Application Discovery over Broadband Explained
Contents

• Application Discovery over Broadcast in Regular HbbTV
• Application Discovery over Broadband
  – Phase 1 – DVB-SI
  – Phase 2 – Watermarking
• Specification Overview
• Looking Forward
Traditionally, HbbTV Terminals “discover” applications using an Application URL carried in an Application Information Table (AIT) received via broadcast.

- Timecode and stream events are found multiplexed into the broadcast stream.
- This is supported in the HbbTV Core Specification (and referenced DVB specs).

The diagram illustrates the distribution of audio, video, timecode, and event information through the MUX and distribution network, leading to the terminal and app server.
• Some broadcast multiplexes don’t include the AIT, timecode, or stream events
• The **Application Discovery over Broadband** specification enables HbbTV Terminals to obtain the AIT and application from broadband in this case using the **DVB Service Information** (DVB-SI) that is always present in a DVB transmission.
Watermarking extends use cases for Application Discovery over Broadband to include:

- Service delivery to the HbbTV Terminal via **HDMI and other non-broadcast interfaces**
- **Timecode**
- **Stream Events**
Specification Overview
Independent Specification

• HbbTV has 2 types of specifications
  – The core HbbTV spec itself (ETSI TS 102 796)
  – A set of related independent specs
• Independent specs are for features not (yet) ready for mass-market adoption in TV sets
  – Operator applications (ETSI TS 103 606)
  – IPTV (ETSI TS 103 555)
  – Application discovery over broadband (ETSI TS 103 464)
  – Targeted advertising
• In time, some of these may be adopted widely enough to justify inclusion in the core spec
Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Scope</td>
<td>2</td>
</tr>
<tr>
<td>References</td>
<td>3</td>
</tr>
<tr>
<td>Definitions and abbreviations</td>
<td>4</td>
</tr>
<tr>
<td>Overview</td>
<td>5</td>
</tr>
<tr>
<td>HbbTV Application Discovery over Broadband</td>
<td>6</td>
</tr>
<tr>
<td>Service and application model</td>
<td>7</td>
</tr>
<tr>
<td>Formats and protocols</td>
<td>8</td>
</tr>
<tr>
<td>Browser application environment</td>
<td>9</td>
</tr>
<tr>
<td>System integration</td>
<td>10</td>
</tr>
<tr>
<td>Capabilities</td>
<td>11</td>
</tr>
<tr>
<td>Security</td>
<td>12</td>
</tr>
<tr>
<td>Privacy</td>
<td></td>
</tr>
<tr>
<td>Annex A (normative): OIPF specification profile</td>
<td></td>
</tr>
<tr>
<td>Annex B (normative): Electronic attachments</td>
<td></td>
</tr>
<tr>
<td>Annex C (informative): Sequence diagrams</td>
<td></td>
</tr>
</tbody>
</table>
Application discovery lifecycle follows a state machine based on detection and loss of audio and video watermarks.

An audio watermark must be present for an application to be launched.

- Video watermarks provide supplemental timing and stream event information and enable lifecycle management during audio interruptions (e.g. audio mute).

The running application is hidden within 2 seconds when watermarks are lost.

- The hidden application is killed when watermarks indicate a channel change or, in any case, after 2 minutes.

Video watermark embedding and detection are not currently mandatory.

- Use of audio and video watermarks in combination enables improved user experience.
A frame-accurate **watermark media timeline** is established by the Terminal using timing information conveyed in the audio and video watermarks.

Applications can reference and access the watermark media timeline using the Media Synchroniser:

- The watermark timeline selector is `urn:hbbtv:sync:timeline:wm`.
- This timeline is accessed via `MediaSynchroniser.currentTime()`.
- No synchronisation of other media elements to the watermark media timeline is currently mandated.
Watermarking provides Terminals with two ways of receiving application stream events

- Stream events can be conveyed in the video watermark
  - Applications “listening” to the `urn:hbbtv:streamevent:a336:video` event stream URL receive stream events sent in the video watermark

- Event notifications can be conveyed in the audio watermark
  - Applications “listening” to the `urn:hbbtv:streamevent:a336:audio` event stream URL receive a stream event containing the latest audio watermark payload data whenever the broadcaster toggles the “query_flag” bit in the audio watermark payload
  - This provides a “push notification” capability from broadcaster to application
• Some HbbTV APIs are modified when Application Discovery over Broadband using watermarking is active
• Some existing APIs are not available for use:
  – selectComponent and unselectComponent
  – setChannel / prevChannel / nextChannel
  – Release
• Behavior is extended or changed for some objects and APIs
  – Application
  – Channel
  – getCurrentActiveComponents
  – fullscreen
  – width / height
• HbbTV application developers need to take this into account!
# Summary of Application Discovery Methods

<table>
<thead>
<tr>
<th>PRIMARY USE CASE</th>
<th>APPLICATION DISCOVERY VIA BROADCAST</th>
<th>APPLICATION DISCOVERY VIA BROADBAND USING DVB-SI</th>
<th>APPLICATION DISCOVERY VIA BROADBAND USING WATERMARKING</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TVs and STBs directly connected to a broadcast network carrying HbbTV signaling.</strong></td>
<td>TVs and STBs directly connected to a broadcast network <em>not</em> carrying HbbTV signaling.</td>
<td>TVs connected via HDMI to a STB where the STB does not support HbbTV.</td>
<td></td>
</tr>
<tr>
<td><strong>TERMINAL TYPES</strong></td>
<td>TV STB</td>
<td>TV STB</td>
<td>TV STB</td>
</tr>
<tr>
<td><strong>TERMINAL INPUT SOURCES</strong></td>
<td>DVB IPTV</td>
<td>DVB IPTV</td>
<td>HDMI DVB IPTV</td>
</tr>
<tr>
<td><strong>AIT DISCOVERY</strong></td>
<td>BROADCAST (DVB)</td>
<td>DVB-SI</td>
<td>A/V WATERMARK</td>
</tr>
<tr>
<td><strong>AIT DELIVERY</strong></td>
<td>BROADCAST (DVB)</td>
<td>BRODBAND (HTTPS)</td>
<td>BRODBAND (HTTPS)</td>
</tr>
<tr>
<td><strong>APP DELIVERY</strong></td>
<td>BRODBAND (HTTPS)</td>
<td>BRODBAND (HTTPS)</td>
<td>BRODBAND (HTTPS)</td>
</tr>
<tr>
<td><strong>TIMECODE DELIVERY</strong></td>
<td>BROADCAST (DVB)</td>
<td>NOT SUPPORTED</td>
<td>A/V WATERMARK</td>
</tr>
<tr>
<td><strong>STREAM EVENT DELIVERY</strong></td>
<td>BROADCAST (DVB)</td>
<td>NOT SUPPORTED</td>
<td>VIDEO WATERMARK</td>
</tr>
</tbody>
</table>
Looking Forwards

• Today HbbTV has the spec and 123 approved unit test descriptions

• Next steps
  – Create test suite based on unit tests
  – Review test suite
  – Run test suite on early implementations
  – All of these need resources and / or money!