Technical Specification

Version 2.5

Status:	Final
Version:	2.5.1
Date:	2nd May 2013
Author:	Smart TV Alliance inc.
Category:	Confidential
Reference:	SDKAPISPEC

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**
2.5.1 Final	**2013-05-02**	**Final version 2.5 for public release**

2. Introduction

2.1. Overview

This document sets out version 2.5 of the Smart TV Alliance specification. It is intended primarily for manufacturers, and describes the technical features to be implemented by end user 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 platforms. 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.

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. This will provide a user friendly environment for developers to create applications that run on the Alliance platform.

This document does not detail individual capabilities of the various members’ platforms, such as all supported codecs or fonts. It specifies the capabilities common to all platforms.

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

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
A/V	Audio / Video
AVC	Advanced Video Codec
CENC	Common Encryption
CFF	Common File Format
CSS3	Cascading Style Sheets
DIAL	DIscovery And Launch
DOM	Document Object Model
DRM	Digital Rights Management
GIF	Graphics Interchange Format
HbbTV	Hybrid Broadcast Broadband Television
HE-AAC	High Efficiency – Advanced Audio Codec
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)
MPEG	Moving Picture Experts Group
MP3	MPEG 1 – Layer 3 audio
MP4	MPEG4
MPD	Media Presentation Description

MPEG2	MPEG2 video codec
MPEG-DASH	MPEG Dynamic Adaptive Streaming over HTTP
OIPF	Open IPTV Forum
PNG	Portable Network Graphics
SD	Standard Definition
SDK	Software Development Kit
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
TLS	Transport Layer Security
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience
VoD	Video on Demand
XML	Extensible Markup Language

2.3. References

[1] Cross Origin Resource Sharing (CORS)
http://dvcs.w3.org/hg/cors/raw-file/tip/Overview.html

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

[3] ECMAscript Language Specification (Fifth Edition), December 2009,
http://www.ecma-international.org/publications/files/ECMA-ST-ARCH/ECMA-262%205th%20edition%20December%202009.pdf

[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-04

[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 Working Draft 29 March 2012
http://www.w3.org/TR/2012/WD-html5-20120329/

[17] HTML5 Server-sent events
http://www.w3.org/TR/2012/WD-eventsource-20120426/

[18] HTML5 Web storage
http://www.w3.org/TR/2011/WD-webstorage-20110208/

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

[20] HTML5 Workers
http://www.w3.org/TR/2011/WD-workers-20110901/

[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 - January 2012
http://www.w3.org/TR/2012/WD-XMLHttpRequest-20120117/

[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 - April 2012
http://www.w3.org/TR/2012/CR-css3-background-20120417/

[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 14496-15:2004, “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
http://www.w3.org/TR/2012/WD-html5-diff-20120329/

[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
http://www.w3.org/TR/2012/WD-css3-transforms-20120403/

[37] CSS3 Animations
http://www.w3.org/TR/2012/WD-css3-animations-20120403/

[38] CSS3 Color
http://www.w3.org/TR/2011/REC-css3-color-20110607

[39] CSS3 Fonts
http://www.w3.org/TR/2011/WD-css3-fonts-20111004/

[40] CSS3 Images
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
http://www.w3.org/TR/2012/WD-css3-text-20120119/

[44] CSS Transistions
http://www.w3.org/TR/2012/WD-css3-transitions-20120403/

[45] CSSOM View Module
http://dvcs.w3.org/hg/csswg/raw-file/tip/cssom-view/Overview.html

[46] HTML5 2D Context
http://www.w3.org/TR/2012/WD-2dcontext-20120329/

[47] HTML5 Web Messaging
http://www.w3.org/TR/2012/CR-webmessaging-20120501/

[48] HTML5 WebSockets
http://www.w3.org/TR/2012/WD-websockets-20120809/

[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

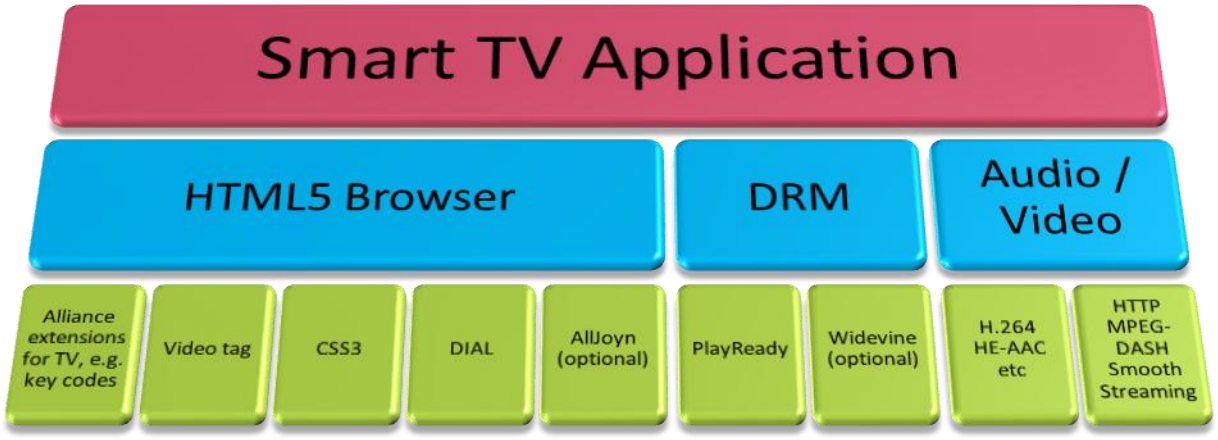
[56] AllJoyn SDK version 3.2
https://www.alljoyn.org/docs-and-downloads

2.4. Trademarks and copyrights

3. Technical Specification

3.1. Introduction

This chapter details the common Smart TV Alliance platform. 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 of the platform such as input methods, video streaming and DRM.
- DRM is supported on the Smart TV Alliance platform, 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 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 technical features of the platform, 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 applications shall not use such features when targeting the Alliance platform.

