



Smart TV[®]
Alliance

Technical Specification

Version 4.0

Status:	Final
Version:	4.0
Date:	4th September 2014
Author:	Smart TV Alliance, Inc.
Category:	Released

© Smart TV Alliance, Inc. 2014

All rights are reserved. Reproduction or transmission in whole or in part, in any form or by any means, electronic, mechanical or otherwise, is prohibited without the prior written consent of the copyright owner

1. CHANGE HISTORY	4
2. INTRODUCTION	5
2.1. OVERVIEW	5
2.2. DEFINITIONS.....	5
2.3. ABBREVIATIONS.....	5
2.4. REFERENCES	6
2.5. TRADEMARKS AND COPYRIGHTS	10
3. TECHNICAL SPECIFICATION	12
3.1. INTRODUCTION	12
3.2. STATUS DEFINITION.....	13
3.3. BROWSER	13
3.3.1. <i>HTML5 profile</i>	13
3.3.1.1. <i>XMLHttpRequest</i>	14
3.3.1.2. <i>CSS3 UI</i>	14
3.3.1.3. <i>CSS3 BG</i>	15
3.3.1.4. <i>CSS3 Media Queries</i>	15
3.3.1.5. <i>CSS3 Transforms</i>	15
3.3.1.6. <i>CSS3 Animations</i>	15
3.3.1.7. <i>CSS3 Color Module</i>	15
3.3.1.8. <i>CSS3 Fonts</i>	16
3.3.1.9. <i>CSS3 Image Values and Replaced Content</i>	16
3.3.1.10. <i>CSS3 Multi-column Layout</i>	16
3.3.1.11. <i>CSS3 Namespaces</i>	16
3.3.1.12. <i>CSS3 Selectors</i>	16
3.3.1.13. <i>CSS3 Text</i>	17
3.3.1.14. <i>CSS3 Text Decoration</i>	17
3.3.1.15. <i>CSS3 Transitions</i>	17
3.3.1.16. <i>CSSOM View</i>	17
3.3.1.17. <i>HTML5 Detail</i>	18
3.3.1.18. <i>HTML5 Common Infrastructure</i>	18
3.3.1.19. <i>HTML5 DOM</i>	18
3.3.1.20. <i>HTML5 Semantics</i>	19
3.3.1.21. <i>HTML5 Media elements</i>	19
3.3.1.22. <i>HTML5 Media Element Events</i>	20
3.3.1.23. <i>HTML5 Forms</i>	20
3.3.1.24. <i>HTML5 Loading web pages</i>	21
3.3.1.25. <i>HTML5 Web application APIs</i>	21
3.3.1.26. <i>HTML5 User interaction</i>	21
3.3.1.27. <i>HTML5 Syntax</i>	21
3.3.1.28. <i>HTML5 Related standards</i>	21
3.3.2. <i>Network Service Discovery</i>	21
3.3.2.1. <i>Overview</i>	21
3.3.2.2. <i>Supported features</i>	22
3.3.2.3. <i>Security recommendation</i>	22
3.3.3. <i>Capabilities</i>	22
3.3.4. <i>Input/key support</i>	23
3.3.5. <i>User Agent String</i>	23
3.3.6. <i>Extended API for Smart TV Alliance</i>	24
3.3.6.1. <i>SmartTvA_API Object</i>	24
3.3.6.2. <i>Methods for SmartTvA_API object</i>	24
3.4. VIDEO/AUDIO STREAMING.....	27
3.4.1. <i>HTML5 video/audio</i>	27
3.4.2. <i>Streaming protocols</i>	27
3.4.2.1. <i>HTTP streaming over SSL</i>	27
3.4.2.2. <i>HLS</i>	27
3.4.2.3. <i>MPEG-DASH</i>	27

3.4.3.	Streaming containers	27
3.4.4.	Streaming codecs	28
3.4.4.1.	Video Codecs	28
3.4.4.2.	Audio Codecs	28
3.4.5.	MIME types for A/V media formats	28
3.4.6.	Subtitles	29
3.4.6.1.	Subtitles using native renderer.....	29
3.4.7.	Multi Audio	30
3.4.7.1.	HTML5 Media Element.....	30
3.4.7.2.	AudioTrackList Interface.....	30
3.4.7.3.	AudioTrack Interface	30
3.4.7.4.	Mapping to Streaming Protocols	30
3.4.7.5.	Smooth Streaming.....	30
3.4.7.6.	MPEG-DASH.....	31
3.4.8.	Ultra High Definition Services	31
3.4.8.1.	Profile	31
3.5.	DIGITAL RIGHTS MANAGEMENT	32
3.5.1.	PlayReady.....	32
3.5.1.1.	PlayReady Features.....	32
3.5.1.2.	PlayReady API	32
3.5.2.	Widevine	32
3.6.	EME AND MSE	33
3.6.1.	Parsing Model.....	33
3.6.2.	EME	33
3.6.2.1.	DRM support through EME	33
3.6.3.	MSE	33
3.7.	MULTISCREEN.....	33
3.7.1.	DIAL	33
3.7.1.1.	Definitions for section 3.7.1	33
3.7.1.2.	DIAL Service Discovery.....	34
3.7.1.3.	DIAL REST Service.....	34
3.7.1.4.	Multiscreen Application Naming Conventions.....	34
3.7.2.	AllJoyn.....	35
3.7.2.1.	Definitions for section 3.7.2	35
3.7.2.2.	Application Launch Specification	35
3.8.	SMART HOME.....	38
4.	HISTORY.....	39
4.1.	CHANGES FROM VERSION 3.0 TO VERSION 4.0	39
4.2.	CHANGES FROM VERSION 2.5 TO VERSION 3.0	39
4.3.	CHANGES FROM VERSION 2.0 TO VERSION 2.5	39
4.4.	CHANGES FROM VERSION 1.0 TO VERSION 2.0	40
ANNEX A.	MULTISCREEN (INFORMATIVE)	42
A.1	RESOLVING URL FROM APPLICATION NAME	42
A.1.1	Resolving Application URL via Internet Server.....	42
A.1.2	Look-up Table of Web Applications	42
A.2	W3C WEBSOCKET API FOR APPLICATION TO APPLICATION COMMUNICATION.....	43
A.2.1	Cloud based app to app communication	43
ANNEX B.	SUBTITLES (INFORMATIVE)	45
B.1	Example TTML file	45

1. Change history

Version	Date	Changes
1.0	2012-06-14	Final
2.0 draft 5	2012-09-17	First draft version 2.0 for public release
2.0.1 Final	2012-12-13	Final version 2.0.1 for public release
2.5	2013-04-01	Final version 2.5 for public release
3.0	2013-09-03	Final version 3.0 for public release
4.0	2014-09-04	Final version 4.0 for public release

2. Introduction

2.1. Overview

This document sets out version 4.0 of the Smart TV Alliance specification. It is intended primarily for manufacturers, and describes the technical features to be implemented by TV Devices.

The Smart TV Alliance's motto is 'build once, run everywhere'. The members' ambition is to align on technology that will allow developers to create apps and successfully run them on all supported Smart TV Alliance TV Devices. These applications will typically be available to users from Smart TV portals.

As far as possible, the specification is built on existing “state of the art” solutions, and this document refers to those. The major building blocks are:

- HTML5;
- MPEG-DASH, Microsoft Smooth Streaming and HTTP Live Streaming;
- H.264 and HE-AAC;
- PlayReady and optionally Widevine DRMs;
- DIAL and optionally AllJoyn for multiscreen applications;
- A specification for ultra-high definition services using HEVC is also included;
- A Smart TV Alliance Smart Home specification is available separately.

Where existing solutions are not available, this document specifies the technical solution developed by the Alliance.

The Alliance will also release a Software Development Kit and developer documentation for this version 4.0 specification. This will provide a user friendly environment for developers to create applications that run on all Smart TV Alliance compatible TV Devices.

This document does not detail the full capabilities of individual TV Devices. It specifies the capabilities common to all Smart TV Alliance TV Devices.

While a lot of care has been taken to ensure the correctness of the information in this document, errors cannot be completely prevented. The latest version of this document, with possible corrections, is always available online. If you have questions and/or remarks regarding these guidelines, please post them through the designated support channels.

2.2. Definitions

TV Device The Smart TV device capable of running applications and receiving services in accordance with this specification. For example, this may be a television, or a device that connects to a television such as a Blu-ray player or set top box.

2.3. Abbreviations

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
A/V	Audio / Video
AVC	Advanced Video Codec (also known as H.264)
CENC	Common Encryption
CSS3	Cascading Style Sheets
DIAL	Discovery And Launch
DOM	Document Object Model
DRM	Digital Rights Management
EME	Encrypted Media Extensions
GIF	Graphics Interchange Format
HbbTV	Hybrid Broadcast Broadband Television
HD	High Definition

HE-AAC	High Efficiency – Advanced Audio Codec
HEVC	High Efficiency Video Coding
HLS	HTTP Live Streaming
HTML	Hypertext Markup Language
HTTP(S)	Hypertext Transport Protocol (Secure)
ISO	International Standards Organization
ISOBMFF	ISO Base Media File Format
JPEG	Joint Photographic Experts Group (compression format)
MIME	Multipurpose Internet Mail Extensions
MPEG	Moving Picture Experts Group
MP3	MPEG 1 – Layer 3 audio
MP4	MPEG4 Part 14 file format – equivalent to ISOBMFF (see above)
MPD	Media Presentation Description
MPEG2	MPEG2 video codec
MPEG-DASH	MPEG Dynamic Adaptive Streaming over HTTP
MSE	Media Source Extensions
OIPF	Open IPTV Forum
PNG	Portable Network Graphics
REST	Representational State Transfer
SD	Standard Definition
SDK	Software Development Kit
SOAP	Simple Object Access Protocol
SSDP	Simple Service Discovery Protocol
SSL	Secure Sockets Layer
STA	Smart TV Alliance
TLS	Transport Layer Security
TS	Transport Stream
TTML	Timed Text Markup Language
UHD	Ultra High Definition
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience
VoD	Video on Demand
W3C	World Wide Web Consortium
XML	Extensible Markup Language

2.4. References

[1] Cross Origin Resource Sharing (CORS)
<http://www.w3.org/TR/2013/CR-cors-20130129/>

[2] Media Queries
<http://www.w3.org/TR/2012/REC-css3-mediaqueries-20120619/>

[3] ECMAScript Language Specification (5.1 Edition), June 2011,
<http://www.ecma-international.org/ecma-262/5.1/>

[4] REC-DOM-Level-2-20001113 Document Object Model (DOM) Level 2 Core Specification, Version 1.0,
<http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113>

[5] REC-DOM-Level-2-20001113 Document Object Model (DOM) Level 2 Style Specification, Version 1.0,
<http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113>

[6] REC-DOM-Level-2-20001113 Document Object Model (DOM) Level 2 Events Specification, Version 1.0,
<http://www.w3.org/TR/2000/REC-DOM-Level-2-Events-20001113>

[7] REC-DOM-Level-2-20030109 Document Object Model (DOM) Level 2 HTML Specification, Version 1.0,
<http://www.w3.org/TR/2003/REC-DOM-Level-2-HTML-20030109>

[8] W3C - CSS Transforms Module Level 3 - April 2012
<http://www.w3.org/TR/2012/WD-css3-transforms-20120403/>

[9] HTTP State Management Mechanism
<http://tools.ietf.org/html/rfc6265>

[10] Persistent Client State: HTTP Cookies
http://wp.netscape.com/newsref/std/cookie_spec.html

