OIPF

Release 1 IPTV Solution

V1.1 Errata 2

[2011-03-10]

Open IPTV Forum

Open IPTV Forum

Postal address

Open IPTV Forum support office address
650 Route des Lucioles - Sophia
Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 43 83
Fax: +33 4 92 38 52 90

Internet

Disclaimer

The Open IPTV Forum accepts no liability whatsoever for any use of this document.

Copyright Notification

Contents

Foreword

This document has been produced by the Open IPTV Forum (OIPF). It contains the second set of errata for the Release 1 V1.1 IPTV Solution specifications.

1 References

[OVIEW]	Open IPTV Forum, “Release 1 Specification, Volume 1 - Overview”, V1.1, October 2009.
[MEDIA]	Open IPTV Forum, “Release 1 Specification, Volume 2 - Media Formats”, V1.1, October 2009.
[META]	Open IPTV Forum, “Release 1 Specification, Volume 3 - Content Metadata”, V1.1, October 2009.
[PROT]	Open IPTV Forum, “Release 1 Specification, Volume 4 – Protocols”, V1.1, October 2009.
[DAE]	Open IPTV Forum, “Release 1 Specification, Volume 5 - Declarative Application Environment”, V1.1, October 2009.
[PAE]	Open IPTV Forum, “Release 1 Specification, Volume 6 - Procedural Application Environment”, V1.1, October 2009.
[CSP]	Open IPTV Forum, “Release 1 Specification, Volume 7 - Authentication, Content Protection and Service Protection”, V1.1, October 2009.
[ERRATA_1]	Open IPTV Forum, “Release 1 IPTV Solution – V1.1 Errata 1”, V1.0, July 2010.

2 Summary of Errata 2

Errata issues with the Release 1 V1.1 IPTV Solution specifications are categorised into one of the following:

- **Editorial ("E")** – where amendments do not affect any normative requirement in the specification.
- **Technical ("T")** - where amendments imply a technical change, but not one that causes any incompatibilities with an earlier revision of the V1.1 specification.
- **Critical ("C")** – where amendments imply a technical change that introduces some element of incompatibility with the published V1.1 specification.

Errata to the IPTV Solution specifications can have one of the following status settings:

- **Acknowledged** – the issue is acknowledged as an erratum and its resolution is under way.
- **Resolved –** the issue has been resolved and the erratum is in preparation.
- **Implemented** – the erratum is specified in the relevant normative section of the present document.

As an erratum is noted, its status can be expected to progress through these states in the indicated order, being updated in successive revisions of the present document. Issues that are notified but subsequently not deemed to be errata are not maintained in this document.

Table 1 below lists the issues addressed in "Errata 2", indicating which specification volume(s) are impacted. Issues are numbered through consecutive V1.1 Errata publications.

The subsequent sections contain the detailed errata for each V1.1 specification volume.

Issue number	Issue	Category	Impacted Volume(s)	Impacted Section(s)	Constituent Errata Issue Reference(s)	Status	Date
24	ProtocolCS	T	3	D.4	5.1	Implemented	13/08/2010
25	DVB-IPTV normative reference	T	4	1.1.1	6.1	Implemented	29/10/2010
26	Content protection in UE profile	C	4 1	D.2 A.4.13	6.2.1 3.1	Implemented	14/06/2010
27	UE profile XML inconsistencies	C	4 1	D.2 A.4.13	6.2.2 3.1	Implemented	14/06/2010
28	Mapping SDP attributes from DVB SD&S information	E	4	E.1	6.3	Implemented	14/06/2010
29	DRM capability in TR-135	T	4	K.1	6.4	Implemented	14/06/2010
30	Table cross-reference	E	5	9.3.10	7.1	Implemented	14/06/2010
31	DVB-GEM normative reference	T	6 2	2.1 1.1.1	4.1 8.2	Implemented Implemented	14/06/2010
32	CI Plus, Marlin and TLS references	T	7	2.1.1	9.1	Implemented	14/06/2010 1/02/2011 1/03/2011
33	CSPG-CI+ Discovery	T	7	4.2.3.3	9.2	Implemented	17/11/2010
34	CSPG-CI+ Control Channel	C	7	4.2.3.4.1.1 4.2.3.10.2	9.3	Implemented	17/11/2010
35	CI+ Registered Service Mode	T	7	4.2.3.9.1	9.5	Implemented	14/06/2010
36	DAE defined MIME types	T	5	7.1.1, 7.1.1.2	7.2	Implemented	18/11/10

37	Certificates	T	5	9.1	7.3	Implemented	18/11/10
38	Duplicated Properties and Methods	T	5	7.4.6.1, 7.8.2.2	7.4	Implemented	18/11/10
39	Gateway discovery	C	5	7.7.1.1, 7.7.1.3, 7.3.3.1	7.5	Implemented	18/11/10
40	getChannelConfig methods	T	5	7.10.1.1, 7.12.1.2, 7.13.11.2	7.6	Implemented	18/11/10
41	States for the <video/broadcast> object	C	5	7.13.1.1, 7.13.1.2, 7.13.1.3	7.7	Implemented	18/11/10, 17/12/10
42	Video and graphics integration model	T	5	4.9, 7.13.1.2, new Annex I	7.8	Implemented	18/11/10
43	Keyset capitalisation	E	5	7.2.5, 9.1	7.9	Implemented	18/11/10
44	Application model and lifecycle	T	5	4.3, 4.3.1, 4.3.2, 4.3.3, 4.3.7, 4.3.8, 4.4.2, 5.1.1.1, 5.1.1.3, 9.1	7.10	Implemented	18/11/10
45	Application types	E	5	5.1.1.1, 5.1.1.2, 5.1.1.3, 5.1.1.6, 8.3	7.11	Implemented	18/11/10
46	Iframe security issues	T	5	5.1.1.3, 5.1.2, new sections 5.1.3 and 10.1.3	7.12	Implemented	18/11/10
47	Method signatures	T	5	7.13.4.2, 7.14.4.2, 7.14.3.1, 7.15.1.2	7.13	Implemented	18/11/10
48	DOM-2 Event for onReadyToPlay	T	5	New section 7.14.8.3	7.14	Implemented	18/11/10
49	Cookies	T	5	9.1	7.15	Implemented	18/11/10
50	Multiple simultaneous applications	T	5	4.3, 4.4, 9.1	7.16	Implemented	18/11/10
51	Media resource management	T	5	7.13.1.1, 7.14.1.1	7.17	Implemented	18/11/10
52	Clarification of Parental Rating Values	T	5	7.16.2.2	7.18	Implemented	18/11/10
53	AVComponent types	T	5	7.13.4.2	7.19	Implemented	18/11/10
54	Data attribute of A/V object	T	5	8.2.2.1, 8.3	7.20	Implemented	18/11/10

55	OITF version inquiry	T	5	7.3.3.1, 8.1.1.1, 7.15.4.1	7.21	Implemented	18/11/10
56	Conflict resolution	E	5	4.1	7.22	Implemented	18/11/10
57	CRID usage	E	5	8.3	7.23	Implemented	18/11/10
58	Video modes	T	5	7.3.5	7.24	Implemented	18/11/10
59	onDRMMessageResult	T	5	7.6.1.1	7.25	Implemented	18/11/10
60	onDRMRightsError	T	5	7.13.6, 7.14.6	7.26	Implemented	18/11/10
61	DVB-MCAST URI Scheme	E	5	8.3, Annex H	7.27	Implemented	18/11/10
62	Return values in 7.9.1.2	T	5	7.9.1.2	7.28	Implemented	18/11/10
63	Merging text on broadcast-independent applications	E	5	5.2.5, 5.2.8	7.29	Implemented	18/11/10
64	Content types in a content catalogue	T	5	7.5.1	7.30	Implemented	18/11/10
65	Playspeeds array	T	5	7.13.2.2, 7.14.4.1	7.31	Implemented	18/11/10
66	A/V control object states and the seek method	T	5	Annex B	7.32	Implemented	18/11/10
67	Remote management API clarification	T	5	7.11.1	7.33	Implemented	18/11/10
68	Managing scheduled recordings	T	5	7.10.4, 7.10.7	7.34	Implemented	18/11/10
69	Duplication of video component selection	T	5	7.13.4, 7.14.5	7.35, Annex B	Implemented	18/11/10
70	Schemas	T	5	8.3	7.36	Implemented	18/11/10
71	Channel change by OIPF specific applications	T	5	7.13.1.1, 7.13.1.2	7.37	Implemented	18/11/10
72	Metadata API changes	T	5	7.12	7.57, Annex A	Implemented	18/11/10
73	DRMAgentPermission	E	6	Appendix E	8.3	Implemented	18/11/10

74	Asynchronous events from the CSP system	T	5	7.6.1.1/7.6.1.3.	7.38	Implemented	08/12/10
75	DRM Message Format Error	T	5	7.6.1.1	7.39	Implemented	25/11/10
76	Parental control unlocking	T	5	7.9.1.2	7.40	Implemented	25/11/10
77	Next and previous channels	T	5	7.13.1.1/2/3	7.41	Implemented	08/12/10
78	The data property of the video/broadcast object	T	5	7.13.1.2	7.42	Implemented	25/11/10
79	Exclude private IP addresses from the Application Boundary	T	5	5.1.3	7.43	Implemented	08/12/10
80	The isSeries property	T	5	7.10.2.2, 7.10.3, 7.10.4.1	7.44	Implemented	25/11/10
81	AVComponent Arrays and Collections	T	5	7.13.4.6, 7.13.4.1.2, 7.14.5.1.2	7.45	Implemented	08/12/10
82	DRM System Name	T	6	G	8.4	Implemented	17/12/10
83	DRM Agent Listener Result	T	6	G	8.5	Implemented	17/12/10
84	DRM Rights Error Event	T	6	G	8.6	Implemented	17/12/10
85	Mismatch between width and height attributes in OIPF and W3C	T	5	B	7.46	Implemented	17/12/10
86	Duplicated parental rating properties	T	5	7.10.2.2, 7.10.5	7.47	Implemented	17/12/10
88	Application Type notation	C	3	3.2.3.3.1	5.2	Implemented	21/12/10
89	Java package naming	C	6	All specification	8.1	Implemented	12/01/11
90	Media component API clarification	T	5	7.16.5.1.3	7.49	Implemented	12/01/11
91	PVR API issues	T	5	7.10.6.2, 7.16.2.5	7.50	Implemented	12/01/11
92	Mismatch between some VK key codes and W3C specification	C	5	B, 9.1	7.51	Implemented	14/01/11
93	Clarifying audio playback from memory	C	5	7.14.10	7.52	Implemented	14/01/11

94	Event handling fixes	T	5	7.5.1.1, 7.5.1.2, 7.5.4, 7.5.8, 7.10.4.1, 7.10.4.3, 7.10.7, 7.13.2.2, 7.13.2.4, 7.15.1.3	7.53	Implemented	14/01/11
95	PVR API clarifications	T	5	7.10.1-7.10.5, 7.13.2, 7.16.2.5	7.54	Implemented	14/01/11
96	Application loading errors	C	5	7.2.1.2, 7.2.1.4	7.55	Implemented	14/01/11
97	setFullScreen reference to CEA-2014	E	5	7.13.1.3	7.56	Implemented	14/01/11
98	Additional media format definitions	T	2	3	4.2	Implemented	14/01/11
99	Service usage of Direct UDP transport	T	2	4.3	4.3	Implemented	14/01/11
100	H.264/AVC GOP structure	T	2	5.1.6	4.4	Implemented	14/01/11
101	SDnS normative reference	T	3	1.1.1	5.3	Implemented	14/01/11
102	Alignment with CI-Plus 1.3	T	7	4.2.3.4.2.1, 4.2.3.7, 4.2.3.8	9.4	Implemented	1/02/2011
103	GBA Authentication	T	7	5.4.4.2	9.6	Implemented	1/02/2011
104	Home Network clarification	T	7	3.3, 5.5.2, 5.5.3, 5.5.4	9.7	Implemented	1/02/2011
105	Terminology – IMS or Communication Services	E	5	5.3.2.3, 7.1.1, 7.1.8, 7.8, 8.2.2.3, 9.2, 9.3.9	7.58	Implemented	01/02/11
106	Clarifying the current channel concept	E	5	New H.4	7.59	Implemented	01/02/11
107	Temporary Channel Objects, Recording and Metadata	T	5	7.13.1.3, 7.10.1.1	7.60	Implemented	01/02/11
108	Mapping from AVComponent to MPEG-2 TS and MP4 FF	T	5	New 8.4	7.61	Implemented	01/02/11
106	Parental rating errors	T	5	7.13.5, 7.14.6	7.62	Implemented	01/02/11
107	AV Control object state diagram clarifications	T	5	7.14.1.1, 4.4.5	7.63	Implemented	01/02/11
108	Feature Tags	E	5	7.8.3	7.64	Implemented	23/02/11

109	Parameter names in getParameter	T	5	7.11.1.2	7.65	Implemented	23/02/11
110	setBufferingStrategy	E	5	7.14.8.2	7.66	Implemented	23/02/11
111	Blocked and locked	E	5	7.16.2.3.1	7.67	Implemented	23/02/11
112	Clarify setChannel mapping for IPTV_SDS	T	5	8.2.2.2.2	7.68	Implemented	28/02/11
113	Missing registerDownloadxxx mappings	T	5	8.2.1.1	7.69	Implemented	28/02/11
114	Default Background Colour	T	5	4.4.6	7.70	Implemented	28/02/11
115	TLS Version	T	5,7	5.2.3 (volume 7), 9.1 (volume 5)	9.8 7.48	Implemented	1/03/2011 2/03/2011

Table 1 Release 1 V1.1 Errata 2 issues summary

3 Errata for Volume 1 - Overview

3.1 UE profile XML schema

The UE Profile XML schema is documented informatively in section A.4.13. It has two errata items against it in this edition of the V1.1 Errata, noted in section 6.1. The resulting new schema definition for UE Profile is specified normatively in section 6.2.2 of the present document.

4 Errata for Volume 2 – Media Formats

4.1 DVB-GEM normative reference

In section 1.1.1, normative references, the reference for DVB-GEM is updated to:

[GEM]	ETSI TS 102 728 V1.1.1 (2010-01), "Globally Executable MHP (GEM) Version 1.2.2"

4.2 Additional Media Format definitions

Additional media formats definitions applying optional audio codecs are identified in section 3:

- *For 25Hz video systems, to include media formats using MPEG-1 Layer 2 audio coding, and*
- *For 30Hz video systems, to include a media format using AC3 audio coding with MP4 systems layer.*

Tables 1and 2 are thus amended as follows:

System Format	Video Format	Audio Format	Mime Type
TS	AVC_HD_25 AVC_SD_25	HEAAC AC3 MPEG1_L2	video/mpeg
TTS	AVC_HD_25 AVC_SD_25	HEAAC AC3 MPEG1_L2	video/vnd.dlna.mpeg-tts
MP4	AVC_HD_25 AVC_SD_25	HEAAC AC3 MPEG1_L2	video/mp4
TS	MPEG2_HD_25 MPEG2_SD_25	AC3 MPEG1_L2	video/mpeg
TTS	MPEG2_HD_25 MPEG2_SD_25	AC3 MPEG1_L2	video/vnd.dlna.mpeg-tts

Table 1 A/V Media Formats for 25Hz video system

System Format	Video Format	Audio Format	Mime Type
TS	AVC_HD_30 AVC_SD_30	HEAAC AC3	video/mpeg
TTS	AVC_HD_30 AVC_SD_30	HEAAC AC3	video/vnd.dlna.mpeg-tts
MP4	AVC_HD_30 AVC_SD_30	HEAAC AC3	video/mp4

Table 2 A/V Media Formats for 30Hz video system

4.3 Service usage of Direct UDP transport

In section 4.3, the transport protocol for Scheduled Content and Streamed CoD are extended to include the option to use Direct UDP, as already specified in Volume 4.

The second clause of section 4.3 is amended to:

Unicast CoD services using the Direct UDP or RTP/UDP transport protocols SHALL use either the TS or the TTS systems layer format.

Table 11 is amended to:

Service	Transport protocol	Systems layer format
Scheduled content	Direct UDP or RTP/UDP	TS, TTS
Streamed CoD	Direct UDP or RTP/UDP	TS, TTS
Streamed CoD	HTTP	TS, TTS, MP4
Download CoD	HTTP	TS, TTS, MP4

Table 6 Systems layer formats for content services

4.4 H.264/AVC GOP structure

The H.264/AVC GOP structure constraints are relaxed so that additional slice types may be used.

The redundant clause on Reference B fields is removed.

The full text of section 5.1.6 is amended to:

All AVC format content provided in IPTV services SHALL conform to the following constraints in GOP structure:

- All slices in the same picture SHALL be of the same type.
- I picture: A picture with *slice_type*=7 or slice_type=2 for all the slices composing that picture or IDR picture
- P picture: A picture with *slice_type*=5 or slice_type=0 for all the slices composing that picture.
- B picture: A picture with *slice_type*=6 or slice_type=1 for all the slices composing that picture.
- Decoding order among I or P pictures SHALL be kept in their display order.
- P picture SHALL NOT refer to B pictures.
- Complementary reference field pair that includes I/P field SHALL NOT include B field.
- Reference B picture SHALL refer to the following.
    - I or P frames or complementary reference field pairs of I or P pictures that immediately precedes/follows in display order.
- Non-reference B picture SHALL refer to the following.
    - I or P frames or complementary reference field pairs of I or P pictures that immediately precedes/follows in display order.

- A reference B frame or a complementary reference field pair of reference B pictures that immediately precedes/follows in display order and is present between “pic1” and “pic2” in display order. Here, “pic1” is immediately preceding I or P picture and “pic2” is immediately following I or P picture.

5 Errata for Volume 3 – Content Metadata

5.1 ProtocolCS

The ProtocolCS schema specified in section D.4 was missing protocol entries for "Unmanaged Scheduled Content", i.e. for termId's "igmp-rtp-udp" and "igmp-udp". The labels for "Managed Scheduled Content…" and the referenced table number in Volume 4 are also corrected, and the version is revised.

The amended text for section D.4 is:

The following Classification Scheme is introduced according to the protocols defined Table 64 of Annex F.1 in Protocols Specification [PROT].

```
<?xml version="1.0" encoding="UTF-8"?>
<ClassificationScheme uri="urn:oipf:cs:ProtocolCS:2010">
    <!-- schema filename is  cs-ProtocolCS.xml -->
    <!--
        This schema is copyrighted by the Open IPTV Forum ("OIPF") and distributed in conjunction
        with Release 1 of the IPTV Solution Specification.

        Disclaimer
        The Open IPTV Forum members accept no liability whatsoever for any use of this document.
        This specification provides multiple options for some features. The Open IPTV Forum Profiling
        specification will complement the Release 1 specifications by defining the Open IPTV Forum
        implementation and deployment profiles. Any implementation based on Open IPTV Forum
        specifications that does not follow the Profiling specifications cannot claim Open IPTV Forum
        compliance.

        Copyright Notification
        No part may be reproduced except as authorized by written permission.
        Any form of reproduction and/or distribution of these works is prohibited.
        Copyright 2009 © Members of the Open IPTV Forum
        All rights reserved.
    -->
    <Term termId="sip-igmp-rtp-udp">
        <Name xml:lang="en">sip-igmp-rtp-udp </Name>
        <Definition xml:lang="en">Scheduled Content over RTP</Definition>
    </Term>
    <Term termId="sip-igmp-udp">
        <Name xml:lang="en">sip-igmp-udp</Name>
        <Definition xml:lang="en">Scheduled Content over UDP</Definition>
    </Term>
    <Term termId="sip-rtsp-rtp-udp">
        <Name xml:lang="en">sip-rtsp-rtp-udp</Name>
        <Definition xml:lang="en">Managed CoD Streaming over RTP</Definition>
    </Term>
    <Term termId="sip-rtsp-udp">
        <Name xml:lang="en">sip-rtsp-udp</Name>
        <Definition xml:lang="en">Managed CoD Streaming over direct UDP</Definition>
    </Term>
    <Term termId="igmp-rtp-udp">
        <Name xml:lang="en">igmp-rtp-udp </Name>
        <Definition xml:lang="en">Unmanaged Scheduled Content over RTP</Definition>
    </Term>
    <Term termId="igmp-udp">
        <Name xml:lang="en">igmp-udp</Name>
        <Definition xml:lang="en">Unmanaged Scheduled Content over UDP</Definition>
    </Term>
    <Term termId="rtsp-rtp-udp">
        <Name xml:lang="en">rtsp-rtp-udp</Name>
        <Definition xml:lang="en">Unmanaged CoD Streaming over RTP</Definition>
    </Term>
    <Term termId="http-get">
        <Name xml:lang="en">http-get</Name>
        <Definition xml:lang="en">Managed/Unmanaged CoD Streaming/Download over HTTP</Definition>
    </Term>
</ClassificationScheme>
```

5.2 Application Type notation

The MIME types used to signal application types in section 3.2.3.3.1 are revised to those registered for this purpose. This errata supersedes section 5.2 of [ERRATA_1].

Section 3.2.3.3.1 is changed to:

3.2.3.3.1 Type Element of ApplicationDescriptor

The type element of the application descriptor defines the actual application environment that is used by the application [TS102809]. The MIME type of the application is carried in the OtherApp element of the type element and takes one of the following values:

- for DAE CE-HTML applications this value SHALL be “`application/vnd.oipf.dae.xhtml+xml`”
- for DAE SVG applications this value shall “`application/vnd.oipf.dae.svg+xml`”
- for PAE applications this value SHALL be “`application/vnd.oipf.pae.gem`”

5.3 SDnS normative reference

In section 1.1.1 the normative reference to SD&S is changed to the relevant ETSI specification:

[SDNS]	ETSI, TS 102 034 V1.4.1 (2009-08), “Digital Video Broadcasting (DVB);Transport of MPEG-2 TS Based DVB Services over IP Based Networks”

6 Errata for Volume 4 – Protocols

6.1 DVB-IPTV normative reference

In section 1.1.1 the normative reference to the DVB-IPTV specification is amended to:

[TS102034]	ETSI, TS 102 034 V1.4.1 (2009-08), “Digital Video Broadcasting (DVB);Transport of MPEG-2 TS Based DVB Services over IP Based Networks”

6.2 UE profile XML structure

This erratum consists of two separate issue that imply changes to the UE Profile XML structure. The first adds content protection support information and the second resolves one problem found with the UE Profile schema syntax, and replaces the multiple extension mechanisms used with a single, common approach.

6.2.1 Content protection in UE profile

The capability to provide information about the supported content protection methods (via the DVB CA_System_ID), also CSPG Gateway type, is added to the UE Profile XML schema. The revised XML schema for the UE Profile in Annex D.2 of Volume 4, and its repetition in Annex A.4.13.in Vol. 1, is shown in the next sub-section, incorporating that erratum as well.

6.2.2 UE Profile XML inconsistencies

This erratum conveys two improvements in the UE Profile XML structure:

- *Incorrect use of the “final” attribute*
- *Adoption of a common single approach for extensions of je XML schema,*

The amended UE Profile XML structure, including the changes outlined in the previous sub-section, is;

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:oipf:iptv:UEProfile:2008-1" xmlns:tns="urn:oipf:iptv:UEProfile:2008-1"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tva="urn:tva:metadata:2007" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <!-- schema filename is  iptv-UEProfile.xsd -->
    <xs:annotation>
        <xs:documentation xml:lang="en">
            This schema is copyrighted by the Open IPTV Forum ("OIPF") and distributed in conjunction
            with Release 1 of the IPTV Solution Specification.

            Disclaimer
            The Open IPTV Forum members accept no liability whatsoever for any use of this document.
            This specification provides multiple options for some features. The Open IPTV Forum Profiling
            specification will complement the Release 1 specifications by defining the Open IPTV Forum
            implementation and deployment profiles. Any implementation based on Open IPTV Forum
            specifications that does not follow the Profiling specifications cannot claim Open IPTV Forum
            compliance.