Status	Definition	Remarks
M	Mandatory, Fully Supported. All 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. **M(CSS2)** indicates mandatory support at CSS2 level.
C-M	Conditionally Mandatory. Implementation of this feature is optional, but devices that do implement this feature SHALL comply with this specification.	
P	Partially Supported.	
O	Optional. Details are defined and devices MAY support this feature.	

3.3. Browser

This chapter describes the level of support from the referenced standards that the platform browser shall meet.

3.3.1. HTML5 profile

High Level View of Support Status

The platform has support for the profile as listed below. Where partial is indicated, the detailed support is described.

Standard	Reference	Status	Remark
HTML5 working draft	[16]	P	See section 3.3.1.16.
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	
ECMAscript-262 5th ed.	[3]	M	
XMLHttpRequest Object (2)	[22]	P	See section 3.3.1.1
Cookies	[10], [9]	M	
CSS3 UI	[23]	P	See section 3.3.1.2
CSS3 BG	[24]	P	See section 3.3.1.3
CSS3 Media Queries	[2]	P	See section 3.3.1.4
CSS2.1	[11]	M	Only mandatory items from that specification are supported.
CSS3 Transforms	[8]	P	See section 3.3.1.5
CSS3 Animations	[37]	P	See section 3.3.1.6
CSS3 Color Module	[38]	P	See section 3.3.1.7
CSS3 Fonts	[39]	P	See section 3.3.1.8
CSS3 Image Values and		P	See section 3.3.1.9

Replaced Content			
CSS3 Multi-column Layout	[41]	P	See section 3.3.1.10
CSS3 Namespaces	[42]	P	See section 3.3.1.11
CSS3 Selectors	[34]. [35]	P	See section 3.3.1.12
CSS3 Text	[43]	P	See section 3.3.1.13
CSS3 Transitions	[44]	P	See section 3.3.1.14
CSSOM View	[45]	P	See section 3.3.1.15

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 4.3	XMLHttpRequest()	M
Event handlers			
	[22] section 4.5	onreadystatechange	M
States			
	[22] section 4.6	readystate	M
Request			
	[22] section 4.7.1	open()	M
	[22] section 4.7.2	setRequestHeader()	M
	[22] section 4.7.6	send()	M
	[22] section 4.7.8	abort()	M
Response			
	[22] section 4.8.1	status	M
	[22] section 4.8.2	statusText	M
	[22] section 4.8.3	getResponseHeader()	M
	[22] section 4.8.4	getAllResponseHeaders()	M
	[22] section 4.8.9	responseText	M
	[22] section 4.8.10	responseXML	M
Events			
	[22] section 4.9	readystatechange	M

3.3.1.2. CSS3 UI

Section	Reference	Description	Support
User interface selectors - pseudo classes			
	[34] section 5.11.3	:hover	M(CSS2)
	[34] section 5.11.3	:active	M(CSS2)
	[34] section 5.11.3	:focus	M(CSS2)
	[35] section 6.6.4.1	:enabled	M
	[35] section 6.6.4.1	:disabled	M
	[23] section 4.1.3	:default	M
	[23] section 4.1.4	:valid	M
	[23] section 4.1.4	:invalid	M
	[23] section 4.1.5	:in-range	M
	[23] section 4.1.5	:out-of-range	M
	[23] section 4.1.6	:required	M
	[23] section 4.1.6	:optional	M
	[23] section 4.1.7	:read-only	M
	[23] section 4.1.7	:read-write	M
	[35] section 5.11.2	:visited	M(CSS2)

Box Model addition			
	[23] section 6.1	box-sizing	P
Outline properties			
	[23] section 7.1	outline	M
	[23] section 7.2	outline-width	M
	[23] section 7.3	outline-style	M (CSS2)
	[23] section 7.4	outline-color	M
	[23] section 7.5	outline-offset	M
Resizing and overflow			
	[23] section 8.2	text-overflow	P
Pointing devices and keyboards			
	[23] section 9.2.2	nav-left, nav-right, nav-up, nav-down	M

3.3.1.3. CSS3 BG

Section	Reference	Description	Support
Backgrounds			
	[24] section 3.2	background-color	M
	[24] section 3.3	background-image	M
	[24] section 3.4	background-repeat	M
	[24] section 3.5	background-attachment	M
	[24] section 3.6	background-position	P
	[24] section 3.7	background-clip	M
	[24] section 3.8	background-origin	M
	[24] section 3.9	background-size	M
	[24] section 3.10	background	M
borders			
	[24] section 4.1	border-color	M
	[24] section 4.2	border-style	M
	[24] section 4.3	border-width	M
	[24] section 4.4	border	M
rounded corners			
	[24] section 5.1	border-radius	M
miscellaneous effects			
	[24] section 7.2	box-shadow	M

3.3.1.4 . CSS3 Media Queries

Section	Reference	Description	Support
media features			
	[2] section 4.1	width	P
	[2] section 4.2	height	P
	[2] section 4.3	device-width	P
	[2] section 4.4	device-height	P
	[2] section 4.5	orientation	P
	[2] section 4.6	aspect-ratio	P
	[2] section 4.7	device-aspect-ratio	P

Normal operation is fully supported, but behavior in certain erroneous conditions is not specified.

