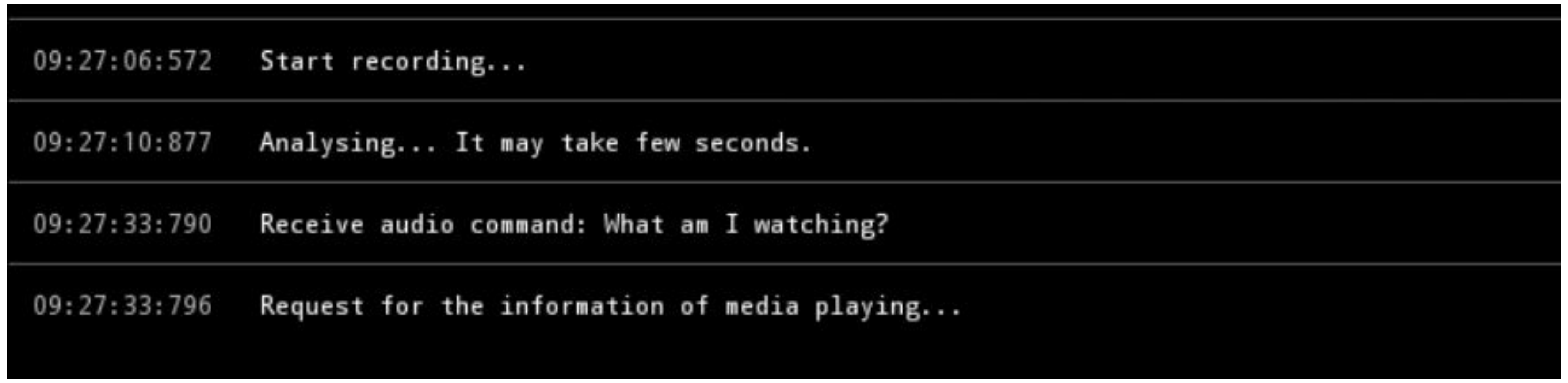
```
{
    "jsonrpc": "2.0",
    "method": "org.hbbtv.app.intent.media.seek-content",
    "params": {
        "anchor": "start",
        "offset": 30
    }
    "id": "2021-04-28T18:50:00Z - 485628"
}
```

## Messages exchanged

| Terminal -> App intent | App -> Terminal messages |
|---|---|
| media.pause | voice.ready |
| media.play | state.media |
| media.fast-forward | |
| media.fast-reverse | |
| media.stop | |
| media.seek-content | |
| media.seek-relative | |
| media.seek-live | |
| media.seek-wallclock | |
| media.search | |
| display* | |
| playback* | |

## Other Voice Triggered Activity

- Remote control key presses
- Text field text input



```
09:27:06:572   Start recording...

09:27:10:877   Analysing... It may take few seconds.

09:27:33:790   Receive audio command: What am I watching?

09:27:33:796   Request for the information of media playing...
```