[11] W3C - CSS 2.1 - April 2008
<http://www.w3.org/TR/2011/REC-CSS2-20110607/>

[12] HTTP Live Streaming - IETF draft - 2011-03
<http://tools.ietf.org/html/draft-pantos-http-live-streaming-06>

[13] Microsoft Smooth Streaming -
<http://www.iis.net/download/smoothstreaming>

[14] PlayReady DRM overview
<http://download.microsoft.com/download/b/8/3/b8316f44-e7a9-48ff-b03a-44fb92a73904/Microsoft%20PlayReady%20Content%20Access%20Technology-Whitepaper.docx>

[15] W3C - RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1 - June 1999
<http://www.ietf.org/rfc/rfc2616.txt>

[16] HTML5 Candidate Recommendation 17 December 2012
<http://www.w3.org/TR/2012/CR-html5-20121217/>

[17] HTML5 Server-sent events Candidate Recommendation 11 December 2012
<http://www.w3.org/TR/2012/CR-eventsourcing-20121211/>

[18] HTML5 Web storage Proposed Recommendation 9 April 2013
<http://www.w3.org/TR/2013/PR-webstorage-20130409/>

[19] HbbTV root certificates
<http://www.hbbtv.org/spec/certificates.html>

[20] HTML5 Web Workers Candidate Recommendation 1 May 2012
<http://www.w3.org/TR/2012/CR-workers-20120501/>

[21] REC-DOM-Level-2-20001113 Document Object Model (DOM) Level 2 Views Specification, Version 1.0
<http://www.w3.org/TR/2000/REC-DOM-Level-2-Views-20001113>

[22] W3C - XMLHttpRequest Level 2 - Draft - December 2012
<http://www.w3.org/TR/2012/WD-XMLHttpRequest-20121206/>

[23] W3C - CSS3 - Basic User Interface Module Level 3 - January 2012
<http://www.w3.org/TR/2012/WD-css3-ui-20120117/>

[24] W3C - CSS3 - Backgrounds and Borders Module Level 3 - July 2012
<http://www.w3.org/TR/2012/CR-css3-background-20120724/>

[25] ISO/IEC 23009-1 : Information technology –Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats

[26] ETSI TS 102 796 v1.2.1 Hybrid Broadcast Broadband TV
http://www.etsi.org/deliver/etsi_ts/102700_102799/102796/01.02.01_60/ts_102796v010201p.pdf

- [27] ISO/IEC 14496-12 ISO Base File Format
- [28] ISO/IEC 13818-1 MPEG-2 Part 1: Systems
- [29] ISO/IEC 14496-15:2010, "Information Technology - Coding of Audio-Visual Objects - Part 15: Advanced Video Coding (AVC) file format", International Standards Organization
- [30] ISO/IEC 14496-3:2009, "Information Technology – Coding of audio-visual objects – Part 3: Audio"
- [31] ETSI TS 102 366 V1.2.1 (2008-08), "Digital Audio Compression (AC-3, Enhanced AC-3) Standard"
- [32] ISO/IEC 11172-3:1993/Cor 1:1996, "Information Technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 3: Audio"
- [33] W3C - HTML5 Differences from HTML4 – May 2013
<http://www.w3.org/TR/2013/WD-html5-diff-20130528/>
- [34] CSS2 Selectors
<http://www.w3.org/TR/CSS2/>
- [35] CSS3 Selectors
<http://www.w3.org/TR/2011/REC-css3-selectors-20110929/>
- [36] CSS3 2D Transforms – September 2012
<http://www.w3.org/TR/2012/WD-css3-transforms-20120911/>
- [37] CSS3 Animations – February 2013
<http://www.w3.org/TR/2013/WD-css3-animations-20130219/>
- [38] CSS3 Color – June 2013
<http://www.w3.org/TR/2011/REC-css3-color-20110607>
- [39] CSS3 Fonts – February 2013
<http://www.w3.org/TR/2013/WD-css3-fonts-20130212/>
- [40] CSS3 Image values and replaced content
<http://www.w3.org/TR/2012/CR-css3-images-20120417/>
- [41] CSS3 Multi-column Layout Module
<http://www.w3.org/TR/2011/CR-css3-multicol-20110412/>
- [42] CSS3 Namespace
<http://www.w3.org/TR/2011/REC-css3-namespace-20110929/>
- [43] CSS Text Level 3 – November 2012
<http://www.w3.org/TR/2012/WD-css3-text-20121113/>
- [44] CSS Transitions – February 2013
<http://www.w3.org/TR/2013/WD-css3-transitions-20130212/>
- [45] CSSOM View Module
<http://dev.w3.org/csswg/cssom-view/>
- [46] HTML5 2D Context – Candidate Recommendation December 2012
<http://www.w3.org/TR/2012/CR-2dcontext-20121217/>
- [47] HTML5 Web Messaging
<http://www.w3.org/TR/2012/CR-webmessaging-20120501/>

[48] HTML5 WebSockets – Candidate Recommendation September 2012
<http://www.w3.org/TR/2012/CR-websockets-20120920/>

[49] DOM Range
<http://dvcs.w3.org/hg/editing/raw-file/tip/editing.html#selections>

[50] Popcorn Javascript library
<http://popcornjs.org/>

[51] HbbTV Content Protection using Microsoft PlayReady v1.0 and DASH Content Protection using Microsoft PlayReady v1.0

[52] Open IPTV Forum Release 1.1 Volume 5 – Declarative Application Environment
<http://www.oipf.tv/specifications/root/uncategorised/volume--5-declarative-application-environment51/download>

[53] Open IPTV Forum Release 2.1 Volume 2 – Media Formats
<http://www.oipf.tv/specifications/root/solution-specification-volume-2-media-formats/download>

[54] Widevine API Mapping – Version 1.0
Smart TV Alliance

[55] DIAL Discovery And Launch protocol specification Version 1.6.4
<http://www.dial-multiscreen.org/dial-protocol-specification>

[56] AllJoyn SDK version 3.2
<https://www.alljoyn.org/docs-and-downloads>

[57] Recommendation ITU-T H.265 “High efficiency video coding”

[58] ISO14496-15:2010/DAM2 "Carriage of high-efficiency video coding (HEVC)"

[59] ISO/IEC 13818-1:2013/DAMd 3 "Transport of HEVC video over MPEG-2 systems"

[60] DOM Level 3 Events
<http://www.w3.org/TR/2012/WD-DOM-Level-3-Events-20120906/>

[61] DOM4
<http://www.w3.org/TR/2012/WD-dom-20121206/>

[62] [unused]

[63] CSS Flexible Box Layout Module
<http://www.w3.org/TR/2012/CR-css3-flexbox-20120918/>

[64] CSS Style Attributes
<http://www.w3.org/TR/2010/CR-css-style-attr-20101012/>

[65] CSS Text Decoration Module Level 3
<http://www.w3.org/TR/2013/WD-css-text-decor-3-20130103/>

[66] CSS Values and Units Module Level 3
<http://www.w3.org/TR/2013/CR-css3-values-20130404/>

[67] DOM Parsing and Serialization
<http://www.w3.org/TR/2012/WD-DOM-Parsing-20120920/>

[68] Web IDL
<http://www.w3.org/TR/2012/CR-WebIDL-20120419/>

- [69] Typed Array Specification
<https://www.khronos.org/registry/typedarray/specs/latest/>
- [70] RFC3986 Uniform Resource Identifier (URI): Generic Syntax
<http://www.ietf.org/rfc/rfc3986.txt>
- [71] RFC3987 Internationalized Resource Identifiers (IRIs)
<http://www.ietf.org/rfc/rfc3987.txt>
- [72] RFC3490 Internationalizing Domain Names in Applications (IDNA)
<http://www.ietf.org/rfc/rfc3490.txt>
- [73] RFC6454 The Web Origin Concept
<http://tools.ietf.org/html/rfc6454>
- [74] [unused]
- [75] Timed Text Markup Language (TTML) 1.0 (Second Edition)
<http://www.w3.org/TR/ttaf1-dfxp>/
<http://www.w3.org/TR/ttaf1-dfxp/>
- [76] The D-Book Part B version 2.0
<http://www.dtg.org.uk/>
- [77] Smart TV Alliance SDK Diversity Handling Guidelines 4.0.0
<https://developers.smarttv-alliance.org/sdk-documentation>
- [78] TrueType Specification V1.x - Microsoft
<http://www.microsoft.com/typography/SpecificationsOverview.mspx>
- [79] WOFF File Format 1.0
<http://www.w3.org/TR/2012/REC-WOFF-20121213/>
- [80] Smart TV Alliance SDK Development Guidelines v4.0.0
<https://developers.smarttv-alliance.org/sdk-documentation>
- [81] ETSI TS 101 154 v1.11.1 Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream
- [82] Network Service Discovery
<http://www.w3.org/TR/2014/WD-discovery-api-20140220/>
- [83] Encrypted Media Extensions
<https://dvcs.w3.org/hg/html-media/raw-file/eme-v0.1b/encrypted-media/encrypted-media.html>
- [84] Media Source Extensions
<http://www.w3.org/TR/2014/CR-media-source-20140109/>
- [85] Smart TV Alliance Smart Home Technical Specification v1.0
<https://developers.smarttv-alliance.org/specification>

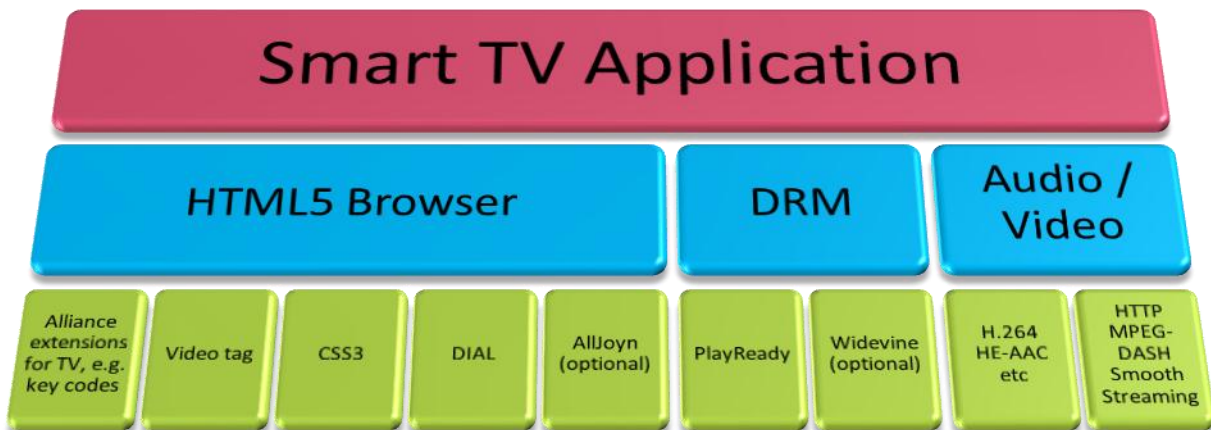
2.5. Trademarks and copyrights

All trademarks and copyrights are the property of their respective owners.

3. Technical Specification

3.1. Introduction

This chapter introduces the common Smart TV Alliance specification. It is divided into logical blocks. See the picture below for an overview of the most significant technologies:



Some important terms are explained as follows:

- Smart TV Applications are written in HTML5 and use the APIs specified in this document. Examples include VoD such as movie rental, catch-up TV and 3D services, social networking, games and news applications.
- The HTML5 Browser implements the HTML5 and APIs profiled for Smart TVs, allowing access to features such as input methods, video streaming and DRM.
- DRM is supported, enabling a wide range of business models for content monetization.
- APIs allow the application to control audio and video streaming and presentation to the user.
- Multiscreen allows web applications to run on the TV Device and related applications on a second screen (such as a smart phone or tablet), and for these applications to discover, launch and communicate with each other.