3.3.1.5. CSS3 Transforms

Section	Reference	Description	Support

	[36] section 6	Transform	M*
	[36] section 8	transform-origin	M*
2D Transform Functions			
	[36] section 13.1	matrix()	M
	[36] section 13.1	translate()	M
	[36] section 13.1	translateX()	M
	[36] section 13.1	translateY()	M
	[36] section 13.1	scale()	M
	[36] section 13.1	scaleX()	M
	[36] section 13.1	scaleY()	M
	[36] section 13.1	rotate()	M
	[36] section 13.1	skewX()	M
	[36] section 13.1	skewY()	M
Transform Function Lists			
	[36] section 14	transform function lists	M*

M*: Mandatory via browser extension, details to be included in developer guidelines.

3.3.1.6. CSS3 Animations

Section	Reference	Description	Support
Keyframes			
	[37] section 3.1	@keyframes	M
	[37] section 3.2	animation-name	M
	[37] section 3.3	animation-duration	M
	[37] section 3.4	animation-timing-function	M
	[37] section 3.5	animation-iteration-count	M
	[37] section 3.6	animation-direction	M
	[37] section 3.7	animation-play-state	M
	[37] section 3.8	animation-delay	M
	[37] section 3.9	animation-fill-mode	M
	[37] section 3.10	animation	M

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 3.1	font-family	M
	[39] section 3.4	font-style	M
	[39] section 3.5	font-size	M
	[39] section 3.7	font	M
Font Resources			
	[39] section 4.6	font-variant	M (CSS2)

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 full support	P

3.3.1.10. CSS3 Multi-column Layout

Section	Reference	Description	Support
The number and width of columns			
	[41] section 3.1	column-width	M
	[41] section 3.2	column-count	M
	[41] section 3.3	columns	P
Column gaps and rules			
	[41] section 4.1	column-gap	M
	[41] section 4.2	column-rule-color	M
	[41] section 4.3	column-rule-style	M
	[41] section 4.4	column-rule-width	M
	[41] section 4.5	column-rule	M
Spanning columns			
	[41] section 6.1	column-span	M

3.3.1.11. CSS3 Namespaces

Section	Reference	Description	Support
	[42]	@namespace	M

3.3.1.12. CSS3 Selectors

Section	Reference	Description	Support
Simple selectors			
	[34] section 5.4	Type selector (h1)	M (CSS2)
	[35] section 6.1.1	Type selectors and namespaces (ns|E)	M
	[34]	Universal selector (*)	M (CSS2)
	[34] section 5.8	Attribute presence and value selectors [att] [att=val] [att~=val] [att|=val]	M (CSS2)
	[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 (CSS2)
	[34] section 5.9	ID selectors (#)	M (CSS2)
Pseudo classes			
	[34] section 5.11.1	:link	M (CSS2)
	[34] section 5.11.1	:visited	M (CSS2)
	[34] section 5.11.3	:hover	M (CSS2)
	[34] section 5.11.3	:active	M (CSS2)
	[34] section 5.11.3	:focus	M (CSS2)
	[35] section 6.6.2	:target	M
	[34] section 5.11.4	:lang	M (CSS2)
	[35] section 6.6.4.1	:enabled	M
	[35] section 6.6.4.1	:disabled	M
	[35] section 6.6.4.2	:checked	M
	[35] section 6.6.5.1	:root	M
	[35] section 6.6.5.2	:nth-child()	M
	[35] section 6.6.5.3	:nth-last-child()	M
	[35] section 6.6.5.4	:nth-of-type()	M
	[35] section 6.6.5.5	:nth-last-of-type()	M
	[34] section 5.11.1	:first-child	M (CSS2)
	[35] section 6.6.5.7	:last-child	M
	[35] section 6.6.5.8	:first-of-type	M

	[35] section 6.6.5.9	:last-of-type	**M**
	[35] section 6.6.5.10	:only-child	**M**
	[35] section 6.6.5.11	:only-of-type	**M**
	[35] section 6.6.5.12	:empty	**M**
	[35] section 6.6.7	negation pseudo class :not(X)	**M**
Pseudo Elements			
	[34] section 5.12.1	::first-line	**M (CSS2)**
	[34] section 5.12.2	::first-letter	**M (CSS2)**
	[34] section 5.12.3	::before	**M (CSS2)**
	[34] section 5.12.3	::after	**M (CSS2)**
Combinators			
	[34] section 5.5	Descendant selectors	**M (CSS2)**
	[34] section 5.6	Child selectors	**M (CSS2)**
Sibling combinators			
	[34] section 5.7	Adjacent sibling combinator (+)	**M (CSS2)**

3.3.1.13. CSS3 Text

Section	Reference	Description	Support
Alignment and Justification			
	[43] section 7.1	text-align	**M**
Edge Effects			
	[43] section 10.3	text-shadow	**M**

3.3.1.14. CSS3 Transitions

Section	Reference	Description	Support
Transitions			
	[44] section 2.1	transition-property	**M***
	[44] section 2.2	transition-duration	**M***
	[44] section 2.3	transition-timing-function	**M***
	[44] section 2.4	transition-delay	**M***
	[44] section 2.5	Transition	**M***
Transition Events			
	[44] section 5	TransitionEvent	**M***
	[44] section 5	propertyName	**M**
	[44] section 5	elapsedTime	**M**
Animation of Property Types Support			
	[44] section 6	Color	**M**
	[44] section 6	Length	**M**
	[44] section 6	Percentage	**M**
	[44] section 6	integer	**M**
	[44] section 6	font weight	**M**
	[44] section 6	number	**M**
	[44] section 6	transform list	**M**
	[44] section 6	rectangle	**M**
	[44] section 6	visibility	**M**
	[44] section 6	shadow	**M**
	[44] section 6	gradient	**M**
	[44] section 6	list of above types	**M**
Properties from CSS			
	[44] section 7.1	background-color	**M**