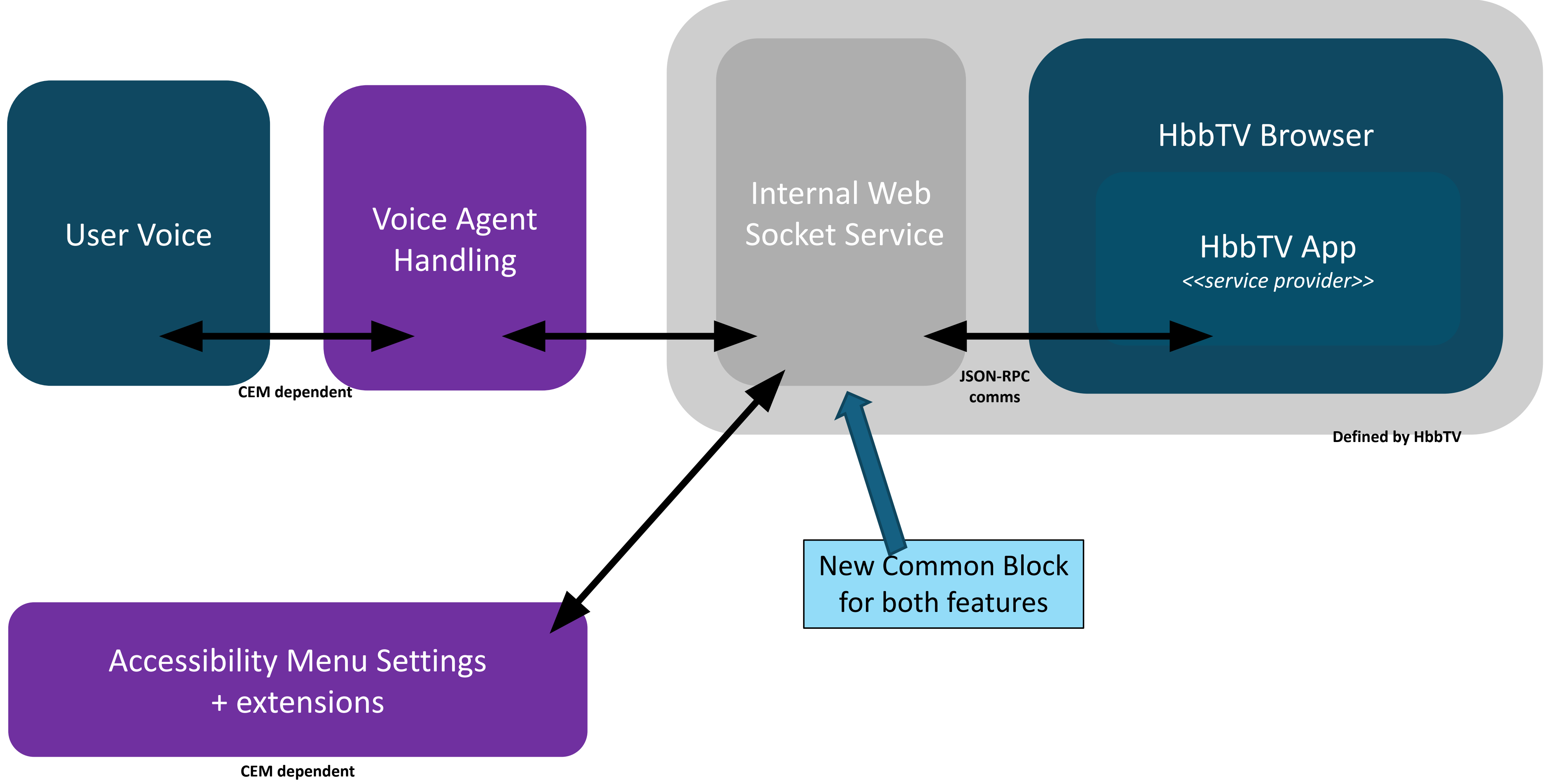
# Accessibility

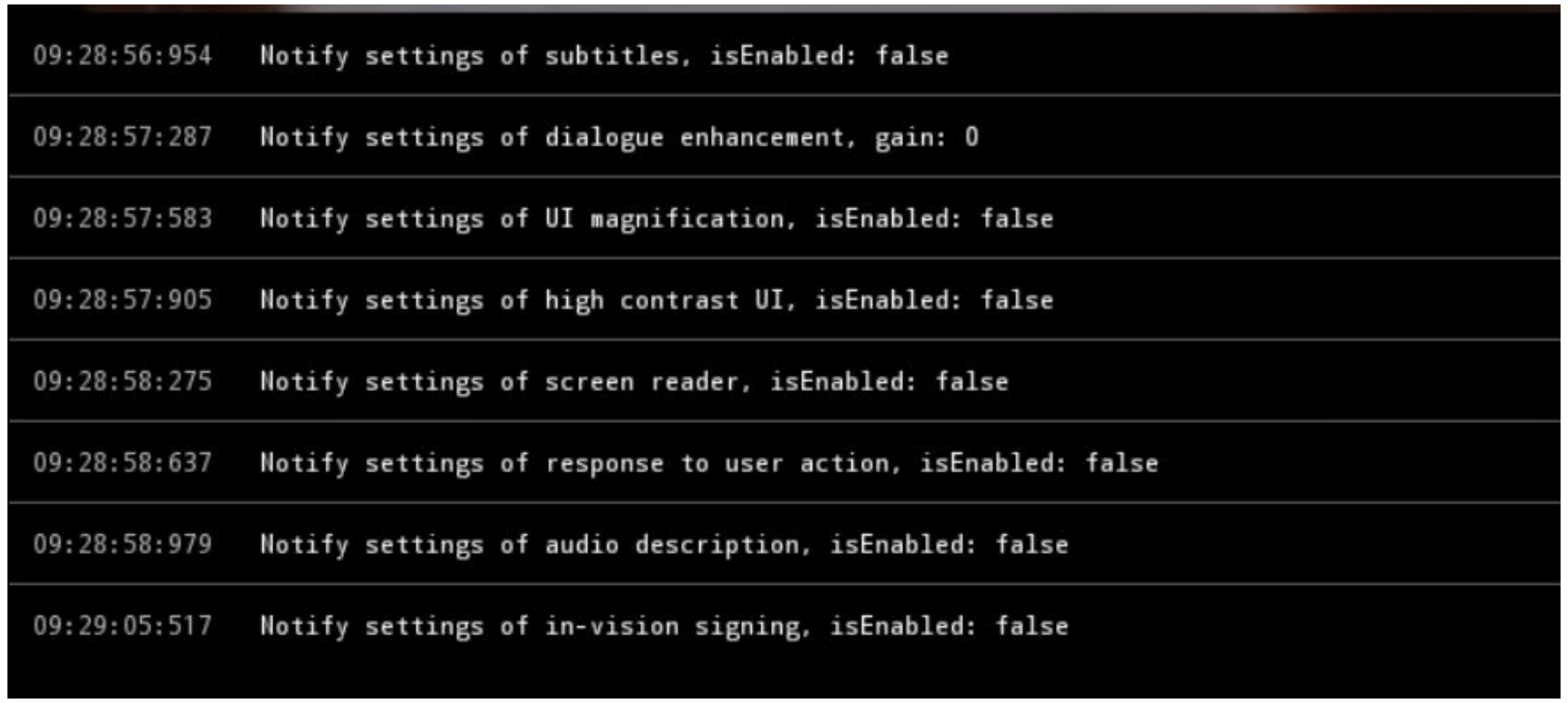
- Supports 8 core features:
  - Subtitles
  - Dialogue Enhancement
  - Magnification UI
  - High Contrast UI
  - Screen Reader
  - Response to a User Action
  - Audio Description
  - In Vision Sign Language