            Copyright Notification
            No part may be reproduced except as authorized by written permission.
            Any form of reproduction and/or distribution of these works is prohibited.
            Copyright 2010 © Members of the Open IPTV Forum
            All rights reserved.
        </xs:documentation>
    </xs:annotation>
    <xs:import namespace="urn:tva:metadata:2007" schemaLocation="imports/tva_metadata_3-1_v141.xsd"/>
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Defines the capabilities of the UE that is currently
            associated with the user
        </xs:documentation>
    </xs:annotation>
    <xs:element name="UEInformation" type="tns:tUEProfile"/>
    <xs:complexType name="tUEProfile">
        <xs:sequence>
            <xs:element name="UserEquipmentID" type="tns:tUEID"/>
            <xs:element name="UserEquipmentClass" type="tns:tUserEquipmentClass"/>
            <xs:element name="Resolution" type="tns:tResolution" minOccurs="0"/>
            <xs:element name="SupportedEncodings" type="tns:tSupportedEncodings"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="SupportedContentProtection" type="tns:tSupportedContentProtection"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="IPEncapsulations" type="tns:tIPEncapsulations"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="tUEID" final="list restriction">
        <xs:annotation>
            <xs:documentation>
                <xs:label xml:lang="en">User Equipment ID</xs:label>
                <xs:definition xml:lang="en">
                    Unique Identifier for the UE(to be specified)
                </xs:definition>
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:minLength value="0"/>
            <xs:maxLength value="16"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tUserEquipmentClass" final="list restriction">
        <xs:annotation>
            <xs:documentation>
                <xs:label xml:lang="en">User Equipment class</xs:label>
                <xs:definition xml:lang="en">
                    Specifies the type of UE
                </xs:definition>
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:enumeration value="OITF-TV"/>
            <xs:enumeration value="OITF-STB"/>
        </xs:restriction>
```

```
</xs:simpleType>
<xs:complexType name="tResolution">
    <xs:attribute name="HorizontalSize" type="xs:integer">
        <xs:annotation>
            <xs:documentation>
                horizontal size in pixels of the screen
            </xs:documentation>
        </xs:annotation>
    </xs:attribute>
    <xs:attribute name="VerticalSize" type="xs:integer">
        <xs:annotation>
            <xs:documentation>
                vertical size in pixels of the screen
            </xs:documentation>
        </xs:annotation>
    </xs:attribute>
    <xs:attribute name="Rotate" type="xs:boolean">
        <xs:annotation>
            <xs:documentation>
                set to TRUE if the screen can be rotated (horizontal
                becomes vertical)
            </xs:documentation>
        </xs:annotation>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="tSupportedContentProtection">
    <xs:annotation>
        <xs:documentation>
            <xs:label xml:lang="en">Content Protection</xs:label>
            <xs:definition xml:lang="en">
                Specifies the supported content protection system (eg. "urn:dvb:casystemid:19188")
                with optionally the gateway (eg. "CI+" or "DTCP-IP") and supported protected formats
            </xs:definition>
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="ProtectedFormat" type="tns:tProtectedFormat"
            minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="ContentProtectionSystemID" type="xs:string" use="required"/>
    <xs:attribute name="CSPG" type="tns:tCSPG" use="optional"/>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
<xs:simpleType name="tCSPG">
    <xs:annotation>
        <xs:documentation>
            <xs:label xml:lang="en">CSPG type</xs:label>
            <xs:definition xml:lang="en">
                Specifies the type of CSPG
            </xs:definition>
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:enumeration value="OIPF-CI+"/>
        <xs:enumeration value="OIPF-DTCP-IP"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tProtectedFormat">
    <xs:annotation>
        <xs:documentation>
            <xs:label xml:lang="en">Protected Format</xs:label>
            <xs:definition xml:lang="en">
                Specifies the supported Protected Format
            </xs:definition>
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:enumeration value="BBTS"/>
        <xs:enumeration value="PF"/>
        <xs:enumeration value="PDCF"/>
        <xs:enumeration value="MPIMP"/>
        <xs:enumeration value="DCF"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="tSupportedEncodings">
    <xs:annotation>
        <xs:documentation>
            <xs:label xml:lang="en">encodings</xs:label>
            <xs:definition xml:lang="en">
                Specifies the supported audio and video encodings
                (eg. MPEG2,H264 AC3, AAC etc)
            </xs:definition>
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="AudioEncoding" type="tns:tAudioEncoding"
            minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="VideoEncoding" type="tns:tVideoEncoding"
            minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="tAudioEncoding">
    <xs:annotation>
        <xs:documentation>
            <xs:label xml:lang="en">Audio Encoding</xs:label>
            <xs:definition xml:lang="en">
                Specifies supported audio encoding Properties
            </xs:definition>
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Encoding" type="tva:ControlledTermType"/>
        <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

```
    <xs:complexType name="tVideoEncoding">
        <xs:annotation>
            <xs:documentation>
                <xs:label xml:lang="en">Video Encoding</xs:label>
                <xs:definition xml:lang="en">
                    Specifies supported video encoding properties
                </xs:definition>
            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="Encoding" type="tva:ControlledTermType"/>
            <xs:element name="SupportedFrameRate" type="tva:FrameRateType"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="tIPEncapsulations">
        <xs:annotation>
            <xs:documentation>
                <xs:label xml:lang="en">encapsulation</xs:label>
                <xs:definition xml:lang="en">
                    Specifies the IP encapsulation that is supported on
                    the device (UDP/RTP, UDP/M2TS, UDP/RTP/M2TS)
                </xs:definition>
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:enumeration value="UDP/RTP"/>
            <xs:enumeration value="UDP/M2TS "/>
            <xs:enumeration value="UDP/RTP/M2TS"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

6.2.3 UE Profile XML schema

With the updates to the UE Profile schema defined above in section 6.2.2, the namespace for this schema is updated to “urn:oipf:iptv:UEProfile:2008-1”. Other XML schemas that import the UE profile should be appropriately amended to use this new namespace.

6.3 Mapping SDP attributes from DVB SD&S information

In Annex E.1 an informative note is added to explain the mapping for Bandwidth to MaxBitrate. The new table row is:

Bandwidth b=AS:<bandwidth>	MaxBitrate (Note: The “MaxBitrate” attribute in SD&S is calculated according to the TIAS bandwidth modifier defined in RFC 3890, but expressed in kb/s. The OITF should do the necessary conversion to express the bandwidth in the SDP as b=AS:<bandwidth>.

6.4 DRM capability in TR-135

The list of possible CA/DRM systems supported by an OITF and signalled in the TR-135 Remote Management Object data model is clarified and extended to include all possible options. The row entries for the “DRMSystems” attribute in “.STBService.{i}.Capabilities.DRM.” of Table 65 in Annex K.1 are amended to:

.STBService.{i}.Capabilities.DRM.		**This object describes the characteristics of the Conditional Access and/or Digital Rights Management of the OITF.**
DRMSystems	R	Comma-separated list of unique identifiers of OIPF supported Content Protection systems Each item is an enumeration: "urn:dvb:casystemid:19188" “OIPF-DTCP-IP” “OIPF-CI+” "urn:dvb:casystemid:456 OIPF-CI+" "urn:dvb:casystemid:12345 OIPF-DCTP-IP"

7 Errata for Volume 5 – Declarative Application Environment

7.1 Table cross-reference

*In section 9.3.10 on DRM Capability Indication, the clause on DRMSystemID should refer to **Table 10** of Section 3.3.2 of [META].*

7.2 DAE Defined MIME types

In section 7.1.1; replace the definition of the mimeType argument of the isObjectSupported() method in section 7.1.1 with the following;

The mimeType may have any of the MIME types defined in tables 1 to 4 of [MEDIA] or any of the DAE defined mime types listed below.

Mime Type
application/notifsocket
application/oipfApplicationManager
application/oipfCapabilities
application/oipfCodManager
application/oipfCommunicationServices
application/oipfConfiguration
application/oipfDownloadManager
application/oipfDownloadTrigger
application/oipfDrmAgent
application/oipfGatewayInfo
application/oipfMDTF
application/oipfParentalControlManager
application/oipfRecordingScheduler
application/oipfRemoteManagement
application/oipfSearchManager
application/oipfStatusView

```
video/broadcast
```

In section 7.1.1.2, change the order of the methods to be as follows;

```
Object oipfObjectFactory.createApplicationManagerObject()

Object oipfObjectFactory.createCapabilitiesObject()

Object oipfObjectFactory.createCodManagerObject()

Object oipfObjectFactory.createConfigurationObject()

Object oipfObjectFactory.createDownloadManagerObject()

Object oipfObjectFactory.createDownloadTriggerObject()

Object oipfObjectFactory.createDrmAgentObject()

Object oipfObjectFactory.createGatewayInfoObject()

Object oipfObjectFactory.createIMSObject()

Object oipfObjectFactory.createMDTFObject()

Object oipfObjectFactory.createNotifSocketObject()

Object oipfObjectFactory.createParentalControlManagerObject()

Object oipfObjectFactory.createRecordingSchedulerObject()

Object oipfObjectFactory.createRemoteManagementObject()

Object oipfObjectFactory.createSearchManagerObject()
```

7.3 Certificates

In section 9.1 ", Minimum DAE capability requirements", the list introduced by "In their SSL/TLS implementation, OITFs SHALL support" shall be extended with the following;

- for verifying server certificates, at least these root certificates:

    1. Thawte Personal Basic CA
    2. Thawte Personal Freemail CA
    3. Thawte Personal Premium CA
    4. Thawte Premium Server CA
    5. Thawte Server CA
    6. Thawte Timestamping CA
    7. VeriSign, Inc. Class 1-3 Public Primary Certification Authority G1
    8. VeriSign, Inc. Class 1-4 Public Primary Certification Authority G2

9. VeriSign, Inc. Class 1-4 Public Primary Certification Authority G3
10. Equifax Secure CA
11. Entrust.net CA
12. Entrust.net CA 2048
13. Entrust.net Client CA
14. GTE CyberTrust Global Root
15. Microsoft Root Authority

7.4 Duplicated Properties and Methods

Remove the first definition of drmContentID from section 7.4.6.1 and the second definitions of the allowContact() and blockContact() methods in section 7.8.2.2.

7.5 Gateway Discovery

In section 7.7.1.1;

1) *Rename the properties ISDiscovery to isIGDiscovered, AGDiscovery to isAGDiscovered*
2) *Rename CSPGDiscovery to CSPGDTCPDiscovered and replace the reference to a CSP Gateway to a CSPG-DTCP Gateway*
3) *Rename cspgURL to cspgDTCPURL and replace the reference to a CSP Gateway to a CSPG-DTCP Gateway*
4) *Rename onDiscoverCSPG to onDiscoverCSPGDTCP, replace the reference to a CSP Gateway to a CSPG-DTCP Gateway and extend the description with* "The CSPG-DTCP gateway SHALL be discovered using a UPnP Discovery mechanism described in [OIPF_PROT] section 10.1.1.3. The actual status of the gateway (discovered or not) can be determined by reading the isCSPGDTCPDiscovered property.".
5) *Extend the description of the interval property with* "When the interval property is set, an UPnP Discovery mechanism is executed."
6) *Extend the description of the onDiscoverIG property with* "which uses a UPnP Discovery mechanism described in [OIPF_PROT] section 10.1.1.1. The actual status of the gateway (discovered or not) can be determined by reading the isIGDiscovered property."
7) *Extend the description of the onDiscoverAG property with* "which uses a UPnP Discovery mechanism described in [OIPF_PROT] section 10.1.1.2. The actual status of the gateway (discovered or not) can be determined by reading the isAGDiscovered property."
8) *Add the following new properties;*

`readonly Boolean` **`isIGSupported`**
readonly property that indicates whether an IMS Gateway is supported or not.

`readonly Boolean` **`isAGSupported`**

readonly property that indicates whether an Application Gateway is supported or not.

`readonly Boolean` **`isCSPGCIPlusSupported`**
readonly property that indicates whether a CSPG-CI+ Gateway is supported or not.

`function` **`onDiscoverCSPGCIPlus`**
read-write property that specifies the function that SHALL be called when a CSPG-CI+ Gateway is discovered or lost by the OITF (including any change to the DRM systems supported by that gateway). The CSPG-CI+ Gateway SHALL be discovered as defined in [OIPF_CSP]. The actual status of the gateway (discovered or not) can be determined by reading the isCSPCIPlusGDiscovered property.

Add a new section 7.7.1.3 as follows

7.7.1.3 Events

For the intrinsic events listed in the table below, a corresponding DOM level 2 event SHALL be generated, in the following manner:

Intrinsic event	Corresponding DOM 2 event	DOM 2 Event properties
`onDiscoverIG`	`DiscoverIG`	Bubbles: No Cancelable: No
`onDiscoverAG`	`DiscoverAG`	Bubbles: No Cancelable: No
`onDiscoverCSPGDTCP`	`DiscoverCSPGDTCP`	Bubbles: No Cancelable: No
onDiscoverCSPGCIPlus	DiscoverCSPGCIPlus	Bubbles: No Cancelable: No

NOTE: the above DOM 2 event is directly dispatched to the event target, and will not bubble nor capture. Applications SHOULD not rely on receiving a `DiscoverIG, DiscoverAG, DiscoverCSPGDTCP and DiscoverCSPGCIPlus` event during the bubbling or the capturing phase. Applications that use DOM 2 event handlers SHALL call the `addEventListener()` method on the `application/oipfGatewayInfo` object. The third parameter of `addEventListener`, i.e. “`useCapture`”, will be ignored.

Add the following property to 7.3.3.1;

readonly Boolean **ciplusEnabled**
Flag indicating whether the platform has CI+ capability.

7.6 getChannelConfig methods

In sections 7.10.1.1 and 7.12.1.2 extend the description of the getChannelConfig method with the following;

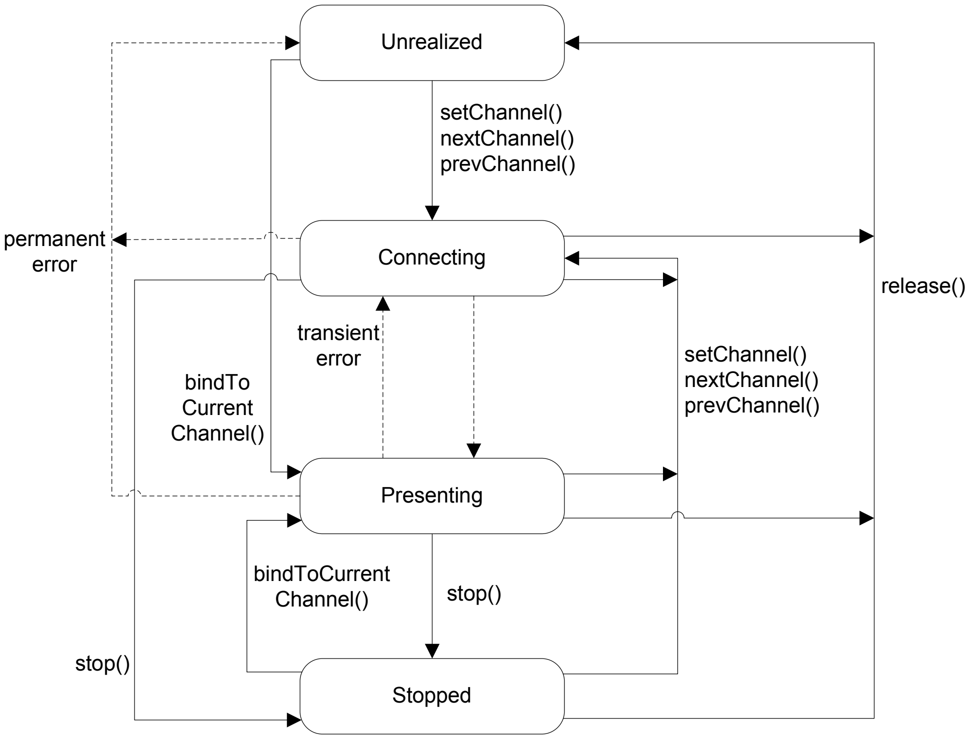
The ChannelConfig object returned from this function SHALL be identical to the ChannelConfig object returned from the getChannelConfig() method on the video/broadcast object as defined in 7.13.3.

In section 7.13.11.2, Channel, add the following extra property;

readonly Boolean **recordable**
Flag indicating whether the channel is available to the recording functionality of the OITF. If the value of the pvrEnabled property on the application/oipfConfiguration object as defined in 7.3.3.1 is false, this property SHALL also be false for all Channel objects.

7.7 States for the <video/broadcast> object

In section 7.13.1.1, replace Figure 1: "State diagram for embedded video/broadcast objects" with the following;



Immediately after the revised figure, replace "When the bindToCurrentChannel() method is called from the unrealized state, or the setChannel(), nextChannel() or prevChannel() method is called from any state, the object will transition to the connecting state, in which the OITF attempts to connect to the media stream." with the following;

When the `setChannel()`, `nextChannel()` or `prevChannel()` method is called from the unrealized, connecting or presenting states, the object shall transition to the connecting state, in which the terminal attempts to connect to the broadcast stream

When the `bindToCurentChannel()` method is called from the unrealized or stopped states, the object shall transition directly to the presenting state.

Insert the following paragraph immediately before the paragraph "Applications can use the playState property of the video/broadcast object to read its current state".

Calling the `stop()` method SHALL stop video and audio presentation and cause the video/broadcast object to transition to the stopped state. This SHALL have no effect on access to non-media broadcast resources such as EIT information. Calling the `bindToCurrentChannel()` method while in the stopped state SHALL result in video and audio presentation being restarted. Calling the `setChannel()`, `nextChannel()` or `prevChannel()` methods while in the stopped state shall result in the terminal attempting to select the new service. Applications can use the `playState` property of the video/broadcast object to read its current state.

In section 7.13.1.2, modify the description of the possible values of the playState property by adding the underlined text as follows;

Value	Description
0	unrealized; the user (or application) has not made a request to start presenting a channel or has stopped presenting a channel and released any resources. <u>The content of the video/broadcast object SHALL be an opaque black rectangle.</u>
1	connecting; the receiver is connecting to the media source in order to begin playback. Objects in this state may be buffering data in order to start playback. <u>The content of the video/broadcast object SHALL be either the last frame of decoded video (e.g. in the case of transient errors or changing channels), or an opaque black rectangle</u>.
2	presenting; the media is currently being presented to the user. The object is in this state regardless of whether the media is playing at normal speed, paused, or playing in a trick mode (e.g. at a speed other than normal speed).
<u>3</u>	<u>stopped; the terminal is not presenting media, either inside the video/broadcast object or in the logical video plane. The logical video plane is disabled. The content of the video/broadcast object SHALL be an opaque black rectangle. Control of media presentation is under the control of the application, as defined in Section 8.4</u>

Add the following method to section 7.13.1.3;

`void stop()`	
`Description`	Stop presenting broadcast video. If the video/broadcast object is in any state other than the unrealized state, it SHALL transition to the stopped state and stop video and audio presentation. This SHALL have no effect on access to non-media broadcast resources such as EIT information. Calling this method from the unrealized state SHALL have no effect. See figure 11 in section 7.13.1.1 for more information of its usage.

7.8 Video and Graphics Integration Model

Insert a new Annex I "Display model" as follows;

I.1 Logical plane model

Digital TV terminals typically have multiple planes for displaying graphics, subtitles, video and background color. This section defines a logical plane model for OITFs. Figure 14 shows the ordering of these logical planes.

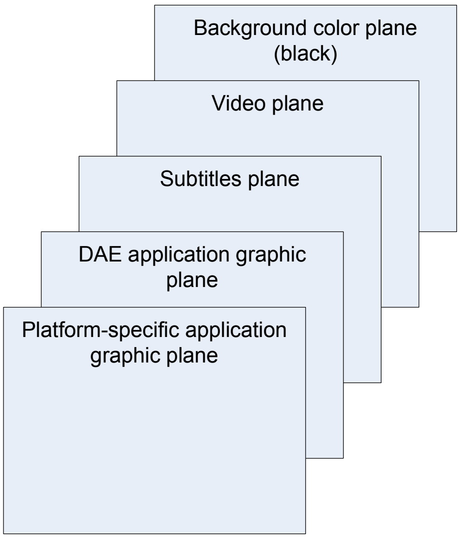

Background color plane (black)
Video plane
Subtitles plane
DAE application graphic plane
Platform-specific application graphic plane

Figure 14: Logical plane model

This logical plane model does not imply any particular physical implementation. For instance, the presence of two graphic planes and a subtitle plane does not imply a requirement for three hardware graphic planes.

The logical planes are defined as follows:

- The “Background color plane” displays a single uniform color which shall be black. This plane is at the bottom of the logical display stack.
- The “Video plane” is used to display video. This plane is on top of the background color plane in the logical display stack. The interaction between the “video plane” and the video/broadcast object is described in clause 10.1.2. Streamed video may appear to be presented in a plane other than the logical video plane. The present document is intentionally silent about the mechanism used by an OITF to achieve this behaviour
- The “Subtitles plane” is used to display subtitles. This plane is on top of the video plane in the logical display stack.
- The “DAE application graphic plane” is used to display any running DAE applications. This plane is on top of the subtitles plane in the logical display stack. The logical resolution of this plane is given by the <width> and <height> elements of the capability description.
- The “Platform-specific application graphic plane” is used to display applications specific to the OITF such as native system menus, banners or pop-ups. This plane is on top of the DAE application graphic plane in the logical display stack.

For subtitles, the following rules apply:

- OITFs SHOULD support simultaneous display of application and subtitles. In that case, the OITF SHALL display the application over the subtitles (as shown in figure 14). If the video is rescaled, the subtitles SHOULD be rescaled/repositioned appropriately or not displayed at all.
- If the presentation of subtitles is requested prior to the launch of an application, then OITFs which cannot support simultaneous display of applications and subtitles SHALL display subtitles in preference to running the application. The OITF may offer the end-user the opportunity to disable subtitles and run the application instead.

- If the presentation of subtitles is requested while an application is running, OITFs which cannot support simultaneous display of applications and subtitles shall display applications in preference to the presentation of subtitles.

I.2 Interaction with the video/broadcast and A/V Control objects

The behaviour of the video/broadcast object is defined in section 7.13.1.1. When no video/broadcast object is instantiated, or when all video/broadcast objects are in the Unrealized state, broadcast video presentation SHALL be under the control of the OITF. When video is under the control of the OITF:

- Any broadcast video being presented SHALL be displayed in the logical video plane.
- The complete logical video plane SHALL be filled.
- The OITF MAY scale and/or position video, for example to remove black bars.

For broadcast related applications as defined in section 5.2.3, broadcast video presentation SHALL initially be under the control of the OITF. Applications wanting to control video presentation SHALL create a video/broadcast object.

When a video/broadcast object is in any state other than the Unrealized state, broadcast video presentation SHALL be under the control of the application. When video is under the control of the application:

- When the video/broadcast object or AV Control object is not in “full-screen mode”, any video being presented SHALL be scaled and positioned to fit the object. The area of the video plane not containing video SHALL be transparent.
- When the video/broadcast object or AV Control object is in “full-screen mode”, presented video SHALL be scaled to fill the entire logical video plane. The OITF MAY further scale and/or position video, for example to remove black bars.
- Depending on the Z index of the video/broadcast or AV Control object with respect to other HTML elements (regardless of whether the object is in “fullscreen mode” or not), presented video may fully or partially obscure other HTML elements with a lower Z index, and may in turn be fully or partially obscured by HTML elements with a higher Z index. As a result of this, video may appear to be presented in a plane other than the logical video plane. This specification is intentionally silent about the mechanism used by an OITF to achieve this behaviour.
- Calling the Application.hide() method SHALL cause video (and any subtitles) being presented under the control of that application to be hidden, and any audio being presented by the video/broadcast or AV Control object under the control of that application to be muted. Calling Application.show() SHALL cause video and audio presentation to be restored.

If the release() method is called on a video/broadcast object, or if the object is garbage collected, control of broadcast video presentation SHALL be returned to the OITF and video SHALL be re-scaled and re-positioned (if necessary).

I.3 Graphic safe area (informative)

Figure 15 shows the recommended safe area for content authoring for the OITF_HD_UIPROF default profile:

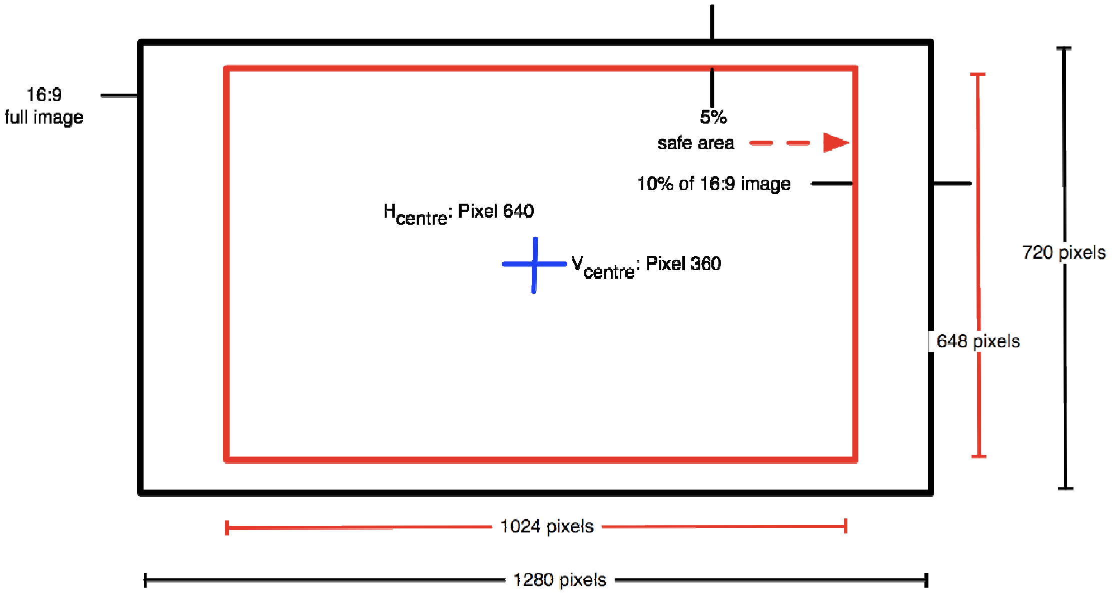


Figure 15: Graphic safe area

In 7.13.1.2, change the definitions of the playState property and the state argument of the onPlayStateChange property as follows;

Value	Description
0	unrealized; the application has not made a request to start presenting a channel or has stopped presenting a channel and released any resources. The content of the video/broadcast object is transparent. Control of media presentation is under the control of the OITF, as defined in section I.2
1	connecting; the terminal is connecting to the media source in order to begin playback. Objects in this state may be buffering data in order to start playback. Control of media presentation is under the control of the application, as defined in section I.2. The content of the video/broadcast object is transparent.
2	presenting; media is currently being presented to the user. The object is in this state regardless of whether the media is playing at normal speed, paused, or playing in a trick mode (e.g. at a speed other than normal speed). Control of media presentation is under the control of the application, as defined in section I.2. The video/broadcast object contains the video being presented.

Add the following text as a new section 4.9;

4.9 Display Model

Annex I describes the logical display model of an OITF and the relationship between DAE application graphics and video.

7.9 Keyset capitalization

Modify the title of section 7.2.5 to use the 'Keyset' capitalization instead of 'KeySet'.

In section 9.1, replace "KeySet" with "Keyset" in the following paragraph;

Because physical color keys may not always be available on remote controls, DAE applications which use the colour keys SHOULD make the same feature, function or link accessible through a button in their user interface which can be navigated to by `up`, `down`, `left` and `right` and selected with `enter` / `OK` and SHOULD make their intended usage known through the `Keyset` object as defined in Section 7.2.5.

7.10 Application model and lifecycle

In section 4.3, delete the following from the second paragraph;

It is accessed over TLS and authenticated with an X509 certificate. Access to privileged capabilities can be requested through extensions to the X509 certificate (see section 10.1).

and

which provides Javascript properties and methods that a DAE application possesses that exceed those of traditional "web pages"

Also in section 4.3, change the first sentence of the third paragraph to say;

The difference between a DAE application and a traditional web page is that web pages are stand-alone with no formal concept of a group of pages or a context within which a group of pages are loaded and execute.

Change section 4.3.1 to say;

DAE applications are comprised of pages which are conceptually no different from traditional web pages. Both pages in a DAE application and traditional web pages can include the contents of other documents. These included documents can have a variety of types, including Cascading Style Sheets (CSS), JavaScript, SVG, JPEG, PNG and GIF.

A dynamic DOM, combined with XMLHttpRequest, permits AJAX-style changes to the current page in a DAE application or web page without necessarily replacing the entire document.

Replace the first paragraph of section 4.3.2 with the following;

A DAE application provides shared context and state common to a number of pages – a concept which doesn't formally exist in the web. Loading and unloading pages within the context of a DAE application is the same as loading and unloading web pages.

Change the second paragraph of section 4.3.2 to say the following;

The application context includes information about the state of an application from the platform's perspective – permissions, priority (for example, which to terminate first in the event of insufficient resources) and similar information that spans all documents within an application during the lifetime of that application.

Change the second sentence of section 4.3.3 to say the following;

Using the createApplication() method as defined in Section 7.2.2.2, applications can be either be started as child nodes of the application or as a sibling of the application (i.e. added as an additional childof this application's parent).

Delete the second paragraph of section 4.3.7 - "Only web pages running as DAE applications (e.g. from a known provider and loaded via TLS) have access to an Application object (via the application/oipfApplicationManager object)."

In section 4.3.8, modify the second and third paragraphs to say the following;

An application is activated through calling the activateInput() method of the application node. This marks an application as active and SHALL insert the application at the start of the active application list (removing it from the list first if it is already present).

An application is deactivated through the deactivateInput() method of the application node. This marks an application inactive and SHALL remove it from the active application list.

In section 4.4.2, delete "For packaged applications, the entire package SHALL be retained (in either packaged or unpackaged form) until the application has terminated."

Move the following 4 paragraphs from 5.1.1.3 to 4.4.2 after the paragraph starting "Any areas of the browser area outside the DOM Window that become visible when it is resized SHALL be transparent" and before the paragraph starting "The default background color of the root of the document" with the changes marked by revisions;

~~Each application has an associated DOM Window object by default. This Window object is~~ Broadcast-related and service provider related applications SHALL initially be created as invisible~~marked hidden~~ to avoid screen flicker during application start-up. Broadcast-independent applications SHALL initially be created as visible. Once loaded (as might be indicated through an onload event handler), the application then typically calls the show() method of its parent Application object.

Broadcast-related and service provider related applications SHALL initially be created as invisible to avoid screen flicker during application start-up. Broadcast-independent applications SHALL initially be created as visible. Once loaded (as might be indicated through an onload event handler), the application then typically calls the show() method of its parent Application object.

If the application does not ever need to be visible, then its DOM Window object will never be shown. In that case, the application should take steps to avoid being formatted to reduce computation and memory overheads. This is typically accomplished by setting the default CSS style of the document's BODY element to display: none.

Because all applications have associated DOM Window objects, it is possible to make any application visible even if it is not normally intended to be visible. This is of particular benefit during debugging of hidden service type applications.

The DOM Window for an application cannot interact with other DOM Window objects of other applications in the system except through the application API. In other words, scripts that are part of the document being displayed inside a DOM Window object cannot discover other applications without going through the application API, which acts as a single point of security control.

Move the following 2 paragraphs from 5.1.1.3 to the end of 5.1.1.1;

All HTML, JavaScript and SVG files that comprise an application SHALL be retrieved from the same Fully-Qualified Domain Name (FQDN). If the application attempts to access files of these types from another domain, this access SHALL fail as if the content did not exist. Files with other MIME types supported by OITF may be retrieved from other domains.

If the document of an application is modified (or even replaced entirely by other pages of the same FQDN), the Application object is retained. This means that the permission set granted when the application is created applies to all "edits" of the document or other pages in the application, until the application is destroyed.

In section 9.1, prefix the following paragraph with "When the CEA-2014 notification framework (see section 5.3.1) is supported, ";

OITFs SHALL support at least 2 DAE applications being visible at one time, one application showing a notification in the notification window (as defined in Section 5.6.3 of CEA-2014-A) and one in the main browser area.

7.11 Application types

In section 5.1.1.1,

1) Delete the reference to "Web applications - pages loaded directly from a URL"

2) Modify the reference to applications loaded from SD&S to read "Service provider related applications (from SD&S signalling)"

3) Insert the following;

- Broadcast-related applications (either be from SD&S signalling or from broadcast signalling in a hybrid device)
- Broadcast independent applications

Rename section 5.1.1.2 to "Broadcast-independent applications" and change the contents to say;

Broadcast-independent applications are started by fetching the first page of the application from a URL.

In section 5.1.1.6, replace the reference to "web applications" with "broadcast-independent applications"

Add the following at the end of section 5.1.1.3;

The URL passed to the createApplication method SHALL be one of the following;

- An HTTP or HTTPS URL referring to an XHTML page as defined by section 6.1 of this specification.
- The DVB URI for launching service provider related applications signalled through SD&S as defined in section 8.3 of this specification
- The DVB URI for launching broadcast-related applications from the current service signalled through SD&S as defined in section 8.3 of this specification

In section 8.3, change the comments entry for the DVB URI scheme to say the following;

Dvb	Application launching	Locator for applications in signalled in SD&S as defined by section 6.3.3 of [TS 102 851].	The orgid and appid encoded in the DVB URI are compared with the applications signalled in SD&S to identify one with the same orgid and appid.

7.12 iframe security issues

In section 5.1.1.3, delete the following paragraph;

All HTML, JavaScript and SVG files that comprise an application SHALL be retrieved from the same Fully-Qualified Domain Name (FQDN). If the application attempts to access files of these types from another domain, this access SHALL fail as if the content did not exist. Files with other MIME types supported by OITF may be retrieved from other domains.

In the first paragraph of section 5.1.2, delete the sentence "Applications SHALL also be destroyed when following a link to a page loaded from a different domain.".

Add a new section 5.1.3 as follows;

5.1.3 Application Boundaries

All of the pages that make up an application are contained within its application boundary. This is the "fully qualified domain name" (FQDN) of the initial page of the application in the absence of an application_boundary_descriptor.

If an applicationBoundary element is present in the SD&S signalling for an application as defined in [A137], the application boundary SHALL also include the FQDNs listed in the applicationBoundary element. If this element is not present, then the application boundary SHALL consist of the FQDN of the initial page of the application.

For files requested with XMLHttpRequest,the Same-Origin Policy SHALL be extended using the application domain; i.e. any domain in the application domain SHALL be considered of same origin.

The OITF SHALL remove any IP address in the application boundary which is within the private address space as defined in [RFC1918], before launching the application.

Add a new section 10.1.3 as follows;

10.1.3 Loading documents from different domains.

The contents of an <iframe>, <embed> or <object> element may be retrieved from an FQDN other than the one from which the top-level document is loaded. In this case, the OITF SHALL enforce security restrictions between the contents of the element and the parent document. These restrictions may be based on the nested browsing context as defined in clause 6.1.1 of [HTML5] and the security restrictions formalised in clause 6.3.1 of [HTML5],excluding the features not included in this specification.

Documents SHALL be assigned the permissions associated with the FQDN from which they were loaded, as defined in section 10.1.1, rather than the permissions associated with the initial document of the application. For example documents loaded in an <iframe> element may be granted a different set of permissions from the top-level document that contains the <iframe> element. Similarly, following a link to a document from a different FQDN may result in the newly-loaded document having a different set of permissions than those granted to the previous document even though they are within the same application boundary.

As described in section 5.1.3, for files requested with XMLHttpRequest, the Same-Origin Policy SHALL be extended using the application domain as defined in section 5.1.3.

Add a new normative reference as follows;

[RFC1918]	IETF, RFC 1918 “Address Allocation for Private Internets”, February 1996

7.13 Method signatures

Move section 7.13.4.2 to 7.16.5 and delete section 7.14.4.2.

In section 7.14.3.1, change the return type of the onPlayPositionChanged method from script to function.

In section 7.15.1.2,

1. *change the signature of the method “void addFLUTEListenerTags(String multicastAddress, String[] tags, String downloadCallBack)” to change the tags argument to be a string rather than an array of strings.*
2. *Change the return type of the method “String[] getFLUTEListeners()” to StringCollection.*
3. *Change the getTags method to have a return type String (not String[]) and the description of the method to say “Returns a comma-separated list of the tags associated with a particular multicast address.”*

7.14 DOM-2 Event for onReadyToPlay

A new section 7.14.8.3 shall be included as follows;

7.14.8.3 DOM 2 events for A/V object

For the intrinsic event “`onReadyToPlay`”, a corresponding DOM level 2 event SHALL be generated, in the following manner:

Intrinsic event	Corresponding DOM 2 event	DOM 2 Event properties
`onReadyToPlay`	`ReadyToPlay`	Bubbles: No Cancelable: No Context Info: None

Note: these DOM 2 events are directly dispatched to the event target, and will not bubble nor capture. Applications SHOULD NOT rely on receiving these events during the bubbling or the capturing phase. Applications that use DOM 2 event handlers SHALL call the `addEventListener()` method on the CEA-2014 A/V Control object. The third parameter of `addEventListener`, i.e. “`useCapture`”, will be ignored.

7.15 Cookies

In section 9.1, “9.1 Minimum DAE capability requirements”, the following paragraph shall be replaced;

OITFs SHALL support at least 100 cookies with a maximum of 20 per domain and a maximum size for any individual cookie of 4K.

With

OITF SHALL support at least 100 cookies with a maximum of 20 per domain and a maximum size for any individual cookie of 4096 bytes (as measured by the sum of the lengths of the cookie's name, value, and attributes). If the cookie is bigger than 4096 bytes it SHALL be discarded, not truncated.

7.16 Multiple Simultaneous Applications

In section 4.3.2, change the third and fourth paragraphs as shown;

An OITF ~~SHALL~~MAY support the execution of more than one application simultaneously. Applications MAY share the same screen estate in a defined and controlled fashion. This differs from multiple web pages, which are typically handled through different browser “windows” or “tabs” and may not share the same screen estate concurrently (although the details of this behaviour are often browser-dependent). This also differs from the use of frames, which, apart from iframes, do not support overlapping screen estate. Where simultaneous execution of more than one application is supported, ~~B~~both foreground and background applications SHALL be supported simultaneously.

Where simultaneous execution of more than one application is supported, ~~A~~applications SHALL be recorded within a hierarchy of applications. Each object representing an application possesses an interface that provides access to methods and attributes that are uniquely available to applications. For example, facilities to create and destroy applications can be accessed through such methods.

Prefix the first paragraph in section 4.3.3 with “Where simultaneous execution of more than one application is supported”

Remove “Multiple” at the start of the first paragraph of section 4.3.4.

In section 4.3.7.1, the following sentence shall be prefixed with “Where simultaneous execution of more than one application is supported*”;*

Any calls to methods on an Application object from pages not running as part of an application from the same provider SHALL throw an error as defined in section 10.1.1.

In section 4.3.8, replace the first paragraph with the following;

Where simultaneous execution of more than one application is supported, the OITF SHALL maintain a list of application nodes ordered in a "most recently activated" order – the active applications list. This list is used by the cross-application event dispatch algorithm as defined in Section and is not directly visible to applications.

In section 4.4.1, prefix the first paragraph and postfix the second paragraph with "Where simultaneous execution of more than one application is supported".

In section 4.4.6, modify the start of item #3 to read as follows;

> 3) Only one application is visible at any time; switching to a different application either hides the currently-visible application (where simultaneous execution of more than one application is supported) or terminates the currently visible application (where simultaneous execution of more than one application is not supported).

In section 4.4.7, insert "Where simultaneous execution of more than one application is supported" into the first paragraph as follows;

As defined in [DOM 2 Events], standard DOM events are raised on a specific node within a single document. This specification extends the event capability of the OITF through cross-application events handling, but does not change the DOM2 event model for dispatching events within documents. Where simultaneous execution of more than one application is supported, an OITF SHALL implement the cross-application events and cross-application event handling model described in this section.

In section 9.1, replace;

OITFs SHALL support multiple simultaneous applications loaded and running in the browser.

With

OITFs MAY support multiple simultaneous applications loaded and running in the browser.

7.17 Media Resource Management

In section 7.13.1.1, replace;

Section 4.4.4 describes the effect on scarce resources when a video/broadcast object is removed from the DOM tree.

With

When a video/broadcast object is destroyed (e.g. by the video/broadcast object being garbage collected), or when the release() method is called, control of broadcast video shall be returned to the terminal. If an application has modified the set of components being presented (e.g. changing the audio or subtitle stream being presented) then the same set of components will continue to be presented.

When a video/broadcast object is destroyed due to a page transition within an application, terminals may delay this operation until the new page is fully loaded in order to avoid display glitches if a video/broadcast object is also present in the new page. Presentation of broadcast video or audio shall not be interrupted in either case.

Add the following at the end of section 7.14.1.1;

9) When an AV Control object is destroyed (e.g. by the AV Control object being garbage collected, or because of a page transition within the application), presentation of streamed audio or video shall be terminated.

7.18 Clarification of Parental Rating Values

In 7.16.2.2, in the description of the parentalRating property, make the following changes;

1. *Qualify (prefix) the existing text "the initial value of this property (upon creation of the Programme object) is an instance of the ParentalRatingCollection object" with "For instances of the Programme class created by the createProgramme() method defined in section 7.10.1.1,"*
2. *Insert the following additional paragraph between the current first and second paragraphs;*

For instances of the Programme class returned through the metadata APIs defined in section 7.12 or through the programmes property of the video/broadcast object defined in section 7.13.3, the initial value of this property SHALL include the parental rating value(s) carried in the metadata or DVB-SI entry describing the programme, if this information is included.

7.19 AVComponent types

In section 7.13.4.2 "The AVComponent class", the following paragraph shall be added following the first paragraph.

For forward compatibility the DAE application SHALL check the value of the type property to ensure that it is accessing an AVComponent object of the correct type.

7.20 Data attribute of A/V object

In section 8.2.2.1, replace the first paragraph in the procedure for the data property as follows;

This property holds the URL that identifies the content, as defined in Section 4.7.1. See [PROT] section 6.2.2.1.1 'Protocol over UNIS-8' for details on CoD URI.

In section 8.3, add new rows to the table between CRID and RTSP as follows;

sip	COD streaming ("sip-rtsp-rtp-udp")		
	COD streaming("sip-rtsp-udp")		

7.21 OITF Version Inquiry

Add the following properties to 7.3.3.1;

`readonly Integer` **`releaseVersion`**

Release version of the OIPF specification implemented by the OITF.

For instance, if the OITF implements release 2 version "1.0", this property should be set to 2.

`readonly Integer` **`majorVersion`**
Major version of the OIPF specification implemented by the OITF. For instance, if the OITF implements release 2 version “1.0”, this property should be set to 1.

`readonly Integer` **`minorVersion`**
Minor version of the OIPF specification implemented by the OITF. For instance, if the OITF implements release 2 version “1.0”, this property should be set to 0.

`readonly String` **`oipfProfile`**
Profile of the OIPF specification implemented by the OITF. Valid profiles are “EMP”, “BMP” and “OIP” (as defined in XXXX).

In section 8.1.1, 1 replace