3.2. Status Definition

This document specifies the common technical features of the Smart TV Alliance TV Devices, using the terms defined in the table below. Items not listed are not supported by this version of the specification. Individual products may support extra features, but Smart TV Alliance applications shall not use such features.

Status	Definition	Remarks
M	Mandatory, Fully Supported. All TV Devices SHALL support this feature in order to comply with this specification.	Such explicitly defined feature overrides any such feature included from older included specifications.
C-M	Conditionally Mandatory. Implementation of this feature is optional, but TV Devices that do implement this feature SHALL comply with this specification.	
P	Partially Supported. Details are defined in a separate table.	
O	Optional. Details are defined and TV Devices MAY support this feature.	

3.3. Browser

This chapter describes the level of support from the referenced standards that the TV Device's browser shall meet. All item support (M) is defined on tag/property level only.

3.3.1. HTML5 profile

The table below gives a high level view of the profile that TV Devices shall implement. Where partial is indicated, the detailed support is described.

Standard	Reference	Status	Remark
HTML5 candidate recommendation	[16]	M	See section 3.3.1.18
HTTP 1.1	[15]	M	
SSL / TLS		M	HbbTV server certificates shall be supported.
DOM Level 2 Core	[4]	M	
DOM Level 2 Style	[5]	M	
DOM Level 2 Events	[6]	M	including MouseEvent
DOM Level 2 HTML	[7]	M	
DOM Level 2 Views	[21]	M	
DOM Level 3 Events	[60]	P	Support for Keyboard Events insofar required by this specification, other parts supported as common subset with DOM Level 2.
DOM Level 4	[61]	P	Supported insofar required by supported HTML5 profile.
DOM Parsing and Serialization	[67]	M	
ECMAScript-262 5.1 ed.	[3]	M	
XMLHttpRequest Object (2)	[22]	M	See section 3.3.1.1
Cookies	[10], [9]	M	
CSS3 UI	[23]	M	See section 3.3.1.2
CSS3 BG	[24]	M	See section 3.3.1.3
CSS3 Media Queries	[2]	M	See section 3.3.1.4
CSS2.1	[11]	M	Only mandatory items from that

			specification are supported.
CSS3 Transforms	[8]	M	See section 3.3.1.5
CSS3 Animations	[37]	M	See section 3.3.1.6
CSS3 Color Module	[38]	M	See section 3.3.1.7
CSS3 Fonts	[39]	M	See section 3.3.1.8
CSS3 Image Values and Replaced Content	[40]	M	See section 3.3.1.9
CSS3 Multi-column Layout	[41]	M	See section 3.3.1.10
CSS3 Namespaces	[42]	M	See section 3.3.1.11
CSS3 Selectors	[34]. [35]	M	See section 3.3.1.12
CSS3 Text	[43]	M	See section 3.3.1.13
CSS3 Transitions	[44]	M	See section 3.3.1.15
CSSOM View	[45]	M	See section 3.3.1.16
CSS3 Flexbox	[63]	M*	
CSS3 Style Attributes	[64]	M	
CSS3 Text Decoration	[65]	M	See section 3.3.1.14
CSS3 Values and Units	[66]	M	
Web IDL	[68]	M	
Typed Arrays	[69]	M	
URIs, IRIs, IDNA	[70] [71] [72]	M	
Web Origin	[73]	M	
Network Service Discovery	[82]	O	See section 3.3.2
Encrypted Media Extensions	[83]	O	See section 3.6.2
Media Source Extensions	[84]	O	See section 3.6.3

M*: Mandatory via browser extension, details to be included in Software Development Kit guidelines document [77].

3.3.1.1. XMLHttpRequest

All section references are to [22] except where explicitly noted.

Section	Reference	Description	Support
CORS	[1]	CORS	M
Constructors	[22] section 4.2	XMLHttpRequest()	M
Event handlers	[22] section 4.4	onreadystatechange	M
States	[22] section 4.5	Readystate	M
Request	[22] section 4.6	open(), setRequestHeader(), withCredentials, send(), abort()	M
Response	[22] section 4.7	status, statusText, getResponseHeader(), getAllResponseHeaders(), responseText, responseXML	M

3.3.1.2. CSS3 UI

Section	Reference	Description	Support
User interface selectors - pseudo classes	[23] section 4.1	:default, :valid, :invalid, :in-range, :out-of-range, :required, :optional, :read-only, :read-write	M
Box Model addition	[23] section 6	box-sizing	M
Outline properties	[23] section 7	outline, outline-width, outline-style, outline-color, outline-offset	M
Resizing and overflow	[23] section 8.2	text-overflow	M

Pointing devices and keyboards	[23] section 9.2	nav-left, nav-right, nav-up, nav-down	M
--------------------------------	------------------	---------------------------------------	---

3.3.1.3. CSS3 BG

Section	Reference	Description	Support
Backgrounds	[24] section 3	background-color, background-image, background-repeat, background-attachment, background-position, background-clip, background-origin, background-size, background	M
Borders	[24] section 4	border-color, border-style, border-width, border	M
rounded corners	[24] section 5	border-radius	M
miscellaneous effects	[24] section 7	box-decoration-break, box-shadow	M*

M*: Mandatory via browser extension, details to be included in Software Development Kit guidelines document [77].

3.3.1.4. CSS3 Media Queries

Section	Reference	Description	Support
media features	[2] section 4	width, height, device-width, device-height, orientation, aspect-ratio, device-aspect-ratio	M

3.3.1.5. CSS3 Transforms

Section	Reference	Description	Support
	[36] section 7	transform	M*
	[36] section 8	transform-origin	M*
2D Transform Functions	[36] section 15	matrix, translate, translateX, translateY, scale, scaleX, scaleY, rotate, skew, skewX, skewY	M
Transform Function Lists	[36] section 16	transform function lists	M*

M*: Mandatory via browser extension, details to be included in Software Development Kit guidelines document [77].

3.3.1.6. CSS3 Animations

Section	Reference	Description	Support
Keyframes	[37] section 4	@keyframes, animation-name, animation-duration, animation-timing-function, animation-iteration-count, animation-direction, animation-play-state, animation-delay, animation-fill-mode, animation	M*

M*: Mandatory via browser extension, details to be included in Software Development Kit guidelines document [77].

3.3.1.7. CSS3 Color Module

Section	Reference	Description	Support
CSS3 Color Module	[38]	CSS3 Color Module Full Support	M

3.3.1.8. CSS3 Fonts

Section	Reference	Description	Support
Basic font properties	[39] section 3	font-family, font-style, font-size, font	M
	[39] section 4.3	src	M
Font rules	[39] section 4.1	@font-face ¹	M

(1): Font format support is defined in 3.3.3.

3.3.1.9. CSS3 Image Values and Replaced Content

Section	Reference	Description	Support
CSS3 Image Values and Replaced Content	[40]	CSS3 Image Values and Replaced Content	M

3.3.1.10. CSS3 Multi-column Layout

Section	Reference	Description	Support
The number and width of columns	[41] section 3	column-width, column-count, columns	M*
Column gaps and rules	[41] section 4	column-gap, column-rule-color, column-rule-style, column-rule-width, column-rule	M*
Column breaks	[41] section 5	break-before, break-after, break-inside	M*
Spanning columns	[41] section 6	column-span	M*

M*: Mandatory via browser extension, details to be included in Software Development Kit guidelines document [77].

3.3.1.11. CSS3 Namespaces

Section	Reference	Description	Support
	[42] section 3	@namespace	P

3.3.1.12. CSS3 Selectors

Section	Reference	Description	Support
Simple selectors			
	[34] section 5.4	Type selector (h1)	M
	[35] section 6.1.1	Type selectors and namespaces (ns E)	M
	[34] section 5.3	Universal selector (*)	M
	[34] section 5.8	Attribute presence and value selectors [att] [att=val] [att~=val] [att =val]	M
	[35] section 6.3.2	Substring matching attribute selectors [att^=val] [att\$=val] [att*=val]	M
	[35] section 6.3.3	Attribute selectors and namespaces (attr)	M
	[34] section 5.8.3	Class selectors (.)	M
	[34] section 5.9	ID selectors (#)	M
Pseudo classes			
	[34] section 5.11	:link, :visited, :hover, :active, :focus, :lang, :first-child	M
	[35] section 6.6	:target, :enabled, :disabled, :checked, :root, :nth-child(), :nth-last-child(), :nth-of-type(), :nth-last-of-type(), :last-child:, :first-of-type, :last-of-type, :only-child, :only-of-type, :empty, :not()	M
Pseudo Elements	[34] section 5,12	:first-line, :first-letter, :before, :after	M

Combinators			
	[34] section 5.5	Descendant selectors	M
	[34] section 5.6	Child selectors	M
Sibling Selectors	[34] section 5.7	Adjacent sibling selector (+)	M

3.3.1.13. CSS3 Text

Section	Reference	Description	Support
Alignment and Justification	[43] section 7.1	text-align	M
Spacing	[43] section 8	word-spacing, letter-spacing	M
Edge Effects			
	[43] section 9.1	text-indent	M

3.3.1.14. CSS3 Text Decoration

Section	Reference	Description	Support
	[65]	text-decoration, text-shadow	M

3.3.1.15. CSS3 Transitions

Section	Reference	Description	Support
Transitions	[44] section 2	transition-property, transition-duration, transition-timing-function, transition-delay, transition	M*
Transition Events	[44] section 5	TransitionEvent, propertyName, elapsedTime	M*
Animation of Property Types Support	[44] section 6	color, length, percentage, integer, font weight, number, rectangle, visibility, shadow, gradient, list of above types	M
(Animatable) properties from CSS	[44] section 7.1	background-color, border-bottom-width, border-left-width, border-right-width, border-spacing, border-top-width, bottom, color, font-size, height, left, letter-spacing, line-height, margin-bottom, margin-left, margin-right, margin-top, max-height, max-width, min-height, min-width, opacity, outline-color, outline-width, padding-bottom, padding-left, padding-right, padding-top, right, text-indent, top, vertical-align, visibility, width, word-spacing, z-index	M

M*: Mandatory via browser extension, details to be included in Software Development Kit guideline document [77].

3.3.1.16. CSSOM View

Section	Reference	Description	Support
Extensions to the Window interface	[45] section 5.1	matchMedia(), screen, innerWidth, innerHeight, scrollX, pageXOffset, scrollY, pageYOffset, scroll(), scrollTo(), scrollBy(), screenX, screenY, outerWidth, outerHeight	M
The MediaQueryList Interface	[45] section 5.2	MediaQueryList, media, matches, addListener(), removeListener(), MediaQueryListListener	M

The Screen Interface	[45] section 5.3	Screen, availWidth, availHeight, width, height, colorDepth, pixelDepth	M
Extensions to the Document Interface	[45] section 6	elementFromPoint()	M
Extensions to the Element Interface	[45] section 7	getClientRects(), getBoundingClientRect(), scrollIntoView(), scrollTop, scrollLeft, scrollWidth, scrollHeight, clientTop, clientLeft, clientWidth, clientHeight	M
Extensions to the HTMLElement Interface	[45] section 8	offsetParent, offsetTop, offsetLeft, offsetWidth, offsetHeight	M
Extensions to the Range Interface	[45] section 9	getClientRects(),getBoundingClientRect()	M
Extensions to the MouseEvent Interface	[45] section 10	screenX, screenY, pageX, pageY, clientX, clientY, x, y, offsetX, offsetY	M
The ClientRectList Interface	[45] section 11.1	ClientRectList, length, item()	M
The ClientRect Interface	[45] section 11.2	ClientRect, top, right, bottom, left, width, height	M

3.3.1.17. HTML5 Detail

As HTML5 is still being defined, some of the supported API's are subject to change. Also refer to [33]. Where needed, partial support for certain parts is indicated and details are described below - section references are included for each item where possible.

