To Mark Richer, President, ATSC:

In response to your recent enquiry regarding activities in the W3C Web and TV Interest Group that are of relevance to ATSC 3.0 standardization, we have prepared this document in the hopes that it will address your needs and answer any questions you may have about our work.

Motivation

The W3C Web and TV IG (http://www.w3.org/2011/webtv/) strongly believes that specifications for the ATSC 3.0 runtime environment should only include Web APIs (i.e. HTML APIs, CSS APIs, JavaScript APIs, DOM APIs, etc.) that are based on W3C specs and not include any externally-defined Web APIs: The IG is happy to work with ATSC to analyze your use cases against the Web standards and help W3C working groups extend Web APIs if we find gaps between existing APIs and your use cases. Among the reasons are that:

1. All the functional areas that the ATSC is interested in are either already supported by standardized W3C APIs or are being addressed by active W3C groups.

2. All W3C APIs are available under the W3C Royalty-Free (RF) license, while proprietary APIs are oftentimes royalty-bearing. Leveraging W3C APIs will make ATSC specifications cheaper and easier to adopt.

3. The widespread public participation of all the major browser vendors, Web developers, and many of the largest Web users guarantees that the quality of the W3C APIs will exceed APIs defined by smaller, private groups.

4. Implementations of W3C APIs are available in multiple widely-used, free, open-source libraries. This will ensure quality and lower the implementation costs for ATSC specifications.

5. The open Web and all leading PC, mobile and device Web browsers closely follow W3C Web standards. Creating an ATSC spec that includes non-W3C APIs will cause a split of ATSC devices away from the open Web and towards a proprietary world.

Functional Area Alignment

The correspondence from the ATSC highlighted six areas of potential alignment between the ATSC and the W3C. The Web & TV Interest Group completed its first round of gap analysis between HTML5 technologies and TV-related services' requirements last year, and we address each of the six functional areas here.

Interactivity Runtime Environment

In addition to standard HTML mechanisms that allow for end users to interact with a Web service (e.g. <input> tags), HTMLMediaElements have been introduced as part of HTML5 to enable end users to directly interact with media. Examples of this include embedded <video> tags and <audio> tags. Furthermore, the Web Applications WG defines the Service Workers API to enhance the Web application runtime environment. Service Workers API (http://www.w3.org/TR/service-workers/) enables event-driven background processing of Web applications. Its cache management can

address not only offline use cases for mobile TV devices but also instant launch of interactive applications for broadcast services.

The W3C has recently created the TV Control API Community Group (http://www.w3.org/community/tvapi/) tasked with the specification of a TV control API. The output of this group would be a suitable Web API to allow for Web applications to be able to interact with TV services.

In addition, the System Applications Working group has been dealing with runtime environment features from a broader viewpoint. The mission of the WG is to define a runtime environment, security model, and associated APIs for building Web applications with comparable capabilities to native applications.

Personalization, Targeted Ads/Content

Several API's have been defined by the W3C that can assist in targeting and personalization that move beyond the Web's traditional use of cookies for persistent storage of user data and allow Web apps to access ads and content metadata.

- The IndexedDB API (http://www.w3.org/TR/IndexedDB/) allows for Web pages to create transactional databases that are sandboxed (i.e. cannot only be accessed by other Web pages in the active browser context based on the same-origin policy).
- The Web Storage API (http://www.w3.org/TR/webstorage/) provides a simple mechanism wherein a Web page can create a local store of arbitrary data. Unlike IndexedDB, Web Storage allows for storing unstructured data but is typically limited in memory allocation.
- The File API (http://www.w3.org/TR/FileAPI/) allows a Web page to access file data that is strictly selectable by the end user.
- The HTML5 track API (http://www.w3.org/TR/html/embedded-content-0.html#the-track-element) allows Web apps to be bound to ads and/or content shown in a video element by providing metadata associated with the media tracks.
- The Sourcing In-band Media Resource Tracks from Media Containers into HTML (http://dev.w3.org/html5/html-sourcing-inband-tracks/) allows Web apps to access in-band media resources such as transport stream descriptors that contain network, service, and program information and more.

Companion Screens

The W3C has standardized several solutions that allow for primary devices (e.g. set-top boxes) and secondary devices to communicate.

- The WebSocket API (http://www.w3.org/TR/websockets/), which allows for Web applications to leverage a TCP-based persistent connection between the client device and a server.
- The WebRTC API (http://www.w3.org/TR/webrtc/), which allows a Web application to form a peer-to-peer UDP-based connection. This API offers a data channel capability, which provides applications with an alternative to WebSockets for point-to-point data transfer.

- The Web Apps Working Group and the Device API Working Group are working on APIs for access to LAN services.
- The Presentation API (http://webscreens.github.io/presentation-api/) is an early-stage proposal that would allow a Web application to display over multiple screens.
- The TV control API resulting from the TV Control API Community Group would also apply to companion devices for control of TV services.
- The W3C is in a process to form a new working group that aims to define an API which enables Web pages to use secondary screens to display Web content. (http://www.w3.org/2014/secondscreen/)

Content Protection/DRM

- Encrypted Media Extensions (http://www.w3.org/TR/encrypted-media/) provides APIs to control playback of protected content.
- The Web Cryptography API (http://www.w3.org/TR/WebCryptoAPI/) provides a means for a Web application to perform basic cryptographic operations along with generating and/or managing encryption keys.

Service Protection

Transport layer security (TLS) is possible in all API's defined by the W3C related to HTTP transport. These include WebRTC, WebSockets, and XML-over-HTTP Request (XHR – see http://www.w3.org/TR/XMLHttpRequest/). Service protection related to DRM is covered in the EME specification.

The System Applications Working Group has been working on a security model and associated APIs for building Web applications with comparable capabilities to native applications.

New HTML5 Mechanisms for Content Delivery

- The Media Source Extensions (http://www.w3.org/TR/media-source/) feature enables Web applications to support adaptive HTTP streaming solutions.
- WebRTC can potentially be used as a distribution mechanism for content.

Testing and Conformance

The efforts of the Interest Group around testing are summarized in the Testing Task Force wiki: http://www.w3.org/2011/webtv/wiki/Testing. The test requirements document is found at http://www.w3.org/2011/webtv/Testing/WebTVTestingRequirements20130813.html.

Further Collaboration between the ATSC and W3C

Specification work within the W3C is an ongoing effort, and some specifications of interest in the area of Broadcast TV are in early stages and require broad industry participation. The W3C strongly encourages the ATSC and/or ATSC members to directly engage with the relevant groups within the W3C to help shape such specifications, help them progress, and also generate implementer interest.