```
OIPF/1.1.0 (<capabilities>; [<vendorName>]; [<modelName>]; [<softwareVersion>];

[<hardwareVersion>]; <reserved>) [<appName>[/<appVersion>]]
```

with

```
OIPF-<oipfProfile>/<releaseVersion>.<majorVersion>.<minorVersion> (<capabilities>;
[<vendorName>]; [<modelName>]; [<softwareVersion>];  [<hardwareVersion>]; <reserved>)
[<appName>[/<appVersion>]]
```

2) and insert the following

- the `<oipfProfile>` field identifies the profile implemented by the OITF as defined in the specification of the `oipfProfile` property of the `LocalSystem` class (in “7.3.3 The LocalSystem class”).
- the `<releaseVersion>`, `<majorVersion>` and `<minorVersion>` fields identify the version of the specification implemented by the OITF as defined in “7.3.3 The LocalSystem class” with properties of the same name.

In section 7.15.4.1, insert “Otherwise, it SHALL be an empty string.” at the end of the descriptions of the appName and appVersion properties.

7.22 Conflict Resolution

The following extra point shall be included at the end of section 4.1 – immediately before heading 4.1.1.

- In case of a conflict between a CEA-2014 requirement and a normative statement in the DAE specification, the normative statement in the DAE specification SHALL have priority.

7.23 CRID Usage

In section 8.3 change the rows for CRID to read as follows;

crid	Programme identification via BCG	Section 4.2.3 of [META]	
	COD Streaming	Section 4.2.3 of [META]	

7.24 Video modes

In section 7.3.5, replace the description of the videoMode property with the following;

- Read or set the video format conversion mode, for which hardware support MAY be available on the device. Valid values are:

- normal

- stretch

- zoom

The following table provides guidance as to the relationship between videoMode, aspectRatio (output) and the aspectRatio (input) of the AVVideoComponent class.

- aspectRatio (input/output) value	- videoMode value		
	- Normal	- Stretch	- Zoom
- 16:9 input / - 4:3 output	- Black bars at top and bottom, all video visible	- No black bars, picture stretched vertically	- No black bars, picture clipped on left and right sides
- 4:3 input / - 16:9 output	- Black bars on left and right, all video visible	- No black bars, picture stretched horizontally	- No black bars, picture clipped top and bottom
- 4:3 input / - 4:3 output	- No change	- No change	- No change
- 16:9 input / - 16:9 output	- No change	- No change	- No change

The DAE application graphical layer is unaffected by the videoMode.

For audio-only outputs, setting this property SHALL have no effect.

Also in section 7.3.5, modify the description of the tvAspectRatio property as shown by the change tracking which follows;

Indicates the output display aspect ratio of the display device connected to this output for which hardware support MAY be available on the device. Valid values are:

- 4:3

- 16:9

~~- 14:9~~

~~- 4:3letterbox~~

For audio-only outputs, setting this property SHALL have no effect.

Also in section 7.3.5, replace the description of the supportedVideoModes property with the following;

Read the video format conversion modes that may be used when displaying a 4:3 input video on a 16:9 output display or 16:9 input video on a 4:3 output display. The assumption is that the hardware supports conversion from either format and there is no distinction between the two. See the definition of the videoModes property for valid values.

For audio outputs, this property will have the value null.

7.25 onDRMMessageResult

In section 7.6.1.1, in the description of the resultCode argument for the property "onDRMMessageResult";

1) *The references to "SendDRMMessage" shall be changed to "sendDRMMessage" and*

2) *A new result code shall be added as follows;*

5	Unknown DRM system	`sendDRMMessage()` failed, because the specified DRM System in DRMSystemId is unknown

7.26 onDRMRightsError

In both sections 7.13.6 and 7.14.6, in the description of the onDRMRightsError property,

The following text shall be replaced;

The function that is called when a DRM licensing error occurs during playback, recording or timeshifting of DRM protected AV content inside the embedded object.

With

The function that is called:

- Whenever a rights error occurs for the A/V content (no license, license invalid), which has lead to blocking consumption of the content.

- Whenever a rights change occurs for the A/V content (license valid), which leads to unblocking the consumption of the content.

This may occur during playback, recording or timeshifting of DRM protected AV content

The description of the errorState argument shall be extended as follows;

0: no license, consumption of the content is blocked.
1: invalid license, consumption of the content is blocked
2: valid license, consumption of the content is unblocked

7.27 DVB-MCAST URI Scheme

In section 8.3 "URI Schemes and their usage", the reference to Annex H in the row for the dvb-mast URI scheme shall be replaced with the following;

DVB-MCAST URI scheme as defined by Annex A1 of [TS 102 539]

Annex H is removed. Normative references "[A086r8]" and "[RFC4607]" are removed and the following normative reference is added;

[TS 102 539]	ETSI TS 102 539, "Digital Video Broadcasting (DVB); Carriage of Broadband Content Guide (BCG) information over Internet Protocol (IP)

7.28 Return values in 7.9.1.2

In the description of the "Integer setParentalControlStatus(String pcPIN, Boolean enable)" method, the following table shall be inserted at the end;

Value	Description
0	The PIN is correct.
1	The PIN is incorrect.
2	PIN entry is locked because an invalid PIN has been entered too many times. The number of invalid PIN attempts before PIN entry is locked is outside the scope of this specification.

In the description of the "Integer unlockWithParentalControlPIN(String pcPIN, Object target, Integer duration)" method,

1) the following table shall be inserted at the end replacing the sentence "The return value indicates the success of the operation, and will take one of the values listed for setPIN()";

Value	Description
0	The PIN is correct.
1	The PIN is incorrect.
2	PIN entry is locked because an invalid PIN has been entered too many times. The number of invalid PIN attempts before PIN entry is locked is outside the scope of this specification.

2) the description of the target parameter shall be replaced with the following;

> The object to be unlocked. The value of this parameter SHALL be an instance of one of the following classes: Channel, Programme, Recording, CODAsset, or Download. If the value of this parameter is not an instance of one of these classes, a TypeError exception SHALL be thrown.

In the description of the "Integer setBlockUnrated(String pcPIN, Boolean block)" method, the following table shall be inserted at the end;

Value	Description
0	The PIN is correct.
1	The PIN is incorrect.
2	PIN entry is locked because an invalid PIN has been entered too many times. The number of invalid PIN attempts before PIN entry is locked is outside the scope of this specification.

7.29 Merging text on broadcast-independent applications

The following text from section 5.2.8 is moved to section 5.2.5 replacing the sentence "They are not under the control of any specific application signalling."

> They do not require any signalling.

Section 5.2.8 is deleted.

7.30 Content types in a content catalogue

Section 7.5.1 shall be replaced with the following;

7.5.1. The application/oipfCodManager embedded object

OITFs that have indicated <clientMetadata> with value "true" and a "type" attribute with value "bcg" SHALL implement an "application/oipfCodManager" embedded object with the following interface.

Content is organised into catalogues, where each catalogue contains a hierarchy of folders that are used to organise individual content items. The structure of the catalogue SHALL be determined by the server managing that catalogue and SHALL be reflected in the structure of the metadata passed to the OITF.

The three types of content in a CoD catalogue are:

- Assets, represented by the CODAsset class. A CODAsset is a user-level description of a piece of CoD content, and so it is more concerned with information such as the price, rental period, description and parental rating rather than detailed technical information about the asset such as encoding format. A CoD asset MAY represent a single movie, or a bundle of movies offered for a single price.
- Folders, represented by the CODFolder class.
- Services, represented by the CODService class. CODService objects are a specific type of container representing subscription VoD (SVOD) services, where users purchase a group of assets which may change over time rather than a single movie or TV show.

The CODAsset, CODFolder and CODService classes define a type property that allows these classes to be distinguished by applications. For each class, this property SHALL take the value defined below:

7.31 playspeeds array

In 7.13.2.2, clarify the definition of the playspeeds property as follows;

Returns the ordered list of playback speeds, expressed as values relative to the normal playback speed (1.0), at which the currently specified A/V content can be played (as a time-shifted broadcast in the video/broadcast object), or undefined if the supported playback speeds are not ~~(yet)~~ known or the video/broadcast object is not in timeshift mode.

If the video/broadcast object is in timeshift mode, the playSpeeds array SHALL always include at least values 1.0 and 0.0.

In 7.14.3.1, modify the definition of the playspeeds property as follows;

Returns ~~the~~an ordered list of playback speeds, expressed as values relative to the normal playback speed (1.0), at which the currently specified A/V content can be played (either through an CEA-2014 audio or video object), or undefined if the supported playback speeds are not (yet) known. ~~Note that the latter may happen at the start of playback of a video when the speeds supported by the server are not yet known.~~

7.32 A/V control object states and the seek method

In annex B, in the section relating to changes to section 5.7, the following restriction is removed;

The method does not have any effect when being called whilst the player is in any of the other states.

7.33 Remote management API clarification

In section 7.11.1 "The application/oipfRemoteManagement embedded object", the descriptions of the properties shall be clarified as follows;

Property	Clarification
vendorName	The value of this property SHALL be the same as the value of the LocalSystem.vendorName property (see section 7.3.3.2)
modelName	The value of this property SHALL be the same as the value of the LocalSystem.modelName property (see section 7.3.3.2)
softwareVersion	The value of this property SHALL be the same as the value of the LocalSystem.softwareVersion property (see section 7.3.3.2)
hardwareVersion	The value of this property SHALL be the same as the value of the LocalSystem.hardwareVersion property (see section 7.3.3.2)

7.34 Managing scheduled recordings

In 7.10.4 "Extension to application/oipfRecordingScheduler for control of recordings";

- *The recordings property shall be of type ScheduledRecordingCollection not RecordingCollection.*
- *The parameter of the method "void remove (Recording recording)" shall be of type ScheduledRecording.*

In 7.10.7 "The PVREvent class", the recording property shall be of type ScheduledRecording not Recording.

7.35 Duplication of Video Component Selection

Replace section 7.13.4 as follows

> To support the selection of specific A/V components for playback (e.g. a specific subtitle language, audio language, or camera angle), the classes defined in Sections 7.16.5.2 – 7.16.5.5 SHALL be supported and the constants, properties and methods defined in Section 7.16.5.1 SHALL be supported on the video/broadcast object.

Replace section 7.14.5 as follows

> To support the selection of specific A/V components for playback (e.g. a specific subtitle language, audio language, or camera angle), the classes defined in Sections 7.16.5.2 – 7.16.5.5 SHALL be supported and the constants, properties and methods defined in Section 7.16.5.1 SHALL be supported on the A/V Control object.

Insert new sections 7.16.5 as defined in Annex B.

7.36 Schemas

In section 8.3, add a new row to the table as follows;

igmp	Scheduled content	Annex F of [PROT].	The transport IP Multicast Address to access the service as defined in [TS 102 034].

In section 8.3, in the HTTP section of the table, replace entries in the usage column as follows;

URI scheme	Original Usage Entry	Replacement usage entry
http and https	COD streaming("http-get") COD download("http-get")	COD streaming
crid	COD streaming ("sip-rtsp-rtp-udp") COD streaming("sip-rtsp-udp")	COD streaming

rtsp	COD streaming (“rtsp-rtp-udp”) COD Streaming(“rtsp-udp”)	COD streaming

7.37 Channel change by OITF specific applications

In section 7.13.1.1, insert the following paragraph after “When the bindToCurrentChannel() method is called from the unrealized or stopped states, the object shall transition directly to the presenting state.”

> If the channel currently being presented changes due to an action outside the application (for example, the user pressing the CH+ key on the remote) then any video/broadcast object presenting that channel (e.g. as the result of a call to bindToCurrentChannel()) SHALL perform the same state transitions and dispatch the same events as if the channel change operation was initiated by the application.

In 7.13.1.2, extend the first paragraphs of the descriptions of the following properties as follows;

Property	Extension
onChannelChangeError	This function may be called either in response to a channel change initiated by the application, or a channel change initiated by the OITF (see section 7.13.1.1).
onPlayStateChange	This function may be called either in response to an initiated by the application, an action initiated by the OITF or an error (see section 7.13.1.1).
onChannelChangeSucceeded	This function may be called either in response to a channel change initiated by the application, or a channel change initiated by the OITF (see section 7.13.1.1).

7.38 Asynchronous Errors from the CSP System

The following is added to section 7.6.1.1;

function **onDRMSystemMessage**(String msg, String DRMSystemID)
The function that is called when the underlying DRM system has a message to report to the current HTML document. The specified function is called with two arguments `msg` and `DRMSystemID` which are defined as follows: • `String msg` – DRM system specific message. • `String DRMSystemID` – argument that specifies the DRM System ID of the DRM system that generated the event as defined by element DRMSystemID in Table 8 of Section 3.3.2 of [META].

The table of events in section 7.6.1.3 has an extra row added at the end as follows;

Intrinsic event	Corresponding DOM 2 event	DOM 2 Event properties
`onDRMSystemMessage`	`DRMSystemMessage`	▪ Bubbles: No ▪ Cancelable: No ▪ Context Info: DRMSystemID, msg

7.39 DRM Message Format Error

In section 7.6.1.1, in the description of "function onDRMMessageResult(String msgID, String resultMsg, Integer resultCode)", the following row shall be added to the table of result codes;

6	Wrong Format	`sendDRMMessage()` failed, because the specified message in `msg` has a wrong format.

7.40 Parental control unlocking

In section 7.9.1.2, the description of the method "Integer unlockWithParentalControlPIN(String pcPIN, Object target, Integer duration)" shall be revised as follows;

1. *The duration parameter on the method is deleted.*
2. *The sentence* "This operation SHALL be protected by the parental control PIN (if PIN entry is enabled)." *is removed.*
3. *The following row is added to the table of return values*

3	Invalid object.

4. *The following text is added;*

The object type of target can be one of the following:

- Channel object, in which case the broadcast channel currently being presented SHALL be unlocked as long as the ccid of the object matches the broadcast channel. If the channel object does not match the broadcast channel, an Invalid Object error SHALL be returned. The channel SHALL remain unlocked until the broadcast video channel is changed to a different one or has stopped being presented (e.g. the OITF being powered off or put in standby).
- Video/broadcast object, in which case the content being presented through this object SHALL be unlocked until a new channel is selected.
- A/V control object, in which case the content being presented through this object SHALL be unlocked until a new item of content is played using this object

Otherwise an Invalid Object error SHALL be returned.

7.41 Next and Previous Channels

In section 7.13.1.1, change the paragraph after the state diagram figure as follows;

When the `setChannel()`~~`, nextChannel() or prevChannel()`~~ method is called from the `unrealized`, `connecting` or `presenting` states or the `nextChannel()` or `prevChannel()` methods are called from the `connecting` or `presenting` states, the object ~~shall~~SHALL transition to the `connecting` state, in which the terminal attempts to connect to the broadcast stream. If `setChannel()` is called with a null parameter, the object ~~shall~~SHALL transition to the `unrealized` state.

In section 7.13.1.2, in the description of the property "function onChannelChangeError(Channel channel, Number errorState)", replace the description error code 10 with the following;

channel cannot be changed by the nextChannel()/prevChannel() methods either because the OITF does not maintain a favourites or channel list or because the video/broadcast object is in the Unrealized state.

In section 7.13.1.3, add the following paragraph to the end of the description of the nextChannel and prevChannel methods;

Calls to this method are valid in the Connecting, Presenting and Stopped states. They are not valid in the Unrealized state and SHALL fail.

7.42 The "data" property of the Video/Broadcast Object

In 7.13.1.2, add the following additional property of the <video/broadcast> object.

`String` **`data`**
Setting the value of the data property SHALL have no effect on the video/broadcast object. If this property is read, the value returned SHALL always be the empty string.

7.43 Exclude private IP addresses from the Application Boundary

The following additional normative reference is added;

[RFC1918]	IETF, RFC 1918 "Address Allocation for Private Internets", February 1996

The following paragraph is added to section 5.1.3;

The OITF SHALL remove any IP address in the application boundary which is within the private address space as defined in [RFC1918], before launching the application.

7.44 The isSeries property

In section 7.10.2.2, remove the isSeries property.

In section 7.10.3, remove the note "Note: Where a series is being recorded, every recorded episode will exist as an independent entry. Only the scheduledepisode will carry the isSeries property."

In section 7.10.4.1, in the description of the recordings property, remove the note “Note: Where a series is being recorded, every recorded episode SHALL exist as an independent entry. Only the scheduled recording SHALL carry the isSeries property.

7.45 AVComponent Arrays and Collections

Add a new section 7.13.4.6 as follows;

7.13.4.6 The AVComponentCollection class

An `AVComponentCollection` object represents a read-only collection of `AVComponent` objects. Next to the properties and methods defined below an `AVComponentCollection` object SHALL support the array notation to access the AV components in this collection.

Properties

readonly Integer **length**
The number of items in the collection.

Methods

AVComponent **item**(Integer index)		
Description	Return the item at position `index` in the collection.	
Arguments	*index*	The index of the item to be returned

In sections 7.13.4.1.2 and 7.14.5.1.2, the return types of the getComponents and getActiveComponents methods are changed from AVComponent[] to AVComponentCollection.

7.46 Mismatch between width and height attributes in OIPF and W3C

In annex B, the following additional text shall be added to the end of the list of “Changes to section 5.7”.

Requirement 5.7.1.g SHALL be modified as follows:

- [Req. 5.7.1.g] The following properties and methods SHALL be supported for video objects:

1) ~~Number~~ String width [RW]; the width of the area used for rendering the video object. This property is only writable if property fullScreen has value false. The effect of changes to width SHALL be in accordance with [Req. 5.7.1.c].

2) ~~Number~~ String height [RW]; the height of the area used for rendering the video object. This property is only writable if property fullScreen has value false. The effect of changes to height SHALL be in accordance with [Req. 5.7.1.c].

7.47 Duplicated parental rating properties

In section 7.10.2.2, the parentalRating property shall be renamed parentalRatings and have the following text added to the description "Note that this property was formerly called "parentalRating" (singular not plural)."

In section 7.10.5.2, the parentalRatings property shall be removed.

In section 7.10.5 (before heading 7.10.5.1), the following changes shall be made;

- *The text* "This class is a subclass of ScheduledRecording (see section 7.10.2)." *shall be replaced with* "This class implements the ScheduledRecording interface (see Section 7.10.2)."
- *The paragraph starting with* "Values of properties in the Recording object SHALL be obtained from metadata about the recorded programme" *shall have the following text appended to the end* "In the event of a conflict between the metadata in the Programme and that in the broadcast channel, the conflict resolution is implementation dependent."
- *The following text shall be added at the end;*
  NOTE: The property "parentalRatings" formerly defined as part of this class is now redundant following the renaming of the "parentalRating" property in the ScheduledRecording class to "parentalRatings".

7.48 TLS Version

The following additional normative references shall be added;

[RFC2246]	IETF RFC 2246 : "The Transport Layer Security (TLS) Protocol Version 1.0".
[RFC4346]	IETF RFC 4346 : "The Transport Layer Security (TLS) Protocol Version 1.1".
[RFC5246]	IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".
[RFC5746]	IETF RFC 5746 : "Transport Layer Security (TLS) Renegotiation Indication Extension".

The following text shall be added to section 9.1 immediately before the line "In their SSL/TLS implementation, OITFs SHALL support".

TLS/SSL – TLS 1.2 [RFC5246] SHOULD be supported, if not then TLS 1.1 [RFC4346] SHOULD be supported, otherwise TLS 1.0 [RFC2246] SHALL be supported. The OITF SHALL support TLS Renegotiation Extension as described in [RFC5746].

7.49 Media component API clarification

In section 7.16.5.1.3,

1) *in the description of the getComponents method, the first paragraph shall have text added and removed as follows;*

~~Returns~~ If the set of components is known, returns a collection of AVComponent values representing the components of the specified type in the current stream. If componentType is set to null or undefined then all the currently active components are returned if the set of active components is known.

2) *In the description of the getCurrentActiveComponents method, the first paragraph shall have text added and removed as follows;*

~~Returns~~ If the set of components is known, returns a collection of AVComponent values representing the currently active components of the specified type that are being rendered. Otherwise returns undefined.

3) *The following additional paragraph shall be added to the description of both the getComponents and getCurrentActiveComponents methods between the current first and second paragraphs;*

For a video/broadcast object, the set of components SHALL be known if the video/broadcast object is in the presenting state and MAY be known if the object is in other states. For an AV Control object, the set of components SHALL be known if the AV Control object is in the playing state and MAY be known if the object is in other states.

7.50 PVR API Issues

In section 7.10.6.2,the return type of the item(Integer index) *method shall be changed from* Object *to* ScheduledRecording.

In section 7.16.2.5;

1) *The description of the recordings property shall have text added as shown;*

The list of in-progress or completed recordings associated with this programme, sorted by start time in increasing order.

2) *The description of the scheduledRecording property shall have text removed as follows;*

If available, this property represents the scheduled recording ~~or an in-progress or completed recording~~ associated with this programme. Has value `undefined` if this programme has no scheduled recording ~~or an in-progress or completed recording~~ associated with it.

7.51 Mismatch between some VK key codes and W3C specification

The following text shall be added at the end of section 9.1.

Note that VK_* key codes defined by CEA2014-A Annex F are OPTIONAL for this specification as specified in Annex B.

In Annex B, in the bullet point "Changes to Section 5.4 ", sub-bullet point "Add keypress events to Requirement 5.4.1.a in the following way: ",

1) *add the following text before the paragraph starting* "For "keydown" and "keyup" events, the key code as specified*";*

Key constant values defined in Annex F are OPTIONAL for this specification. An OITF SHALL map VK_* constants to an internal OITF specific value. A DAE application SHALL NOT rely on the internal OITF specific key code and SHALL use the VK_* key constant literals instead.

2) *In the paragraph starting* "For "keydown" and "keyup" events, the key code as specified in Annex F", "key code as specified in Annex F" *shall be replaced by* "internal OIPF specific key code"

In Annex B, make the following changes to the section "Changes to the Annexes";

1) *Add the following text between the change to annex C and the change to annex F.*

In Annex F, the constant values defined by CEA2014-A are OPTIONAL for this specification. An OITF SHALL map VK_* constants to an internal OITF specific value. A DAE application SHALL NOT rely on the internal OITF specific key code and SHALL use the VK_* key constant literals instead.

2) *In the change to annex F concerning VK_PLAY_PAUSE, remove "= 463"*

7.52 Clarifying audio playback from memory

Section 7.14.10 shall be replaced with the following;

7.14.10 Playback of audio from memory

This section describes how an AV Control object can be used for the playback of audio from memory.

7.14.10.1 Usage of CE-HTML tags

The AV Control object SHALL be used to play audio clips from memory. The value of the AV Control object's type attribute SHALL be set to one of the values defined in Section 8.2.1 of [MEDIA]. The <object> element representing the AV Control object MAY contain <param> elements to set the value of parameters affecting the playback of the clip, For audio from memory, valid parameters are:

- cache – a case-sensitive value of "true" indicates that the audio clip should be played from memory. This parameter SHALL be included for all clips to be played from memory. For formats which can not be played from memory, or for values of the parameter other than "true", this parameter SHALL have no effect. The default value of this parameter SHALL be "false".

loop – indicates the number of times the audio clip SHALL be played when play() is called. The value SHALL be positive integers or the case sensitive string "infinite", which SHALL play the audio clip continuously until stop() is called or the data property is set to null. The default value of this parameter SHALL be "1".

Simultaneous playback of multiple audio clips from memory, or simultaneous playback of audio clips from memory and streaming audio or video presentation SHALL follow the behaviour described in section 4.4.5.

7.14.10.2 Usage of the DOM interface

For AV Control objects used to play audio from memory, the following properties and methods SHALL be supported:

- The properties data, playState, error and onPlayStateChange, as defined in Req. 5.7.1.f of [CEA-2014-A].

- The methods play() and stop(), as defined in Req. 5.7.1.f of [CEA-2014-A].

When the play() method is called, if a <param> element as described above is present where the cache parameter is set to the value "true", the OITF SHALL:

- attempt to pre-load the audio clip specified by the value of its data property and play the audio clip from memory. If the terminal cannot pre-load the audio clip due to insufficient memory, the terminal SHALL play the clip as streaming audio.

- attempt to retain the audio clip in its cache once playback has finished, until the AV Control object's data property is modified or the AV Control object is destroyed.

If the AV Control object's data property refers to a file in a format other than those listed in section 7.14.12.1, the AV Control object SHALL NOT attempt to play the file from memory.

The <param> element as defined in Section 7.14.2.1 of this document SHALL be made accessible through a DOM HTMLParamElement object.

7.14.10.3 Example usage (Informative)

The following HTML document shows an example of a script to start the playback of memory audio:

```
<head>
:
<script type=”text/javascript”>
     function startBGM() {
         document.getElementById(“aid1”).play(1);
     }
     :
</script>
</head>
<body>
<object type="audio/mp4" id="aid1" data="http://www.avsource.com/audio/bgm.aac">
<param name=”cache” value=”true” />
<param name=”loop” value=”infinite”/>
</object>
 :
<div id=”btn1” onclick=”startBGM()”><img src=”start1.gif” /></div>
 :
</body>
```

The following HTML document shows an example of a script to stop the playback of memory audio:

```
<head>

  :
<script type=”text/javascript”>
     function stopBGM() {
         document.getElementById(“aid1”).stop();
     }
  :
</script>
</head>
<body>
<object type="audio/mp4" id="aid1" data="http://www.avsource.com/audio/bgm.aac">
<param name=”cache” value=”true” />
<param name=”loop” value=”infinite”/>
</object>
 :
<div id=”btn2” onclick=”stopBGM()”><img src=”stop1.gif” /></div>
  :
</body>
```

7.53 Event handling fixes

In section 7.5.1.1, replace onContentCatalogueEvent and onContentAction with the following;

function **onContentCatalogueEvent(Integer Action)**
This function is the DOM 0 event handler for events relating to changes in a content catalogue collection. The specified function is called with the argument `action`: `Integer action` - The type of event. For current versions of the specification, this property SHALL always have the value 0 to indicate a change in the list of available catalogues.

function **onContentAction(Integer action, Integer result, Object item, ContentCatalogue catalogue)**
This function is the DOM 0 event handler for events relating to actions carried out on an item in a content catalogue. The specified function is called with the following arguments: `Integer action` - The type of action that the event refers to. Valid values are:

Value	Description
0	An operation to browse a content collection (e.g. getting a page from the collection).
1	Indicates that more information is available about this item (e.g. that more information has

	been retrieved from the server).

`Integer result` - The result of the action. Valid values are:

Value	Description
`0`	The operation succeeded.
`1`	The item no longer exists in the catalogue.
`2`	The server has not responded in the timeout period.
`3`	Communication with the server has been interrupted.

`Object item` - The item in the catalogue that the event refers to.

`ContentCatalogue catalogue` - The parent catalogue of the affected object.

Add a new section 7.5.1.2 as follows;

7.5.1.2 Events

For the intrinsic events listed in the table below, a corresponding DOM level 2 event SHALL be generated in the following manner:

Intrinsic event	Corresponding DOM 2 event	DOM 2 Event properties
`onContentCatalogueEvent`	`ContentCatalogueEvent`	Bubbles: No Cancelable: No Context Info: `action`
`onContentAction`	`ContentAction`	Bubbles: No Cancelable: No Context Info: `action`, `result`, `item`, `catalogue`

NOTE: the above DOM 2 events are directly dispatched to the event target, and will not bubble nor capture. Applications SHOULD not rely on receiving the events listed above during the bubbling or the capturing phase. Applications that use DOM 2 event handlers SHALL call the `addEventListener()` method on the `LocalSystem` object. The third parameter of `addEventListener`, i.e. “`useCapture`”, will be ignored.

Remove sections 7.5.4 and 7.5.8.

In section 7.10.4.1, replace the description of the onPVREvent function with the following;

function **`onPVREvent( Integer state, ScheduledRecording recording )`**

This function is the DOM 0 event handler for notification of changes in the state of recordings. The specified function is called with the following arguments:

`Integer state` – The current state of the recording. One of:

Value	Description
1	The recording has started.
2	The recording has stopped, having completed.
3	The recording sub-system is unable to record due to resource limitations.
4	There is insufficient storage space available. (Some of the recording may be available).
6	The recording has stopped before completion due to unknown (probably hardware) failure.
7	The recording has been newly scheduled.
8	The recording has been deleted (for complete or in-progress recordings) or removed from the schedule (for scheduled recordings).
9	The recording is due to start in a short time.
10	The recording has been updated. For performance reasons, OITFs SHOULD NOT dispatch events with the state when only the duration of an in-progress recording has changed.

`ScheduledRecording recording` – The recording to which this event refers.

In section 7.10.4.3, in the description of the DOM 2 event corresponding to onPVREvent, remove the reference to section 7.10.8.

Remove section 7.10.7.

In section 7.13.2.2, replace the description of the onRecordingEvent property with the following;

function **`onRecordingEvent( Integer state, Integer error, String recordingId )`**

This function is the DOM 0 event handler for notification of state changes of the recording functionality. The specified function is called with the following arguments:

`Integer state` - The current state of the recording. One of:

Value	Description

0	Unrealized: user/application has not requested timeshift or recordNow functionality for the channel shown. No timeshift or recording resources are claimed in this state.
1	Recording has been newly scheduled.
2	Recording is about to start .
3	Acquiring recording resources (incl. media connection).
4	Recording has started.
5	Recording has been updated.
6	Recording has successfully completed.
10	Acquiring timeshift resources (incl. media connection).
11	Timeshift mode has started.

`Integer error` - If the state of the recording has changed due to an error, this field contains an error code detailing the type of error. One of:

Value	Description
0	The recording sub-system is unable to record due to resource limitations.
1	There is insufficient storage space available. (Some of the recording may be available).
2	Tuner conflict (e.g. due to conflicting scheduled recording).
3	Recording not allowed due to DRM restrictions.
4	Recording has stopped before completion due to unknown (probably hardware) failure.
10	Timeshift not possible due to resource limitations.
11	Timeshift not allowed due to DRM restrictions.
12	Timeshift ended due to unknown failure.

If no error has occurred, this argument SHALL take the value `undefined`.

`String recordingId` - The identifier of the recording to which this event refers, This SHALL be equal to the value of the `id` property for the affected recording, if the event is associated with a specific recording.

In section 7.13.2.4, in the description of the DOM 2 event corresponding to onRecordingEvent, remove the reference to section 7.13.2.4.1 and remove section 7.13.2.4.1.

Add a new section 7.15.1.3 as follows;

7.15.1.3 Events

For the intrinsic events listed in the table below, a corresponding DOM level 2 event SHALL be generated in the following manner:

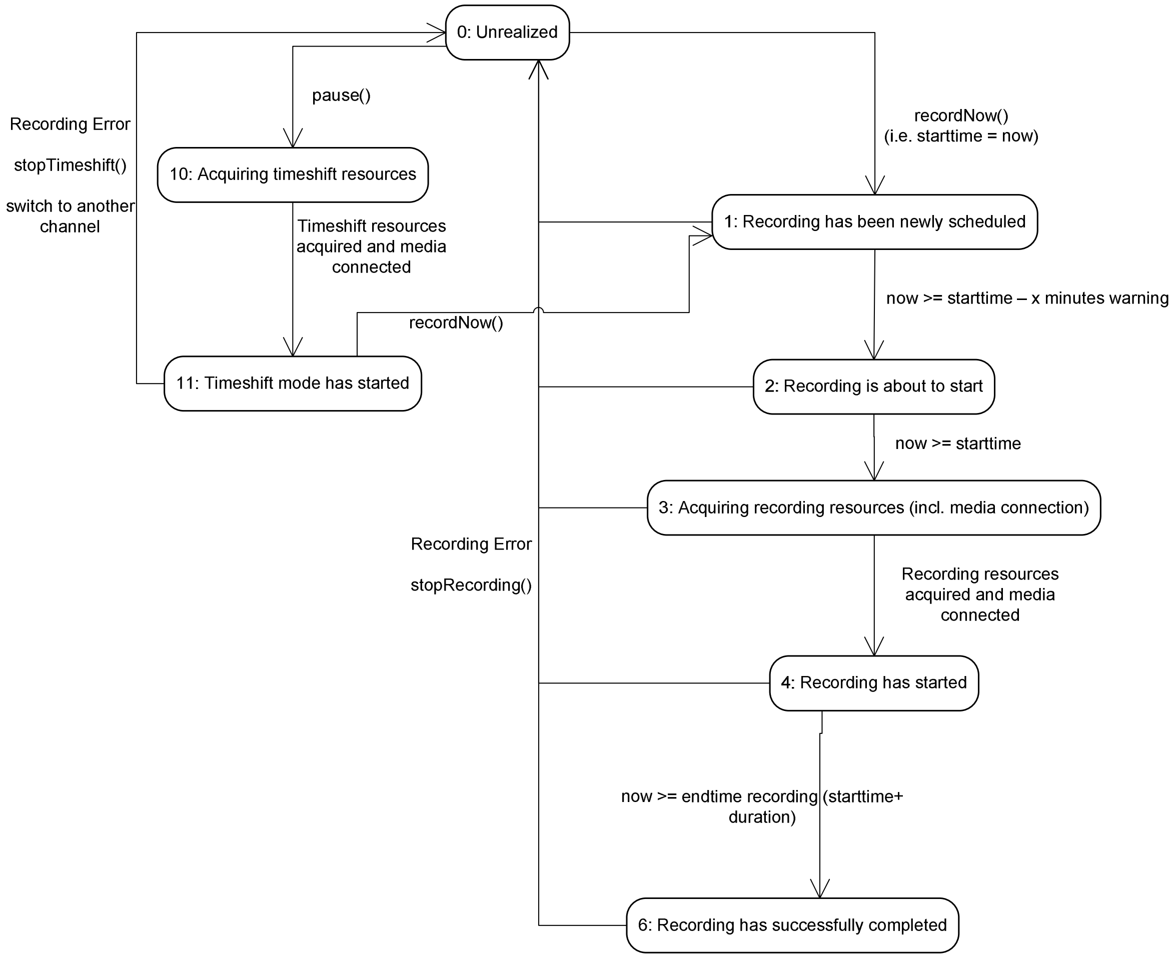
Intrinsic event	Corresponding DOM 2 event	DOM 2 Event properties
onFLUTEListenerResult	FLUTEListenerResult	Bubbles: No Cancelable: No Context Info: `multicastAddress`, `resultMsg`

NOTE: the above DOM 2 event is directly dispatched to the event target, and will not bubble nor capture. Applications SHOULD not rely on receiving a `ReceiveRemoteMessage` event during the bubbling or the capturing phase. Applications that use DOM 2 event handlers SHALL call the `addEventListener()` method on the application/oipfRemoteControlFunction object. The third parameter of `addEventListener`, i.e. “`useCapture`”, will be ignored.

7.54 PVR API clarifications

In section 7.10, make the changes shown in annex C to sections 7.10.1 to 7.10.5 inclusive.

In section 7.13.2, replace figure 12 “PVR States for recordNow and timeshifting using video/broadcast” with the following;


0: Unrealized
pause()
Recording Error
stopTimeshift()
switch to another channel
10: Acquiring timeshift resources
Timeshift resources acquired and media connected
recordNow()
11: Timeshift mode has started
recordNow()
(i.e. starttime = now)
1: Recording has been newly scheduled
now >= starttime – x minutes warning
2: Recording is about to start
now >= starttime
3: Acquiring recording resources (incl. media connection)
Recording Error
stopRecording()
Recording resources acquired and media connected
4: Recording has started
now >= endtime recording (starttime+ duration)
6: Recording has successfully completed

Make the following changes in section 7.13.2.3;

1) *In the description of the recordNow(Integer duration) method, change the second and third paragraph as follows;*

~~If~~ Calling `recordNow()`~~ is called~~ while the broadcast that is currently rendered in the video/broadcast object is already being recorded,~~ the total duration of this ongoing recording is extended by the value of the `duration` argument (i.e. the value of the `duration` argument is added onto the remaining recording time).The success or failure and the current state of the recording can be tracked using the `onRecordingEvent` intrinsic event handler as defined in Section 7.13.2.2 or by registering for the respective DOM 2 `RecordingEvent` as defined in Section 1.1.1.~~SHALL have no effect on the recording and SHALL return the value `null`.

In other cases, ~~The~~this method returns a `String` value representing a unique identifier to identify the recording. If the OITF provides recording management functionality through the APIs defined in section 7.10.4, this SHALL be the value of the `id` property of the associated `Recording` object defined in section 7.10.5.1.

2) *In the description of the offset argument of the seek method, add text as shown;*

The offset from the reference position, in seconds. This can be either a positive value to indicate a time later than the reference position or a negative value to indicate time earlier than the reference position.

3) *In the description of the offset parameter of the setChannel method ,make the following changes;*

The optional offset attribute MAY be used to specify the desired offset with respect to the live broadcast in number of seconds from which the OITF SHOULD start playback immediately after the channel switch (whereby offset is given as a ~~positive~~ negative value for seeking to a time in the past).

In section 7.16.2.5, replace the scheduledRecording *and the* recordings *properties with the following;*

`readonly ScheduledRecording` **`recording`**
If available, this property represents the recording associated with this programme (either scheduled, in-progress or completed). Has value `undefined` if this programme has no scheduled recording associated with it.

7.55 Application loading errors

Add the following to section 7.2.1.2;

`function` **`onApplicationLoadError`**`( Application appl )`
The function that is called when the OITF fails to load the file containing the initial HTML document of an application (e.g. due to an HTTP 404 error, an HTTP timeout, being unable to load the file from a DSM-CC object carousel or due to the file not being either an HTML file). All properties of the `Application` object referred to by `appl` SHALL have the value `undefined` and calling any methods on that object SHALL fail.

In section 7.2.1.4, add a new row to the table of events as follows;

Intrinsic event	Corresponding DOM 2 event	DOM 2 Event properties
`onApplicationLoadError`	`ApplicationLoadError`	Bubbles: No Cancelable: No Context Info: `appl`

7.56 setFullScreen reference to CEA-2014

In section 7.13.1.3, the reference to section 5.7.4.f of CEA-2014 shall be replaced with a reference to section 5.7.1.c of CEA-2014.

7.57 Metadata API Changes

In section 2.1, add the following new normative reference;

[TVA-BID]	ETSI, TS 102 822-6-1 V1.4.1 (2007-11), “Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 6: Delivery of metadata over a bi-directional network; Sub-part 1: Service and transport”

Section 7.12 shall be updated as shown in Annex A.

In section 7.13.10.2, add the following property;

`function` **`onChannelListUpdate`**
This function is the DOM 0 event handler for events relating to channel list updates. Upon receiving a ChannelListUpdate event, if an application has references to any Channel objects then it SHOULD dispose of them and rebuild its references. Where possible Channel objects are updated rather than removed, but their order in the ChannelConfig.all collection MAY have changed. Any lists created with ChannelConfig.createFilteredList() SHOULD be recreated in case channels have been removed.

Add a new section 7.13.10.4 as follows;

7.13.10.4 Events

For the intrinsic events "`onChannelScan`" and "`onChannelListUpdate`", corresponding DOM level 2 events SHALL be generated, in the following manner:

Intrinsic event	Corresponding DOM 2 event	DOM 2 Event properties
`onChannelScan`	`ChannelScan`	▪ Bubbles: No ▪ Cancelable: No ▪ Context Info: `type, progress, frequency, signalStrength, channelNumber, channelType, channelCount, transponderCount`
`onChannelListUpdate`	`ChannelListUpdate`	▪ Bubbles: No ▪ Cancelable: No ▪ Context Info: none

Note: the above DOM 2 event is directly dispatched to the event target, and will not bubble nor capture. Applications SHOULD NOT rely on receiving these events during the bubbling or the capturing phase. Applications that use DOM 2 event handlers SHALL call the `addEventListener()` method on the `ChannelConfig` object itself. The third parameter of `addEventListener`, i.e. "`useCapture`", will be ignored.

In sections 7.5.5 and 7.5.6, delete the uid property.

In section 7.5.6,

1) *Extend the description of the uri property as shown;*

The CRID of the asset. The value of this property is given by the `programId` attribute of the BCG `ProgramInformation` element that describes the asset.

2) *Modify the description of the previewUri property as shown;*

The URI used to refer to a preview of the asset.

For assets whose BCG description contains a `RelatedMaterial` element indicating a relationship of `Trailer` or `Preview`, the value of this property is given by the ~~CRID~~ `MediaURI` element of the `MediaLocator` ~~of the asset referred to by~~contained in that element.

For assets without an appropriate `RelatedMaterial` element, the value of this property SHALL be `undefined`.

In section 7.5.7;

1) *In the description of the uri property, replace "serviceUrl" with "ServiceURL"*

2) *In the description of the description property, replace "BCG Synopsis" with "BCG ServiceDescription"*

3) *In the description of the thumbnailUri property, make the following changes to the second paragraph;*

Alternatively, for services whose BCG description contains a `RelatedMaterial` element indicating a relationship of `Promotional Still Image`~~`Trailer or Preview`~~, the value of this property is given by the `MediaURI` element of the `MediaLocator` contained in that element~~CRID of the asset referred to by that element~~.

4) In the description of the previewUri property, make the following changes to the second paragraph;

For services whose BCG description contains a `RelatedMaterial` element indicating a relationship of `Trailer` or `Preview`, the value of this property is given by the `MediaURI` element of the `MediaLocator` contained in that element~~CRID of the asset referred to by that element~~.

7.58 Terminology – IMS or Communication Services

1) *In section 5.3.2.3, "onIMSNotification" shall be replaced by "onNotification" in the following text;*

   The OITF SHALL call the callback function onIMSNotification for the corresponding application. This includes the IMS message.

2) *Rename section 7.8 to "Communication Services APIs"*

3) *In the text between headings 7.8 and 7.8.1, replace IMS with communication services (or remove it) as follows;*

   If an OITF has indicated support for the control of its ~~IMS~~ Communication Services functionality by a server by stating <~~ims~~communicationServices>true</~~ims~~communicationServices> as defined in Section 9.3.9 in its capability description, the OITF SHALL support ~~IMS~~communication services through the use of the following non-visual object:

   <object type="application/~~oipfIMS~~oipfCommunicationServices"/>

   The ~~IMS~~ Communication Services API provides the necessary javascript methods to register new users~~ in the IMS network~~. It also provides methods to register users (`registerUser`), along with the supported feature tags~~, IMS Communication Service Identifier (ICSI) and IMS Application Reference Identifier (IARI), and de-register users (deRegisterUser)~~. A method `getRegisteredUsers` is also defined to view all the registered users. A method `getAllUsers` retrieves all users provisioned in the IG. Once registered it is possible to switch users for using ~~IMS~~communication services by using method `setUser`.

   A property is defined to view the current user to be used for a service (`currentUser`).

   In order to handle the out-of-session ~~IMS~~communication services notifications, namely, the new dialogues, there is a method for subscribing to these events (subscribe~~IMS~~Notification). All new dialogues are communicated through a callback function (on~~IMS~~Notification) to the application instance performing the subscription.

The ~~IMS~~Communication Services APIs apply only to privileged applications and SHALL adhere to the security model as defined in Section 10.

4) *Throughout section 7.8, replaces references to the* "application/oipfIMS embedded object" *with the* "application/oipfCommunicationServices embedded object".

5) *Throughout section 7.8, rename the onIMSNotification method to onNotification and insert the following text at the end of the method description* "NOTE this method was formerly named onIMSNotification."

6) *Throughout section 7.8, replace* "subscribeIMSNotification" *with* "subscribeNotification" *including in the description of the onNotificationResult method and the definition of the subscribeIMSNotification method itself. Add the following text at the end of the description of* *subscribeIMSNotification:* "NOTE: This function was formerly named subscribeIMSNotification."

7) *In section 7.8.1, remove the reference to IMS in the description of the registerUser, deRegisterUser, setUser and subscribeNotification (formerly subscribeIMSNotification) methods.*

8) *Throughout section 7.8.1, replace* "unsubscribeIMSNotification" *with* "unsubscribeNotification" *including in the definition of the method itself. Add the following at the end of the method description* "NOTE: This function was formerly named unsubscribeIMSNotification."

9) *Make the following changes to the start of section 7.8.2.*

7.8.2 Extensions to application/~~oipfIMS~~oipfCommunicationServices for ~~communication~~ presence and messaging services

If a client has indicated support for the control of its ~~Communication Services~~presence and messaging functionality by a server by stating <~~communication_services>~~presence_messaging>true</~~communication_services~~presence_messaging> as defined in Section 9.3.9 in its capability description, the client SHALL support ~~IMS~~ Communication Services through the use of the following non-visual embedded object:

<object type="application/~~oipfIMS~~oipfCommunicationServices"/>

The ~~Communication Services~~presence and messaging API provides for instant messaging, presence and contact list services. The messages can either be in a chat session using MSRP (see [PROT]) or page mode messages sent without a session. The support of ~~Communication Services~~presence and messaging services SHALL follow the OMA specification [PRES], [IM].

The Communication Services API SHALL be supported in combined OITF and IG deployment cases. It MAY be supported in other deployment cases. The use of the HNI-IGI interface is OPTIONAL between the OITF and IG when these are co-deployed.

10) *Rename section 8.2.2.3 from* "IMS APIs" *to* "Communication Services APIs"

11) *In section 9.2, replace IMS with communication services as shown in the following;*

"+~~IMS~~CommunicationServices"	<~~ims~~communicationServices>true</~~ims~~communicationServices>

12) *In section 9.3.9, replace IMS with Communication Services in the name of the section and the first sentence. Change the XML fragments as follows;*

<xs:element name="~~ims~~communicationServices" type="xs:boolean"/>

<xs:element name="~~communication_services~~presenceMessaging" type="xs:boolean"/>

7.59 Clarifying the current channel concept

The following text shall be added as a new section H.4.

H.4 Current Channel (informative)

There are 3 different "current channel" concepts in this specification;

- The current channel of an OITF. This is the most obvious "current channel" to the end-user but the most complex to properly define technically – particularly where more than one channel is being presented at the same time. The bindToCurrentChannel() method implicitly defines this as this the channel whose audio is being presented.
- The current channel of a <video/broadcast> object. This is the easiest to define technically.
- The current channel of a broadcast-related application. This is the channel which is currently the source of the signalling information controlling the lifecycle of a broadcast-related application (as described in section 5.2.3).

In simple situations, all of these may refer to the same channel. In complex situations they may not. Here are some examples;

Scenario	Current Channel of the OITF	Current Channel of <video/broadcast> object(s)	Current channel of broadcast-related application(s)
The OITF is presenting exactly one broadcast video channel, this video is being presented by a <video/broadcast> object (in the Presenting state) which is part of a broadcast-related application which is controlled by signalling information from that broadcast video channel	All 3 current channels reference the same broadcast channel.		
The OITF is presenting exactly one broadcast video channel, this video is under the control of the OITF (as defined in section H.2) and one or more broadcast-related applications are running which are controlled by signalling information from that broadcast video channel none of which have a <video/broadcast> object outside the Unrealized state.	The channel being presented by the OITF	Not relevant	The channel being presented by the OITF
The OITF is presenting exactly one broadcast video channel, this video is under the control of the OITF (as defined in section H.2) and no broadcast-related applications are running.	The channel being presented by the OITF	Not relevant	Not relevant
The OITF is presenting two broadcast video channels, one main channel (responding to channel up and channel down) and a PiP	The main channel (the one responding to channel up /	Not relevant.	Not relevant.

Scenario	Current Channel of the OITF	Current Channel of <video/broadcast> object(s)	Current channel of broadcast-related application(s)
channel.	channel down)		
The OITF is presenting two broadcast video channels, one main channel (responding to channel up and channel down) and a PiP channel. A broadcast-related application is running associated with the main channel. The user swaps the main channel to PiP and vice-versa.	The channel which was previously PiP.	Not relevant.	This specification does not address what happens to broadcast-related applications under these circumstances.
A broadcast-independent or service provider related DAE application has two <video/broadcast> objects, one presenting the channel resulting from a call to bindToCurrentChannel() and the second presenting another channel set by setChannel().	The same as the current channel of the <video/broadcast> object presenting the channel resulting from a call to bindToCurrentChannel()	The two <video/broadcast> objects have different current channels.	Not relevant.

7.60 Temporary Channel Objects, Recording and Metadata

Make the changes shown to the first paragraph of the description of the "Channel createChannelObject(Integer idType, String dsd, Integer sid)" method in 7.13.1.13;

Creates a `Channel` object of the specified `idType`. This method is typically used to create a `Channel` object of type `ID_DVB_SI_DIRECT`. The `Channel` object can subsequently be used by the `setChannel()` method to switch a tuner to a channel that is not part of the channel list which was conveyed by the OITF to the server. The ~~scope of the~~ resulting `Channel` object represents a locally defined channel which ~~is limited to the Javascript environment (incl. `video/broadcast` object) to which the `Channel` object is returned, i.e. it~~ does not get added to the channel list accessed through the ChannelConfig class (see 7.13.10)~~available through method `getChannelConfig()`~~.

Make the changes shown to the first paragraph of the description of the "Channel createChannelObject(Integer idType, Integer onid, Integer tsid, Integer sid, Integer sourceID, String ipBroadcastID)" method in 7.13.1.3;

Creates a `Channel` object of the specified `idType`. The Channel object can subsequently be used by the `setChannel` method to switch a tuner to a channel that is not part of the channel list which was conveyed by the OITF to the server. The ~~scope of the~~ resulting `Channel` object represents a locally defined channel which ~~is limited to the Javascript environment (incl. `video/broadcast` object) to which the `Channel` object is returned, i.e. it~~ does not get added to the channel list accessed through the ChannelConfig class (see 7.13.10)~~available through method `getChannelConfig`~~.

The following text shall be inserted in the description of 7.13.11 (The Channel class) between the first and second sentences.

Channel objects typically represent channels stored in the channel list (see 7.13.10). Channel objects may also represent locally defined channels created by an application using the createChannelObject methods on the video/broadcast embedded object or the channelConfig class or the createChannelList method on the channelConfig class. Accessing the channel property of a ScheduledRecording object or Recording object which is scheduled on a locally defined channel SHALL return a Channel object representing that locally defined channel.

The description of the "ScheduledRecording recordAt(Integer startTime, Integer duration, Integer repeatDays, String channelID)" method shall be amended as shown in annex C of this document.

7.61 Mapping from AVComponent to MPEG-2 TS and MP4 FF

A new section 8.4 shall be inserted as follows;

8.4 Mapping from APIs to Content Formats

8.4.1 AVComponent

The following table shows the mapping from the properties of the AVComponent class to the data carried inside the MPEG-2 TS and MP4 file format.

Property **Name and Type**	**MPEG-2 TS** **With DVB-SI component_descriptor in SDT and/or EIT**	**MPEG-2 TS** **Without DVB-SI SDT and EIT**	**MP4 FF**
Name: componentTag Type: Integer	The contents of the component_tag field in the stream_identifier_descriptor in PMT		Not defined
Name: pid Type: Integer	The PID of the stream in the PMT		trackID
Name: Type Type: One of the following constants COMPONENT_TYPE_VIDEO / COMPONENT_TYPE_AUDIO / COMPONENT_TYPE_SUBTITLE	May be derived as follows: • A value of 0x02 or 0x1B in the stream_type field in the PMT → VIDEO. • A value of 0x03 or 0x11 in the stream_type field in the PMT → AUDIO. • A value of 0x06 in the stream_type field in the PMT and the presence of a DTS_audio_stream_descriptor in the ES loop in the PMT → AUDIO. • A value of 0x06 in the stream_type field in the PMT and the presence of an AC3_descriptor or an Enhanced_AC3_descriptor in the ES loop in the PMT → AUDIO. • A value of 0x06 in the stream_type field in the PMT and the presence of a subtitling_descriptor in the ES loop in the PMT → SUBTITLES.		Track has a VisualSampleEntry (handler_type ="vide") -> COMPONENT_TYPE_VIDEO Track has an AudioSampleEntry (handler_type = "soun") -> COMPONENT_TYPE_AUDIO

Property Name and Type	MPEG-2 TS With DVB-SI component_descriptor in SDT and/or EIT	MPEG-2 TS Without DVB-SI SDT and EIT	MP4 FF
	• A value of 0x06 in the stream_type field in the PMT and the presence of a teletext_descriptor in the ES loop in the PMT and an entry in that descriptor with Teletext_type set to 0x02 or 0x05 → SUBTITLES.		
Name: Encoding Type: A string identifying the video or audio format as defined in section 3 of [OIPF_MEDIA2]	May be derived as follows: • If a video component is present (see mapping for Type above) → "video/mpeg" or "video/mp2t". • If no video is present but an audio component is present: o A value of 0x03 in the stream_type field in the PMT → "audio/mpeg". o A value of 0x11 in the stream_type field in the PMT and the profile_and_level field in that descriptor indicates MPEG-4 AAC or MPEG-4 HE AAC → "audio/mp4". o A value of 0x11 in the stream_type field in the PMT and the profile_and_level field in that descriptor indicates MPEG-4 HE AAC v2 → "audio/aacp". o A value of 0x06 in the stream_type field in the PMT and the presence of a DTS_audio_stream_descriptor in the ES loop in the PMT → "audio/vnd.dts". o A value of 0x06 in the stream_type field in the PMT and the presence of an AC3_descriptor in the ES loop in the PMT → "audio/ac3".		Track has a sample description type "avc1" -> "video/mp4". Track has a sample description type "mp4a" -> "audio/mp4"
Name: Encrypted Type: Boolean	May be derived from any of the following: • Presence of a CA_descriptor in the PMT in the program information loop. • Presence of a CA_descriptor in the PMT in the elementary stream information loop describing the stream.		Not defined
Name: Aspect ratio	Derived from the stream_content and	Undefined	Not defined

Property Name and Type	MPEG-2 TS With DVB-SI component_descriptor in SDT and/or EIT	MPEG-2 TS Without DVB-SI SDT and EIT	MP4 FF
Type: Number containing width divided by height as a decimal Only defined for video components.	component_type fields in the component_descriptor.		
Name: Language Type: String containing an ISO 639 language code Only defined for audio and subtitle components.	For audio components, the contents of the ISO_639_language_code field in the ISO_639_language_descriptor In the ES loop of the PMT unless overridden by the ISO_639_language_code field in the supplementary_audio_descriptor. For subtitles, the contents of the ISO_639_language_code fieldin the subtitling_descriptor or teletext_descriptor, as appropriate.		The contents of the language field in the media header “mdhd” of the track.
Name: Audio Description Type: Boolean - True if is component is an audio description Only defined for audio components.	True if any of the following is true: • There is an audio component with an ISO_639_language_descriptor in the PMT with the audio_type field set to 0x03 • There is a supplementary_audio_descriptor with the editorial_classification field set to 0x01 • There is an ac-3_descriptor or an enhanced_ac-3_descriptor with a component_type field with the service_type flags set to Visually Impaired. Otherwise false.		Not defined
Name: Audio channels Type: Number indicating 5 for 5.1, 7 for 7.1, 2 for stereo Only defined for audio components.			Not defined
Name: Hearing impaired Type: Boolean - Has value true if the stream is intended for the hearing-impaired (e.g. contains a written description of the sound effects), false otherwise. Only defined for subtitle	True if one of the following is true: • There is a subtitling_descriptor with the subtitling_type field set to 0x20, 0x21, 0x22, 0x23 or 0x24. • There is a teletext_descriptor with a teletext_type field with a value of 0x05.		Not defined

Property Name and Type	MPEG-2 TS With DVB-SI component_descriptor in SDT and/or EIT	MPEG-2 TS Without DVB-SI SDT and EIT	MP4 FF
components.			

7.62 Parental rating errors

Make the following changes to the method "" in section 7.13.5 and the method "" in section 7.14.6.

1. *Change the second argument from* "ParentalRating rating" *to* "ParentalRatingCollection ratings" *and modify the description as follows;*

   `ParentalRatingCollection` ~~`ParentalRating`~~ `ratings` – the parental ratings ~~value~~ of the currently playing content. The `ParentalRatingCollection`~~ParentalRating~~ object is defined in Section 7.9.

2. *Make the following changes to the first paragraph of the description of the method;*

   is triggered whenever one or more~~a~~ parental ratings ~~value is~~ are discovered and none of them are valid. ~~for a parental rating system that is not supported by the OITF.~~A valid parental rating is defined as one which uses a parental rating scheme that is supported by the OITF and which has a parental rating value that is supported by the OITF.

3. *In the description of the contentID argument, change* "parental rating change" *to* "parental rating error".

4. *In the table of events, the context info for "onParentalRatingError" shall have* "rating" *changed to* "ratings".

7.63 AV Control object state diagram clarifications

The following changes shall be made to the state diagram figure in 7.14.1.1;

1) *change the labelling of the transition from paused to stopped from "stop()" to "stop() or (data or type changed) or setSource()"*

2) *change the labelling of the transition from playing to playing from "play(x), x<>0" to "play(x), x<>0 or looping memory audio"*

3) *change the labelling of the transition from error to stopped from "automatic" to "stop() or data(or type) attribute has changed or setSource()"*

4) *add a transition from error to connecting labelled "play(x)"*

5) *add a transition from stopped back to stopped labelled identically to that from error to stopped.*

Also in section 7.14.1.1, item numbered 2 in the numbered list shall have a reference to the finished state included as follows;

> Scarce resources for playback using the A/V Control object SHALL be released when state 6 ('error') or 0 ('stopped') or 5 ('finished') are reached.

Also in section 7.14.1.1, a new item shall be added to the numbered list as follows;

9) When not presenting video, the AV Control object SHALL be rendered as an opaque black rectangle.

In section 4.4.5, the following sentence shall be changed as shown;

- The interrupted presentation shall be resumed automatically by the terminal when the interrupting audio ends (i.e. when the AV Control object transitions to the stopped, finished or error states).

7.64 Feature Tags

In section 7.8.3, in the description of the featureTags property, the start of the description shall be rephrased as follows;

> The feature tag data is optional. It carries a collection of feature tag objects associated with an application.

7.65 Parameter names in getParameter

In section 7.11.1.2, in the description of the getParameter(String parameterName) method, parameter name arguments shall be case-insensitive.

7.66 setBufferingStrategy

In section 7.14.8.2, in the description of the setBufferingStrategy method, strategy names shall be case-insensitive.

7.67 Blocked and locked

In section 7.16.2.3.1, in the description of the blocked property, the table showing how the blocked and locked properties work together shall have an extra row for blocked being false and locked being true identifying that this is an invalid combination of states.

7.68 Clarify setChannel mapping for IPTV_SDS

In 8.2.2.2.2, in the section on procedures, the two conditions;

> If the channel has an idType of ID_IPTV_URI

and

> If the channel has an idType of ID_IPTV_SDS

shall be replaced with

> Unless the channel is an IMS based IPTV service (see below)

and

> If the channel is an IMS based IPTV service (ie, if it is of type ID_IPTV_SDS and if the corresponding service has a "sip-igmp-rtp-udp" or "sip-igmp-udp" file format specified in its SD&S BDR record)

respectively.

7.69 Missing registerDownloadxxx mappings

The following shall be added to section 8.2.1.1.

registerDownload(String URL, String contentType, Date downloadStart)	API described in Section 7.4.1.1 to download the content identified by the given URL. If the OITF includes the Content Download functional entity, the URL is passed to the Content Download functional entity to download content over UNIT-17 using HTTP as described in Section 5.2.3.1 of [PROT]. As specified in 7.4.1.1, the `contentType` attribute can be used to evaluate if the content type is part of the list of accepted content types of the OITF. If `contentType` has value "`application/vnd.oipf.ContentAccessDownload+xml`", the method SHALL return a download identifier, after which the OITF SHALL immediately fetch the Content Access Download Descriptor, after which the same SHALL happen as if `registerDownload()` had been called.
registerDownload(String CRID, String IMI, Date downloadStart)	API described in Section 7.4.2 to download content described in a BCG record. If the OITF includes the Content Download functional entity, <CRID,IMI> BCG tuple is resolved to an URL as described in 4.3 of [META] and passed to the Content Download functional entity to download content over UNIT-17 using HTTP as described in Section 5.2.3.1 of [PROT].

7.70 Default Background Colour

In section 4.4.6, the following text;

> The default background color of the root of the document (i.e. the <html> rendering 'canvas') SHALL be a nontransparent color and SHOULD be white as most browsers, unless explicitly overriden with the following (or an equivalent) CSS construct to allow the underlying video to be shown for those areas of the screen that are not obscured by overlapping non-transparent (i.e. opaque) children of the <body> element:
>
> html { background-color: transparent; }
>
> body { background-color: transparent; }

shall be replaced with the following;

> Application developers SHOULD explicitly set the background color of the application <body> and <html> elements.
>
> Setting the background color to 'transparent' (e.g. using CSS construct html, body { background-color: transparent; }) will allow the underlying video to be shown for those areas of the screen that are not obscured by overlapping non-transparent (i.e. opaque) children of the <body> element."

8 Errata for Volume 6 – Procedural Application Environment

8.1 Java package naming

Throughout the specification, the OIPF defined Java packages are prefixed with “org.oitf”. *All instances of this prefix shall be changed to* “org.oipf”.

8.2 DVB-GEM normative reference

In section 2.1, normative references, the reference for DVB-GEM is updated to:

[GEM]	ETSI TS 102 728 V1.1.1 (2010-01), “Globally Executable MHP (GEM) Version 1.2.2”

8.3 DRMAgentPermission

In appendix E, in the description of the two constructors for UserAuthenticationPermission, the comment “Creates a new DRMAgentPermission” *shall be* “Creates a new UserAuthenticationPermission”

8.4 DRM System Name

The following additional text shall be included after the heading “Appendix G”;

The DRM Systems names are defined as URNs with the DVB CA System ID (16 bit number) in there. A DRM System name shall be signaled by prefixing the decimal number format of CA_System_ID with "urn:dvb:casystemid:" as defined in Table 10 for the DRMSystemID attribute, [META]. Note that the decimal number format of CA_System_ID SHALL not have leading zeroes.

8.5 DRM Agent Listener Result

In Appendix G, the Interface DRMAgentListener, package org.oipf.drm, the success method should be modified to include the new String parameter resultMsg as follows;