3.3.1.18. HTML5 Common Infrastructure

Section	Reference	Description	Support
HTMLAllCollection	[16] section 2.8.2.1	As required by the referenced interfaces.	M
HTMLFormControlsCollection	[16] section 2.8.2.2	As required by the referenced interfaces.	M
RadioNodeList	[16] section 2.8.2.2	As required by the referenced interfaces.	M
HTMLOptionsCollection	[16] section 2.8.2.3	As required by the referenced interfaces.	M
DOMStringMap	[16] section 2.8.3	As required by the referenced interfaces.	M
Transferable objects	[16] section 2.8.4	As required by the referenced interfaces.	M

3.3.1.19. HTML5 DOM

Section	Reference	Description	Support
Document object			
	[16] section 5.5.3	location	M
	[16] section 5.3.1	domain	M
	[16] section 3.1	referrer, cookie, lastModified, readyState, Title, body, head, images, embeds, plugins, links, forms, scripts, getElementByName()	M
	[16] section 3.2.3.6	Dir	M
	[16] section 3.4	Document.open(), close(), write(), writeln()	M
	[16] section 5.2.8	WindowProxy open()	M

	[16] section 5.2	defaultView	M
	[16] section 7.4.3	activeElement	M
	[16] section 6.1.6.2	onabort, onblur, oncanplay, oncanplaythrough, onchange, onclick, onclose, ondurationchange, onemptied, onended, onerror, onfocus, oninvalid, onkeydown, onkeypress, onkeyup, onload, onloadeddata, onloadedmetadata, onloadstart, onpause, onplay, onplaying, onpopstate, onprogress, onratechange, onseeked, onseeking, onsubmit, onsuspend, ontimeupdate, onwaiting, onreadystatechange	M
	[16] section 6.1.6.2	onmousedown, onmousemove, onmouseout, onmouseover, onmouseup	C-M (1)
HTML Element			
	[16] section 3.1.4	title	M
	[16] section 7.1	hidden	M
	[16] section 7.3	click()	M
	[16] section 7.4.3	focus(), blur()	M
	[16] section 7.6	isContentEditable	M
	[16] section 4.2.6	style	M
Global Attributes (for all HTML Elements)	[16] section 3.2.3	id, title, lang, dir, class, style, data-*, dataset	M

(1): Mandatory if pointer devices are supported for Smart TV Alliance applications.

3.3.1.20. HTML5 Semantics

Section	Reference	Description	Support
The root element	[16] section 4.1		M
Document Metadata	[16] section 4.2	title, base, link, meta, style	M
Scripting	[16] section 4.3	script	M
Sections	[16] section 4.4	body, article, section, nav, aside, h1, h2, h3, h4, h5, h6, hgroup, header, footer, address	M
Grouping content	[16] section 4.5	p, hr, pre, blockquote, ol, ul, li, dl, dt, dd, figure, figcaption, div	M
Text-level semantics	[16] section 4.6	a, em, strong, small, s, cite, q, dfn, abbr, code, var, i, b, u, mark, span, br, wbr	M
Embedded content	[16] section 4.8	img, iframe, embed, object, param, canvas, map, area	M
Tabular data	[16] section 4.9	table, caption, colgroup, col, tbody, thead, tfoot, tr, td, th, TableCellElement	M
Interactive elements	[16] section 4.11	summary	M
Links	[16] section 4.12	alternate, stylesheet	M

3.3.1.21. HTML5 Media elements

Section	Reference	Description	Support
---------	-----------	-------------	---------

The video element	[16] section 4.8.6	width, height, videoWidth, videoHeight, poster	M
The audio element	[16] section 4.8.7		M
The source element	[16] section 4.8.8	src, type, media	M
The track element	[16] section 4.8.9	kind, src, srclang, label, default, Track	M
Media elements	[16] section 4.8.10	error, src, currentSrc, networkState, preload, load(), canPlayType(), readyState, seeking, currentTime, duration, paused, defaultPlaybackRate, playbackRate, played, seekable, ended, autoplay, Loop, play(), pause()	M
	[16] section 4.8.10	audioTracks	M
	[16] section 4.8.10	textTracks	M
MediaError	[16] section 4.8.10.1		M
AudioTrackList	[16] section 4.8.10.10.1	length, AudioTrack(), getTrackById()	M
AudioTrack	[16] section 4.8.10.10.1	id, kind, label, language, enabled	M
TextTrackList	[16] section 4.8.10.12.5	length, TextTrack()	M
TextTrack	[16] section 4.8.10.12.5	kind, label, language, mode	M
TimeRanges	[16] section 4.8.10.14		M

3.3.1.22. HTML5 Media Element Events

Reference	Description	Support
[16] section 4.8.10.15	loadstart, progress, suspend, abort, error, emptied, loadedmetadata, loadeddata, canplay, canplaythrough, playing, waiting, seeking, seeked, ended, durationchange, timeupdate, play, pause, ratechange	M

3.3.1.23. HTML5 Forms

Section	Reference	Description	Support
The form element	[16] section 4.10.3		M
The fieldset element	[16] section 4.10.4		M
The legend element	[16] section 4.10.5		M
The label element	[16] section 4.10.6		M
The input element	[16] section 4.10.7	readonly, size, required, min/max, step, type=hidden/text/password/radio/submit/reset/button	M
The button element	[16] section 4.10.8		M
The select element	[16] section 4.10.9		M
The optgroup element	[16] section 4.10.10		M
The option element	[16] section 4.10.12		M
The textarea element	[16] section 4.10.13		M
The progress element	[16] section 4.10.16		M
The meter element	[16] section 4.10.17		M
Association of controls and forms	[16] section 4.10.18		M
Form validation	[16] section 4.10.1.4		M
The constraint validation API	[16] section 4.10.21.3		M
ValidityState	[16] section 4.10.21.3		M

3.3.1.24. HTML5 Loading web pages

Section	Reference	Description	Support
Window object	[16] section 5.2	WindowProxy, History, Location	M
Event handlers	[16] section 6.1.6.2	Refer to 3.3.1.19	M

(1): Mandatory if pointer devices are supported for Smart TV Alliance applications.

3.3.1.25. HTML5 Web application APIs

Section	Reference	Description	Support
Events	[16] section 6.1.6	oninput	M
WindowBase64	[16] section 6.4		M
The navigator object	[16] section 6.5.1	navigator, NavigatorID	M

3.3.1.26. HTML5 User interaction

Section	Reference	Description	Support
The hidden attribute	[16] section 7.1	hidden	M

3.3.1.27. HTML5 Syntax

Section	Reference	Description	Support
Writing HTML Documents			
	[16] Section 8.1.1	DOCTYPE	M
Parsing HTML Documents	[16] Section 8.2	HTML5 tokenizer, HTML5 tree building	M

3.3.1.28. HTML5 Related standards

The following standards are not directly part of HTML5:

Reference	Description	Support
[46]	2D Context	M
[46] Section 1	Text	M
[47]	Cross-document messaging	M
[17]	Server-sent events	M
[48]	WebSocket	M
[18] Section 4.2	Session Storage	M
[18] Section 4.3	Local Storage	M
[20]	Workers	M
[49]	Text selection	M

3.3.2. Network Service Discovery

3.3.2.1. Overview

The Network Service Discovery API ([82]) allows applications to discover locally available services. The primary use case for this API is the discovery of Smart Home Appliances. Also refer to section 3.8. The subsequent use of the service itself is not part of this API.

Feature	Support
Network Service Discovery	O

If the Network Service Discovery API is supported, the C-M functions in the table below are mandatory.

3.3.2.2. Supported features

TV Devices implementing this API must support the following features:

Section	Reference	Description	Support
Security and privacy considerations	[82] Section 4	Support for the mandatory security requirements of this API, including CORS	C-M*
Requesting networked services			
	[82] Section 5	getNetworkServices(), error.name	C-M
Obtaining networked services			
	[82] Section 6	length, servicesAvailable, services[], getServiceById(), servicefound, servicelost	C-M
Communicating with a networked service			
	[82] Section 7	id, name, type, url, config, online, notify, available, unavailable	C-M
Service Discovery			
	[82] Section 8.2	SSDP protocol	C-M
	[82] Section 8.1	Zeroconf	O
	[82] Section 8.3	Discovery And Launch (DIAL)	O

3.3.2.3. Security recommendation

(*) TV Devices implementing this API should take proper precautions to prevent potential misuse of this API. Next to the support for the mandatory security features of the API, it is recommended that TV Devices enable this API only:

- when authorized by the user of the TV Device (e.g. as part of a user setting or TV Device generated pop-up request)
- for Smart TV Apps specifically requiring this API (e.g. apps made for the Smart Home)
- for discoverable services that implement CORS ([1]). When a CORS implementation is not available for a service, that service should only be discovered (available within the list of available services) if it was specifically enabled on the TV Device by e.g. adding it to a user configurable whitelist of allowed service type tokens ([82] Section 3).

3.3.3. Capabilities

The TV Devices shall adhere to these minimum capabilities:

Capability	Details	Remark
Browser resolution	TV Devices shall support a viewport resolution of 1280x720 pixels. Additionally TV Devices may support higher viewport sizes.	Application developers shall provide a 1280x720 application. They may also provide applications supporting higher viewport sizes, e.g. 1920x1080.
Aspect ratio	TV Devices shall support an aspect ratio of 16:9.	
Color format	32 bits	
Supported downloadable font formats for @font-face element	TV Devices shall support at least one of: <ul style="list-style-type: none"> • WOFF (Web Open Font Format) [79] • TrueType v1.0 (TTF) [78] 	At least one of these font formats shall be supported by the TV Device. The development guidelines [80] provide information on how to reliably specify both font formats.

Supported fonts (or equivalent)	"Tiresias" (Screenfont) TV Devices shall support: <ul style="list-style-type: none"> • Scalable fonts; • Signalling of font size by application as 'pt' and 'em'. 	font-family: sans-serif. (True Type font, Basic Euro Latin Character Set)
Text entry method	Supported	Refer to guidelines for more information; for some TV Devices applications need to provide an on screen keyboard.
Image format	GIF, JPEG and PNG	
Media format	Refer to 3.4.4	

3.3.4. Input/key support

TV Devices shall support DOM Level 3 Keyboard Events ([60]) insofar as required by this specification. This includes the keyCode attribute and support for the following global VK_-key constants:

Key constant	Description	Support
VK_UP		M
VK_DOWN		M
VK_LEFT		M
VK_RIGHT		M
VK_ENTER	Typically mapped to the OK key	M
VK_PLAY		M
VK_PAUSE		M
VK_STOP		M
VK_FAST_FWD		M
VK_REWIND		M
VK_BACK		M
VK_0		C-M
VK_1		C-M
VK_2		C-M
VK_3		C-M
VK_4		C-M
VK_5		C-M
VK_6		C-M
VK_7		C-M
VK_8		C-M
VK_9		C-M
VK_RED		C-M
VK_GREEN		C-M
VK_YELLOW		C-M
VK_BLUE		C-M

Note: some manufacturers may implement these key constants in a Javascript library.