- ORB was extended with websocket communication API to enable communication with the terminal.

- A simple ‘TV interface’ was implemented to exercise the APIs.

- The ‘TV Interface’ supports 3 requests from apps
  - featureSupportInfo
  - featureSuppress
  - featureSettingsQuery

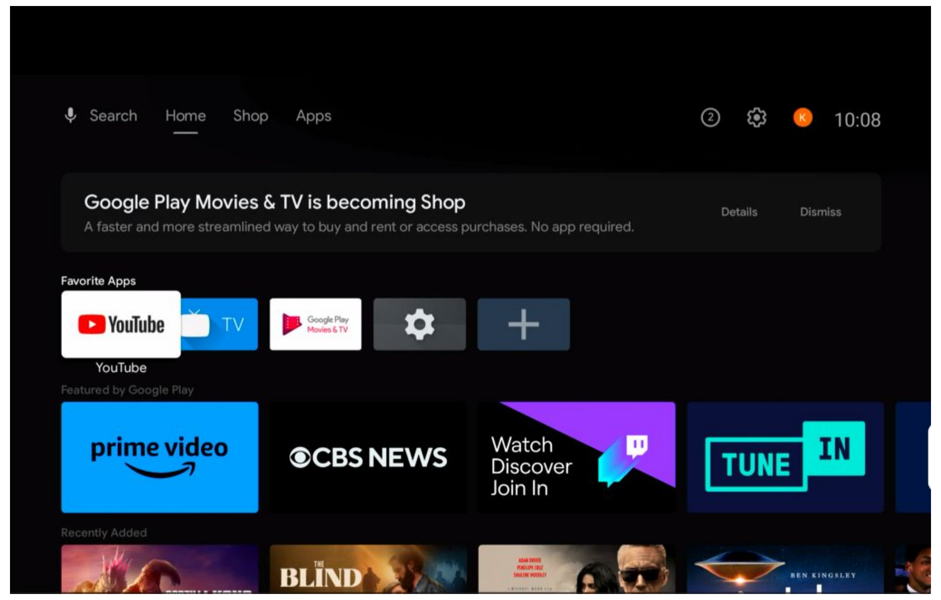


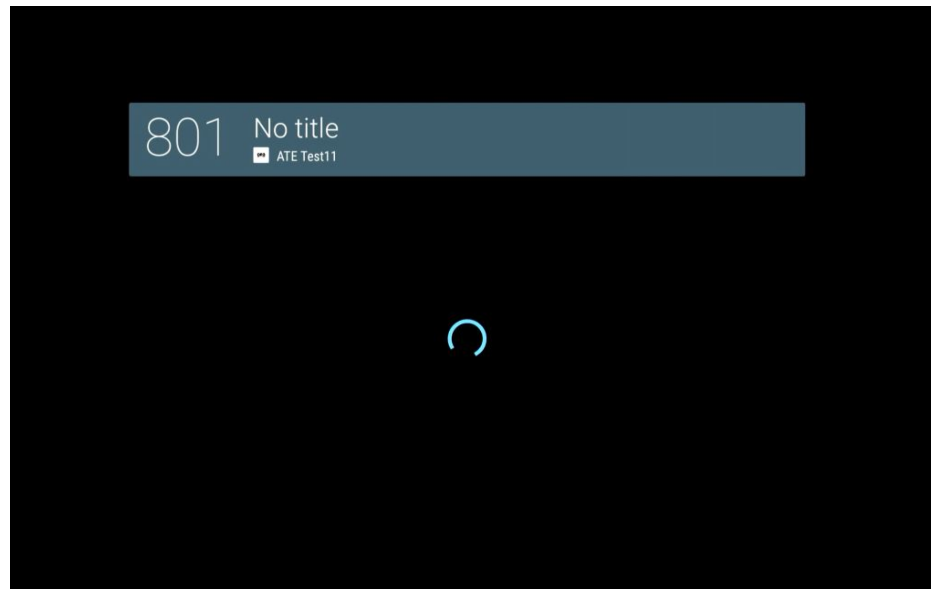


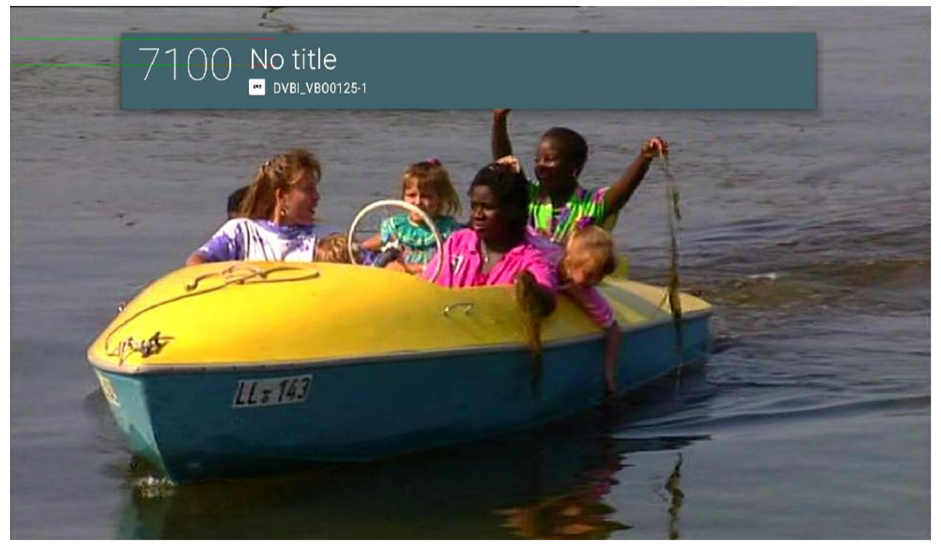
```
09:28:56:954   Notify settings of subtitles, isEnabled: false
09:28:57:287   Notify settings of dialogue enhancement, gain: 0
09:28:57:583   Notify settings of UI magnification, isEnabled: false
09:28:57:905   Notify settings of high contrast UI, isEnabled: false
09:28:58:275   Notify settings of screen reader, isEnabled: false
09:28:58:637   Notify settings of response to user action, isEnabled: false
09:28:58:979   Notify settings of audio description, isEnabled: false
09:29:05:517   Notify settings of in-vision signing, isEnabled: false
```