```
/**
* An operation triggered by a message sent through
* the DRMAgent has succeeded.
* @param resultMsg the DRM specific message result
* @param msgID the message ID provided when the message
* which triggered this operation was sent to the DRM agent
*/
public void success(String resultMsg, String msgID );
```

8.6 DRMRightsErrorEvent

The following additional class shall be added at the end of Appendix G.

```
package org.oipf.drm;
/**
 * Represents an error event raised when a player
 * tries to play a protected content without a license
 * or with an invalid license
 */
public class DRMRightsErrorEvent extends ControllerErrorEvent {

   /**
    *  errorState valid values
    */
   static public final int NOLICENSE = 0;
   static public final int LICENSEINVALID = 1;

   protected Integer errorState;

   /**
    * The unique identifier of the protected content in the scope of the DRM
    * system that raises the error
    */
   protected String contentID;

   /**
    * The DRM System name
    */
   protected String DRMSystemName;

   /**
    * The rightsIssuerUrl optional element indicating the URL that
    * can be used to non-silently obtain the rights for the content
    * item currently being played for which this DRM error is generated
    */
   protected String rightsIssuerUrl;

   /**
    * Constructor
    */
   public DRMRightsErrorEvent (Controller from, String message, Integer errorState,
 String contentID, String DRMSystemID, String rightsIssuerUrl) {
       super(from, message);
   }

   /**
    * Gets the errorState value
    * @return errorState describing the current state (NOLICENSE or LICENSEINVALID).
```

```
     */
    public Integer getErrorState() {return errorState;}

    /**
     * Gets the contentID value
     * @return contentID of the content.
     */
    public Integer getContentID () {return contentID;}

    /**
     * Gets the DRM System name value
     * @return DRM System name describing the DRM System that generated
     * the error.
     */
    public String getDRMSystemID () {return DRMSystemID;}