Note: Digit and color keys are available for developers, but these may not be readily available to users on certain TV Devices. Please see the developer guidelines for additional details.

3.3.5. User Agent String

TV Devices compliant to all mandatory items of this specification shall include the following user agent strings, separated by white space.

SmartTvA/4.0.0

3.3.6. Extended API for Smart TV Alliance

The Smart TV Alliance API provides an interface to the information associated with Smart TV Alliance features. The API is accessible by a JavaScript application running on the TV Device.

3.3.6.1. SmartTvA_API Object

SmartTvA_API object provides Smart TV Alliance specific methods and properties. The TV device shall create this object and then create properties and methods for this object defined in this specification. TV Devices shall allow JavaScript software running on the TV to extend this object with other properties and methods.

3.3.6.2. Methods for SmartTvA_API object

3.3.6.2.1. hasCapability

hasCapability			
Description	Query an optional or conditionally mandatory function on the TV Device.		
Arguments	Name	Type	Description
	Query	String	This string is the optional or conditionally mandatory function name to query support for. Note: this string is case sensitive.
	Params	String	This string contains additional query information.as a variable number of param arguments (param1, ..., paramN). The number of arguments depends on the query string (refer to hasCapability method arguments table). Note: this string is case sensitive.
Return value	Boolean	It is set to true if the given function is supported, otherwise false. For any value of Query and/or Params not specified by this document, false is returned.	

hasCapability method arguments table

Query	Param1	Param2	Description	Remark
3DSupport			Return value is set to true if TV Device can display 3D video in side-by-side and top-bottom formats, otherwise false.	See section 3.4.4.1
Key	numerickeys		Return value is set to true if global VK_ key constants corresponding to all defined numeric keys (VK_0, VK_1, ..., VK_9) are supported, otherwise false.	See section 3.3.4
	colorkeys		Return value is set to true if global VK_ key corresponding to all color keys (VK_RED, VK_GREEN, VK_YELLOW, and VK_BLUE) are supported, otherwise false.	See section 3.3.4
Multiscreen	AllJoyn		Return value is set to true if AllJoyn is supported, otherwise false.	See section 3.7.2
DRM	PlayReady	DASH	Return value is set to true if PlayReady in combination with MPEG-DASH are supported according to this	See section 3.5

			specification, otherwise false.	
	Widevine	AdaptiveStreaming	Return value is set to true if Widevine in combination with Widevine Adaptive Streaming and Widevine API are supported according to this specification, otherwise false. Note: No space is included in "AdaptiveStreaming".	See section 3.5
Multiaudio			Return value is always set to true. (Support is mandatory)	See section 3.4.7
TTML	inband		Return value is always set to true. (Support for TTML subtitles delivered via Smooth Streaming is mandatory)	See section 3.4.6.1.3
	outofband		Return value is set to true if TTML out of band subtitles are supported according to this specification, otherwise false.	See section 3.4.6.1.4
UHD			UHD is set to true if supported according to this specification, otherwise false.	See section 3.4.8
UHD	MainTier	5.1	Return value is set to true if Video codecs level up to Main Tier 5.1 is supported, otherwise false.	See section 3.4.8
UHD	HLS		Return value is set to true if HLS/MPEG2-TS supported, otherwise false.	See section 3.4.8
NSD			Network Service Discovery is supported according to this specification. This includes support for the mandatory discovery protocols.	See section 3.3.2
NSD	zeroconf		Network Service Discovery and zeroconf are supported according to this specification. If this flag is set to true, the simple query for 'NSD' shall also return true.	See section 3.3.2.2
NSD	DIAL		Network Service Discovery and DIAL are supported according to this specification. If this flag is set to true, the simple query for 'NSD' shall also return true.	See section 3.3.2.2
EMEMSE			Return value is set to true if EME/MSE are supported according to this specification, otherwise false.	See section 3.6.2 See section 3.6.3
EMEMSE	Widevine		Return value is set to true if Widevine EME is supported according to this specification, otherwise false.	See section 3.6.2

Usage: SmartTvA_API.hasCapability(query[,param1[,paramN]]);

3.3.6.2.2. *exit*

Exit	
Description	Notify the TV Device that an application is ready to exit and returns control back to the TV Device. Note: The end user's expectation is to navigate back to the source application that called the application. TV Device is expected to meet this user experience, but there could be exceptions and the behavior following this call is device dependent.
Arguments	<i>None</i>
Return value	<i>None</i>

Usage: SmartTvA_API.exit();

3.4. Video/audio streaming

3.4.1. HTML5 video/audio

See HTML Video Element and Media Element Events table in section 3.3

Note: The *src* element shall be set to the URL of the Smooth Streaming manifest or the MPEG-DASH MPD, or playlist file of HLS.

3.4.2. Streaming protocols

The following streaming protocols shall be implemented by TV Device:

Function	Detail	A/V content	Audio-only content	Reference
General	HTTP 1.1 with Range request	M	M	-
	HTTP streaming over SSL	M	M	3.4.2.1
Adaptive	HTTP Live Streaming	M		3.4.2.2
	Microsoft Smooth Streaming	M		[13]
	MPEG-DASH (ISOBMFF & CENC) according to HbbTV version 1.2.1 profile [26]	M		3.4.2.3

For audio only content, TV Device are only required to support progressive download.

3.4.2.1. HTTP streaming over SSL

TV Devices shall support reception of streams via HTTPS, with HbbTV root certificates for server authentication [19].

3.4.2.2. HLS

HTTP Live Streaming specification version 6, equivalent to protocol version 3, as specified in [12], shall be supported, with the following exceptions:

Tags	Reference	Status
EXT-X-PROGRAM-DATE-TIME	[12] Section 3.3.4	O
EXT-X-ALLOW-CACHE	[12] Section 3.3.5	O
EXT-X-DISCONTINUITY	[12] Section 3.3.9	O
All other tags	[12]	M

3.4.2.3 MPEG-DASH

TV Devices shall implement the MPEG-DASH [25] ISOBMFF Live profile, as further defined by HbbTV version 1.2.1 [26].

TV Devices shall support MPEG-DASH for unencrypted content. Details of MPEG-DASH support for encrypted content are specified in the DRM chapter of this document.

3.4.3. Streaming containers

The following container formats are supported:

Format	Detail	Status	Reference
MP4 File Format	Used in combination with HTTP, MPEG-DASH and Smooth Streaming.	M	[27]

MPEG2 Transport Stream	Used in combination with HTTP streaming and HLS.	M	[28]
------------------------	--	---	------

3.4.4. Streaming codecs

The supported media formats/codecs are described in this section.

3.4.4.1. Video Codecs

TV Devices shall support decoding of H.264 [29] encoded video and constrained according to [81], up to the following:

- 50 Hz H.264/AVC HDTV IRD and Bitstream ([81] section 5.7.4); and
- 60 Hz H.264/AVC HDTV IRDs and Bitstreams ([81] section 5.7.5)

This includes support for the following H.264 profiles and levels:

- Constrained Baseline Profile up to Level 3
- Main Profile up to Level 4.2
- High Profile up to Level 4.2

TV Devices advertising 3D capability according to section 3.3.5.2.1 shall support fullscreen 3D video in side-by-side and top-bottom formats.

3.4.4.1.1. Resolution and Aspect Ratio

TV Devices shall support the resolutions and aspect ratios as specified in tables 8, 10 and 11 of [81].

TV Devices should decode video with any resolution up to 1920x1080. However, it is not possible to test every possible resolution. Therefore, it is recommended that applications use the resolutions specified in tables 8, 10 and 11 of [81].

3.4.4.2 Audio Codecs

Codec	Detail	Status	Reference
HE-AAC	For A/V and audio only services.	M	[30]
AC-3	For A/V services only. Not supported for audio only services.	M	[31]
MP3	For audio only services. Not supported for A/V services.	M	[32]

3.4.5. MIME types for A/V media formats

MIME types shall apply as follows:

Format	MIME Type	Reference	Comment
MPEG-DASH	application/dash+xml	[25], [26]	Used for all DASH content.
Smooth Streaming	application/vnd.ms-sstr+xml	[13]	Used for all Smooth Streaming content.
HLS	application/vnd.apple.mpegurl	[12]	Used for all HLS content.
MP4 file format	video/mp4	[53]	Used for non-adaptive streaming only.
MPEG2-TS	video/mpeg	[53]	Used for non-adaptive streaming only.
HE-AAC	audio/mp4	[53]	Supported for progressive download only.
MP3	audio/mpeg	[53]	Supported for progressive download only.

These MIME types are used in the HTML5 video object.

3.4.6. Subtitles

This section specifies subtitle support.

3.4.6.1. Subtitles using native renderer

This specification supports the use of TTML [75] for subtitles, and a minimum profile is defined.

Implementation of this profile is mandatory for TV Devices. TV Devices are only required to support subtitles for full screen video.

3.4.6.1.1. Profile

Each timed text entry shall be contained as a <p> item. Timing information is specified as follows:
 tt:begin="hh:mm:ss:ff" tt:end="hh:mm:ss:ff".

Support of all other tags is optional. TV Devices may interpret other tags and attributes if they are present. If present, tags and attributes not supported by the TV Device shall be ignored. <p> items with valid timing information shall be displayed even if they contain attributes not supported by the TV Device.

An example TTML file is shown in Annex B.

3.4.6.1.2. Delivery

TTML subtitles may be delivered either in band or out of band. TV Device supporting this profile shall support in band method.

For in band delivery, Smooth Streaming is used, according to [13].

For out of band delivery, the TTML file is delivered using HTTP, independent of the streaming protocol.

3.4.6.1.3. In band method

Applications control subtitles using HTML5 TextTrack element. The following table shows the Microsoft Smooth Streaming mapping to HTML5 TextTrack element.

Support for this method is mandatory for TV Devices.

HTML5		Microsoft Smooth Streaming
Interface	Attribute	Mapping Info
TextTrackList	length	Number of StreamIndex tags of Text
TextTrack	label	Name attribute of StreamIndex tag of Text
TextTrack	language	Language attribute of StreamIndex tag of Text

TextTrack.mode (disabled / hidden / showing) is used for selecting the subtitle language.
 The TextTrack object list order shall be the same as the order of StreamingIndex.

3.4.6.1.4. Out of band method

Applications control subtitles using HTML5 TextTrack element. The native client in the TV Device acquires the TTML file by making an HTTP request to the URL given in the src attribute.

Support for this method is optional for TV Devices.

HTML5		
Interface	Attribute	Mapping Info
TextTrackList	length	Number of text tracks
TextTrack	language	Language of the text track
HTMLTrackElement	src	Address of the TTML file

See [76] for an informative explanation of the use of out of band subtitle delivery.

3.4.7. Multi Audio

TV Devices provide an interface for the content to provide multiple track audio streams.

3.4.7.1. HTML5 Media Element

Reference	Attribute	Support
[16] section 4.8.10	audioTracks	M

3.4.7.2. AudioTrackList Interface

Reference	Attribute	Support
[16] section 4.8.10.10.1	length	M
[16] section 4.8.10.10.1	[index]	M
[16] section 4.8.10.10.1	getTrackById	M

The length attribute must return the number of tracks represented by their objects at the time of request. TV Devices shall align the [index] list and the track list.
e.g.)