	[44] section 7.1	border-bottom-width	M
	[44] section 7.1	border-left-width	M
	[44] section 7.1	border-right-width	M
	[44] section 7.1	border-spacing	M
	[44] section 7.1	border-top-width	M
	[44] section 7.1	bottom	M
	[44] section 7.1	color	M
	[44] section 7.1	font-size	M
	[44] section 7.1	height	M
	[44] section 7.1	left	M
	[44] section 7.1	letter-spacing	M
	[44] section 7.1	line-height	M
	[44] section 7.1	margin-bottom	M
	[44] section 7.1	margin-left	M
	[44] section 7.1	margin-right	M
	[44] section 7.1	margin-top	M
	[44] section 7.1	max-height	M
	[44] section 7.1	max-width	M
	[44] section 7.1	min-height	M
	[44] section 7.1	min-width	M
	[44] section 7.1	opacity	M
	[44] section 7.1	outline-color	M
	[44] section 7.1	outline-offset	M
	[44] section 7.1	outline-width	M
	[44] section 7.1	padding-bottom	M
	[44] section 7.1	padding-left	M
	[44] section 7.1	padding-right	M
	[44] section 7.1	padding-top	M
	[44] section 7.1	right	M
	[44] section 7.1	text-indent	M
	[44] section 7.1	top	M
	[44] section 7.1	vertical-align	P
	[44] section 7.1	visibility	M
	[44] section 7.1	width	M
	[44] section 7.1	word-spacing	M
	[44] section 7.1	z-index	M

M*: Mandatory via browser extension, details to be included in developer guidelines.

3.3.1.15. CSSOM View

Section	Reference	Description	Support
Extensions to the Window interface			
	[45] section 4	matchMedia()	M
	[45] section 4	screen	M
	[45] section 4	innerWidth	M
	[45] section 4	innerHeight	M
	[45] section 4	scrollX	M
	[45] section 4	pageXOffset	M
	[45] section 4	scrollY	M
	[45] section 4	pageYOffset	M
	[45] section 4	scroll()	M
	[45] section 4	scrollTo()	M
	[45] section 4	scrollBy()	M
	[45] section 4	screenX	M
	[45] section 4	screenY	M

	[45] section 4	outerWidth	M
	[45] section 4	outerHeight	M
The Screen Interface			
	[45] section 4.2	Screen	M
	[45] section 4.2	availWidth	M
	[45] section 4.2	availHeight	M
	[45] section 4.2	width	M
	[45] section 4.2	height	M
	[45] section 4.2	colorDepth	M
	[45] section 4.2	pixelDepth	M
Extensions to the Document Interface			
	[45] section 5	elementFromPoint()	M
Extensions to the Element Interface			
	[45] section 6.1	getClientRects()	M
	[45] section 6.1	getBoundingClientRect()	M
	[45] section 6	scrollIntoView()	M
	[45] section 6	scrollTop	M
	[45] section 6	scrollLeft	M
	[45] section 6	scrollWidth	M
	[45] section 6	scrollHeight	M
	[45] section 6	clientTop	M
	[45] section 6	clientLeft	M
	[45] section 6	clientWidth	M
	[45] section 6	clientHeight	M
Extensions to the HTMLElement Interface			
	[45] section 7	offsetParent	M
	[45] section 7	offsetTop	M
	[45] section 7	offsetLeft	M
	[45] section 7	offsetWidth	M
	[45] section 7	offsetHeight	M
Extensions to the Range Interface			
	[45] section 8	getClientRects()	M
	[45] section 8	getBoundingClientRect()	M
Extensions to the MouseEvent Interface			
	[45] section 9	screenX	M
	[45] section 9	screenY	M
	[45] section 9	pageX	M
	[45] section 9	pageY	M
	[45] section 9	clientX	M
	[45] section 9	clientY	M
	[45] section 9	x	M
	[45] section 9	y	M
	[45] section 9	offsetX	M
	[45] section 9	offsetY	M
The ClientRectList Interface			
	[45] section 10.1	ClientRectList	M
	[45] section 10.1	length	M
	[45] section 10.1	item()	M
The ClientRect Interface			

	[45] section 10.2	ClientRect	M
	[45] section 10.2	top	M
	[45] section 10.2	right	M
	[45] section 10.2	bottom	M
	[45] section 10.2	left	M
	[45] section 10.2	width	M
	[45] section 10.2	height	M

3.3.1.16. HTML5 detail

As HTML5 is still being defined, some of the supported API's are subject to change. This specification assumes that the platform browser has support for the mandatory items from HTML4 and thus is not exhaustive where it pertains to this minimum level of standard support. 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.