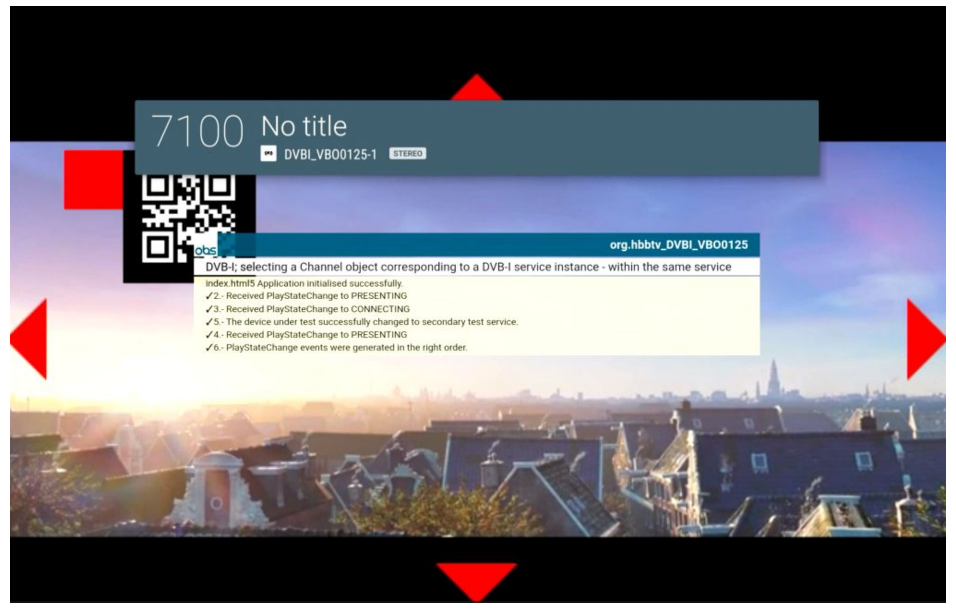
# DVB-I

- Updated VBO in ORB to support required DVB-I features:
  - Broadband playback states
  - Service Instances
- Extended ORB
  - 2nd browser instance for broadband playback
- Implemented a simple DVB-I client to exercise features:
- Validated with associated test cases (OBS) and streams (Sofia Digital)