[Media Container File]	
<Track#1>	audioTracks [0] = Track#1
<Track#2>	audioTracks [1] = Track#2
<Track#3>	audioTracks [2] = Track#2
:	:
:	:

3.4.7.3. AudioTrack Interface

Reference	Attribute	Support
[16] section 4.8.10.10.1	id	M
[16] section 4.8.10.10.1	kind	M
[16] section 4.8.10.10.1	label	M
[16] section 4.8.10.10.1	language	M
[16] section 4.8.10.10.1	enabled	M

The *id* attribute shall return the identifier of the track, if it has one, or an empty string otherwise. The *kind* attribute shall return the category of the track, if it has one, or an empty string otherwise. The *label* attribute shall return the label of the given track. The *language* attribute shall return the RFC5646/BCP 47 language tag of the track, if it has one, or an empty string otherwise. The *enabled* attribute shall be set to true to select a track. The enabled attribute must return true if the track is currently selected, and false otherwise.

Limitation:

TV Devices do not need to support mixing of multiple audio tracks. Multiple audio tracks cannot be enabled (= true) simultaneously.

3.4.7.4. Mapping to Streaming Protocols

Multiaudio is supported for the following protocols.

Streaming Protocol	Support
Microsoft Smooth Streaming	M
MPEG-DASH (ISO BMFF)	M

3.4.7.5. Smooth Streaming

Interface	Attribute	Mapping
AudioTrackList	length	Number of StreamIndex[13] tags of Audio
AudioTrack	label	Name attribute of StreamIndex tag (1)

AudioTrack	language	Language attribute of StreamIndex tag (2)
------------	----------	---

(1): The *Name* attribute specifies the name of the track. This attribute is usually used to discriminate between multiple tracks of the same type, e.g. AAC, MP3.

(2): The *Language* attribute specifies the ISO639-2/T code of the used language, which consists of a three-letter code, e.g. eng, jpn.

3.4.7.6. MPEG-DASH

Interface	Attribute	Mapping
AudioTrackList	length	Number of Adaptation[25] tags of Audio
AudioTrack	language	The lang attribute of Adaptation tag (3)

(3): The *lang* attribute specifies the RFC5646/BCP 47 code of the used language, which consists of a two-letter code, e.g. en, jp.

3.4.8. Ultra High Definition Services

This section specifies features that are mandatory for TV Devices which are capable of decoding and rendering Smart TV Alliance Ultra High Definition video services. Video shall be presented in Ultra High Definition resolution only when the video is shown in full screen. When the video is not in full screen, the application resolution remains as specified in section 3.3.3.

Feature	Support
Ultra High Definition	O

Note: If UHD is supported, C-M functions are mandatory.

3.4.8.1. Profile

Function	Details	Support
Video format	Resolution: 3840 x 2160p	C-M
	Frame rate: 24, 25, 30 fps	C-M
	Frame rate: 48, 50, 60 fps (in conjunction with Main Tier 5.1)	O
Video codecs	HEVC [57] Profile: Main 10 Support for Tiles and WPP (wavefront parallel processing) are optional. (1)	C-M
	Level: Main Tier 5.0 (24, 25, 30 fps)	C-M
	Level: Main Tier 5.1 (48, 50, 60 fps)	O
Color space (2)	BT709	C-M
	BT2020	O
Audio codecs	As in section 3.4.4.2	C-M
Systems layer	[58] specifies the carriage of HEVC in MP4 file format Used in combination with HTTP, MPEG-DASH	C-M
	[59] specifies the carriage of HEVC in MPEG2 Transport Stream Used in combination with HLS	O
Content Protection	Combination of PlayReady + MPEG-DASH	C-M
Progressive and Adaptive Streaming	MPEG-DASH	C-M
	HLS	O

Notes:

1) The TV Device does not need to support streams where either `tiles_enable_flag` or `entropy_conding_sync_enabled_flag` (both in `picture_parameter_set`) is set to 1.

2) Indicates decoder and display capability of the TV Device.

3.5. Digital Rights Management

This chapter describes the Digital Rights Management methods to be supported by TV Devices. Note that this table does not apply for UHD – see section 3.4.8.

The following table shows the supported combinations of DRM and streaming format.

DRM	Detail	Status	Reference
PlayReady	According to Microsoft requirements. In combination with Microsoft Smooth Streaming.	M	[14]
PlayReady	According to Microsoft requirements. In combination with MPEG-DASH.	O	[14] [51]
Widevine	In combination with Widevine Adaptive Streaming.	O	[54]

3.5.1. PlayReady

This chapter describes the mandatory features of PlayReady, and the API provided to use PlayReady from an application.

3.5.1.1. PlayReady Features

The following table shows which features of PlayReady [14] are mandatory for TV Devices.

PlayReady feature	Status	Reference
Reactive license acquisition / License post-delivery	M	[14]
Proactive license acquisition / License pre-delivery	M	[14]
Domains	O	[14]
Metering	O	[14]
License query	O	[14]
License server URL override	M	[14]
Set Challenge CustomData	M	[14]
Set Challenge SOAP Header	O	[14]
Set Challenge HTTP Header	O	[14]

3.5.1.2. PlayReady API

Applications interface with PlayReady via the following interfaces:

- HTML5 video object
 - The *src* element shall be set to the URL of the Smooth Streaming manifest or the MPEG-DASH MPD.
 - The Smooth Streaming manifest shall include PlayReady signalling as specified in [14].
 - The MPEG-DASH MPD shall include PlayReady signalling as specified in [51].
 - In case of an error, the error attribute of the HTML5 video object shall be set to *MEDIA_ERR_DECODE*.
- OIPF DRM Agent
 - Applications shall use the OIPF DRM Agent API [52], as applied to PlayReady in [51]. The *oipfDrmAgent.sendDRMMessage* method is used to pass requests to PlayReady, and results are returned via *onDRMMessageResult*.

3.5.2. Widevine

The API provided to use Widevine from an application is described in [54]. Widevine DRM is optional for TV Devices.

3.6. EME and MSE

Note: this section is subject to change in future versions of this specification.

The Encrypted Media Extensions (EME) and the Media Source Extensions (MSE) are HTMLMediaElement extensions of W3C standard.

Smart TV Alliance TV Devices optionally support the EME/MSE. Applications which support EME/MSE must use both together. The use of either EME or MSE in isolation is not supported.

3.6.1. Parsing Model

TV Devices support the EME/MSE in the model that parse the adaptive streaming manifest file by JavaScript library.

3.6.2. EME

EME is a JavaScript API for HTMLMediaElement. The EME extends the video and audio elements to enable Digital Rights Management (DRM) protected content.

Function	Version	Status	Reference
EME	0.1b	○	[83]

3.6.2.1. DRM support through EME

The table below shows the required DRM support for TV Devices supporting EME.

DRM	Details	Status
PlayReady	EME with PlayReady.	C-M ^{*1}
Widevine	EME with Widevine.	○

*1: Mandatory if TV Device supports EME/MSE.

3.6.3. MSE

MSE is an interface to extend the video and audio elements in W3C. It is able to manage a media stream as a HTMLMediaElement.

Function	Version	Status	Reference
MSE	Candidate Recommendation	○	[84]

3.7. Multiscreen

3.7.1. DIAL

This chapter describes the use of the DIAL protocol (see [55]) for Smart TV Alliance applications. Two components are mandatory – DIAL Service Discovery and DIAL REST Service.

DIAL Service discovery enables a client to discover DIAL servers on its local network segment and obtain access to the DIAL REST Service on those devices.

The DIAL REST Service enables the client to query, launch and optionally stop applications on a Host Device and to retrieve the Application Instance URL.

3.7.1.1. Definitions for section 3.7.1

Definitions	Description
Companion Device	Mobile device (e.g. smart phone or a tablet) This is equivalent to the 2 nd screen in DIAL.

Client	Companion Device
Host	Smart TV Alliance TV Device
Host application	A Smart TV Alliance application or native application running on the Host
Client application	Application running on the Client
Look-up table	A table matching application name to web application URL (e.g. mySTAAApp -> www.mySTAAApp.com)
App-to-app communication	Communication between an application running on the companion device and a Smart TV Alliance Application running on the TV Device.

3.7.1.2. DIAL Service Discovery

Service discovery is defined in [55]. Refer to chapter 5 “Dial Service Discovery”.

3.7.1.3. DIAL REST Service

3.7.1.3.1. Application Resource

The Application Resource URL is defined in [55]. Refer to section 6.1 “Application Resource”.

3.7.1.3.2. Launching an Application

Launching an application is defined in the DIAL specification [55]. Refer to section 6.1.1 “Launching an Application”

The launch parameters shall be added to the application URL as a query string.

E.g. if the launch parameters are param1=value1¶m2=value2 and the url that has to be launched is

"www.mystaapp.com". The following URL shall be opened

"www.mystaapp.com?param1=value1¶m2=value2" (See also annex A2.1)

The host has access to a look-up table of the combination of the Application Name and the URL of the web application. The implementation of this look-up table is not specified in this document. The host browser accesses the URL in the look-up table. If the Application Name is not in the look-up table, then the host returns an HTTP response with response code 404 Not Found. Otherwise, refer to section 6.1.1.2 “Server Response” in [55].

Optionally the user may be notified the first time an application is launched through DIAL.

3.7.1.3.3. Stopping an Application

Stopping an application is defined in the DIAL specification [55]. Refer to section 6.1.2 “Stopping an Application”.

Support for stopping an application is optional.

3.7.1.3.4. Querying for Application Information

Querying for application information is defined in the DIAL specification [55]. Refer to section 6.1.3.1 “Client request”.

3.7.1.4. Multiscreen Application Naming Conventions

The application name of a multiscreen application shall be prefixed by the multiscreen Smart TV Alliance prefix.

The multiscreen Smart TV Alliance prefix shall be “org.smarttv-alliance”.

The maximum total size of the Smart TV Alliance Multiscreen Application Name shall be 277 bytes. The first 21 bytes are reserved for Smart TV Alliance prefix as above (“org.smarttv-alliance” + “.”). The rest of 256 bytes are used for the application name.

3.7.2. AllJoyn

This chapter describes the use of AllJoyn.
 The implementation of AllJoyn is optional.

The AllJoyn framework provides a mechanism that enables both members and application developers to create peer-to-peer applications to interact between one or more mobile devices, and the television. These applications include DIAL like functionality, as well as other more interactive multi-screen experiences.

AllJoyn enables applications to publish their functionality on the network using object oriented APIs. These APIs are discovered using either explicit advertising/discovery or finer grain announcements. These announcements enable services to advertise their capabilities as they are defined in the interfaces they expose. Applications and services are defined by these interfaces as they provide the mechanism for interaction over the network, they are the API definitions services expose. These APIs can also be marked as secured which will enable authentication and encryption between the applications.

AllJoyn is provided via an Open Source Project which is available on GitHub at the following URL: <http://alljoyn.github.com/download-source.html>, the implementation serves as the specification. There is also tutorial and general documentation available (see [56]).

3.7.2.1. Definitions for section 3.7.2

Definitions	Description
Peer to peer	The ability to communicate directly without having to mediate that communication via a server. Applications are said to be peer applications when they implement both service and client side functionality; that is, neither is the server or the client, but each is both.
Service	An application exposing APIs on the network
Client	An application using APIs published by a service
Interface	An API definition that is used by services to expose their functionality on the network. It can be thought of as a contract stating that the service will honor the functionality defined in the interface definition.
Method	A member of an interface. Allows a client to interact via the service and receive a reply
Signal	A member of an interface. Allows the asynchronous delivery of information from a service to a client.
Property	A member of an interface. A way of publishing some data on the network.