3.3.1.17. HTML5 Elements

Section	Reference	Description	Support
Elements in the DOM			
	[16] Section 3.2.2	HTMLElement	P
Global Attributes			
	[16] Section 3.2.3	Global Attributes	P
	[16] Section 3.2.3.9	Embedding custom non-visible data (data-*)	M
Dynamic markup insertion			
	[16] Section 3.4	Dynamic markup insertion	M
	[16] Section 3.5.6	outerhtml	M
Sections			
	[16] Section 4.4.2	Section elements	M
Grouping content			
	[16] Section 4.5.11	figure	M
	[16] Section 4.5.12	figcaption	M
Text-level semantics			
	[16] Section 4.6.19	mark	M
	[16] Section 4.6.27	wbr	M
Embedded content			
	[16] Section 4.8.7	Audio element	P
	[16] Section 4.8.6	Video element	P
	[16] Section 4.8.11	Canvas element	M
Interactive elements			
	[16] Section 4.11.2	summary	M
	[16] Section 4.11.4	menu element of type list	M

3.3.1.18. HTML5 Video element

Reference	Description	Support
[16] section 4.8.6	error	M
[16] section 4.8.6	src	M
[16] section 4.8.6	currentSrc	M
[16] section 4.8.10	networkState	M
[16] section 4.8.10	preload	M
[16] section 4.8.10	buffered (TimeRanges)	M
[16] section 4.8.6	canPlayType()	M

[16] section 4.8.10	readyState	M
[16] section 4.8.10	seeking	M
[16] section 4.8.6	currentTime	M
[16] section 4.8.6	duration	M
[16] section 4.8.10	paused	M
[16] section 4.8.10	defaultPlaybackRate	M
[16] section 4.8.10	playbackRate	M
[16] section 4.8.10	seekable (TimeRanges)	M
[16] section 4.8.10	ended	M
[16] section 4.8.10	autoplay	M
[16] section 4.8.10	loop	M
[16] section 4.8.6	play()	M
[16] section 4.8.6	pause()	M
[16] section 4.8.6	width	M
[16] section 4.8.6	height	M
[16] section 4.8.6	videoWidth	M
[16] section 4.8.6	videoHeight	M
[16] section 4.8.6	poster	M
[16] section 4.8.6	MediaError	M
[16] section 4.8.6	Media Element Events	P

3.3.1.19. HTML5 Media Element Events

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

3.3.1.20. HTML5 Loading web pages

Section	Reference	Description	Support
Window object			
	[16] section 5.2	Window object	M
Session history and navigation			
	[16] section 5.2	history	M
Offline web applications			
		Offline web applications	M

3.3.1.21. HTML5 Web application APIs

Section	Reference	Description	Support
Events			
	[16] section 6.1.6.2	oninput	M
	[16] section 6.1.6.2	onchange	M
System state and capabilities			
		The navigator object	M

3.3.1.22. HTML5 User interaction

Section	Reference	Description	Support
The hidden attribute			
	[16] section 7.1	hidden	M
Editing			
	[16] section 7.5	contenteditable	M

3.3.1.23. HTML5 Forms

Section	Reference	Description	Support
The fieldset element			
	[16] Section 4.10.4	name	M
	[16] Section 4.10.4	type	M
	[16] Section 4.10.4	elements	M
The label element			
	[16] Section 4.10.6	form	M
	[16] Section 4.10.6	htmlFor	M
The input element			
	[16] Section 4.10.7	autofocus	M
	[16] Section 4.10.7	form	M
	[16] Section 4.10.7	formAction	M
	[16] Section 4.10.7	formEnctype	M
	[16] Section 4.10.7	formMethod	M
	[16] Section 4.10.7	formNoValidate	M
	[16] Section 4.10.7	formTarget	M
	[16] Section 4.10.7	type=search	M
	[16] Section 4.10.7	type=tel	M
	[16] Section 4.10.7	type=url	M
	[16] Section 4.10.7.1	type=email	M
	[16] Section 4.10.7.1	type=number	M
	[16] Section 4.10.7.1	type=range	M
	[16] Section 4.10.7.1	type=checkbox	M
Common input element attributes			
	[16] Section 4.10.7.3.1	autocomplete	M
	[16] Section 4.10.7.3.2	dirname	M
	[16] Section 4.10.7.3.7	multiple	M
	[16] Section 4.10.7.3.9	pattern	M
	[16] Section 4.10.7.3.12	placeholder	M
The select element			
	[16] Section 4.10.9	select	M
The datalist element			
	[16] Section 4.10.10	datalist	M
The output element			
	[16] Section 4.10.15	output	M
The progress			

Section	Reference	Description	Support
element			
	[16] Section 4.10.16	progress	M
The meter element			
	[16] Section 4.10.17	meter	M
Association of controls and forms			
	[16] Section 4.10.18	Association of controls and forms	M
Form validation			
	[16] Section 4.10.21.3	willValidate	M
	[16] Section 4.10.21.3	validity	M
	[16] Section 4.10.21.3	validationMessage	M
	[16] Section 4.10.21.3	checkValidity()	M
	[16] Section 4.10.21.3	setCustomValidity()	M

3.3.1.24. 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	M
	[16] Section 8.2	HTML5 tree building	M

3.3.1.25. HTML5 Related standards

Below 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. Capabilities

The Smart TV platform adheres to these minimum capabilities:

Capability	Details	Remark
Screen resolution	1280 times 720 pixels (within safe area)	Static resolution - refer to the development guidelines for safe screen area information.
Color format	32 bits	
Supported fonts (or equivalent)	"Tiresias" (Screenfont) minimum size 18pts	font-family: sans-serif. (True Type font, Basic Euro Latin Character Set)
Text entry method	Supported	Refer to guidelines for more information; for some platforms an on screen keyboard needs to be implemented.
Image format	GIF, JPEG and PNG	

Media format	Refer to 3.4.4	

3.3.3. Input/key support

The platform supports DOM level 2 KeyEvents ([6]) and supports the following global VK_-key constants corresponding to key codes.
(for some manufacturers, key constants are implemented in a Javascript library):

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

3.3.4. User Agent String

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

SmartTvA/2.5.0

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 receivers:

Function	Detail	Status	Reference
General	HTTP 1.1 with Range request	M	-
	HTTP streaming over SSL	M	3.4.2.1
Adaptive	HTTP Live Streaming version 4	M	3.4.2.2
	Microsoft Smooth Streaming	M	[13]
	MPEG DASH (CFF & CENC) per HbbTV1.5 profile	M	3.4.2.3

3.4.2.1. HTTP streaming over SSL

Receivers 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 4, equivalent to protocol version 2, is mandatorily supported with the following tag exceptions:

Tags NOT Supported	Reference
EXT-X-PROGRAM-DATE-TIME	[12] Section 3.2.4
EXT-X-ALLOW-CACHE	[12] Section 3.2.5
EXT-X-DISCONTINUITY	[12] Section 3.2.8

Furthermore, it is recommended that service providers make content available at a minimum of SD resolution

3.4.2.3 MPEG-DASH

Receivers shall implement the MPEG-DASH [25] ISOBMFF Live profile, as further defined by HbbTV version 1.5 [26].

Receivers 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

Codec	Detail	Status	Reference
AVC	Supported profiles: BP@L3, MP@L3, HP@L4.	**M**	[29]

3D displays shall support 3D video in side-by-side and top-bottom formats.

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:

- For MPEG DASH content, as defined in [25] and [26]
- For Microsoft Smooth Streaming, as defined in [13]
- For HLS, as defined in [12]
- For non-adaptive HTTP streaming,
  - For MP4 file format: “video/mp4” [53]
  - For MPEG2-TS: “video/mpeg” or “video/mp2t” [53]
- For non-adaptive HTTP streaming audio only,
  - For HE-AAC: “audio/mp4” [53]
  - For MP3: “audio/mpeg” [53]

3.4.6. Subtitles

Subtitling of video content is supported through the use of a Javascript library in the application. The Popcorn library [50] is recommended, and receivers shall support the use of this library.

3.5. Digital Rights Management

This chapter describes the Digital Rights Management methods supported on the platform.

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]
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 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 devices.

3.6. Multiscreen

3.6.1. DIAL

This chapter decribes 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.

3.6.1.1. Definitions for section 3.6.1

Definitions	Description
Companion Device	Mobile device (e.g. smart phone or a tablet) This is equivalent to the 2nd screen in DIAL.
Client	Companion Device
Host	Smart TV Alliance 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. mySTAApp -> www.mySTAApp.com)
App-to-app communication	Communication between an application running on the companion device and a Smart TV Alliance Application running on the TV.

3.6.1.2. DIAL Service Discovery

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

3.6.1.3. DIAL REST Service

3.6.1.3.1. Application Resource

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

3.6.1.3.2. 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&param2=value2 and the url that has to be launched is "www.mystaapp.com". The following URL shall be opened
“www.mystaapp.com?param1=value1&param2=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.

3.6.1.3.3. Stopping an Application

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

3.6.1.3.4. Querying for Application Information

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

3.6.1.4. 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.

3.6.2. 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]).

3.6.2.1. Definitions for section 3.6.2

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.

3.6.2.2. 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.

3.6.2.2.1. Overview

The org.alljoyn.launch interface is implemented by an AllJoyn service on a target device such as a smartTV. 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.

3.6.2.2.2. 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.

3.6.2.2.3. *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”)*

3.6.2.2.4. 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 specifed 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.

3.6.2.2.5. Signals

None

3.6.2.2.6. Properties

None

3.6.2.2.7. Related AllJoyn Data Types

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

3.6.2.2.8. 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.

3.6.2.2.9. *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.

3.6.2.2.10. 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 (platform specific)
applicationResponse	STRING (s)	Application specific response to start request

3.6.2.2.11. *StopAppResponse*

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

3.6.2.2.12. InstallAppResponse

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

3.6.2.2.13. AllJoyn Introspection XML

The following XML defines the org.alljoyn.launch interface.

```
<node name="/anyobject"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="http://www.alljoyn.org/schemas/introspect.xsd">
    <interface name="org.alljoyn.launch">
        <method name="GetAppInfo">
            <arg name="appName" type="s" direction="in"/>
            <arg name="appInfo" type="(sqa{sv})" direction="out"/>
        </method>
        <method name="StartApp">
            <arg name="appName" type="s" direction="in"/>
            <arg name=”appArgs” type=”sy” direction=”in”/>
            <arg name=”options” type=”(bb)” direction=”in”/>
            <arg name="response" type="(ys)" direction="out"/>
        </method>
        <method name="StopApp">
            <arg name="appName" type="s" direction="in"/>
            <arg name=”response” type=”(y)” direction=”out”/>
        </method>
        <method name="InstallApp">
            <arg name="appName" type="s" direction=”in”/>
            <arg name=”response” type=”(y)” direction=”out”/>
        </method>
    </interface>
</node>
```

4. History

4.1. 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: - 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		Include corrections for v2.0 errata 1. Editorial corrections and clarifications.

4.2. Changes from version 1.0 to version 2.0

	v1.0	v2.0
HTML5	Partial Support for: - HTML5 working draft: - audio tag - video tag	Extended/Additional Support for: - 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 - CSS3 UI - CSS3 BG - CSS3 Media Queries	Extended/Additional Support for: - 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: - ECMAscript-262 5th edition	Full and Mandatory Support for: - ECMAscript-262 5th edition
AJAX	Support for: XMLHttpRequest	Extended/Additional Support for: - XMLHttpRequest CORS - XMLHttpRequest Constructors - XMLHttpRequest Event Handlers - XMLHttpRequest States - XMLHttpRequest Request - XMLHttpRequest Response - XMLHttpRequest Events
AV Streaming	Support for: - WMV, VC-1, WMA	- Support for: MPEG_DASH Removed Support for: - WMV, VC-1, WMA
DRM	Support for: - PlayReady but with no streaming protocols	Support for: - PlayReady with OIPF DRM Agent Optional Support for: - 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)

Annex 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 host 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 manufacturer.