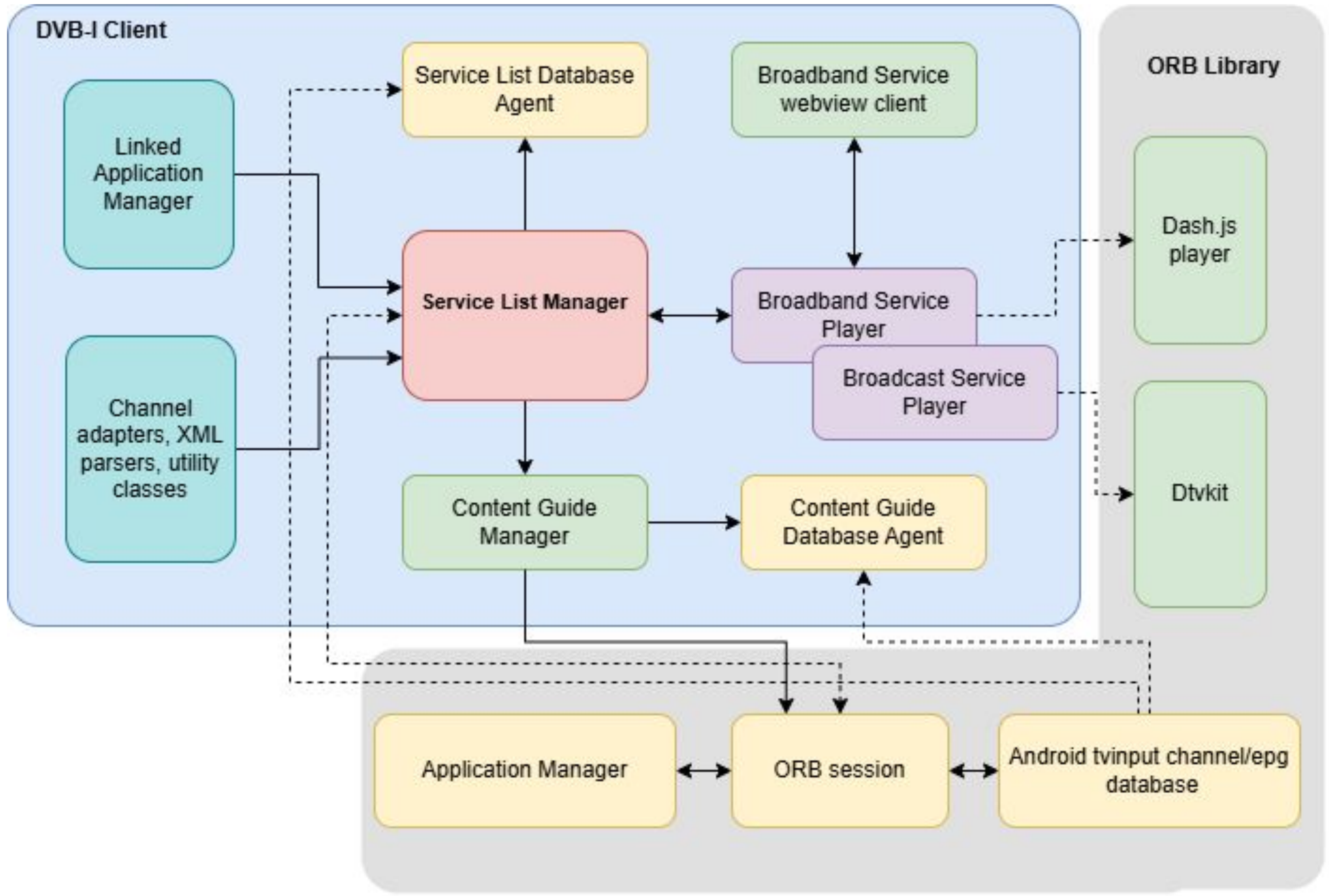




org.hbbtv_DVBI_VBO0125

DVB-I; selecting a Channel object corresponding to a DVB-I service instance - within the same service

index.html5 Application initialised successfully.
✓2.- Received PlayStateChange to PRESENTING
✓3.- Received PlayStateChange to CONNECTING
✓5.- The device under test successfully changed to secondary test service.
✓4.- Received PlayStateChange to PRESENTING
✓6.- PlayStateChange events were generated in the right order.



DVB-I Client
Linked Application Manager
Channel adapters, XML parsers, utility classes
Service List Database Agent
Service List Manager
Content Guide Manager
Broadband Service webview client
Broadband Service Player
Broadcast Service Player
Content Guide Database Agent
ORB Library
Dash.js player
Dtvkit
Application Manager
ORB session
Android tvinput channel/epg database

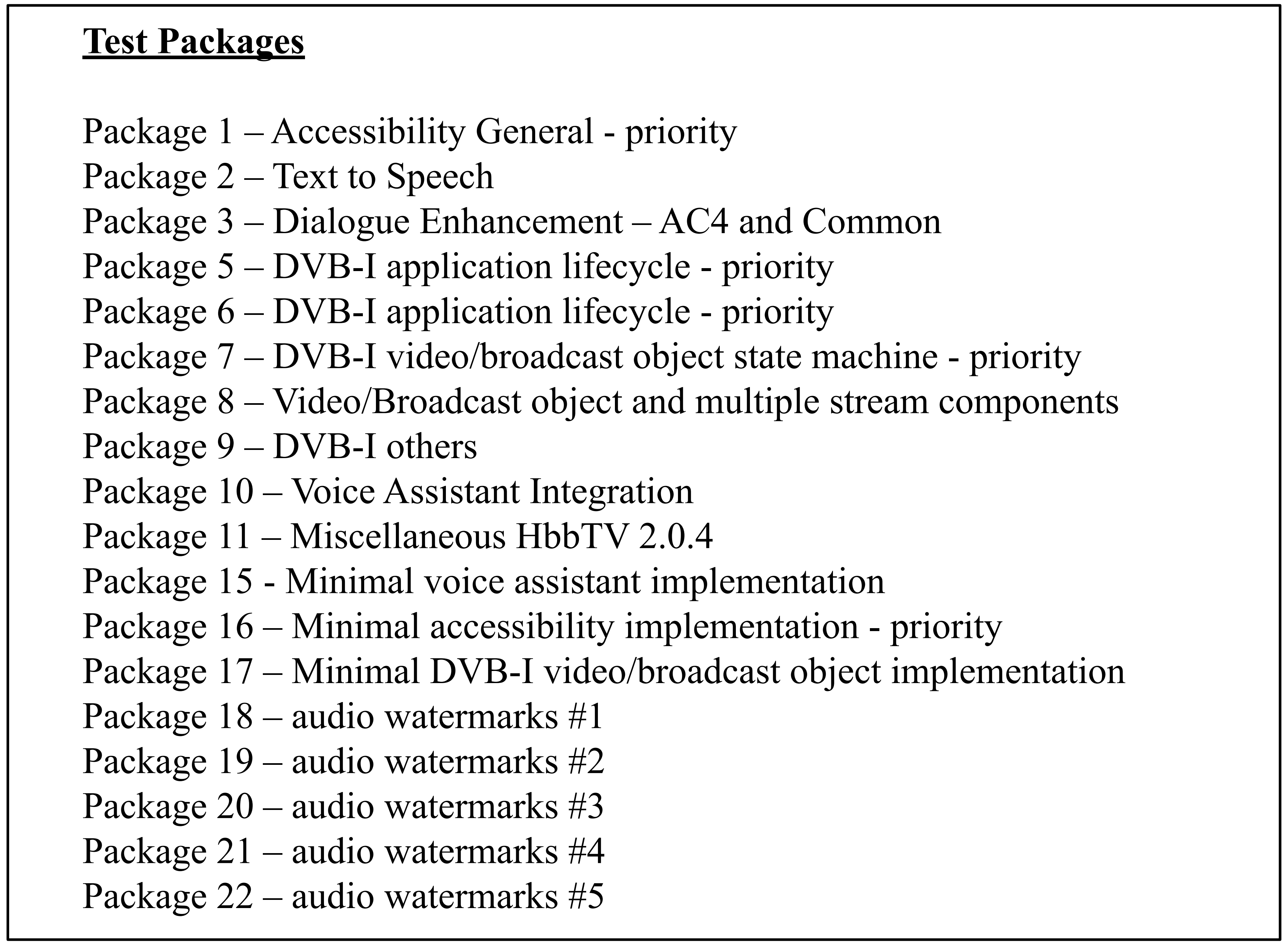
- **Linked application manager**: Where a service includes a linked application, this is responsible for identifying if (a version of) the application can be presented and if so, interfacing to the appropriate engine to make the presentation happen. Note that some services may require a linked application to be started before the video and audio of the service are presented.
- **Service list manager**: This is responsible for discovering and querying service list server(s) and handling the service list(s) that are returned. When a DVB-I service is selected, it is responsible for instructing the service player to play the service.
- **Content guide manager**: This is responsible for accessing content guide server(s) and handling the content guide data that is returned. There is no assumption that this caches content guide data on the DVB-I client in the same way that content guide data would be cached in a broadcast device.
- **Broadband service player**: This is responsible for the complete lifecycle of playback of a service delivered on a broadband network. It controls the DVB-DASH player, any secondary OTT players and the linked application manager as appropriate. For services where the media content is described by a playlist, this is responsible for processing the playlist.