3.7.2.2. Application Launch Specification

The following is the specification for an application launch service similar to DIAL. It defines the *org.alljoyn.launch* standard AllJoyn interface.

3.7.2.2.1. Overview

The *org.alljoyn.launch* interface is implemented by an AllJoyn service on a target device such as a smart TV. The target device advertises the existence of its launch service by publishing a sessionless signal that declares the capabilities of the service. Client devices such as smart phones and tablets discover the existence of the service by receiving this sessionless signal. Once a client device has received the signal, it can present the information to the user who may then choose to connect in order access the service offered by the *org.alljoyn.launch* interface.

3.7.2.2.2. Security

The org.alljoyn.launch interface exposes methods that greatly affect the usability and user experience of the device. Therefore, it is assumed that some device manufacturers will want to control access to the org.alljoyn.launch interface to a restricted set of client devices. To provide this control, the org.alljoyn.launch interface is defined to be secure. Any of the standard AllJoyn authentication mechanisms (Pin Code, username/password or certificate based) can be used to authenticate the identity of connecting clients. Certain implementations may choose to loosen this requirement and provide non-secure access to the org.alljoyn.launch interface. However, client implementations that want to be interoperable with all implementations of org.alljoyn.launch should assume that authentication will be required.

3.7.2.2.3. Advertised Capabilities

AllJoyn nodes that wish to implement the org.alljoyn.launch interface should advertise their existence over AllJoyn by sending an “org.alljoyn.capabilities.Capabilities” sessionless signal that declares at least one BusObject (of any path name) that implements the org.alljoyn.launch interface.

AllJoyn nodes may also include vendor specific metadata in the org.alljoyn.capabilities.Capabilities sessionless signal that they emit. This metadata may be used to identify the brand and model of the device in addition to any other data that the vendor wishes to include. Although implementors are free to put any information they want in the metadata, there are some conventions for “typical” metadata entries. These “typical” metadata entries are:

Vendor: A String representation of the vendor name

Product: A String representation of the product name

FriendlyName: A String representation of a potentially user configurable name (e.g. “Living Room TV”)

3.7.2.2.4. Methods

The following methods are exposed by a BusObject that implements the org.alljoyn.launch interface:

GetAppInfo(in STRING appName, out AppInfo appInfo)

Inputs:

appName: Reverse domain name style application name (e.g. “com.company.appname”)

Output:

appInfo: Describes given appName (See AppInfo definition below)

Description:

Receive information about an application with a given application name

StartApp(in STRING appName, in BYTE[] appArgs, in StartAppOptions options, out StartAppResponse response)

Inputs:

appName: Reverse domain name style application name (e.g. “com.company.appname”)

appArgs: Application specific arguments passed to running app instance

options: See definition of StartAppOptions below

Outputs:

response: See StartAppResponse definition below.

Description:

Start the named application with given appArgs and options.

When StartApp is called, the current state of the requested application is obtained by the service in an implementation specific way. Then, based on the StartAppOptions that were specified when the application was originally started, one of the following actions are taken. (Please refer to section entitled “**Related AllJoyn Data Types**” for details.)

Current App State	startOptions.allowControl	Action
Installed but not running	N/A	App is started with given args and options
Running	true	appArgs are passed to running app instance
Running	false	No action. StartAppResponse indicates failure reason.

Not Installed	N/A	No action. StartAppResponse indicates failure reason.
AppName is unknown	N/A	No action. StartAppResponse indicates failure reason.

StopApp(in STRING appName, out StopAppResponse response)

Inputs:

appName: Reverse domain name style application name (e.g. "com.company.appname")

Output:

response: See StopAppResponse definition below.

Description:

Stop the named application.

InstallApp(in STRING appName, out InstallAppResponse response)

Inputs:

appName: Reverse domain name style application name (e.g. "com.company.appname")

Output:

response: See InstallAppResponse definition below.

Description:

Install the named application.

3.7.2.2.5. Signals

None

3.7.2.2.6. Properties

None

3.7.2.2.7. Related AllJoyn Data Types

The following struct data types are used in the Methods, Signals and Properties of the org.alljoyn.launch interface.

3.7.2.2.8. AppInfo

Member Name	Type	Description
Name	STRING (s)	Reverse domain name style name of application (e.g. "com.company.appname")
runState	UINT8 (y)	0 = Unknown Application 1 = Application is known but is not installed 2 = Application is installed but not running 3 = Application is running
applicationSpecific	DICTIONARY (a{sv})	Application specific data. Valid keys and value types are specified by the application. None are required.

3.7.2.2.9. StartAppOptions

Member Name	Type	Description
allowControl	BOOLEAN (b)	Set to true if other clients are allowed to "restart" or "stop" the application.
autoClose	BOOLEAN (b)	Set to true if the service should automatically close the application when the client/service session is closed.

3.7.2.2.10. StartAppResponse

MemberName	Type	Description
Status	UINT8 (y)	0 = Success (other fields of this struct contain valid info) 1 = Unknown App name 2 = App not installed 3 = App already started (restart not allowed) 4 = App failed to start (TV Device specific)

applicationResponse	STRING (s)	Application specific response to start request
----------------------------	------------	--

3.7.2.2.11. *StopAppResponse*

Member Name	Type	Description
Status	UINT8 (y)	0 = Success 1 = Unknown App name 2 = App not running

3.7.2.2.12. *InstallAppResponse*

Member name	Type	Description
Status	UINT8 (y)	0 = Success 1 = Unknown App name 2 = App failed to install

3.8. Smart Home

The Smart Home specification [85] is based on this version of the Smart TV Alliance technical specification. It includes methods to discover, monitor and control various kinds of smart home appliances. The specification is published as a separate document alongside this version of Smart TV Alliance technical specification.

Smart Home functions are optional for compliance with this version of the Smart TV Alliance specification.

4. History

4.1. Changes from version 3.0 to version 4.0

	v3.0	v4.0
HTML5		HTML5 section reorganised and clarified and updated.
CSS3		Clarified CSS3 section.
JavaScript		Added support for: <ul style="list-style-type: none"> - Network Service Discovery. - EME and MSE
AV Streaming	Audio track selection added to specification.	Audio track selection made mandatory. Updated, more detailed video codecs specification. Updated UHD section.
Subtitles	TTML subtitles added to specification.	TTML subtitle support with inband delivery via Smooth Streaming made mandatory.
Smart Home		Aligned with Smart Home v1.0 specification.

4.2. Changes from version 2.5 to version 3.0

	v2.5	v3.0
HTML5	Only DOM level 2 referenced.	DOM Level 3 Events and 4 referenced, as required by the supported HTML5 profile. Major enhancements to HTML5 profile.
CSS3		Major enhancements to CSS3 profile.
JavaScript		New APIs for device capability and application exit.
AV Streaming	Support for services up to HD resolution. HLS v2	Support for Ultra High Definition services. HLS v3 Support for audio track selection by application.
Subtitles		Support for subtitles using TTML.
Smart Home		Specification to be published in 2014.

4.3. Changes from version 2.0 to version 2.5

	v2.0	v2.5
CSS2	Obsolete CSS2.1 DRAFT specification referenced.	Latest CSS2.1 RECOMMENDATION referenced.
Multiscreen	No specific support.	Support added for: <ul style="list-style-type: none"> - DIAL - AllJoyn (optional) - Annex on the use of Websockets for app to app communication
HTTP		Added Smart TV Alliance identifier with version information in user agent string.
Misc		Editorial corrections and clarifications.

4.4. Changes from version 1.0 to version 2.0

	v1.0	v2.0
HTML5	Partial Support for: <ul style="list-style-type: none"> - HTML5 working draft: - audio tag - video tag 	Extended/Additional Support for: <ul style="list-style-type: none"> - HTML5 Elements - HTML5 Video Element - HTML5 Media Element Events - HTML5 Loading web pages - HTML5 Web application APIs - HTML5 User interaction - HTML5 Forms - HTML5 Syntax - HTML5 Related standards
CSS3	Partial Support for <ul style="list-style-type: none"> - CSS3 UI - CSS3 BG - CSS3 Media Queries 	Extended/Additional Support for: <ul style="list-style-type: none"> - CSS3 UI - CSS3 BG - CSS3 Media Queries - CSS3 Transforms - CSS3 Animations - CSS3 Color Module - CSS3 Fonts - CSS3 Image Values and Replaced Content - CSS3 Multi-column Layout - CSS3 Namespaces - CSS3 Selectors - CSS3 Text - CSS3 Transitions - CSSOM View
JavaScript	Partial Support for: <ul style="list-style-type: none"> - ECMAScript-262 5th edition 	Full and Mandatory Support for: <ul style="list-style-type: none"> - ECMAScript-262 5th edition
AJAX	Support for: <ul style="list-style-type: none"> - XMLHttpRequest 	Extended/Additional Support for: <ul style="list-style-type: none"> - XMLHttpRequest CORS - XMLHttpRequest Constructors - XMLHttpRequest Event Handlers - XMLHttpRequest States - XMLHttpRequest Request - XMLHttpRequest Response - XMLHttpRequest Events
AV Streaming	Support for: <ul style="list-style-type: none"> - WMV, VC-1, WMA 	<ul style="list-style-type: none"> - Support for: MPEG_DASH Removed Support for: <ul style="list-style-type: none"> - WMV, VC-1, WMA
DRM	Support for: <ul style="list-style-type: none"> - PlayReady but with no streaming protocols 	Support for: <ul style="list-style-type: none"> - PlayReady with OIPF DRM Agent Optional Support for:

		<ul style="list-style-type: none">- PlayReady with MPEG_DASH- Widevine with Widevine Adaptive Streaming
UI/UX Guideline	Combined device and app support list	Separated device and app support list

Annex A. Multiscreen (Informative)

A.1 Resolving URL from Application Name

Annex A describes examples of use cases for a server run by the manufacturer being used to resolve an Application URL using a look-up table, and application to application communication using W3C WebSocket API.

A.1.1 Resolving Application URL via Internet Server

DIAL Client in the companion device would request to launch an application in the host. The DIAL server in the TV Device needs to resolve the valid Application URL according to the requested Application Name. The Manufacturer's server provides the look-up table.

The form of look-up table and connections between look-up module and Manufacturer's Server(s) are dependent on the TV Device manufacturer.