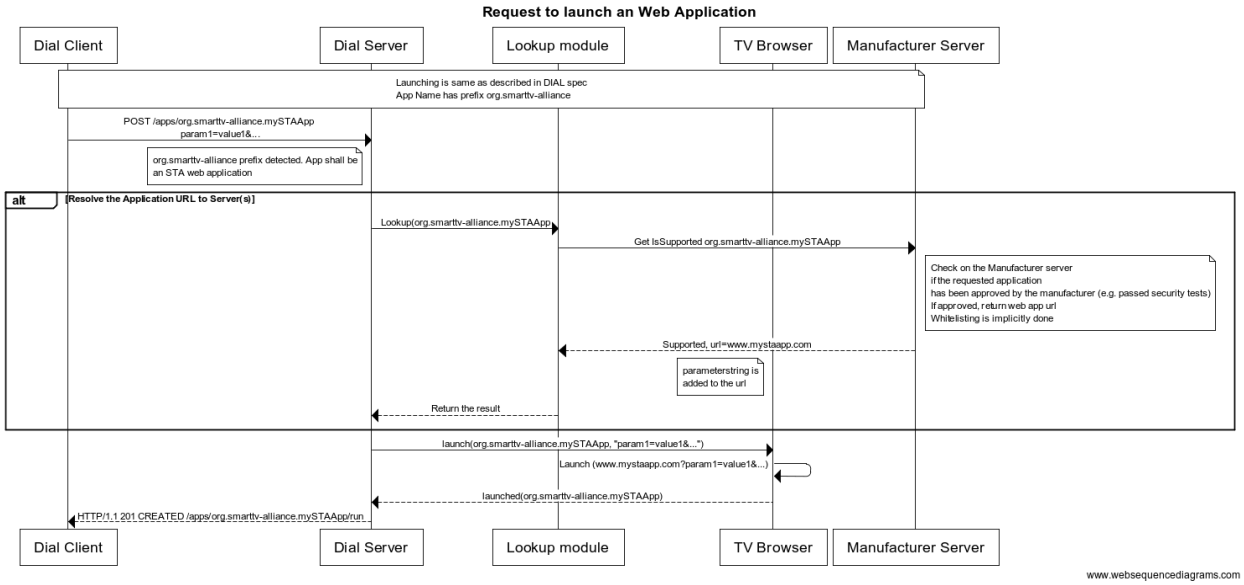


Figure A. 1 An example sequential diagram to retrieve an Application URL from Manufacturer's server

In figure A.1, DIAL Client requests to launch "org.smarttv-alliance.mySTAApp" application in the host. DIAL Server in the host 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 receiver

A.1.2 Look-up Table of Web Applications

The Lookup module in the Smart TV Alliance Receiver can download one or more look-up tables from the Manufacturer's Server(s). This means the receiver 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 Smart TV Alliance Receiver. 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 Smart TV Alliance Receiver(s), the lookup module shall download changes to the table.

