Companion Screens & Streams

Expanding the TV Experience with Companion Screens

What is DVB-CSS

DVB-CSS enables devices such as tablets, smart phones, and other similar Companion Screens or 'Second Screens' to interact with the TV or set-top box and become part of the overall TV experience. To achieve this, DVB-CSS defines a framework that allows applications on smart devices to construct an integrated, synchronized content experience with the TV over the home network, potentially drawing together many components from multiple sources.

Background

Various bespoke applications have been created to allow smart personal devices to extend, interact and respond to content being watched on the main TV. The wide range of these applications creates a burden on all parties in the chain (content providers and distributors, hardware manufactures and app developers). Against this background, DVB has developed a set of standardized protocols to ease implementation by defining the interface between an application and another display, or the device driving that display.

The desire was to provide reliable, accurate and timely interactions over the home network. To further simplify use and deployment, widely used and supported protocols and environments were favored as the technology building blocks.

How does it work?

Figure 1 shows an architectural overview of the specification. The key non-interface components are the broadcaster that provides content, using any of the standard DVB delivery mechanisms, and protocols to a television device, an application on a Companion Device (e.g. a tablet) and a new service, the MRS (that can be based in the cloud). The dotted interface between the broadcaster and MRS is outside the scope of DVB-CSS.

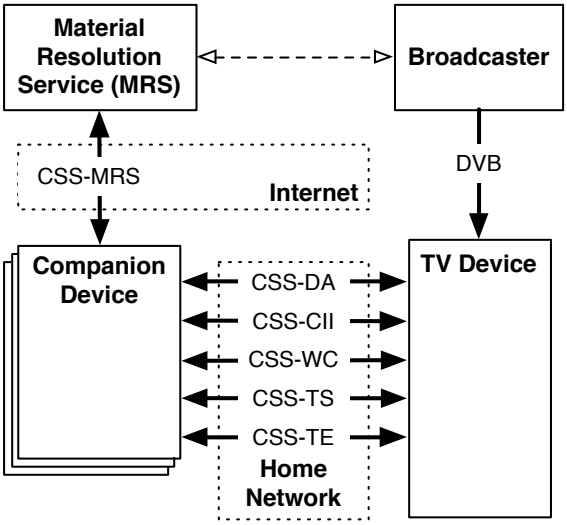
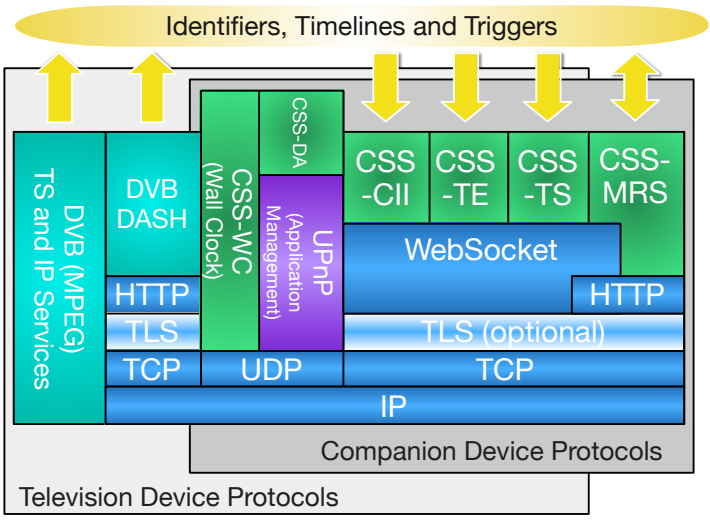


Figure 1 also shows the various interfaces offered by the TV Device over the home network to the companion device:

- CSS-DA: Discovery and Association to allow the companion device to find the TV and the CSS-CII interface address.
- CSS-CII: Content Identification that conveys details of the content being displayed (using information contained within the DVB broadcast) and updates to that content, and the addresses or locations of the other interfaces offered by the TV.
- CSS-WC: Wall Clock that allows the TV and Companion Device application to have a common clock for reference.
- CSS-TS: Time Stamp that conveys details of the time position of the content being displayed.
- CSS-TE: Trigger Events that provides access to "triggers" in broadcast, e.g. DSM-CC "Do It Now" Events.

Finally, Figure 1 shows the CSS-MRS interface provides a translation service between arbitrary identifiers and time lines. This allows for any identifiers or timelines present in the DVB broadcast to be translated into, for example, common reference values. This decoupling helps to simplify design and deployment of companion device applications.

Figure 2 show the protocols defined or used by DVB-CSS. The majority of interfaces are built over WebSockets or HTTP. These interfaces define JSON-formatted messages to convey information, making it very friendly for HTML-5 environments.



The WebSocket-based interfaces work by opening a connection to the relevant address, usually followed by sending an initial JSON-formatted message to configure the connection (e.g. to specify the timeline of interest), and then receiving JSON-formatted messages with the necessary information, that are updated with changed information as appropriate.

The CSS-MRS interface is based on an HTTP REST interface, where the details of the content to be resolved are passed in a tightly defined form as a URI parameter. This approach means that typical caching techniques can be used to assist with scalability of the MRS interface to nation-sized populations and request loads. The response message is again JSON-formatted. The response can also include details of how updates to this information can be received; here a range of methods are defined, including a WebSocket-based solution, to address various potential issues and differing use-cases.

The CSS-WC mechanism is based on UDP and conveys details of the internal reference clock of the TV Device (against which the content timestamps are described). The method is similar to NTP: a message is sent by the Companion Device that requests the current value of the TV Device reference clock, to which the TV device responds. An optional follow-up message can also be sent by the TV device that includes the actual transmission time of the previous response message to improve accuracy of the estimate of the TV Device clock made by the Companion Device. The UDP protocol allows for a more accurate clock to be constructed than if a TCP-based protocol was used.

The CSS-DA interface uses the network service discovery mechanisms and the Application Management service template specified by the UPnP Forum and is defined in part 3; the other interfaces are defined in part 2.

Market Deployment

Use of the DVB-CSS specifications is expected to start with the deployment of devices complying to the HbbTV 2.0 specification. Part of HbbTV 2.0 includes much of the DVB-CSS specification family. Although the specification can make use of additional signaling in broadcast, it is possible to make use of the specification without any alterations to broadcasts. Thus with the deployment of suitably equipped devices, such as HbbTV 2.0 compliant devices, use of these specifications comes with a very low barrier to market.

Next Steps for DVB-CSS

DVB is currently monitoring the market development for CSS. If new features are requested, DVB will provide the necessary specifications.

Links

www.dvb.org/standards:

DVB-CSS part 1	Overall architecture	DVB BlueBook A-167-1	ETSI TR 103 286-1
DVB-CSS part 2	Content ID and Sync	DVB BlueBook A-167-2	ETSI TS 103 286-2
DVB-CSS part 3	Discovery	DVB BlueBook A-167-3	ETSI TS 103 286-3

www.hbbtv.org	Clause 13 of HbbTV 2.0 references the DVB-CSS specifications.
bbc.in/1z2JNg5	Reference testing tool software for the CSS-CII, CSS-WC and CSS-TS interfaces.
bit.ly/2uh8eHq	Collection of test streams, open source tools and other resources.

Produced by the DVB Project Office – dvb@dvb.org