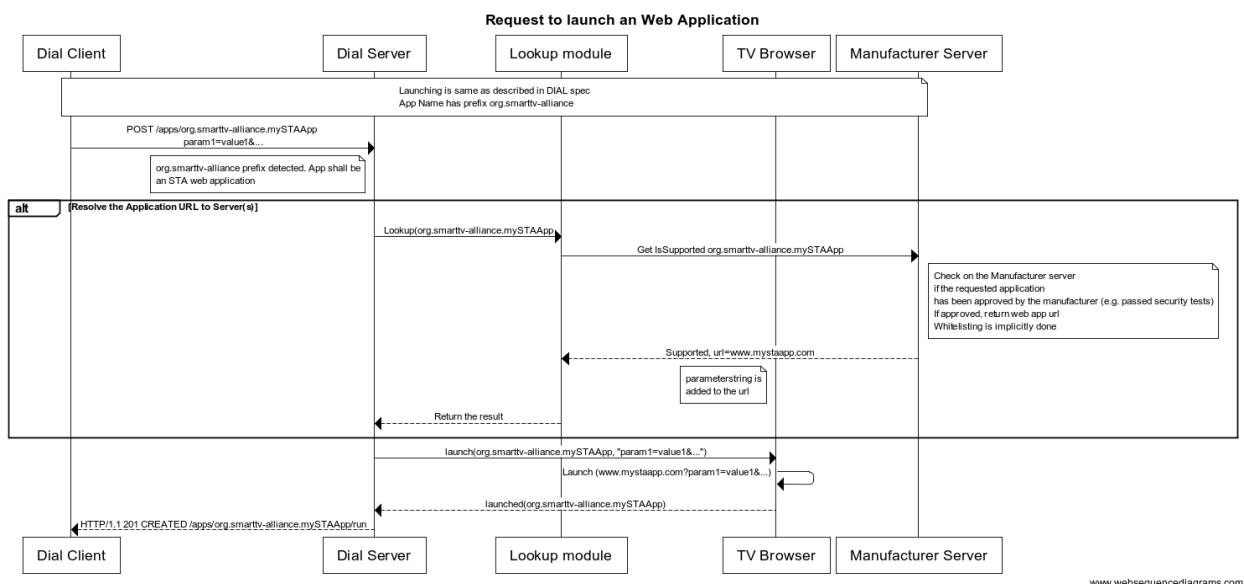


Figure A. 1 An example sequential diagram to retrieve an Application URL from Manufacturer's server

In figure A.1, DIAL Client requests to launch "org.smarttv-alliance.mySTAAApp" application in the host. DIAL Server in the TV Device receives a HTTP POST message. This calls a function of Lookup module which sends a request with Application Name to Manufacturer's Server(s), Manufacturer's Server(s) will response this message with one or more valid Application URL(s) to Lookup module. The Lookup module returns the results to DIAL Server. DIAL Server will launch the Application with the Application URL when the Application Name is a valid to this TV Device.

A.1.2 Look-up Table of Web Applications

The Lookup module in the TV Device can download one or more look-up tables from the Manufacturer's Server(s). This means the TV Device can maintain both the valid Application Name and Application URLs in local storage.

There are two ways to download look-up tables from the Manufacturer's Server(s).

- 1) Pull table
- 2) Notify the version change of table

The first approach is the easier way to update the local look-up table in the TV Device. The lookup module downloads the whole table or parts of the table in the Manufacturer's Server(s). Notification of look-up table version changes is recommended. As soon as the Manufacturer's Server notifies the update of look-up table to TV Device(s), the lookup module shall download changes to the table.

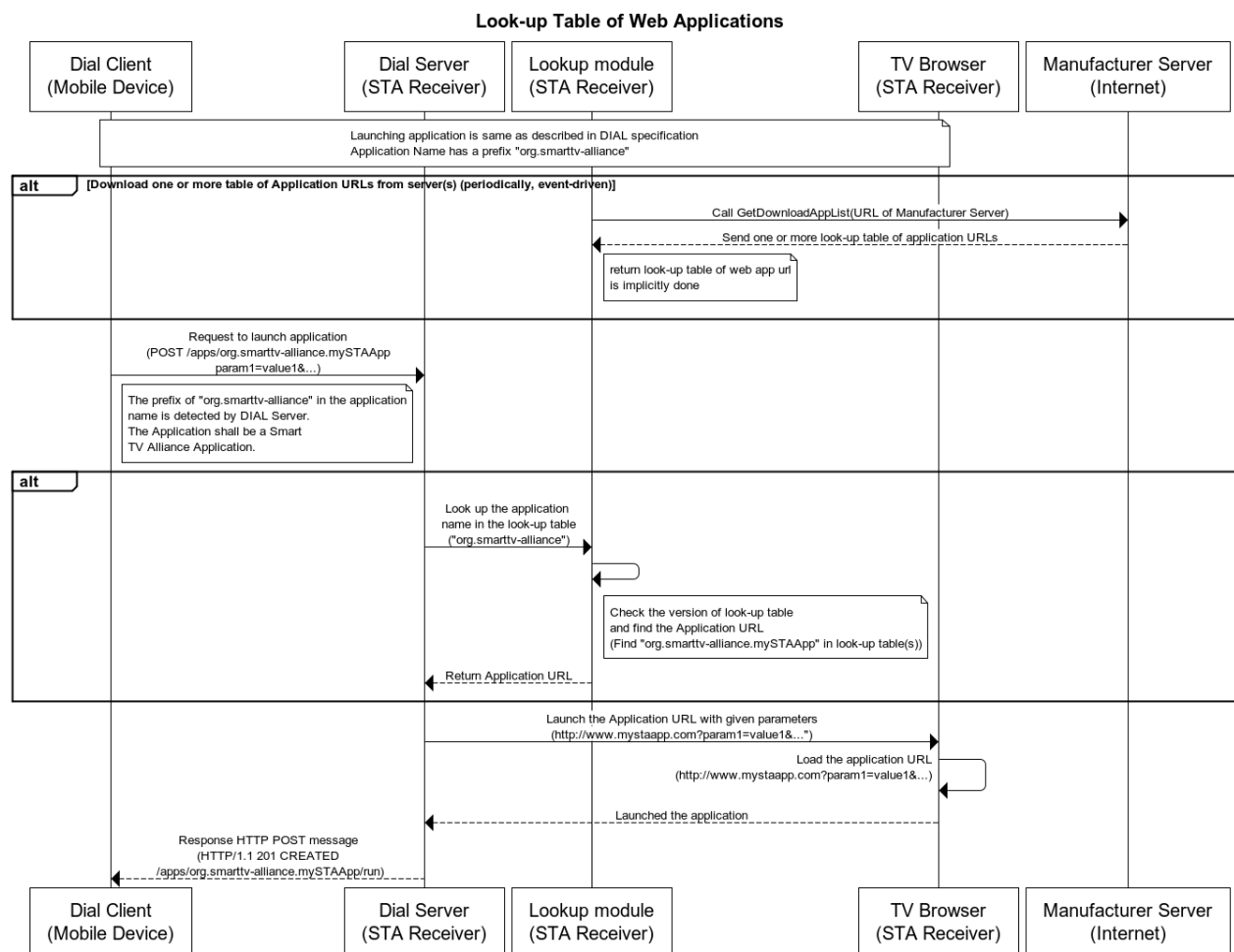


Figure A. 2 Update local look-up table(s) of Application URL(s) from Manufacturer's Server(s)

Figure A.2 shows the mechanisms of the look-up table update. The lookup module makes a connection to the Manufacturer Server to check the version of table in the Manufacture Server periodically. The Manufacturer Server gives the latest version of the table that is available to the lookup Module when the TV Device has an old version.

A.2 W3C WebSocket API for Application to Application Communication

This section describes the Application to Application Communication for Multiscreen Applications using DIAL and W3C WebSocket API.

A.2.1 Cloud based app to app communication

App to app communication is possible via a server in the cloud. A Service Provider can run its own WebSocket Server on the Internet. Both mobile and TV applications are provided with the address of this server by the Service Provider, and can make a WebSocket connection to this server. They then communicate with each other via this server. The server is responsible routing messages between the applications, using for example an ID communicated between the applications via the DIAL protocol.

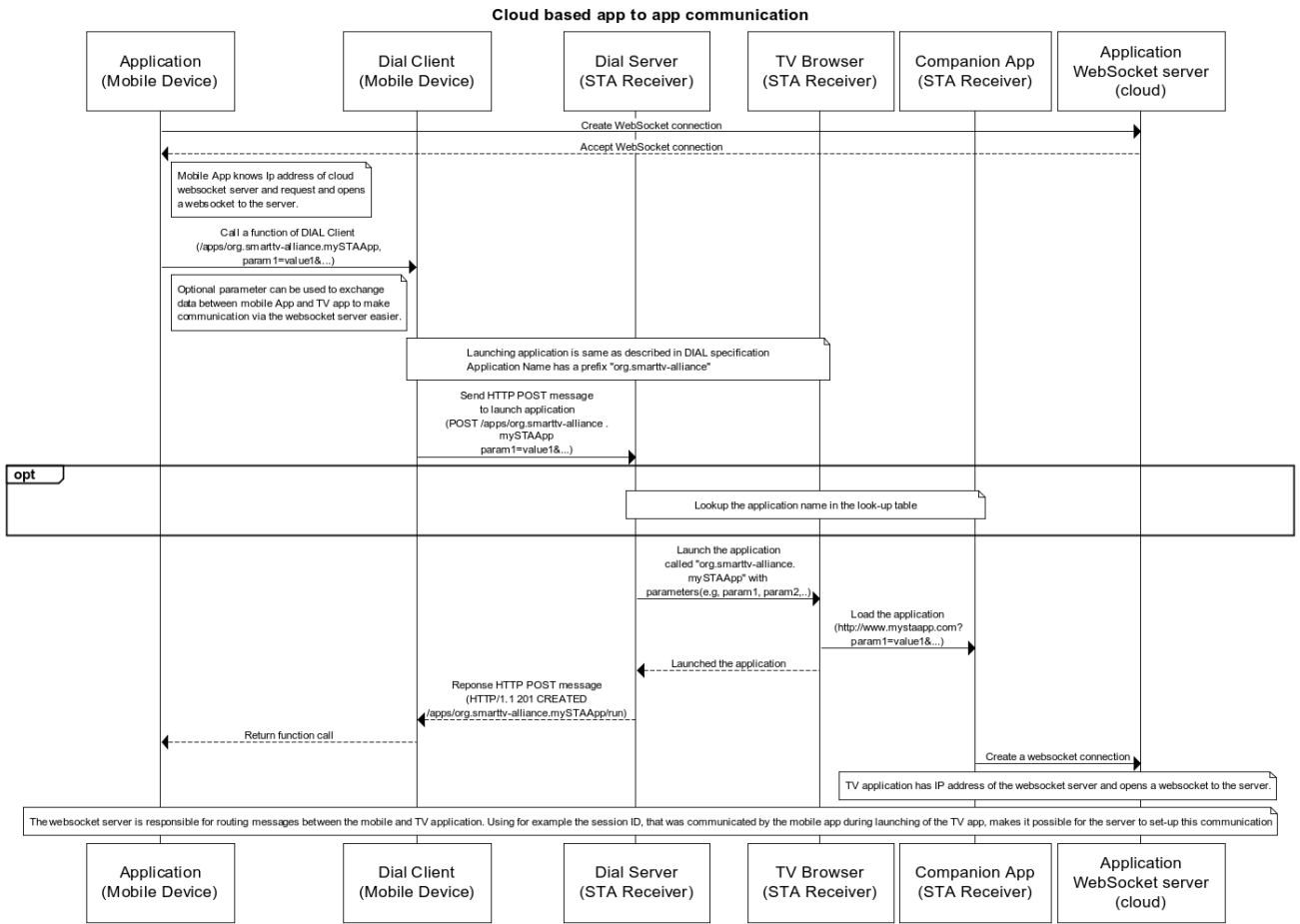


Figure A. 3 Cloud based app to app communication

Annex B. Subtitles (Informative)

B.1 Example TTML file

The following is an example TTML file using only features required by this specification.

```
<?xml version="1.0" encoding="utf-8"?>
<tt xml:lang="en"
    xmlns="http://www.w3.org/ns/ttml"
    xmlns:tts="http://www.w3.org/ns/ttml#styling"
    xmlns:ttm="http://www.w3.org/ns/ttml#metadata">
  <head>
</head>
  <body>
    <div>
      <p begin="00:00:01:00" end="00:00:04:00">First text.</p>
      <p begin="00:00:05:00" end="00:00:09:00">Second text.</p>
    </div>
  </body>
</tt>
```