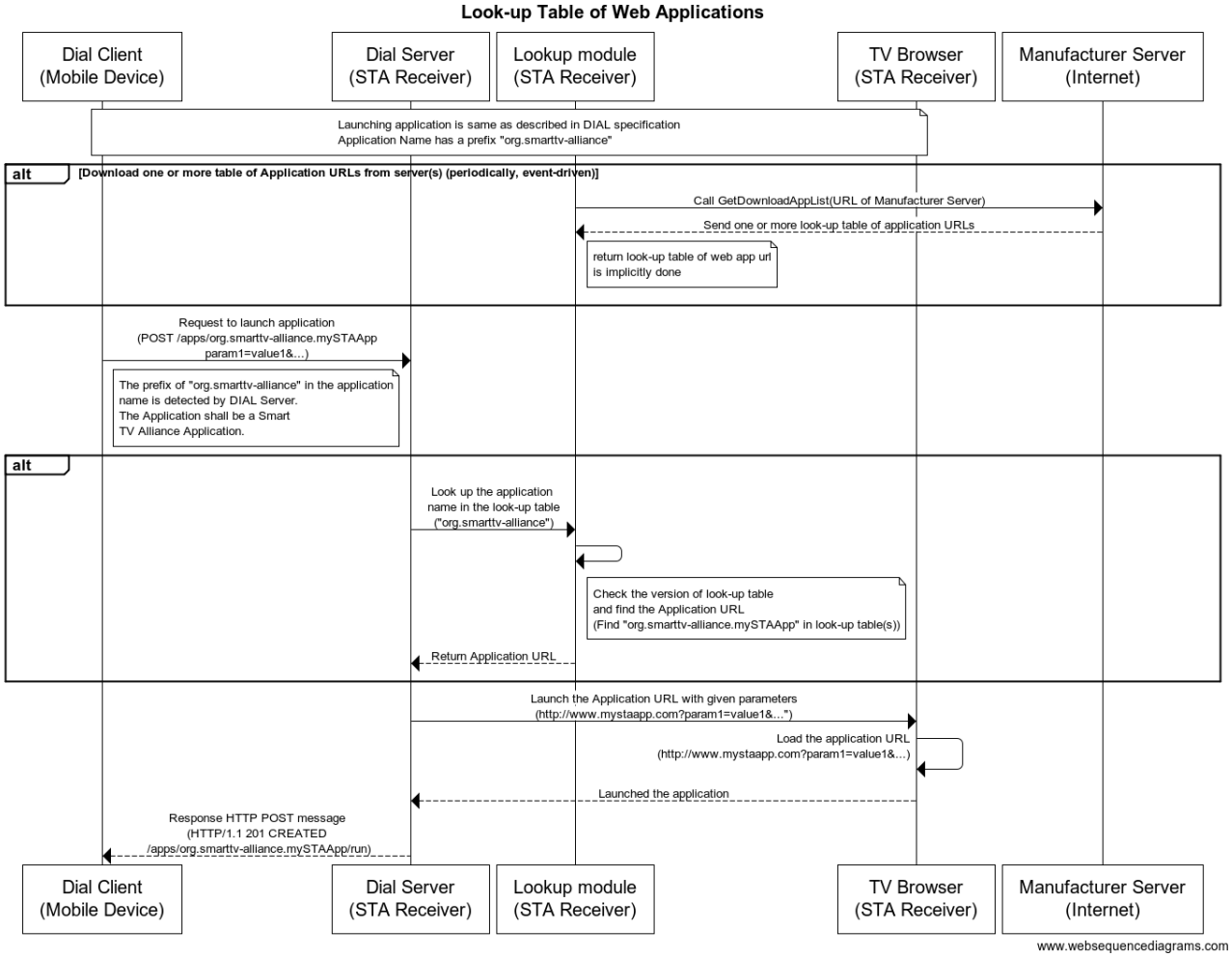


Figure A. 2 Update local look-up table(s) of Application URL(s) from Manufacturer's Server(s)

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 Smart TV Alliance Receiver has an old version.

Annex 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 (informative)

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.

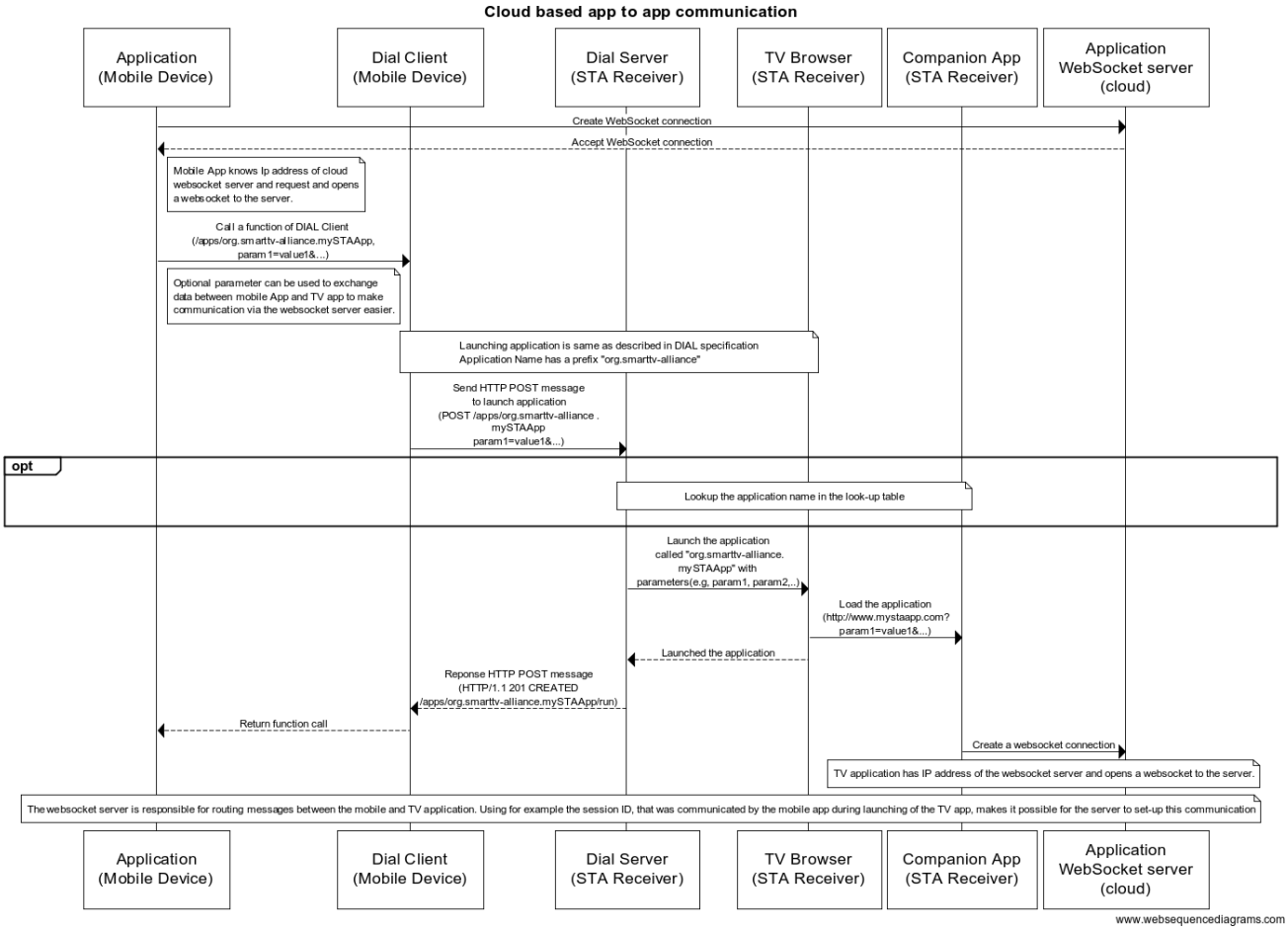


Figure A. 3 Cloud based app to app communication