    /**
     * Gets the rightsIssuerUrl value
     * @return rightsIssuerUrl for license retrieval.
     */
    public String getRightsIssuerUrl () {return rightsIssuerUrl;}
}
```

9 Errata for Volume 7 – Authentication, Content Protection and Service Protection

9.1 CI Plus, Marlin and TLS references

The normative references to CI Plus and Marlin specifications are revised to the following:

[CI+]	CI Plus LLP, CI Plus Specification, “Content Security Extensions to the Common Interface”, V1.3 (2011-01).
[MRL BBTS]	Marlin Developer Community, “Marlin Broadband Transport Stream Specification”, Version 1.0
[MRL BNSP]	Marlin Developer Community, “Marlin – Broadband Network Service Profile Specification”, Version 1.1.
[MRL CORE]	Marlin Developer Community, “Marlin - Core System Specification”, Version 1.3
[MRL FF]	Marlin Developer Community, “Marlin – File Formats Specification”, Version 1.1
[OMARLIN]	Marlin Developer Community, “OMArlin Specification”, Version 1.0
[RFC5746]	IETF, RFC 5746, “Transport Layer Security (TLS) Renegotiation Indication Extension”.

9.2 CSPG-CI+ Discovery

This erratum clarifies the discovery of the CSPG-CI+ CAM module according to the CI+ specification. Section 4.2.3.3 now reads as follow.

The CSPG-CI+ discovery SHALL be performed at OITF start-up and CSPG-CI+ initialization. The setup of the session to the [CI+] Specific Application Support (SAS) resource and the connection to the Open IPTV Forum private application are described in section 4.2.3.4.1.1. A successful connection means that a CSPG-CI+ has been discovered.

9.3 CSPG-CI+ Control Channel

9.3.1 CSPG-CI+ Control Channel

This erratum replaces section 4.2.3.4.1.1 as follow.

4.2.3.4.1.1 Control Channel

OITF controls the CSPG-CI+ using resources defined in [DVB-CI] as well as resources as defined in section 11 of [CI+].

OITF and CSPG-CI+ SHALL use the SAS resource, defined in [CI+], section 11.4, to handle messages as specified in this section.

The CSPG-CI+ SHALL create one session to the SAS resource as soon as it has completed its Application Information phase of initialization. The OITF SHALL send a ***SAS_connect_rqst()*** APDU [CI+] to the CSPG-CI+ with the specific Open IPTV Forum private_host_application_ID defined in Table 8. The CSPG-CI+ SHALL acknowledge the connection by sending back a ***SAS_connect_cnf()*** APDU [CI+].

private_host_application_ID	Value (64bits)
OIPF_APPLICATION_ID	0x0108113101190000

Table 8 Open IPTV Forum private_host_application_ID

Then any further exchanges between the OITF and the CSPG-CI+ are completed through the use of the ***SAS_async_msg()*** APDU. Syntax of this APDU is reminded in Table 9.

Syntax	No. of Bits	Mnemonic
`SAS_async_msg() {`		
`   SAS_async_msg_tag`	24	uimsbf
`   length_field()`		
`   message_nb`	8	uimsbf
`   message_length`	16	uimsbf
`   for (i=0; i<message_length; i++) {`		
`      message_byte`	8	uimsbf
`   }`		
`}`		

Table 9 *SAS_async_msg*() APDU syntax

4.2.3.4.1.1.1 Specific messages

The OITF and CSPG-CI+ SHALL support the messages listed in Table 11. For each of the messages the message_byte payload takes the generic syntax given in Table 10. The message data may be broken into a number of records containing the same or different types of data identified by the datatype_id.

Syntax	No. of Bits	Mnemonic
`message_byte() {`		
`   command_id`	8	uimsbf
`   ca_system_id`	16	uimsbf
`   transaction_id`	32	uimsbf
`   send_datatype_nbr`	8	uimsbf
`   for (i=0; i<send_datatype_nbr; i++) {`		
`      datatype_id`	8	uimsbf
`      datatype_length`	16	uimsbf
`      data_type()`	8 * datatype_length	bslbf
`   }`		
`}`		

Table 10 Generic message_byte() syntax

command_id 8-bit value that identifies the message. The values are defined in Table 11.

ca_system_id 16-bit integer that identifies the CA system being queried.

transaction_id A 32-bit value, generated by the OITF, provided in a message to the CSPG-CI+ that will be returned in any corresponding reply message from the CSPG-CI+. The transaction_id allows the OITF to match the CSPG-CI+'s replies with the corresponding requests. The OITF SHOULD increment the value, modulo 2^{32}, with every message it sends. The transaction_id should be ignored in messages sent spontaneously (events) by the CSPG-CI+ (i.e. rights_info, parental_control_info, system_info).

send_datatype_nbr 8-bit integer that gives the number of data type items included in the message.

datatype_id 8-bit integer that identifies the type of the data contained in the data type loop. The values are defined in Table 12.

datatype_length 16-bit integer that gives the length of the data_type() field in bytes.

data_type Data type payload. The data type loop shall only contain the specified data type, but may contain multiple records of the same type, the number of records may be determined by computation of the datatype_length field.

Message	command_id value (hexadecimal)	Direction	
		OITF	CSPG-CI+
send_msg	0x01	→	
reply_msg	0x02	←	
parental_control_info	0x03	←	
rights_info	0x04	←	
system_info	0x05	←	
(reserved)	0x06-0x7F		
(user defined)	0x80-0xFF		

Table 11 OIPF specific messages and command_id values

Data type	datatype_id value (hexadecimal)
oipf_ca_vendor_specific_information	0x01
oipf_country_code	0x02
oipf_parental_control_url	0x03
oipf_rating_type	0x04
oipf_rating_value	0x05
oipf_rights_issuer_url	0x06
oipf_access_status	0x07
oipf_status	0x08
(reserved)	0x09-0x7F
(user defined)	0x80-0xFF

Table 12 OIPF specific datatype_id values

4.2.3.4.1.1.2 Mapping of messages to DAE API or Events

The OITF SHALL map the specific messages listed in Table 11 to DAE API or Events as described in Table 13:

Message	DAE API or Event
send_msg	sendDRMMessage
reply_msg	onDRMMessageResult
parental_control_info	onParentalRatingChange, onParentalRatingError
rights_info	onDRMRightsError
system_info	onDRMSystemMessage

Table 13 Mapping to DAE API or Events

The DRMSystemID attribute in DAE API or Events are mapped to the ca_system_id field in the SAS_async_msg APDU. The ca_system_id field is filled by extracting the numeric value from the DRMSystemID string, such that "urn:dvb:casystemid:" is removed and the remaining number is converted from a string to a 16 bit integer. The DRMSystemId is build by prefixing the 16 bit integer converted to a decimal number string with "urn:dvb:casystemid:" as described in [OIPF_META].

Private data are array of bytes encoded for DAE API or Events attributes in a string using a hexadecimal representation, as defined for xs:hexBinary type used in XML schemas. In CI+ SAS_async_msg fields, the private data is encoded in bytes.

Precise mapping of DAE API or Events and attributes are described in the following sections.

4.2.3.4.1.1.3 send_msg

A native application or DAE application SHOULD use the ***send_msg*** message to provide DRM specific messages to the CSPG-CI+.

When requested by either a native or DAE application, the OITF SHALL send the ***send_msg*** message to the CSPG-CI+ to exchange DRM messages. Examples of usage are:

- Service Provider handles the purchase of content at the server side and then uses the ***send_msg*** message via a DAE application to ask the CSPG-CI+ to retrieve the associated license.
- Service provider sends the ***send_msg*** message via a DAE application to the CSPG-CI+ to force the CSPG-CI+ to purchase a specific program.

The data types for the ***send_msg*** message are listed in the following table.

Syntax	Occurrence number
`oipf_ca_vendor_specific_information`	1

Table 14 *send_msg* message data types

oipf_ca_vendor_specific_information Vendor specific information. The maximum length is 65000 bytes.

When a DAE application calls the sendDRMMessage API with msgType set to the MIME type "application/vnd.oipf.cspg-hexbinary" and a DRMSystemId set to a ca system id supported by the CSPG-CI+, the OITF SHALL send a ***send_msg*** message to the CSPG-CI+.

The prototype of the sendDRMMessage API defined in [OIPF_DAE] is recalled here:

String sendDRMMessage(String msgType, String msg, String DRMSystemID)

The OITF SHALL map the attributes of the called DAE API as follows:

- the DRMSystemId attribute is mapped to the ca_system_id field as described in section 4.2.3.4.1.1.2.
- the private data in msg attribute encoded in a string using a hexadecimal representation, as defined for xs:hexBinary type used in XML schemas is decoded to bytes before passing it to ***send_msg*** message in the oipf_ca_vendor_specific_information field as described in section 4.2.3.4.1.1.2.

4.2.3.4.1.1.4 reply_msg

The CSPG-CI+ SHALL send the ***reply_msg*** message to the OITF to provide the status of the ***send_msg*** message.

The data types for the ***reply_msg*** message are listed in the following table.

Syntax	Occurrence number
`oipf_status`	1
`oipf_ca_vendor_specific_information`	0..1

Table 15 *reply_msg* message data types

oipf_status If equal to 0, the ***send_msg*** message has been successfully handled by the CSPG-CI+ and a oipf_ca_vendor_specific_information may be available.
If equal to 1, the ***send_msg*** message failed because an unspecified error occurred.

	If equal to 2, the ***send_msg*** message failed because the CSPG-CI+ was unable to complete the necessary computations in the time allotted. If equal to 3, the ***send_msg*** message failed because oipf_ca_vendor_specific_information has a wrong format. If equal to 4, the ***send_msg*** message failed because user consent is needed for that action. If equal to 5, the ***send_msg*** message failed because the specified CA system in ca_system_id is unknown. Unspecified status values SHOULD be considered as, message failed because an unspecified error occurs.
oipf_ca_vendor_specific_information	Vendor specific information. The maximum length is 65000 bytes.

NOTE: A service provider should not provide a DRM Message in metadata (BCG, SD&S, CAD) and expect a response in oipf_ca_vendor_specific_information of ***reply_msg*** message, if these metadata are handled by a native application. The native application sending the DRM message to the CSPG-CI+ will not know how to handle a response.

When receiving a ***reply_msg*** message with a transaction_id mapping to a send_msg message issued from a DAE application call to sendDRMMessage, the OITF SHALL issue an onDRMMessageResult event to the DAE application

The prototype of the onDRMMessageResult event defined in [OIPF_DAE] is recalled here:

function onDRMMessageResult(String msgID, String resultMsg, Integer resultCode)

The OITF SHALL set the attributes of the issued DAE event as follows:

- the msgID attribute set to the value returned to the called sendDRMMessage.
- the resultCode attribute is mapped to oipf_status field as follows:

oipf_status field	Description	resultCode attribute	Description
0	Successful	0	Successful
1	Unspecified error	1	Unknown error
2	Out of time	2	Cannot process request
3	Wrong format	6	Wrong format
4	User Consent Needed	4	User Consent Needed
5	Unknown DRM system	5	Unknown DRM system

- the resultMsg attribute set to the private data in oipf_ca_vendor_specific_information encoded in a string as described in section 4.2.3.4.1.1.2.

4.2.3.4.1.1.5 parental_control_info

The CSPG-CI+ SHALL send a ***parental_control_info*** message to advise the OITF whenever the selected program's rating changes. If the new rating does not meet the parental rating criterion (e.g. rating is at or above a certain threshold, for a rating system that is ordered from lower viewer age to higher viewer age), the program is not descrambled anymore . If the new rating meets the parental rating criterion (e.g. rating is under a certain threshold, for a rating system that is ordered from lower viewer age to higher viewer age), the program is descrambled again.

The data types for the ***parental_control_info*** message are listed in the following table.

Syntax	Occurrence number
`oipf_access_status`	1
`oipf_rating_type`	1
`oipf_rating_value`	1
`oipf_country_code`	0..n
`oipf_parental_control_url`	0..1

Table 16 *parental_control_info* message data types

oipf_access_status If equal to 0, the program is no longer being descrambled, access conditions to the program are no longer being met. A oipf_parental_control_url may be provided.
If equal to 1, the program is descrambled again.

oipf_rating_type Rating_type as defined in the parental_rating access_criteria_descriptor in [IEC62455].

oipf_rating_value 1-byte rating_value as defined in the parental_rating access_criteria_descriptor in [IEC62455].

oipf_country_code 2-byte optional country_codes as defined in the parental_rating access_criteria_descriptor in [IEC62455].

oipf_parental_control_url Optional url for connecting to the service provider, for unlocking the parental control.

The OITF SHALL support at least the parental rating system identified by the oipf_rating_type 0, which maps to the parental rating system in DVB Systems [DVB-SI].

If an oipf_parental_control_url is provided and the event is raised to a native application, the native application SHOULD launch the DAE with the oipf_parental_control_url that might allow to unlock parental control in the CSPG-CI+.

When the *parental_control_info* message is received and a DAE application is launched, the OITF SHALL issue the relevant event to the DAE application:

- onParentalRatingChange event, if the parental rating system specified by the oipf_rating_type is supported by the OITF.
- onParentalRatingError event, if the parental rating system specified by the oipf_rating_type is not supported by the OITF.

The prototype of the onParentalRatingChange and onParentalRatingError events defined in [OIPF_DAE] are recalled here:

function onParentalRatingChange(String contentID, ParentalRating rating, String DRMSystemID, Boolean blocked)

function onParentalRatingError(String contentID, ParentalRating rating, String DRMSystemID)

The OITF SHALL set the attributes of the issued event as follows:

- the contentId attribute is set to null or undefined.
- the rating attribute (ParentalRating object) is initialized as follows:

  - If the oipf_rating_type is supported by the OITF, the oipf_rating_type field is mapped into the scheme property of the ParentalRating object. If the oipf_rating_type is not supported by the OITF, the scheme is set to null or undefined.
  - The oipf_rating_value field is mapped into the value property of the ParentalRating object. If the oipf_rating_type is supported by the OITF, the name property of the ParentalRating object is filled with the string representation of the parental rating value. If the oipf_rating_type is not supported by the OITF, the name property is set to null or undefined.
  - The oipf_country_code field is mapped into the region property of the ParentalRating object

- the DRMSystemID attribute is mapped to the ca_system_id field as defined in section 4.2.3.4.1.1.2.
- The blocked attribute is mapped to oipf_access_status as follows

oipf_access_status field	Description	Blocked attribute	Description
0	program not descrambled	True	Content blocked
1	Program descrambled	False	Content not blocked

A DAE application SHOULD use a proprietary method using sendDRMMessage to unlock parental control.

If the program is no longer being descrambled (oipf_access_status=0), the OITF SHALL blank the video decoder output. The native or DAE application SHOULD not stop playing the program, as the program may become descrambled again later (access criteria change, parental unlocking etc).

If the program being played is descrambled again (oipf_access_status=1), the OITF SHALL display the video again.

4.2.3.4.1.1.6 rights_info

The CSPG-CI+ SHALL send a ***rights_info*** message to advise the OITF that access conditions or rights changed and that the CSPG-CI+ is no longer able or is able again to descramble all requested elementary streams. Once this message is received and if a DAE application is launched, the OIPF SHALL send the relevant event onDRMRightsError, as defined in [OIPF_DAE] sections 7.13.6 and 7.14.7, to the DAE application.

If the program is descrambled again, the OITF SHOULD display the program again. If the program is no longer being descrambled, the OITF MAY decide to stop the program and SHOULD use the oipf_rights_issuer_url, which may provide for the CSPG-CI+ information to let it retrieve missing rights.

The data types for the ***rights_info*** message are listed in the following table.

Syntax	Occurrence number
`oipf_access_status`	1
`oipf_rights_issuer_url`	0..1

Table 17 *rights_info* message data types

oipf_access_status If equal to 0, the program is no longer being descrambled, access conditions to the program are no longer being met. A oipf_rights_issuer_url may be provided. If equal to 1, the program is descrambled again.

oipf_rights_issuer_url Optional url for connecting to the service provider.

The prototype of the onDRMRightsError event defined in [OIPF_DAE] is recalled here:

> *function onDRMRightsError(Integer errorState, String contentID, String DRMSystemID, String rightsIssuerURL)*

When the *right_info* message is received and a DAE application is launched, the OITF SHALL issue the *onDRMRightsError* event to the DAE application.

The OITF SHALL set the attributes of the issued event as follows:

- The errorState attribute is mapped to oipf_access_status field as follows:

oipf_access_status field	Description	errorState attribute	Description
0	program not descrambled	0	No license
1	Program descrambled	2	Valid license

- The contentId attribute is set to null or undefined.
- The DRMSystemID attribute is mapped to the ca_system_id field as defined in section 4.2.3.4.1.1.2.
- The rightsIssuerURL is mapped to oipf_rights_issuer_url if this field is present. If the oipf_rights_issuer_url is not present, rightIssuerURL is set to null or undefined.

If the program is no longer being descrambled (oipf_access_status=0), the OITF SHALL blank the video decoder output. The native or DAE application SHOULD not stop playing the program, as the program may become descrambled again later (access criteria change, rights update etc).

If the program being played is descrambled again (oipf_access_status=1), the OITF SHALL display the video again.

4.2.3.4.1.1.7 system_info

The CSPG-CI+ SHALL send a ***system_info*** message to advise the OITF of any DRM related event, e.g. the removal of a smartcard. Once this message is received and if a DAE application is launched, the OIPF SHALL send the relevant event onDRMSystemMessage, as defined in [OIPF_DAE] section 7.6.1, to the DAE application.

The data types for the ***system_info*** message are listed in the following table.

Syntax	Occurrence number
`oipf_ca_vendor_specific_information`	1

Table 18 *system_info* message data types

oipf_ca_vendor_specific_information Vendor specific information. The maximum length is 65000 bytes.

When the *system_info* message is received and if a DAE application is launched, the OITF SHALL issue the onDRMSystemMessage event to the DAE application.

The prototype of the onDRMSystemMessage event defined in [OIPF_DAE] is recalled here:

function onDRMSystemMessage(String DRMSystemID, String msg)

The OITF SHALL set the attributes of the issued event as follows:

- The DRMSystemID attribute is mapped to the ca_system_id field as defined in section 4.2.3.4.1.1.2.
- The msg attribute set to the private data in oipf_ca_vendor_specific_information encoded in a string as described in section 4.2.3.4.1.1.2.

9.3.2 Metadata – DRM Control Information

Then section 4.2.3.10.2 is updated accordingly. Table 20 (on DRMContentID and RightsIssuerURL) and Table 21 are updated as follow.

Element / Attribute Name	**Element / Attribute Mapping for CSPG-CI+**
DRMControlInformation	
DRMSystemID	SHALL be set to the value defined for the specific protection system in the CSPG-CI+, in section 4.2.3.10.1
DRMContentID	Vendor specific information.
RightsIssuerURL	SHOULD be set to the RightsIssuerURL which is provided in the ***rights_info*** message defined in section 4.2.3.4.1.1.6.
SilentRightsURL	MAY be set to an URL allowing retrieval of a message to be forwarded to the CSPG-CI+ in order to silently get updated rights. The MIME type or the HTTP response SHALL be "application/vnd.oipf.cspg-hexbinary" and the body of the HTTP response SHALL be an hexadecimal string as described in section 4.2.3.4.1.1.2.
PreviewRightsURL	MAY be set to an URL allowing retrieval of a message to be forwarded to the CSPG-CI+ in order to get preview rights. The MIME type or the HTTP response SHALL be "application/vnd.oipf.cspg-hexbinary" and the body of the HTTP response SHALL be a hexadecimal string as described in section 4.2.3.4.1.1.2.
DoNotRecord	Vendor specific mapping
DoNotTimeShift	Vendor specific mapping
DRMPrivateData	DRMPrivateData structure SHALL be substituted by the HexBinaryPrivateData structure.
mimeType	SHALL be set to the mime type of the DRMPrivateData. For CSPG-CI+, it SHALL therefore be set to the following MIME type: "application/vnd.oipf.cspg-hexbinary"

Table 20 DRMControlInformation Mapping for CSPG-CI+

Element / Attribute Name	Element / Attribute Description
HexBinaryPrivateData	
Message	A hexadecimal encoded sequence of bytes to be sent to the CSPG-CI+ using ***send_msg*** message

Table 21 HexBinaryPrivateData Structure

9.4 Alignment with CI-Plus 1.3

9.4.1 Change in section reference

A reference to section of the CI-Plus specification is updated in section 4.2.3.4.2.1 as follow.

The OITF SHALL support the Low-Speed Communications resource with IP extension as specified in [CI+], section 14.2.

9.4.2 Removing reference to CI-Plus PVR resource

The reference to the PVR resource is removed from sections 4.2.3.7 and 4.2.3.8 which now read as follow.

4.2.3.7 Personal Video Recorder

PVR functionality is supported by using URI (Usage Rule Information) as defined in [CI+], section 5.7.

When the OITF is asked to store content, it SHALL send the content to CSPG-CI+. The content is returned from CSPG-CI+ and recorded in accordance with the URI associated with the content.

4.2.3.8 Time Shifting

Time Shifting functionality is supported by using URI (Usage Rule Information) as defined in [CI+], section 5.7.

When the OITF is asked to time shift content, it SHALL store the content returned from CSPG-CI+ before rendering in accordance with the URI associated to the content.

9.5 CI+ Registered Service Mode

The requirement to support the CI+ Registered Service Mode is relaxed to being recommended, as it should be a deployment decision as to whether this feature is used. Thus the text of section 4.2.3.9.1 is amended to:

As the network offered in the Open IPTV Forum context is a bi-directional communication channel, the optional Registered Service Mode (RSM) in the CI+ specification [CI+] is RECOMMENDED to be supported by CSPG-CI+.

9.6 GBA Authentication

The Ks_NAF key used in OIPF specifications is named Ks_(ext)_NAF in 3GPP specifications as this key Ks_(ext)_NAF could be computed by the IG (Ks_NAF) or the ISIM (Ks_ext_NAF). So name Ks_NAF is changed into Ks_(ext)_NAF in section 5.4.4.2 which now reads as follow.

5.4.4.2 Re-use of GBA Authentication – Using HTTP Digest Authentication

The key Ks that was established during the GBA registration MAY be used later on for authentication between OITF functions and services (i.e., Application Servers). Each time an OITF needs to access a service offered by an AS (i.e., NAF) that requires GBA Authentication, a specific key Ks_(ext)_NAF SHALL be derived by the IG and the server side GBA Single Sign-on function (the BSF). This generated key SHALL be conveyed to the OITF function in the residential network by the IG, and to the AS by the server side GBA Single Sign-on function (the BSF). The key Ks_(ext)_NAF SHALL then be used for authentication between the OITF function and the AS, using HTTP Digest authentication as specified by [3GPP24.109].

If the OITF has registered to an IG which supports GBA Authentication, the OITF SHALL act as a User Equipment in [3GPP24.109] and therefore SHALL signal in its User Agent that it supports GBA Authentication.

When a SAA (acting like a NAF in [3GPP24.109]) requests GBA Authentication, the OITF SHALL retrieve GBA Credentials for the specified NAF_ID from the IG as specified in [OIPF_PROT], and SHALL perform HTTP Digest authentication as specified by [3GPP24.109].

As a pre-requisite to this procedure, the GBA registration (cf. 5.4.4.1) MUST have been successfully completed.

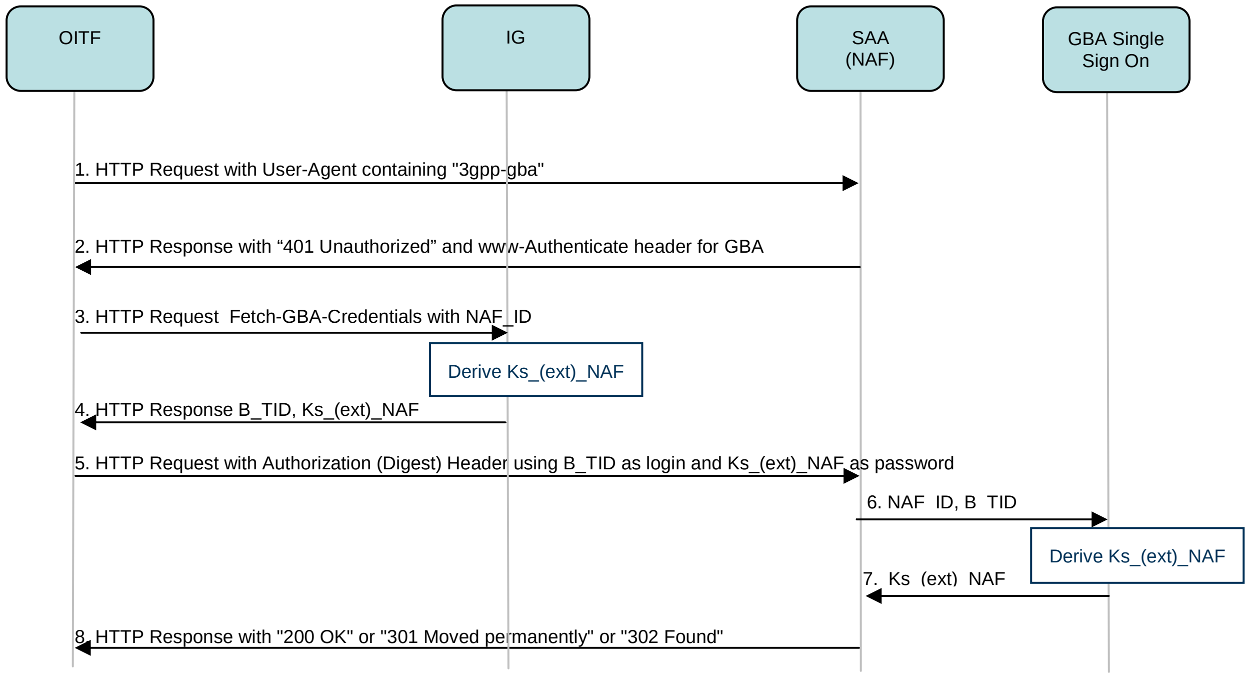

OITF
IG
SAA (NAF)
GBA Single Sign On
1. HTTP Request with User-Agent containing "3gpp-gba"
2. HTTP Response with “401 Unauthorized” and www-Authenticate header for GBA
3. HTTP Request Fetch-GBA-Credentials with NAF_ID
Derive Ks_(ext)_NAF
4. HTTP Response B_TID, Ks_(ext)_NAF
5. HTTP Request with Authorization (Digest) Header using B_TID as login and Ks_(ext)_NAF as password
6. NAF ID, B TID
Derive Ks_(ext)_NAF
7. Ks (ext) NAF
8. HTTP Response with "200 OK" or "301 Moved permanently" or "302 Found"

Figure 25 Authentication between an OITF and an SAA Based on GBA Keys

Figure 25 shows the message sequence for authentication between an OITF function and an SAA based on the previously established GBA key. It contains the following steps:

1. OITF function sends a request for a resource (e.g., service) to the SAA (NAF). It is assumed here that the resource requires authentication. The User-Agent string in the HTTP request contains "3gpp-gba" indicating to the SAA that it supports GBA authentication.

2. The SAA (NAF) returns a 401 Unauthorized message, the realm indicates that 3GPP bootstrapping is used and provides the NAF FQDN as defined in [3GPP24.109].

3. OITF sends a request including the NAF_ID to the IG to retrieve GBA credentials, and IG generates Ks_NAF in case of GBA_ME or Ks_ext_NAF with the co-operation of the ISIM in case of GBA_U (Ks_(ext)_NAF). Note: according to [3GPP33.220], the NAF_ID is constructed as follows: NAF_ID = FQDN of the NAF || Ua security protocol identifier. The identifier for Ua security protocol HTTP Digest authentication according to [3GPP24.109] is (0x01,0x00,0x00, 0x00,0x02). The request format is specified in [OIPF_PROT], section 5.3.6.2, step 1.

Ks_(ext)_NAF is computed as Ks_(ext)_NAF = KDF (Ks, "gba-me", RAND, IMPI, NAF_ID), where KDF is the key derivation function as specified in Annex B of [3GPP33.220], and the key derivation parameters consist of the user's IMPI, the NAF_ID and RAND.

4. IG returns Ks_(ext)_NAF, B-TID and the lifetime of the key Ks_(ext)_NAF to OITF. The lifetime indicates the expiry time of the key Ks_(ext)_NAF and is equal to the lifetime of the key Ks (which was specified by the BSF during the GBA bootstrapping procedure). The response format is specified in [OIPF_PROT], section 5.3.6.2, step 2.
5. The OITF function repeats the request with an Authorisation header, using B-TID as username and Ks_(ext)_NAF as password.
6. SAA (NAF) sends B-TID and its NAF_ID to the GBA Single Sign-on function in provider network, the GBA Single Sign-on function retrieves Ks and calculates Ks_(ext)_NAF.
7. The GBA Single Sign-on function in provider network returns Ks_(ext)_NAF, together with its lifetime, to SAA (NAF).

   Note the key lifetime returned by the GBA Single Sign-on function is equal to the lifetime of the corresponding Ks. But the SAA (NAF) may choose a shorter key lifetime based on local policy and/or application-specific needs.
8. If Ks_(ext)_NAF has expired, the SAA (NAF) shall send a suitable bootstrapping renegotiation request to the OITF, according to [3GPP33.220]. Otherwise the SAA (NAF) uses Ks_(ext)_NAF to authenticate the request. Upon successful authentication, the SAA (NAF)/service serves the request or redirects the OITF to the service. The response may contain session management information (cookie, URL parameter).

The message format for steps 3 and 4 are specified in the section 5.3.6.2 of [OIPF_PROT].

9.7 Home Network clarification

The definition of HN (Home Network) is removed and the following definition is added in section 3.3 as follow.

SPP	Service Platform Provider

The use of HN or Home Network ambiguous terminology is clarified in sections 5.5.2, 5.5.3 and 5.5.4 which now read as follows.

5.5.2 Prerequisites

Prior to the first IMS Registration (and hence prior to the first SIP Digest or IMS AKA) protocol execution, the following parameters MUST be provisioned:

- to the IG[1]:
    - for SIP Digest:
        - one or more IP Multimedia Private Identities (IMPI),
        - one or more IP Multimedia Public Identities (IMPU), each associated to one or more IMPIs,
        - one or more passwords, each assigned to one and only one of the IMPIs provisioned to the IG,

[1] In case of IMS AKA, these parameters are in a UICC with an ISIM or USIM application.

        - a Service Platform Provider Network Domain Name.
    - for IMS AKA, an ISIM or a USIM application shall always be used for authentication , as described in [3GPP33.203]. For the purpose of this document, the ISIM is a term that indicates a collection of IMS security data and functions on a UICC.
        - The ISIM SHALL include :
            - one IMPI.
            - one or more IP Multimedia Public Identities (IMPU), associated with the IMPI
            - a SPP Network Domain Name referred as Home Network Domain Name in 3GPP specifications
            - Support for sequence number checking in the context of IMS Domain
            - An Authentication key
            - The same framework for algorithms as specified for USIM
        - There shall only be one ISIM for each IMPI.
- and to the User Database, the IMS subscription information comprising:
    - the IMPI(s) and IMPU(s) provisioned to the IG,
    - the association of the IMPU(s) to the IMPI(s),
    - and for SIP Digest the password(s) provisioned to the IG. The User Database stores each password against the IMPI it is assigned to.
    - And for IMS AKA the Authentication Key contained and protected within the UICC in the IG. The User Database stores each Authentication Key against the IMPI it is assigned to.

Methods for provisioning these parameters to IG and User Database functional entities are out of scope of this specification.

5.5.3 SIP Digest Message Flows

Figure 28 shows the message flow for SIP Digest authentication, which is interlaced into IMS Registration messages:

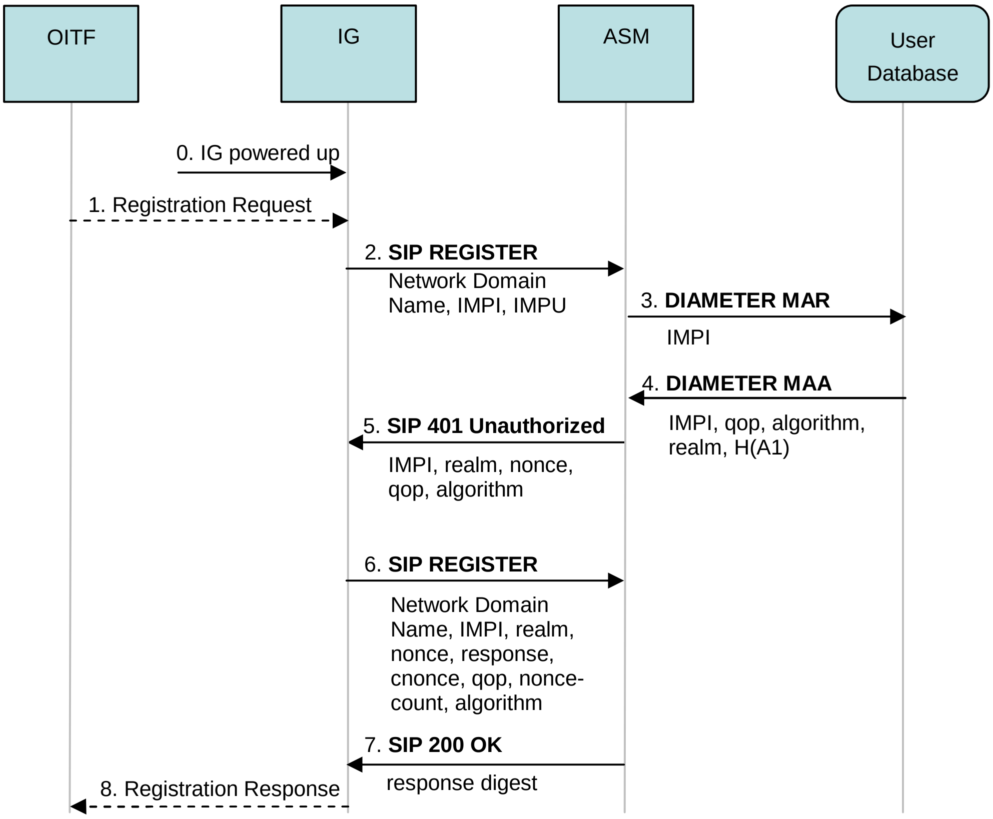


Figure 28 SIP Digest Message Flow Interlaced into IMS Registration

0. The IG is powered up. This can initiate the execution of steps 2 – 7.

1. **OITF to IG: Registration Request**
The OITF sends a request for registration to the IMS Gateway (IG), when needed (the end user explicitly logs on for personalized services).

2. **IG to ASM: SIP REGISTER**
This request contains the SPP Network Domain Name of the IG's IMS home network, an IMPI and an IMPU. If the ASM has a valid SIP Digest authentication vector (SD-AV) for the specific IMPI, steps 3, 4 and 5 are omitted.

3. **ASM to User Database: DIAMETER MULTIMEDIA AUTH REQUEST (MAR)**
The ASM requests a SD-AV from the User Database with respect to the IMPI received in step 2.

4. **User Database to ASM: DIAMETER MULTIMEDIA AUTH ANSWER (MAA)**
Along with the IMPI, the User Database sends a SD-AV to the ASM containing the following data: qop value (quality of protection), the authentication algorithm, realm, and a hash value H(A1) of the IMPI, realm, and password. [RFC2617] provides additional information on the values in the authentication vector for SIP Digest based authentication. Upon reception of the MAA message, the ASM stores the H(A1) value and generates the nonce value needed to challenge the IG.

5. **ASM to IG: SIP 401 Unauthorized**
The ASM denies the IG authentication but sends a SIP 401Unauthorized message to the IG in order to challenge the IG. This message contains the IMPI, the nonce, the authentication algorithm, and the realm and qop values.

6. **IG to ASM: SIP REGISTER**
After reception message 5, the IG generates a client nonce (cnonce) and calculates an authentication response value using this cnonce and other values received in step 5 (see [RFC2617]). The IG sends a new SIP REGISTER request to the ASM, this time with the authentication response along with the parameters IMPI, realm, nonce, response, cnonce, qop, nonce-count, and algorithm.

7. **ASM to IG: SIP 200 OK** (successful case)
After reception of the SIP REGISTER message containing the authentication response value, the ASM calculates the *expected* response value using the previously stored H(A1) and the stored nonce value together with other parameters (see [RFC2617]). If the response value received from the IG equals the expected response value, the IG has been authenticated and the IMPU is registered in the ASM. In this successful case, the ASM sends the SIP 200 OK from ASM to the IG, enabling the IG to authenticate the SPP Network. This SIP 200 OK message contains a response digest calculated using the cnonce value generated by the IG prior to sending message 6.

8. **IG to OITF: Registration Response**
The IG informs the OITF about the result of the registration procedure (when step 1 was executed).

The details of the messages 2 – 7 are specified in [3GPP24.229].

5.5.4 IMS AKA Message Flows

To support IMS AKA, a UICC with an ISIM or USIM application must be integrated into the IMS Gateway (IG). From the IMS point of view, the IG thereby takes the role of an IMS Subscriber. The UICC stores a long-term secret key K which is shared between the ISIM or USIM application and a User Database belonging to the network operator that provides the ISIM or the USIM. Figure 29 shows the high-level message flows for user identification and authentication based on the IMS AKA procedure

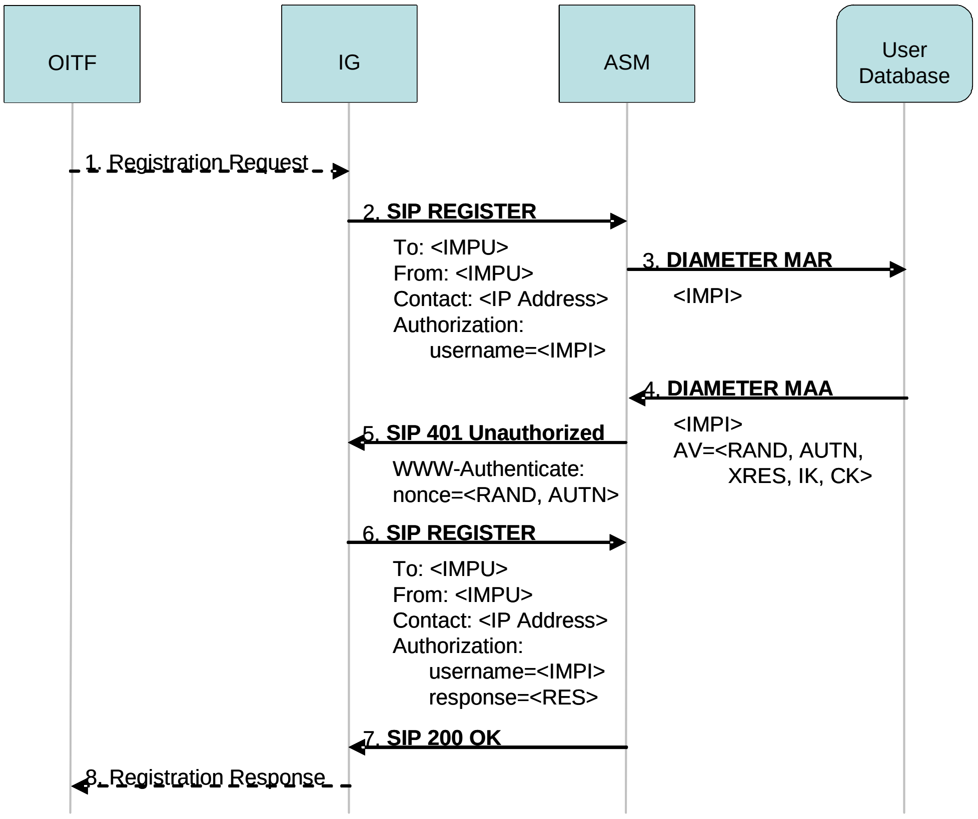


Figure 29 User Identification and Authentication based on the IMS AKA procedure

0. The IG is powered up. This can initiate the execution of steps 2 .7

1. **OITF to IG: Registration Request**
The OITF sends a request for registration to the IMS Gateway (IG), when needed (the end user explicitly logs on for personalized services)

2. **IG to ASM: SIP REGISTER**
This request contains the SPP Network Domain Name of the IG's IMS home network, the IMPI and the IMPU. All this data is read from the ISIM.

3. **ASM to User Database: DIAMETER MULTIMEDIA AUTH REQUEST (MAR)**
ASM requests authentication data from the User Database with respect to the IMPI received in step 2.

4. **User Database to ASM: DIAMETER MULTIMEDIA AUTH ANSWER (MAA)**
The User Database sends an Authentication Vectors (AV) to the ASM containing the following data: random challenge RAND, answer XRES expected by the IG in step 6, network authentication token AUTN, integrity key IK, and ciphering key CK. The authentication token AUTN contains a message authentication code (MAC) enabling the IG to authenticate the SPP Network (see step 5).

5. **ASM to IG: SIP 401 Unauthorized**
At this point in time, the ASM denies the IG authentication. Instead, it sends a SIP Unauthorized message with a WWW-Authenticate header to the IG. This header contains RAND and AUTN. After reception of this message, the IG verifies the message authentication code contained in AUTN thereby authenticating its SPP Network.

6. **IG to ASM: SIP REGISTER**
ISIM computes the value RES on input of its version of the secret key K stored on the UICC of the IG. The

IG sends a new SIP REGISTER request to the ASM, this time with RES as response to the challenge the ASM initiated in step 5.

7. **ASM to IG: SIP 200 OK**
   If RES = XRES (successful case), ASM considers the IG as authenticated, and binds IMPU to the IP address <IP address>.

8. **IG to OITF: Registration Response**
   The IG informs the OITF about the result of the registration procedure. (when step 1 was executed)

In case of success, the ISIM of the IG is able, based on its knowledge of the secret key K and the authentication token AUTN, to calculate the same values of the integrity key IK and the ciphering key CK as those that the ASM received in step 4 from the User Database. The IG and the ASM use IK and CK to establish IPSec Security Associations for protecting SIP signaling messages over the IG – ASM reference point

The details of the messages 2 -7 are specified in [3GPP24.229].

9.8 TLS Version

TLS version is clarified in section 5.2.3 which now reads as follow.

5.2.3 Common Requirements

On both HNI-INI and HNI-IGI interface, the **OITF** SHALL support all of the following mechanisms, redirection, and security for the HTTP protocol and HTML support:

- standard HTTP requirements: HTTP redirection, HTTP cookies
- URL parameters
- HTML forms and HTTP Post in forms
- TLS/SSL – TLS 1.2 SHOULD be supported, if not then TLS 1.1 SHOULD be supported, otherwise TLS 1.0 SHALL be supported. The OITF SHALL support TLS Renegotiation Extension as described in [RFC5746].

To avoid extra message exchanges, the **OITF** SHALL provide in requests, when available (see section 5.6):

- HTTP authentication header (Authorization)
- HTTP cookie header (Cookie)

Annex A: Changes to Section 7.12

7.12 Metadata APIs

This section defines the Javascript APIs used by DAE applications for reading and searching metadata about programmes and channels. This API does not specify whether these query operations are carried out on the OITF or whether they require communication with a server.

The metadata API provides DAE applications with high-level access to metadata about programmes and channels. This document describes the mapping between this API and ~~BCG~~CoD and programme guide metadata. Mappings between this API and other metadata sources are not specified in this document.

This section SHALL apply for OITFs that have indicated `<clientMetadata>` with value “`true`” and a “`type`” attribute with value “`bcg`” or “`dvb-si`” as defined in section 9.3.7 in their capability profile ~~and MAY apply for OITFs that have indicated `<clientMetadata>` with value “`true`” and a “`type`” attribute with value “`dvb-si`”~~.

Note that in order to access the metadata of programmes and channels several extensions to the `Programme` and `Channel` classes have been defined when the OITF has indicated support for `<clientMetadata>`. See sections 7.16.2.3 “Metadata extensions to programme” and 7.13.11.3 “Metadata extensions to channel” for more information).

The functionality as described in this section is subject to the security model of Section 10, (in particular section 10.1.3.6).

7.12.1 The application/oipfSearchManager embedded object

OITFs SHALL implement the “`application/oipfSearchManager`” embedded object. This object provides a mechanism for applications to create and manage metadata searches.

The following example shows how a metadata search can be constructed and executed:

```
// Event handler function for asynchronous search results
function handleSearchResults() {
  if ((state == 0) || (state ==1)) {
    //more results are available, or our search has finished

    // update the results.  Doing this asynchronously means
    // that if we’re working with the current set of results,
    // we get the new results  when it suits the application.
    search.result.update();

    // do stuff with the results
    var myResult = search.result[0];

             //get the next page of results
    search.result.getResults(10, 20);
  }
}

// Function that creates and starts a search
function doSearch() {

  // create a new search for on-demand content
  mySearchManager = document.getElementById(“searchManager”);
  mySearch = mySearchmanager.createSearch(2);

  // search for any programme with “space” in the title as a word
  // or part of a word
  myQuery = mySearch.createQuery(
      “urn:tva:transport:fieldIDs:2002:Title”,
      6,
      “space”);
  mySearch.setQuery(myQuery);
  // return results alphabetically by title
  mySearch.orderBy(“urn:tva:transport:fieldIDs:2002:Title”, true);

  mySearchManager.onMetadataSearch = handleSearchResults;

  if (mySearch.results.getResults(0, 10)) {
    // some results are available immediately, e.g. because
    // they were cached

    // do stuff with the results
    var myResult = mySearch.results[0];
  }
}
```

7.12.1.1 Properties

readonly Integer **guideDaysAvailable**

The number of days for which guide data is available. A value of -1 means that the amount of guide data available is unknown. ~~This information is derived from the `start` and end attributes of the `Schedule` entry in the Programme Location Table.~~

function **onMetadataUpdate**(Integer action, Integer info, Object object)

This function is the DOM 0 event handler for events indicating changes in metadata. This SHALL be raised when changes to the parental control settings change the lock status of an item, or when a new version of the metadata becomes available. The specified function is called with the arguments action, info and object. These arguments are defined as follows:

Integer action – the type of update that has taken place. This field will take one of the following values:

Value	**Description**
1	A new version of metadata is available (see clause 4.1.2.1.2 of [META]) or and applications SHOULD discard all references to Programme objects immediately and re-acquire them.
2	A change to the parental control flags for a content item has occurred (e.g. the user has unlocked the parental control features of the receiver, allowing a blocked item to be played).
3	A flag affecting the filtering criteria of a channel has changed. Applications MAY listen for events with this action code to update lists of favourite channels, for instance.

Integer info – extended information about the type of update that has taken place. If the action argument is set to the value 3, the value of this field SHALL be one or more of the following:

Value	**Description**
1	The list of blocked channels has changed.

2	A list of favourite channels has changed.
4	The list of hidden channels has changed.

If the action argument is set to the value 2, the value of this field SHALL be one or more of:

Value	Description
1	The block status of a content item has changed.
2	The lock status of a content item has changed.

This field is treated as a bitfield, so values MAY be combined to allow multiple reasons to be passed.

Object object – the affected channel, programme, or CoD asset. If more than one is affected, then this argument SHALL take the value null.

`function onMetadataSearch( MetadataSearch search, Integer state )`

This function is the DOM 0 event handler for events relating to metadata searches. The specified function is called with the arguments search and state. These arguments are defined as follows:

MetadataSearch search – the affected search

Number state – the new state of the search

Value	Description
0	Search has finished. This event SHALL be generated when a search has completed or been aborted.
1	More search results are available. Calling `update()` on the `SearchResults` object SHALL update the list of results to include the newly-retrieved data.
2	The data returned by the search is no longer valid, e.g. because of a change in the metadata. Applications that still require the data SHALL repeat the search.

For the intrinsic events “`onMetadataSearch`” and “`onMetadataUpdate`”, corresponding DOM level 2 events SHALL be generated, in the following manner:

Intrinsic event	Corresponding DOM 2 event	DOM 2 Event properties
`onMetadataSearch`	`MetadataSearch`	Bubbles: No Cancelable: No Context Info: ~~see section 7.12.5~~ `search, state`
`onMetadataUpdate`	`MetadataUpdate`	Bubbles: No Cancelable: No Context Info: ~~see section 7.12.6~~ `action, info, object`

These events are targeted at the `application/oipfSearchManager` object.

7.12.1.2 Methods

<table>
<tr><td colspan="3"><code>MetadataSearch <b>createSearch</b>(Integer searchTarget)</code></td></tr>
<tr><td>Description</td><td colspan="2">Create a <code>MetadataSearch</code> object that can be used to search the metadata.</td></tr>
<tr><td>Arguments</td><td><i>searchTarget</i></td><td>The metadata that should be searched.
Valid values of the <code>searchTarget</code> parameter are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Metadata relating to scheduled content shall be searched.</td></tr>
<tr><td>2</td><td>Metadata relating to ~~on-demand~~ content on demand shall be searched.</td></tr>
</table>
These values are treated as a bitfield, allowing searches to be carried out across multiple search targets.</td></tr>
</table>

`ChannelConfig` **`getChannelConfig`**`()`	
Description	Returns the channel line-up of the tuner in the form of a `ChannelConfig` object as defined in Section 7.13.8. This includes the favourite lists. The `ChannelConfig` object returned from this function SHALL be identical to the `ChannelConfig` object returned from the `getChannelConfig()` method on the video/broadcast object as defined in 7.13.3.

7.12.2 The MetadataSearch class

A `MetadataSearch` object represents a query of the BCG and SD&S metadata about available programmes. Applications can create `MetadataSearch` objects using the `createSearch()` method on the `application/oipfSearchManager` object. When metadata queries are performed on a remote server, the protocol used is defined in section 4.1.2.2 of [META].

Changes to constraints or the ordering of search results SHALL be applied when the `getResults()` method on the corresponding `SearchResults` object is called.

Due to the nature of metadata queries, searches are asynchronous and events are used to notify the application that results are available. `MetadataSearchEvents` SHALL be targeted at the `application/oipfSearchManager` object.

To minimise race conditions, results are updated on request rather than dynamically. Upon receipt of a `MetadataSearchEvent` indicating that more results are available, applications SHALL call `update()` on the corresponding `SearchResults` object to get the new results.

7.12.2.1 Properties

~~`readonly Integer` **`id`**~~

~~The ID of the search. This can be used by applications to match asynchronous events to the search that generated them. The value of this field is generated automatically and is implementation-dependent.~~

`readonly Integer` **`searchTarget`**

The target(s) of the search. Valid values of the `searchTarget` parameter are:

Value	Description
1	Metadata relating to scheduled content SHALL be searched.
2	Metadata relating to on-demand content SHALL be searched.

These values SHALL be treated as a bitfield, allowing searches to be carried out across multiple search targets.

`Query` **`query`**

The query that will be carried out by this search.

`readonly SearchResults` **`result`**

The results found so far~~, sorted by logical channel number and time~~.

~~This property MAY only be valid after a call to `update()`.~~ The values within result MAY change after ~~subsequent~~ calls to its `update()` method.

7.12.2.2 Methods

`void setQuery( Query query )`		
Description	Set the query terms to be used for this search, discarding any previously-set query terms. Calling this method when a search is in progress SHALL: 1. Abort any outstanding requests for results (equivalent to calling `results.abort()`). 2. Invalidate any existing search results and dispatch a `MetadataSearch` event with a value of 2 for the state argument.	
Arguments	*query*	The query terms to be used

`void addRatingConstraint( ParentalRatingScheme scheme, Integer threshold )`		
Description	Constrain the search to only include results whose parental rating value is below the specified threshold.	
Arguments	*scheme*	The parental rating scheme upon which the constraint SHALL be based. If the value of this argument is `null`, any existing parental rating constraints SHALL be cleared.
	threshold	The threshold above which results SHALL NOT be returned. If the value of this argument is `null`, any existing constraint for the specified parental rating scheme SHALL be cleared.

`void addCurrentRatingConstraint()`	
Description	Constrain the search to only include results whose parental rating value is below the threshold currently set by the user.

`void addChannelConstraint( ChannelList channels )`		
Description	Constrain the search to only include results from the specified channels. If a channel constraint has already been set, subsequent calls to `addChannelConstraint()`SHALL add the specified channels to the list of channels from which results should be returned. For CoD searches, adding a channel constraint SHALL have no effect.	
Arguments	*channels*	The channels from which results SHALL be returned. If the value of this argument is `null`, any existing channel constraint SHALL be removed.

void **addChannelConstraint**(Channel channel)		
Description	Constrain the search to only include results from the specified channel. If a channel constraint has already been set, subsequent calls to `addChannelConstraint()` SHALL add the specified channel to the list of channels from which results should be returned. For CoD searches, adding a channel constraint SHALL have no effect.	
Arguments	*channel*	The channel from which results SHALL be returned. If the value of this argument is `null`, any existing channel constraint SHALL be removed.

void **orderBy**(String field, Boolean ascending)		

Description	Set the order in which results SHOULD be returned in future. Any existing search results SHALL not be re-ordered. Subsequent calls to `orderBy()` will apply further levels of ordering within the order defined by previous calls. For example:
	`orderBy("ServiceName", true);`
	`orderBy("PublishedStart", true);`
	will cause results to be ordered by service name and then by start time for results with the same channel number.~~Set the order in which results SHOULD be returned. Repeated calls to `orderBy()` allow more complex orderings to be set.~~	

Arguments	*field*	The name of the field by which results SHOULD be sorted. A value of `null` indicates that any currently-set order SHALL be cleared and the default sort order should be used.
	ascending	Flag indicating whether the results SHOULD be returned in ascending or descending order.

<table>
<tr><td colspan="3">Query <b>createQuery</b>(String field, Integer comparison, String value)</td></tr>
<tr><td>Description</td><td colspan="2">Create a metadata query for a specific value in a specific field within the metadata. Simple queries MAY be combined to create more complex queries. Applications SHALL follow the ECMAScript type conversion rules to convert non-string values into their string representation, if necessary.</td></tr>
<tr><td rowspan="2">Arguments</td><td><i>field</i></td><td>The name of the field to compare. Fields are identified by the fieldIDs defined in annex B.2 of [TVA-BID], or using simplified XPath notation. The '/' operator is the only permitted XPath operator.</td></tr>
<tr><td><i>comparison</i></td><td>The type of comparison. One of:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>True if the specified value is equal to the value of the specified field.</td></tr>
</table>
</td></tr>
</table>

		1	True if the specified value is not equal to the value of the specified field.
		2	True if the value of the specified field is greater than the specified value.
		3	True if the value of the specified field is greater than or equal to the specified value.
		4	True if the value of the specified field is less than the specified value.
		5	True if the value of the specified field is less than or equal to the specified value.
		6	True if the string value of the specified field contains the specified value. This operation SHALL be case insensitive, and SHALL match parts of a word as well as whole words (e.g. a value of "term" will match a field value of "Terminator").
		7	True if the specified field exists.
	value	The value to check. Applications SHALL follow the ECMAScript type conversion rules to convert non-string values into their string representation, if necessary	

`~~Boolean~~ void findProgrammesFromStream( Channel channel, Integer startTime, Integer count )`		
Description	Retrieve guide data for a specified number of programmes from a given channel from metadata contained in the stream as defined in section 4.1.3 of [META]. Searches made using this method will implicitly remove any existing constraints, ordering or queries created by prior calls to methods on this object. ~~Results may be returned both synchronously and asynchronously, depending on whether data is available from the cache. If `findProgrammesFromStream()` returns `false`, results are not available until the notification events have been returned and `result.update()` has been called. If `findProgrammesFromStream()` returns `true`, results are available immediately, and the application need not wait for `COMPLETE` events or call `result.update()` to obtain the results.~~	
Arguments	*channel*	The channel for which programme information should be found.
	startTime	The start of the time period for which results should be returned measured in seconds since midnight (GMT) on 1/1/1970. The start time is inclusive; any programmes starting at the start time, or which are showing at the start time, will be included in the search results. If `null`, the search will start from the current time.
	count	The number of programmes for which information should be returned.

7.12.3 The Query class

The `Query` class represents a metadata query that the user wants to carry out. This may be a simple search, or a complex search involving Boolean logic. Queries are immutable; an operation on a query SHALL return a new `Query` object, allowing applications to continue referring to the original query.

The examples below show how more complex queries can be constructed:

```
Query qa = ~~MetadataSearch~~mySearch.createQuery("Title", 6, "Terminator");
Query qb = ~~MetadataSearch~~mySearch.createQuery ("SpokenLanguage", 0, "fr-CA");
Query qc = qb.and(qa.negate());
```

7.12.3.1 Properties

Void.

~~`readonly String` **`field`**~~

~~The name of the field to compare. Fields are identified by the fieldIDs defined in annex B.2 of TS 102-822-6-1, or using simplified XPath notiation. The '/' operator is the only permitted XPath operator.~~

~~`readonly Integer` **`comparison`**~~

~~The type of comparison. One of:~~

~~**Value**~~	~~**Description**~~
~~0~~	~~True if the specified value is equal to the value of the specified field.~~
~~1~~	~~True if the specified value is not equal to the value of the specified field.~~
~~2~~	~~True if the value of the specified field is greater than the specified value.~~
~~3~~	~~True if the value of the specified field is greater than or equal to the specified value.~~
~~4~~	~~True if the value of the specified field is less than the specified value.~~
~~5~~	~~True if the value of the specified field is less than or equal to the specified value.~~
~~6~~	~~True if the string value of the specified field contains the specified value. This operation is case insensitive, non whole word.~~
~~7~~	~~True if the specified field exists.~~

~~`readonly String` **`value`**~~

~~The value to check. Applications SHALL follow the ECMAScript type conversion rules to convert non-string values into their string representation, if necessary.~~

7.12.3.2 Methods

`Query **and**( Query query )`		
Description	Create a query based on the logical AND of the predicates represented by the current query and the argument query.	
Arguments	*query*	The second predicate for the AND operation.

`Query **or**( Query query )`		
Description	Create a query based on the logical OR of the predicates represented by the current query and the argument query.	
Arguments	*query*	The second predicate for the OR operation.

`Query **not**()`	
Description	Create a new query that is the logical negation of the current query.

7.12.4 The SearchResults class

The `SearchResults` class represents the results of a metadata search. Since the result set may contain a large number of items, applications request a 'window' on to the result set, similar to the functionality provided by the OFFSET and LIMIT clauses in SQL.

Applications MAY request the contents of the result in groups of an arbitrary size, based on an offset from the beginning of the result set. The data SHALL be fetched from the appropriate source, and application SHALL be notified when the data is available.

Next to the properties and methods defined below a `SearchResults` object SHALL support the array notation to access the results in this collection.

7.12.4.1 Properties

`readonly Integer **length**`
The number of items in the currently available results. If results are fetched asynchronously, the value of this property SHALL be zero until after `update()` has been called.

`readonly Integer **offset**`
The current offset into the total result set.

readonly Integer **totalSize**
The total number of items in the result set. If results are fetched asynchronously, the value of this property SHALL be undefined until `getResults()` has been called and a `MetadataSearchEvent` notifying the application that results are available has been dispatched.~~The total number of items in the result set. This MAY be `undefined` until `getResults()` has been called.~~

7.12.4.2 Methods

Object **item**(Integer index)		
Description	Return the item at position index in the collection of currently available results, or `undefined` if no item is present at that position. This function SHALL only return objects that are instances of `Programme` when searching metadata for scheduled content, or `CODAsset`, `CODFolder`, or `CODService` when searching CoD metadata.	
Arguments	*index*	The index into the result set.

Boolean **getResults**(Integer offset, Integer count)		
Description	~~Retrieve~~ Perform the search and retrieve a subset of the items that match the query. Results MAY be returned both synchronously and asynchronously, depending on whether data is available from the cache. If `getResults()` returns `false`, results are not available until the notification events have been returned and `update()` has been called. If `getResults()` returns `true`, results are available immediately, and the application need not wait for `MetadataSearchEvents` indicating that results are available~~`COMPLETE` events~~ or call `update()` to obtain the results. For results returned as a result of the same call to `getResults()`, the full result set may build up over time – the availability of new entries in the result set will be indicated by notification events. Subsequent calls to `getResults()` will clear the result set, so only results fetched for the most recent call to `getResults()` will be available to applications.	
Arguments	*offset*	The number of items at the start of the result set to be skipped before data is retrieved.
	count	The number of results to retrieve.

void **abort**()	
Description	Abort any outstanding request for results. Items currently in the collection SHALL be removed (i.e. the value of the `length` property SHALL be 0 and any calls to `item()`SHALL return `undefined`).

void **update**()	
Description	~~Update the results available.~~Through the update method new results are made available to applications. When a call to getResults() has returned false and results are fetched asynchronously, this method must be called after an application has received a notification event informing it that new results are available. The results may be delivered over multiple notification events. Until this method is called, results returned by asynchronous requests SHALL NOT be available to applications. This ensures that applications have a consistent view of the available results, without the result set changing asynchronously. This enables applications to (for example) iterate over the current result set and update their UI before retrieving the new results which have been returned to the OITF but are not yet available to applications.

7.12.5 The MetadataSearchEvent class

Void

~~A `MetadataSearchEvent` notifies the application about the status of a search through the guide data. These will be targeted at the `application/oipfSearchManager` object.~~

~~To receive these events, applications MAY add listeners for "`MetadataSearch`" events.~~

~~7.2.5.1 Properties~~

~~`readonly Integer` **`state`**~~

~~The type of event. This SHALL take one of the following values:~~

~~Value~~	~~Description~~
~~0~~	~~Search has stopped. This event SHALL be generated at the termination of every search.~~
~~1~~	~~More search results are available. Calling `update()` on the `SearchResult` object SHALL update the list of results to include the newly-retrieved data.~~
~~2~~	~~The data returned by the search is no longer valid, e.g. because of a change in the metadata. Applications that still require the data SHALL repeat the search.~~

~~`readonly Integer` **`id`**~~

~~The ID of the metadata search with which this event is associated. This MAY be used by applications to match events to the search that generated them.~~

7.12.6 The MetadataUpdateEvent class

Void.

~~The `MetadataUpdateEvent` object indicates a change in the state of a channel or programme that may require applications to re-build their displays. A `MetadataUpdateEvent` will be raised when the user changes the parental control settings (changing the lock status of an item) or when autonomous updates mean that the channel line up or programme database has changed.~~

~~`MetadataUpdateEvents` are intended to allow applications to update their user interfaces in response to a change in the state of a channel or programme without having to poll all channels or programmes to identify affected items.~~

~~7.12.6.1 Properties~~

~~`readonly Integer` **`action`**~~

~~The type of update that has taken place. This field will take one of the following values:~~

~~**Value**~~	~~**Description**~~
~~1~~	~~The channel line-up has (or may have) changed and that the collection referred to by `ChannelConfig.all` has been updated. If an application has references to any `Channel` objects then it SHOULD dispose of them and rebuild its references. Where possible `Channel` objects are updated rather than removed - but their order in the `ChannelConfig.all` collection MAY have changed. Any lists created with `ChannelConfig.createFilteredList()` SHOULD be recreated in case channels have been removed.~~
~~2~~	~~A new version of metadata is available and applications SHOULD discard all references to `Programme` objects immediately and re-acquire them.~~
~~3~~	~~A change to the parental control flags for a content item has occurred (e.g. the user has unlocked the parental control features of the receiver, allowing a blocked item to be played).~~
~~4~~	~~A flag affecting the filtering criteria of a channel has changed. Applications MAY listen for events with this action code to update lists of favourite channels, for instance.~~

~~Extended information about the type of update that has taken place.~~

~~If the action field is set to the value 4, the value of this field SHALL be one or more of the following:~~

~~**Value**~~	~~**Description**~~
~~1~~	~~The list of blocked channels has changed.~~
~~2~~	~~A list of favourite channels has changed.~~
~~4~~	~~The list of hidden channels has changed.~~

~~If the action field is set to the value 3, the value of this field SHALL be one or more of:~~

~~**Value**~~	~~**Description**~~
~~1~~	~~The block status of a content item has changed.~~
~~2~~	~~The lock status of a content item has changed.~~

~~This field is treated as a bitfield, so values MAY be combined to allow multiple reasons to be passed.~~

~~`readonly Object` **`object`**~~

~~Object indicating the channel or programme that has been affected, or `null` if more than one item has been affected.~~

Annex B: Video component selection

7.16.5 Extensions for playback of selected media components

This section defines APIs for the selection of specific A/V components for playback.

NOTE: The term component may correspond to MPEG_2 components, but is not restricted to that.

7.16.5.1 Media playback extensions

7.16.5.1.1 Constants

The following constants are defined as properties on any objects implementing this section:

Name	Value	Use
`COMPONENT_TYPE_VIDEO`	0	Represents a video component. This constant is used for all video components regardless of encoding.
`COMPONENT_TYPE_AUDIO`	1	Represents an audio component. This constant is used for all audio components regardless of encoding.
`COMPONENT_TYPE_SUBTITLE`	2	Represents a subtitle component. This constant is used for all subtitle components regardless of subtitle format. NOTE: A subtitle component may also be related to closed captioning as part of a video stream.

7.16.5.1.2 Properties

`function` **`onSelectedComponentChanged ( Integer componentType )`**

This function is called when there is a change in the set of components being presented. This may occur if one of the currently selected components is no longer available and an alternative is chosen based on user preferences, or when presentation has changed due to a different component or set of components being selected.

OITFs MAY optimise event dispatch by dispatching a single event in response to several calls to `selectComponent()` or `unselectComponent()` made in rapid succession.

The specified function is called with one argument:

`Integer componentType` - The type of component whose presentation has changed, as represented by one of the constant values listed in section 7.13.4.1.1. If more than one component type has changed, this argument will take the value `undefined`.

7.16.5.1.3 Methods

`AVComponentCollection` **`getComponents`**`( Integer componentType )`		
Description	Returns a collection of `AVComponent` values representing the components of the specified type in the current stream. If componentType is set to null or undefined then all the currently active components are returned. One or more of the components returned MAY be passed back to one of the other methods unchanged (e.g. `selectComponent()`). If property `preferredAudioLanguage` in the `Configuration` object (refer to section 7.3.2.1) is set then a component is by default selected and is considered as an active component. If property `preferredSubtitleLanguage` in the `Configuration` object (refer to section 7.3.2.1) is set and property `subtitleEnabled` in `AVOutput` class (refer to section 7.3.5.1) is enabled then a component is by default selected and is considered as an active component.	
Argument	*componentType*	The type of component to be returned , as represented by one of the constant values listed in section 7.13.4.1.1

`AVComponentCollection` **`getCurrentActiveComponents`**`( Integer componentType )`		
Description	Returns a collection of `AVComponent` values representing the currently active components of the specified type that are being rendered. One or more of the components returned MAY be passed back to one of the other methods unchanged (e.g. `selectComponent()`).	
Argument	*componentType*	The type of currently active component to be returned. represented by one of the constant values listed in section 7.13.4.1.1

`void` **`selectComponent`**`( AVComponent component )`	
Description	Select the component that will be subsequently rendered when A/V playback starts or select the component for rendering if A/V playback has already started. If playback has started, this SHALL replace any other components of the same type that are currently playing. If property `preferredAudioLanguage` in the `Configuration` object (refer to section 7.3.2.1) is set then a component is by default selected and it is not necessary to perform `selectComponent()`. If property `preferredSubtitleLanguage` in the `Configuration` object (refer to section 7.3.2.1) is set and property `subtitleEnabled` in `AVOutput` class (refer to section 7.3.5.1) is enabled then a component is by default selected and it is not necessary to perform `selectComponent()`.

Argument	*component*	A component object available in the stream currently being played.

void **unselectComponent**(AVComponent component)		
Description	Stop rendering of the specified component of the stream. If property preferredAudioLanguage in the Configuration object (see section 7.3.2.1) is set then unselecting a specific component returns to the default preferred audio language. If property preferredSubtitleLanguage in the Configuration object (see section 7.3.2.1) is set and property subtitleEnabled in AVOutput class (see section7.3.5.1) is enabled then unselecting a specific component returns to the default preferred subtitle language. In order to stop rendering subtitles completely it is necessary to disable subtitles with property subtitleEnabled in AVOutput class.	
Argument	*component*	The component to be stopped.

void **selectComponent**(Integer componentType)		
Description	If A/V playback has already started, start rendering the default component of the specified type in the current stream. This SHALL replace any other components of the same type that are currently playing. If A/V playback has not started, the default component of the specified type will be subsequently rendered after calling the setChannel method on the video/broadcast object.	
Argument	*componentType*	The type of component for which the default component should be rendered.

void **unselectComponent**(Integer componentType)		
Description	If A/V playback has already started, stop rendering of the specified type of component. If A/V playback has not started, no components of the specified type will be subsequently rendered after calling the setChannel method on the video/broadcast object.	
Argument	*component*	The type of component to be stopped.

7.16.5.1.4 Events

For the intrinsic event “onSelectedComponentChange”, corresponding DOM level 2 events SHALL be generated, in the following manner:

Intrinsic event	Corresponding DOM 2 event	DOM 2 Event properties
`onSelectedComponentChange`	`SelectedComponentChange`	Bubbles: No Cancelable: No Context Info: `componentType`

7.16.5.2 The AVComponent class

`AVComponent` represents a component within a complete media stream - a single stream of video, audio or data that can be played or manipulated. This is not necessary for basic playback, record or EPG services. However, it provides a mechanism to get at extended streams for enhanced services.

For forward compatibility the DAE application SHALL check the value of the `type` property to ensure that it is accessing an `AVComponent` object of the correct type.

7.16.5.2.1 Properties

`readonly Integer` **`componentTag`**
The component tag identifies a component. The component tag identifier corresponds to the component_tag in the component descriptor in the ES loop of the stream in the PMT [EN 300 468], or `undefined` if the component is not carried in an MPEG-2 TS .

`readonly Integer` **`pid`**
The MPEG Program ID (PID) of the component in the MPEG2-TS in which it is carried, or `undefined` if the component is not carried in an MPEG-2 TS.

`readonly Integer` **`type`**
Type of the component stream. Valid values for this field are given by the constants listed in section 7.16.5.1.1.

`readonly String` **`encoding`**
The encoding of the stream. The value of video format or audio format defined in section 3 of [MEDIA] SHALL be used.

`readonly Boolean` **`encrypted`**
Flag indicating whether the component is encrypted or not.

7.16.5.3 The AVVideoComponent class

The `AVVideoComponent` class implements the `AVComponent` interface.

7.16.5.3.1 Properties

`readonly Number` **`aspectRatio`**
Indicates the aspect ratio of the video or `undefined` if the aspect ratio is not known. Values SHALL be equal to width divided by height, rounded to a float value with two decimals, e.g. 1.78 to indicate 16:9 and 1.33 to indicate 4:3.

7.16.5.4 The AVAudioComponent class

The `AVAudioComponent` class implements the `AVComponent` interface.

7.16.5.4.1 Properties

`readonly String` **`language`**
An ISO 639 language code representing the language of the stream.

`readonly Boolean` **`audioDescription`**
Has value true if the stream contains an audio description intended for people with a visual impairment, false otherwise.

`readonly Integer` **`audioChannels`**
Indicates the number of channels present in this stream (e.g. 2 for stereo, 5 for 5.1, 7 for 7.1).

7.16.5.5 The AVSubtitleComponent class

The `AVSubtitleComponent` class implements the `AVComponent` interface.

7.16.5.5.1 Properties

`readonly String` **`language`**
An ISO 639 language code representing the language of the stream.

`readonly Boolean` **`hearingImpaired`**
Has value true if the stream is intended for the hearing-impaired (e.g. contains a written description of the sound effects), false otherwise.

7.16.5.6 The AVComponentCollection class

An `AVComponentCollection` object represents a read-only collection of `AVComponent` objects. Next to the properties and methods defined below an `AVComponentCollection` object SHALL support the array notation to access the AV components in this collection.

7.16.5.6.1 Properties

`readonly Integer` **`length`**
The number of items in the collection.

7.16.5.6.2 Methods

`AVComponent` **`item`**`( Integer index )`		
Description	Return the item at position `index` in the collection.	
Arguments	*index*	The index of the item to be returned

Annex C: PVR API

7.10.1 The application/oipfRecordingScheduler embedded object

The OITF SHALL support the scheduling of recordings of broadcasts through the use of the following non-visual embedded object:

```
<object type=“application/oipfRecordingScheduler”/>
```

Note that the functionality in this section SHALL adhere to the security model as specified in Section 10.1.

7.10.1.1 Methods

`ScheduledRecording` **`record`**`( Programme programme )`		
Description	Requests the scheduler to schedule the recording of the programme identified by the `programmeID` property of the programme. The other data contained in the programme object is used solely for annotation of the (scheduled) recording. If such programme metadata is provided, it is retained in the `ScheduledRecording` object that is returned if the recording of the programme was scheduled successfully, reflecting the possibility that not all relevant metadata might be available to the scheduler. If the recording could not be scheduled due to a scheduling conflict or lack of resources the value `null` is returned. Note that the actual implementation of this method should enable the scheduler to identify the domain of the service that issues the scheduling request in order to support future retrieval of the scheduled recording through the `getScheduledRecordings` method.	
Arguments	*programme*	The programme to be recorded, as defined in 7.16.2.

`ScheduledRecording` **`recordAt`**`( Integer startTime, Integer duration, Integer repeatDays, String channelID )`	
Description	Requests the scheduler to schedule the recording of the broadcast to be received over the channel identified by `channelID`, starting at `startTime` and ~~continuing for~~ stopping at `startTime` + `duration`~~ minutes~~. If the recording was scheduled successfully, the resulting `ScheduledRecording` object is returned. If the recording could not be scheduled due to a scheduling conflict or lack of resources the value `null` is returned. The OITF SHOULD associate metadata with recordings scheduled using this method. This metadata MAY be obtained from the broadcast being recorded (for example DVB-SI in an MPEG-2 transport stream) or from other sources of metadata. If an application anticipates that the OITF may not be able to obtain any metadata, it SHOULD instead of using this method; - create a Programme object (using the createProgramme() method) containing the channelID, startTime and duration - populate that Programme object with metadata

<table>
<tr><td></td><td colspan="2">- pass that Programme object to the record(Programme) method.

Note that the actual implementation of this method should enable the scheduler to identify the domain of the service that issues the scheduling request in order to support future retrieval of the scheduled recording through the `getScheduledRecordings` method.</td></tr>
<tr><td>Arguments</td><td><i>startTime</i></td><td>The start of the time period of the recording measured in seconds since midnight (GMT) on 1/1/1970. If the start time occurs in the past and the current time is within the specified duration of the recording, the OITF SHALL start recording immediately and MAY record any earlier content from the current programme that is available (e.g. stored in a time-shift buffer).</td></tr>
<tr><td></td><td><i>duration</i></td><td>The duration of the recording in seconds.</td></tr>
<tr><td></td><td><i>repeatDays</i></td><td>Bitfield indicating on which days of the week the recording SHOULD be repeated. Values are as follows:
<table>
<tr><th>Day</th><th>Bitfield Value</th></tr>
<tr><td>Sunday</td><td>0x01 (i.e. 00000001)</td></tr>
<tr><td>Monday</td><td>0x02 (i.e. 00000010)</td></tr>
<tr><td>Tuesday</td><td>0x04 (i.e. 00000100)</td></tr>
<tr><td>Wednesday</td><td>0x08 (i.e. 00001000)</td></tr>
<tr><td>Thursday</td><td>0x10 (i.e. 00010000)</td></tr>
<tr><td>Friday</td><td>0x20 (i.e. 00100000)</td></tr>
<tr><td>Saturday</td><td>0x40 (i.e. 01000000)</td></tr>
</table>
These bitfield values can be ‘OR’-ed together to repeat a recording on more than one day of a week (e.g. weekdays)

A value of 0x00 indicates that the recording will not be repeated.</td></tr>
<tr><td></td><td><i>channelID</i></td><td>The identifier of the channel from which the broadcasted content is to be recorded. Specifies either a ccid or ipBroadcastID (as defined by the `Channel` object in Section 7.3.12)</td></tr>
</table>

<table>
<tr><td colspan="2">`ScheduledRecordingCollection` **`getScheduledRecordings`**`()`</td></tr>
<tr><td>Description</td><td>Returns a subset of all the recordings that are scheduled but which have not yet started. The subset SHALL include only scheduled recordings that were scheduled using a service from the same FQDN as the domain of the service that calls the method.</td></tr>
</table>

ChannelConfig **getChannelConfig()**	
Description	Returns the channel line-up of the tuner in the form of a ChannelConfig object as defined in Section 7.13.8. This includes the favourite lists. The ChannelConfig object returned from this function SHALL be identical to the ChannelConfig object returned from the getChannelConfig() method on the video/broadcast object as defined in 7.13.3.

void **remove**(ScheduledRecording recording)		
Description	Remove a recording (either scheduled, in-progress or completed). For non-privileged applications, recordings SHALL only be removed when they are scheduled but not yet started and the recording was scheduled by the current service. ~~Removes a scheduled recording.~~ As with the record method, only the programmeID property of the scheduled recording SHALL be used to identify the scheduled recording to remove where this property is available. The other data contained in the scheduled recording SHALL NOT be used when removing a ~~scheduled~~ recording scheduled using methods other than recordAt(). For recordings scheduled using recordAt(), the data used to identify the recording to remove is implementation dependent. If the A/V Control object is referring to the indicated recording the state in A/V Control object shall be automatically changed to 6 (the error state).	
Arguments	*recording*	The scheduled recording to be removed.

Programme **createProgrammeObject**()	
Description	Factory method to create an instance of Programme

7.10.2 The ScheduledRecording class

The ScheduledRecording object represents a scheduled programme in the system, i.e. a recording that is scheduled but which has not yet started. . The values of the properties of a ScheduledRecording (except for startPadding and endPadding) are provided when the object is created using one of the record() methods in Section 7.10.1, for example by using a corresponding Programme object as argument for the record() method, and can not be changed for this scheduled recording object (except for startPadding and endPadding).

7.10.2.1 Constants

The following constants are defined as properties of the ScheduledRecording class:

Name	Value	Use
`ID_TVA_CRID`	0	Used in the `programmeIDType` property to indicate that the programme is identified by its TV-Anytime CRID (Content Reference Identifier).
`ID_DVB_EVENT`	1	Used in the `programmeIDType` property to indicate that the programme is identified by a DVB URL referencing a DVB-SI event as enabled by Section 4.1.3 of [META]. Support for this constant is OPTIONAL.

7.10.2.2 Properties

`Integer` **`startPadding`**
The amount of padding to add at the start of a scheduled recording, in seconds. This property is initialised to the value of the `Configuration.pvrStartPadding` property. The default OITF defined start padding MAY be changed through property `pvrStartPadding` of the `Configuration` class as defined in Section 7.3.2. When a recording is due to start, the OITF MAY use a smaller amount of padding in order to avoid conflicts with other recordings. Positive values of this property SHALL cause the recording to start earlier than the specified start time (i.e. the actual duration of the recording shall be increased); negative values SHALL cause the recording to start later than the specified start time (i.e. the actual duration of the recording shall be decreased).

`Integer` **`endPadding`**
The amount of padding to add at the end of a scheduled recording, in seconds. This property is initialised to the value of the `Configuration.pvrEndPadding` property. The default OITF defined end padding MAY be changed through property `pvrEndPadding` of the `Configuration` class as defined in Section 7.3.2. When a recording is in progress, the OITF MAY use a smaller amount of padding in order to avoid conflicts with other recordings. Positive values of this property SHALL cause the recording to end later than the specified end time (i.e. the actual duration of the recording shall be increased); negative values SHALL cause the recording to end earlier than the specified end time (i.e. the actual duration of the recording shall be decreased).

`readonly Integer` **`repeatDays`**

Bitfield indicating on which days of the week the recording SHOULD be repeated. Values are as follows:

Day	Bitfield Value
Sunday	0x01 (i.e. 00000001)
Monday	0x02 (i.e. 00000010)
Tuesday	0x04 (i.e. 00000100)
Wednesday	0x08 (i.e. 00001000)
Thursday	0x10 (i.e. 00010000)
Friday	0x20 (i.e. 00100000)
Saturday	0x40 (i.e. 01000000)

These bitfield values can be 'OR'-ed together to repeat a recording on more than one day of a week (e.g. weekdays)

A value of 0x00 indicates that the recording will not be repeated.

For recordings other than those created using the `recordAt()` method, the value of this property SHALL be `undefined`.

`readonly String` **`name`**

The short name of the scheduled recording, e.g. 'Star Trek: DS9'. For recordings scheduled using the `oipfRecordingScheduler.recordAt()` method, OITFs SHALL set this to an implementation-dependent value (e.g. "Manual Recording").

`readonly String` **`longName`**

The long name of the scheduled recording, e.g. 'Star Trek: Deep Space Nine'. If the long name is not available, this property will be `undefined`.

`readonly String` **`description`**

The description of the scheduled recording, e.g. an episode synopsis. If no description is available, this property will be `undefined`.

`readonly String` **longDescription**
The long description of the programme. If no description is available, this property will be `undefined`.

`readonly Integer` **startTime**
The start time of the scheduled recording, measured in seconds since midnight (GMT) on 1/1/1970. The value for the `startPadding` property can be used to indicate if the recording has to be started before the `startTime` (as defined by the `Programme` class).

`readonly Integer` **duration**
The duration of the scheduled recording (in seconds). The value for the `endPadding` property can be used to indicate how long the recording has to be continued after the specified duration of the recording.

`readonly Channel` **channel**
Reference to the broadcast channel where the scheduled programme is available.

`readonly Boolean` **isSeries**
If true, then when a subsequent episode of a programme becomes available it SHOULD be added to the recording list automatically. Note: Where several episodes of a season are available, then only the latest scheduled recording will carry the `isSeries` flag.

`readonly String` **programmeID**
The unique identifier of the scheduled programme or series, e.g. a TV-Anytime CRID (Content Reference Identifier). For recordings scheduled using the `oipfRecordingScheduler.recordAt()` method, the value of this property MAY be undefined.

`readonly Integer` **programmeIDType**
The type of identification used to reference the programme, as indicated by one of the ID_* constants defined in Section 7.10.2.1. For recordings scheduled using the `oipfRecordingScheduler.recordAt()` method, the value of this property MAY be undefined.

`readonly Integer` **episode**
The episode number for the programme if it is part of a series. This property is `undefined` when the

programme is not part of a series or the information is not available.

readonly Integer **totalEpisodes**
If the programme is part of a series, the total number of episodes in the series. This property is `undefined` when the programme is not part of a series or the information is not available.

readonly ParentalRatingCollection **parentalRating**
A collection of parental rating values for the programme for zero or more parental rating schemes supported by the OITF. The value of this property is typically provided by a corresponding "`Programme`" object that is used to schedule the recording and can not be changed for this scheduled recording object. If no parental rating information is available for this scheduled recording, this property is a `ParentalRatingCollection` object (as defined in Section 7.9.5) with length 0. Note that if the `parentalRating` property contains a certain parental rating (e.g. PG-13) and the broadcast channel associated with this scheduled recording has metadata that says that the content is rated PG-16, then the conflict resolution is implementation dependent.

7.10.3 The ScheduledRecordingCollection class

A `ScheduledRecordingCollection` object represents a read-only collection of `ScheduledRecording` objects. Next to the properties and methods defined below a `ScheduledRecordingCollection` object SHALL support the array notation to access the items in this collection.

7.10.3.1 Properties

readonly Integer **length**
The number of items in the collection.

7.10.3.2 Methods

ScheduledRecording **item**(Integer index)		
Description	Return the item at position `index` in the collection.	
Arguments	*index*	The index of the item to be returned

7.10.4 Extension to application/oipfRecordingScheduler for control of recordings

The OITF SHALL support the following extensions to the `application/oipfRecordingScheduler` object defined in section 7.10.1.

This subsection SHALL apply for OITFs that have indicated support for the extended PVR management functionality by adding the attribute `'manageRecordings = true'` to the `<recording>` element in the client capability description as defined in Section 9.3.3

The functionality as described in this section is subject to the security model of Section 10.

7.10.4.1 Properties

`readonly ScheduledRecordingCollection` **`recordings`**

Provides a list of scheduled and recorded programmes in the system. This property may only provide access to a subset of the full list of recordings, as determined by the value of the `manageRecordings` attribute of the `<recording>` element in the client capability description (see Section 9.3.3).

Note: Where a series is being recorded, every recorded episode SHALL exist as an independent entry. Only the scheduled recording SHALL carry the `isSeries` property.

`readonly DiscInfo` **`discInfo`**

Get information about the status of the local storage device. The `DiscInfo` class is defined in Section 7.16.4.

`function` **`onPVREvent`**`( Integer state, Recording recording )`

This function is the DOM 0 event handler for notification of changes in the state of recordings. See the definition of the corresponding DOM 2 `PVREvent` in Section 7.13.2.2 for more details.

The specified function is called with the following arguments:

`Integer state` – The current state of the recording. One of:

Value	Description
1	The recording has started.
2	The recording has stopped, having completed.
3	The recording sub-system is unable to record due to resource limitations.
4	There is insufficient storage space available. (Some of the recording may be available).
6	The recording has stopped before completion due to unknown (probably hardware) failure.
7	The recording has been newly scheduled.
8	The recording has been deleted (for complete or in-progress recordings) or removed from the schedule (for scheduled recordings).
9	The recording is due to start in a short time.

10	The recording has been updated.

`ScheduledRecording recording` – The recording to which this event refers.

7.10.4.2 Methods

`Recording` **`getRecording`**`( String id )`		
Description	Returns the `Recording` object for which the value of the `Recording.id` property corresponds to the given `id` parameter. If such a `Recording` does not exist, the method returns `null`.	
Arguments	*id*	Identifier corresponding to the `id` property of a `Recording` object.

~~`void` **`remove`** `( ScheduledRecording recording )`~~		
~~Description~~	~~Remove a recording (either scheduled, in-progress or completed).~~ ~~For non-privileged applications, recordings SHALL only be removed when they are scheduled but not yet started and the recording was scheduled by the current service.~~ ~~If the A/V Control object is referring to the indicated recording the state in A/V Control object shall be automatically changed to 6 (the error state).~~	
~~Arguments~~	~~*recording*~~	~~The recording to be removed.~~

`void` **`stop`**`( Recording recording )`		
Description	Stop an in-progress recording. The recording SHALL NOT be deleted.	
Arguments	*recording*	The recording to be stopped.

`void` **`refresh`**`()`	
Description	Update the `recordings` property to show the current status of all recordings.

7.10.4.3 Events

For the intrinsic events listed in the table below, a corresponding DOM level 2 event SHALL be generated in the following manner:

Intrinsic event	Corresponding DOM 2 event	DOM 2 Event properties
`onPVREvent`	`PVREvent`	Bubbles: No Cancelable: No Context Info: `state`, `recording`

Note: the DOM 2 events are directly dispatched to the event target, and will not bubble nor capture. Remote UIs SHOULD NOT rely on receiving these events during the bubbling or the capturing phase. Remote UIs that use DOM 2 event handlers SHALL call the `addEventListener()` method on the `application/oipfScheduledRecording` object itself. The third parameter of `addEventListener`, i.e. “`useCapture`”, will be ignored.

7.10.5 The Recording class

The `Recording` class represents an in-progress or completed recording being made available through the extended PVR management functionality as defined in Section 7.10.4. This class implements the `ScheduledRecording` interface (see Section 7.10.2), with the following changes:

- The `startPadding` property is read only.
- For in-progress recordings, changes to the value of the `endPadding` property SHALL modify the actual duration of the recording. If the value of the `endPadding` property is changed so that the current actual duration of the recording exceeds the new actual duration specified by the sum of the `startPadding`, `duration` and `endPadding` properties, the recording SHALL be stopped immediately. Changing the value of this property for a completed recording SHALL have no effect.

Recordings MAY be “manual” in that they simply record a channel at a certain time, for a period - analogous to a traditional VCR - or alternatively recordings can be programme based.

If an in-progress recording is interrupted and automatically resumed, e.g. due to a temporary power failure, all sections of the recording SHALL be represented by a single `Recording` object.

Values of properties in the `Recording` object SHALL be obtained from metadata about the recorded programme and are typically copied from the Programme used for scheduling a recording by the `record(Programme programme)` method of the `application/oipfRecordingScheduler` object. See Section 7.10.4 for more information about the mapping between the properties of a Programme and the BCG metadata.

7.10.5.1 Properties

`readonly Integer` **`state`**

The state of the recording. One of:

Value	Description
`1`	The recording has started.

2	The recording has stopped, having completed.
3	The recording sub -system is unable to record due to resource limitations.
4	There is insufficient storage space available. (Some of the recording may be available).
5	The recording has not taken place due to unknown (probably hardware) failure.
6	The recording has only partially completed due to a clash or hardware failure. There are three possible conditions for this: 1) The end of the recording is missed. 2) The start of the recording is missed. 3) A piece from the centre of the recording is missed (e.g. due to the receiver rebooting or a transient failure of the network connection).

`readonly String` **`id`**
An identifier for this recording. This value SHALL be unique to this recording and so can be used to compare two recording objects to see if they refer to the same recording. The OITF SHALL guarantee that recording identifiers are unique in relation to download identifiers and `CODAsset` identifiers.

`readonly Boolean` **`isManual`**
`true` if the recording was scheduled using `oipfRecordingScheduler.recordAt()` or using a terminal-specific approach that does not use guide data to determine what to record, `false` otherwise. If `false`, then any fields whose name matches a field in the `Programme` object contains details from the programme guide on the programme that has been recorded. If `true`, only the channel, start time and duration of the recording are valid.

`Boolean` **`doNotDelete`**
If `true`, then this recording should not be automatically deleted by the system.

`Integer` **`saveDays`**
The number of days for which an individual or manual recording SHOULD be saved. Recordings older than this value MAY be deleted. This property is initialised to the value of the `Configuration.pvrSaveDays` property.

`Integer` **`saveEpisodes`**

The number of episodes of a series-link that SHOULD be saved. Older episodes MAY be deleted. This is only valid when set on the latest scheduled recording in the series. This property is initialised to the value of the `Configuration.pvrSaveEpisodes` property.

`readonly Boolean` **`blocked`**

Flag indicating whether the programme is blocked due to parental control settings or conditional access restrictions.

The `blocked` and `locked` properties work together to provide a tri-state flag describing the status of a programme. This can best be described by the following table:

Description	**blocked**	**locked**
No parental control applies.	`false`	`false`
Item is above the parental rating threshold (or manually blocked); no PIN has been entered to view it and so the item cannot currently be viewed.	`true`	`true`
Item is above the parental rating threshold (or manually blocked); the PIN has been entered and so the item can be viewed.	`true`	`false`
Invalid combination – OITFs SHALL NOT support this combination	`false`	`true`

`readonly ParentalRatingCollection` **`parentalRatings`**

A collection of parental rating values for the programme for zero or more parental rating schemes supported by the OITF, defined using the `ParentalRatingCollection` object as specified in Section 7.9.5. If no parental rating information is available for this scheduled recording, this property is a `ParentalRatingCollection` object with length 0.

Note that if the `parentalRatings` property contains a certain parental rating (e.g. PG-13) and the broadcast channel associated with this scheduled recording has metadata that says that the content is rated PG-16, then the conflict resolution is implementation dependent.

`readonly Integer` **`showType`**

Flag indicating the type of show. This field SHALL take one of the following values:

Value	**Description**
0	The show is live.
1	The show is a first-run show.

2	The show is a rerun.

`readonly Boolean` **`subtitles`**

Flag indicating whether subtitles or closed-caption information is available.

`readonly StringCollection` **`subtitleLanguages`**

Supported subtitle languages, indicated by iso639 language codes.

`readonly Boolean` **`isHD`**

Flag indicating whether the programme has high-definition video.

`readonly Boolean` **`isWidescreen`**

Flag indicating whether the programme is broadcast in widescreen.

`readonly Integer` **`audioType`**

Bitfield indicating the type of audio that is available for the programme. Since more than one type of audio may be available for a given programme, the value of this field SHALL consist of one or more of the following values ORed together:

Value	Description
1	Mono audio
2	Stereo audio
4	Multi-channel audio

`readonly Boolean` **`isMultilingual`**

Flag indicating whether more than one audio language is available for this recording.

`readonly StringCollection` **`audioLanguages`**

Supported audio languages, indicated by iso639 language codes.

readonly StringCollection **genres**
A collection of genres that describe this programme.

readonly Integer **recordingStartTime**
The actual start time of the recording, including any padding. This MAY not be the same as the scheduled start time of the recorded programme (e.g. due to a recording starting late, or due to start/end padding).

readonly Integer **recordingDuration**
The actual duration of the recording, including any padding. This MAY not be the same as the scheduled duration of the recording (e.g. due to a recording finishing early, or due to start/end padding).

readonly BookmarkCollection **bookmarks**
A collection of the bookmarks set in a recording. If no bookmarks are set, the collection SHALL be empty.

readonly Boolean **locked**
Flag indicating whether the current state of the parental control system prevents the recording from being viewed (e.g. a correct parental control PIN has not been entered to allow the recording to be viewed).