**DVB-I Test Case**

- Changing instances within the same channel
- The harness starts at channel 801 as usual, then tunes to the channel 7100 (hbbtv_VBO125).
- Then the loaded hbbtv app, finds the dash (broadband) instance and tunes to it (cartoon).
- The test checks that the correct (dash) instance is selected and tuned based on video events.

# Test Cases & Harness Changes

- Close co-operation with Sofia Digital on DVB-I test assets.
- Audio Watermarking test materials with support from Verance on streams.
- Created a simple tool for generating the Accessibility Result Declaration
  - Accessibility tests rely on this to know what the expected result is.



**Test Packages**

Package 1 – Accessibility General - priority
Package 2 – Text to Speech
Package 3 – Dialogue Enhancement – AC4 and Common
Package 5 – DVB-I application lifecycle - priority
Package 6 – DVB-I application lifecycle - priority
Package 7 – DVB-I video/broadcast object state machine - priority
Package 8 – Video/Broadcast object and multiple stream components
Package 9 – DVB-I others
Package 10 – Voice Assistant Integration
Package 11 – Miscellaneous HbbTV 2.0.4
Package 15 - Minimal voice assistant implementation
Package 16 – Minimal accessibility implementation - priority
Package 17 – Minimal DVB-I video/broadcast object implementation
Package 18 – audio watermarks #1
Package 19 – audio watermarks #2
Package 20 – audio watermarks #3
Package 21 – audio watermarks #4
Package 22 – audio watermarks #5

# Emulator Image

- The code for the partial implementations is available from the ORB repos: https://github.com/OpenRedButtonProject/Orb
- A simple way to try it out is to use the ORB emulator image
  - This combines the ORB HbbTV solution with the DTVKit DVB stack running in the Android emulator.
  - Essentially a complete Android TV on your desktop
  - Drag and drop TS files, perform a channel scan and interact with HbbTV Apps as you would on a normal TV with the added benefit of full access to the chromium web developer tools.
- For access to the ORB Emulator please send an email to info@orb-tv.com